A Continuum Electrostatic Approach for Calculating The Binding Energetics of Multiple Ligands

D I S S E R T A T I O N

submitted to
the Faculty of Biology, Chemistry and Geoscience
of the University of Bayreuth, Germany

for obtaining the degree of
Doctor of Natural Sciences

presented by
Timm Essigke

Bayreuth 2007

Die vorliegende Arbeit wurde in dem Zeitraum von Januar 2004 bis Dezember 2007 an der Universität Bayreuth unter der Leitung von Professor G. Matthias Ullmann erstellt.

Vollständiger Abdruck der von der Fakultät Biologie, Chemie und Geowissenschaften der Universität Bayreuth genehmigten Dissertation zur Erlangung des akademischen Grades Doktor der Naturwissenschaften (Dr. rer. nat.).

Erster Prüfer:	Prof. Dr. G. Matthias Ullmann
Zweiter Prüfer:	Prof. Dr. Holger Dobbek
Dritter Prüfer:	Prof. Dr. Thomas Hellweg
Prüfungsvorsitz:	Prof. Dr. Jürgen Senker

Tag der Einreichung:	20.12.2007
Zulassung zur Promotion:	09.01.2008
Annahme der Dissertation:	24.01.2008
Auslage der Arbeit:	24.01.2008 - 06.02.2008
Kolloquium:	20.02.2008

A Continuum Electrostatic Approach for Calculating The Binding Energetics of Multiple Ligands

SUMMARY

Complex biomolecules like proteins or nucleic acids can transiently bind various ligands, *e.g.,* electrons, protons, ions or larger molecules. This property is the key to enzymatic catalysis, regulation and energy transduction in biological systems. Interactions between different ligand binding sites can lead to complex titration behaviors, which can be explained based on a microstate description of the system. Previous approaches to calculate the binding behavior of multiple ligand, only treated sites with one or no ligand bound by using a binary state vector to describe the system. Also only one or two ligand types, *i.e.,* protons or electrons, were used for the calculation.

In this thesis, I derive a more general formulation of the theory of ligand binding to biomolecules. For each site any number of charge forms and rotamer forms are allowed as well as any number of ligands and any number of ligand types can be bound. Charge and rotamer forms of sites can be parameterized by measurements on model reactions in solution or by quantum chemical calculations. An energy function is described, consistently combining experimentally determined contributions and those, which can be calculated by continuum electrostatics, molecular mechanics and quantum chemistry. Programs (*i.e.,* QMPB and Perl Molecule) were developed to perform calculations based on the generalized ligand binding theory. The class library Perl Molecule was developed to write powerful Perl programs, which perform the necessary processing steps, *e.g.,* for the conversion of a pdb-file into the input required for energy calculations. The generated input contains all experimentally determined or molecular mechanically and quantum chemically obtained parameters. The energy calculations are performed by the program QMPB, which uses other programs for the continuum electrostatics calculations to solve the linearized Poisson-Boltzmann equation. The computations scale linearly with the total number of sites of the system and can easily be performed in parallel. From the obtained energies, microscopic ligand binding probabilities can be calculated as function of chemical potentials of ligands in solution, *e.g.,* by a Monte Carlo program. Additionally, microscopic and macroscopic equilibrium constants can be computed.

The usefullness and correctness of the new approach based on a generalized ligand binding theory is demonstrated by a number of studies on diverse examples. Because various groups used Lysozyme as benchmark system for continuum electrostatics, it is chosen to test if previously obtained results can be reproduced with QMPB. Different quantum chemical approaches are applied to the benzoquinone system for parameterizing a site with several protonation and reduction forms. A complex site is also the Cu_B center in Cytochrome *c* oxidase, which is studied to decide if multiple protonation forms of the coordinating histidines are involved in the reaction mechanism. Factors influencing the reduction potential of the electron transfer protein ferredoxin are analyzed using the programs Perl Molecule and QMPB. Here,

in particular conformational changes of a peptide bond in the vicinity of the [2Fe-2S] center are of interest. The protonation form of a neighboring glutamate turns out to influence the reduction potential strongly. Protonation and phosphorylation studies on the protein HPr lead to the development of a four-microstate model to explain conformational changes on a histidine, which can be observed by experiment, molecular dynamics simulation and electrostatic calculations. The phosphorylation and protonation state dependent conformational change can be related to the dual role of the protein in regulation and phosphate-transfer. The new microstate description does not only allow to analyze thermodynamic properties but also paved the road for the study of the kinetics of charge transfer.

ZUSAMMENFASSUNG

Komplexe Biomoleküle, wie Proteine oder Nukleinsäuren, sind in der Lage, vorübergehend verschiedene Liganden zu binden, wie z. B. Elektronen, Protonen, Ionen oder größere Moleküle. Diese Eigenschaft ist der Schlüssel zur enzymatischen Katalyse, zur Regulation und zur Energietransduktion in biologischen Systemen. Die Interaktionen verschiedener Ligandenbindungsstellen können zu komplexem Titrationsverhalten führen, das basierend auf einer Mikrozustandsbeschreibung des Systems erklärt werden kann. Frühere Ansätze, das Bindungsverhalten von mehreren Liganden zu berechnen, behandelten nur Bindungsstellen, die entweder einen oder keinen Liganden gebunden hatten, so dass ein binärer Zustandsvektor zur Beschreibung des Systems benutzt werden konnte. Außerdem wurden nur ein oder zwei Ligandentypen, wie z. B. Elektronen oder Protonen, für die Rechnung zugelassen.

In der vorliegenden Arbeit leite ich eine generellere Formulierung der Ligandenbindungstheorie für Biomoleküle her. Für jede Bindungsstelle sind eine beliebige Anzahl von Ladungs- und Rotamerformen möglich und es können eine beliebige Anzahl von Liganden und eine beliebige Anzahl von Ligandentypen gebunden werden. Ladungs- und Rotamerformen von Bindungsstellen können durch Messungen von Modellreaktionen in Lösung oder durch quantenchemische Rechnungen parameterisiert werden. Eine Energiefunktion wird beschrieben, die experimentell bestimmte Beiträge, sowie Beiträge aus Kontinuumselektrostatik-, Molekularmechanik- und Quantenchemierechnungen konsistent kombiniert. Hierzu wurden Programme, insbesondere QMPB und Perl Molecule, entwickelt, die es ermöglichen, Berechnungen basierend auf der generalisierten Ligandenbindungstheorie durchzuführen. Die Klassenbibliothek Perl Molecule wurde entwickelt, die es erlaubt, leistungsfähige Perl-Programme zu schreiben. Diese Programme führen die Schritte durch, die notwendig sind, um z. B. eine PDB-Datei in die für Energieberechnungen benötigten Eingabe-Dateien umzuwandeln. Die so erzeugten Eingabe-Dateien enthalten alle Parameter, die entweder experimentell bestimmt oder über Molekularmechanik- und Quantenchemierechnungen ermittelt wurden. Die Energieberechnungen werden von dem Programm QMPB durchgeführt, das andere Programme für die Kontinuumselektrostatik-Rechnungen benutzt, die die linearisierte Poisson-Boltzmann Gleichung lösen. Die mit QMPB durchgeführten Rechnungen skalieren linear mit der totalen Anzahl von Bindungsstellen des Systems und können einfach parallel ausgeführt werden. Von den ermittelten Energien können, z. B. mit Hilfe eines Monte Carlo Programmes, mikroskopische Wahrscheinlichkeiten für die Ligandenbindung errechnet werden. Diese Wahrscheinlichkeiten sind abhängig von den chemischen Potentialen der Liganden in Lösung. Zusätzlich können mikroskopische und makroskopische Gleichgewichtskonstanten berechnet werden.

Der Nutzen und die Richtigkeit des hier beschriebenen neuen Ansatzes, basierend auf einer allgemeineren Ligandenbindungstheorie, wird anhand von einigen Untersuchungen an

einer Reihe von Beispielen demonstriert. Da schon verschiedene Gruppen Lysozym als Testsystem für Kontinuumselektrostatikrechnungen genutzt haben, wird es auch in der vorliegenden Arbeit verwendet, um die Reproduzierbarkeit älterer Ergebnisse mit QMPB zu überprüfen. Unterschiedliche quantenchemische Methoden wurden auf das Benzochinon-System zur Parameterisierung als Bindungsstelle mit mehreren Protonierungs- und Reduktionsformen angewendet. Eine komplizierte Bindungsstelle stellt auch das Cu_B Zentrum von Cytochrom *c* Oxidase dar. Es wird untersucht, ob mehrere Protonierungsformen der koordinierenden Histidine am Reaktionsmechanismus beteiligt sind. Mit den Programmen Perl Molecule und QMPB werden Faktoren analysiert, die das Reduktionspotential des Elektronenüberträger-Proteins Ferredoxin beeinflussen. In diesem Zusammenhang sind besonders konformationelle Änderungen einer Peptidbindung in der Nähe des [2Fe-2S] Zentrums von Interesse. Es hat sich herausgestellt, dass die Protonierungsform eines benachbarten Glutamates einen grossen Einfluss auf das Reduktionspotential hat. Untersuchungen der Protonierung und Phosphorylierung des Proteins HPr führen zur Entwicklung eines Vierzustandsmodells. Damit können konformationelle Änderungen eines Histdins erklärt werden, die experimentell, in Molekulardynamiksimulationen und in Elektrostatikrechnungen beobachtet werden können. Die vom Phosphorylierungs- und Protonierungszustand abhängigen konformationellen Änderungen können mit den zwei Aufgaben des Proteins in der Regulation und im Phosphattransfer in Verbindung gebracht werden. Die neue Mikrozustandsbeschreibung erlaubt es nicht nur, thermodynamische Eigenschaften zu analysieren, sondern bereitet auch den Weg für kinetische Untersuchungen von Ladungstransfers.

PREFACE

Acknowledgements

This thesis was done in the last four years in the group of Professor G. Matthias Ullmann at the University of Bayreuth. Matthias has been a friend and mentor for almost the last 10 years. After raising my interest in computational biochemistry and biophysics, he continuously supported me as undergraduate student at Free University Berlin and The Scripps Research Institute, La Jolla, USA, as researcher in Rebecca Wade's group in Heidelberg and as his graduate student here in Bayreuth in the last years. I am particularly grateful to Matthias for also sticking by me in difficult times. His contribution of countless ideas, suggestions and advice was so valueable to my work, that I can not thank him enough. I am also grateful for his contribution of the programs SMT and GMCT to the toolchain developed in this thesis. He provided me with excellent working conditions and substantial computation time, which was important for many applications.

Eva-Maria Krammer tremendously helped during my work. She was the first to test my programs Perl Molecule and QMPB by herself on the bacterial photosynthetic reaction center. On one hand, the complex system challenged my programs, previously only tested on much smaller proteins, and on the other hand using the programs without having in mind how they work in detail highlighted several bugs and pitfalls which could then be removed. Many suggestions of her helped to improve the programs. Our aim to parameterize ubiquinone for the reaction center led to a broad study of quantum chemical methods applied to benzoquinone, duroquinone and ubiquinone. Eva conducted an intense literature research and ran numerous Gaussian calculations of which I can only discuss a small fraction in the application section of this work. I am very grateful to her for proofreading my thesis and the many suggestions and corrections she made. Eva's encouragements and exemplary diligence were a great motivation.

With Thomas Ullmann I had many valueable discussions profitting from his detailed literature knowledge, his insights obtained from running countless calculations with MEAD and ADF and his studies of the source code of these programs to extend them. In many respects he is picking up and extending my work. For example he does studies with sidechain rotamers using QMPB which are much more advanced than my initial tests. Thomas implemented a sophisticated membrane model into MEAD, which allows to study membrane proteins in a very realistic environment including membrane potentials. To use the new helper programs for solving the Poisson-Boltzmann equation only little changes in a script generated by QMPB were required. I am also thankful for his good company at our core hours during night shift.

With Punnagai Munusami I was collaborating to test and use Perl Molecule and QMPB on her system Cytochrome *c* oxidase. A part of this work is reported in the applications section. Punnagai is clearly our Gaussian expert and contributed many helpful suggestions to the quinone project of Eva and me. I appreciated the many discussions we had.

Mirco Till is using QMPB to calculate microstate energies as basis for kinetic studies with DMC. Therefore, his work builds up on mine and extends it into a new direction. The application on gramicidin A is briefly discussed and other applications, *e.g.,* on the reaction center with Eva will follow. I am grateful, that he backed me up in case of computer and network problems. Redundancy is particularly important on OSI layer 8. I enjoyed our computer and programming related discussions.

All the other present and former group members also had their contributions adding their scientific skills and making every day more fun: Dr. Elisa Bombarda brought a lot of vitality into our sometime too quite group. She did the groundwork for including membrane potentials into electrostatic computations as it was implemented by Thomas for QMPB. Frank Dickert was always helpful with his superb technical skills and a driving force for most group activities. He provided the protein of his diploma project, cytochrome *c*2, as my second test case for QMPB after lysozyme. Edda Kloppmann saved me a lot of work providing the LaTeX style files for this document. In the last months I miss our discussions. Dr. Torsten Becker was always a pleasant office mate, helping with his physical knowledge, intuition and humor. The program DMC, he develops together with Mirco, is breaking new ground in simulating long range charge transfer. Dr. Astrid Klingen's study on the histidine treatment procedure in Multiflex was helpful for a better comparison. Supervising the bachelor thesis of Thomas Weinmaier was an instructive experience. I am grateful for his work creating a SWIG interface for MEAD and his suggestions of a number of informatics books. I thank Siriporn Promsri for the clock (Fig. 4.6) stimulating some good ideas. I miss the discussions about ferredoxin, its reductase and other topics with Verónica Dumit and I am looking forward to her joining our group again. I am happy, that Silke Wieninger is the first member of a new generation of PhD students in our group, who will solve all the remaining problems.

Not only inside the group my thesis was supported by a number of inspiring collaborations, but we also had a very fruitful collaboration with Nadine Homeyer and Professor Heinrich Sticht (University of Erlangen) on the regulatory and phosphate transfer protein HPr. Research was a perfect team play, where results of one group guided the next steps of the other group. I very much enjoyed our constructive and motivating discussions.

Thanks to Sabrina Förtsch and Alexander Dotor for proofreading and making helpful suggestions on the informatics chapter.

I would not have come to this point without Professor Ernst-Walter Knapp (Free University Berlin) supporting me during my studies and guiding me into this interesting field of research. Likewise I am deeply grateful for Dr. Rebecca Wade's (EML Research gGmbH) support and discussions during my time in Heidelberg. The research I did with her on ferredoxin and other proteins highlighted shortcomings of previous programs. Therefore, it was the starting point for my thesis work.

My thesis substantially builds up on theoretical works of Professor Donald Bashford and his group. His MEAD library was the basis for electrostatic programs I developed to be run by QMPB. I am thankful, that he realized even before the "open source revolution" that scientific

progress requires liberal licenses like the GPL, which allow that code is distributed and extended free of charge for everyone on the internet. Professor Louis Noodleman and his group did pioneering work on combining quantum chemical and Poisson-Boltzmann calculations, which formed a good basis for my studies.

I acknowledge the progress ADF has made in the last years in particular in terms of speed and scaleability. I am grateful to Dr. Stan van Gisbergen for providing us with this superb software package. I appreciate very much the work of the CHARMM and Gaussian developers, who's programs were of great value during this work. Special thanks also to Dr. Nicolas Calimet for writing and providing Hwire for hydrogen placement and to the developers of VMD for molecular visualization. Several figures in this work were made with VMD, sometimes in combination with povray. Other figures were drawn with xfig, xmgrace, dia, inkscape and chemtool. Typesetting was done with LaTeX using Kile, KBibTex, KBib and Kpdf. Most programming I did in Perl. I deeply appreciate the great work Dr. Larry Wall and many Perl developers contributing by CPAN packages to this work. Alike I acknowledge the work of the developers of C++ and the GNU compiler. I appreciated doing all my work on Linux, deeply grateful to Dr. Linus Torvalds and to many other kernel developers, the GNU Foundation and the Debian project for their great distribution. It was fun and a pleasant distraction to research which features this operation system can provide for us. It is impossible to thank all open source developers for their programs explicitly which were helpful for my thesis.

I acknowledge the support of the compute center of the University of Bayreuth, in particular I thank Dr. Bernhard Winkler for running our compute cluster and backup.

Last but not least, I want to deeply thank my parents Jutta and Walter Essigke for their continuous support and encouragement. They gave me the freedom to choose my way and follow my interests but never failing to be on my side if I needed their advice.

Typesetting Conventions

Italics are used for:

- Expressions in other languages than English (*e.g., i.e., etc.*)
- Organisms (*Bacillus subtilis*, *Escherichia coli*, *E. coli*)
- Equation symbols ($\Delta\Delta G_{\mathrm{Born}}$, $\langle x_i(l_m, \{\mu_\lambda\})\rangle$)

`Constant width` is used for:

- Class, object, method or attribute names
- Words taken from input and output files

Sans serif is used for:

- Program names (QMPB. Perl Molecule, Multiflex)

Publications

1. What Determines the Redox Potential of Ferredoxins?
   Timm Essigke, G. Matthias Ullmann, Rebecca C. Wade
   In: Proceedings of the 13th International Conference on Cytochromes P450, Monduzzi Editore, Bologna: 25-30, **2003**

2. Calculation of the Redox Potential of Iron-Sulfur Proteins
   Timm Essigke, Rebecca C. Wade and G. Matthias Ullmann
   J. Inorg. Biochem., 96(1),127, **2003**

3. Formation and Characterization of an All-Ferrous Rieske Cluster and Stabilization of the $[2\mathrm{Fe}-2\mathrm{S}]^0$ Core by Protonation
   Ellen J. Leggate, Eckhard Bill, Timm Essigke, G. Matthias Ullmann and Judy Hirst
   Proc. Natl. Acad. Sci. USA, 101(30), 10913-10918, **2004**

4. Effect of HPr Phosphorylation on Structure, Dynamics, and Interactions in the Course of Transcriptional Control
   Nadine Homeyer, Timm Essigke, Heike Meiselbach, G. Matthias Ullmann and Heinrich Sticht
   J. Mol. Model., 13(3), 431-444, **2007**

5. Effects of Histidine Protonation and Phosphorylation on Histidine-Containing Phosphocarrier Protein Structure, Dynamics, and Physicochemical Properties
   Nadine Homeyer[1], Timm Essigke[1], G. Matthias Ullmann and Heinrich Sticht
   Biochemistry, 46(43), 12314-12326, **2007**

6. Investigating the Mechanisms of Photosynthetic Proteins Using Continuum Electrostatics
   G. Matthias Ullmann, Edda Kloppmann, Timm Essigke, Eva-Maria Krammer, Astrid R. Klingen, Torsten Becker, Elisa Bombarda
   Submitted to *Photosyn. Res.*, **2007**

7. Does Deprotonation of CuB Ligands Play a Role in the Reaction Mechanism of Cytochrome *c* Oxidase?
   M. Punnagai, Timm Essigke, and G. Matthias Ullmann
   Submitted to *J. Am. Chem. Soc.*, **2007**

8. Simulating the Proton Transfer in Gramicidin A by a Sequential Dynamical Monte Carlo Method
   Mirco S. Till, Torsten Becker, Timm Essigke, and G. Matthias Ullmann
   In preparation for *J. Phys. Chem.*, **2007**

[1] Authors equally contributed

CONTENTS

CHAPTER 1

INTRODUCTION

Computational biochemistry is an emerging field at the crosspoint of biochemistry, computational chemistry and biophysical chemistry. While bioinformatics traditionally has a focus on statistical and algorithmic analysis of sequence data, computational biochemistry studies are usually based on structural data. Many techniques, especially molecular dynamics, developed closely linked to biophysical protocols of structure refinement, *e.g.,* based on NMR spectroscopy or x-ray crystallography. However, the detailed computational analysis of physical properties and dynamics of molecules soon became a discipline on its own, independent of the structure determination process.

As second origin, computational chemistry and quantum chemistry developed as branch of physical chemistry. The initial focus was the calculation of properties and reactions of small organic and inorganic molecules. With the exponential increase of computational power in the last decades, the size of the systems under investigation could grow and soon biological systems were of interest.

As a third origin, biochemistry and molecular biology developed so many powerful techniques to accumulate a wealth of data on many interesting biomolecules. However, often the results are hard to interpret, because there are numerous counter acting effects. For example, some small mutations may hinder a protein to fold. Alternatively, mutations, which are expected to have a large effect may have only little influence on the rest of the protein. Thus replacing a charged sidechain by another with opposite charge may have little effect, because the altered charge is compensated by other groups. It is a logical desire to plan expensive experiments on the computer before performing them in the lab and to use computer programs to guide the interpretation of experimental results, especially when structural data is already available.

The size of biomolecules, their stabilization based on many weak interactions, *i.e.,* hydrogen bonds and van-der-Waals interactions, and dependence on their native environment, an ionic aqueous solution, raised new challenges for the field of computational biochemistry. For example, typical quantum chemical models of an active site of an enzyme consisting of only a few atoms in vacuum turned out to be inadequate to describe the intricate energetics in proteins. The influence of the environment and solvent was found to be crucial to be modeled accurately. The dominant long reaching physical interaction at the scale of molecules is the electrostatic interaction and therefore electrostatics are most important to describe the environment. Thus, also this work is centered on the computation of electrostatic energies.

After giving a short overview about the methods used in this work, *i.e.,* molecular mechanics, quantum chemistry and continuum electrostatics, I will focus on approaches suited for calculation of ligand binding energetics. Finally, the aim of the thesis will be formulated in detail.

1.1 Techniques of Biomolecular Simulation

1.1.1 Molecular Mechanics

Maybe the most widely known method is molecular mechanics (MM), which describes molecules by classical potentials. Building blocks of molecules are parameterized from experimental and quantum chemical data allowing to build a multitude of molecules without additional parameterization. Energies of different structures and different conformations can be qualitatively and in many cases also quantitatively computed. The energy function is cheap enough to compute long-time dynamics of proteins (typically 10 ns, in exceptions 1 ms). Molecular mechanics energy minimizations and molecular dynamics (MD) simulations including experimentally determined constraints, *e.g.,* electron densities in x-ray crystallography or NOE and J-coupling data in NMR spectroscopy, are an essential step of many structure refinement protocols. Also theoretical structure predictions, *e.g.,* homology modeling or threading, often use MD for the final refinement. For protein folding by MD, the timespans which can be covered in the simulation are problematic, but significant progress has been made by a world-wide distributed computing effort [1–6].

Calculations of proteins in vacuum were done in the past (and are still done for refinement with experimental constraints), but generally can lead to undesired behavior like unfolding of the protein. More realistic simulations describe the biomolecule in a box of explicitly modeled water molecules, leading to more reliable and stable results. However, as a drawback a major part of the computational time is spent for simulating water instead of the molecule of interest. Conciderable efforts have been made in development of implicit solvent models (based on solving the Poisson-Boltzmann equation or an approach called Generalized Born). Still it is considered much more reliable to do simulations in a water box.

A drawback of MD is that chemical reactions can not be simulated (no bond-forming and bond-breaking). A semi-classical method was developed, empirical valence bond (EVB, [7, 8]), which parameterizes the chemical reactions by additional force field terms obtained from quantum mechanics and experiment. The method is fast and good results have been obtained. However the major part of the work using this approach is the parameterization, which has to be re-done for each reaction. Other methods use semi-empirical or *ab initio* quantum mechanical (QM) methods to describe the chemical reaction and couple the calculation to MD simulations of the rest of the protein and the solvent (QM/MM, [9]). In this approach, the choice of the boundaries between the QM and the MM part seems to be crucial, because boundary errors seem to be unavoidable.

Usually, no chemicals can leave or enter a MD simulation during the run since the canonical or the isothermal-isobaric ensembles are used. Grand canonical MD allows the number of particles to change, however it suffers form slow convergence. Therefore, it is rarely used.

For non-covalent ligand binding studies, MD is mostly used by placing a ligand in the solvent next to the binding site and analyzing the binding process during a trajectory. Often a constaint between the ligand and the site is increased during the simulation to force the binding. As a result, a detailed understanding of conformational changes during the binding process can be gained. By free energy perturbation studies, exact ligand binding energies can be determined. However, the study of binding of multiple ligands to different sites as a function of the chemical potential of the ligand in solution, as it is intended in this work, is hardly possible with MD simulations.

1.1.2 Quantum Chemistry

Quantum chemistry traditionally studied small molecules due to its computational expense. In the last decades it became increasingly important also for biomolecular systems due to the gain in computational power available. Chemical reactions can be simulated at high accuracy providing insight into reaction pathways, transition state energies and reaction mechanisms. Physical properties can be calculated without the parameterization required for MM, and many properties (*e.g.,* spin populations) do not even conceptionally exist in the classical picture of MM. Not only electrostatic interactions between the atoms under investigation are present, but also electronic polarization is included. Nevertheless, QM is not necessarily more accurate than MM, *e.g.,* for torsion angle parameters, because MM is often intensively parameterized taking spectroscopic data into account. For calculations of atomization energies and ionization potentials, good standard methods (*e.g.,* B3LYP) show average errors of several $\frac{\text{kcal}}{\text{mol}}$ [10] and computational approaches have to try canceling systematic errors by appropriate thermodynamic cycles.

Treatment of whole proteins via QM minimization is possible with semi-empiric methods, but still the methods are far too expensive for quantum dynamics on such large systems. Explicit treatment of solvation is hardly possible with QM because solvation energies or reduction potentials of ions show a very slow convergence [11–14]. For homogeneous solvents a number of continuum approaches exist *e.g.,* COSMO [15–17], PCM [18–20] and many variants of these approaches, which also take polarization of the molecule due to the solvent into account. For inhomogeneous solvents, usually point charges are derived from an electrostatic potential fit method (section 2.3.2) and the solvation energy is obtained from a Poisson-Boltzmann calculation.

1.1.3 Continuum Electrostatics

As pointed out in the beginning, electrostatic interactions are of crucial importance in biological systems. Continuum electrostatics is a very accurate approach to obtain these energies. The solvent is modeled implicitly, which allows to treat systems of basically any size, without computationally expensive dynamic simulations of solvent molecules. QM with good basis sets is limited to a few hundred atoms today (*e.g.,* the large center of ferredoxin in section 5.3). One of the largest MD studies from the point of CPU time was a 1.6 ns simulation of $2.64 \cdot 10^6$ atoms of a solvated ribosome [21], while one of the largest continuum electrostatic calculations was done for $1.25 \cdot 10^6$ atoms of a microtubulule requirering about 2 million times less floating point operations [22]. (Calculations on the ribosome were done in the same work

with $88 \cdot 10^3$ and $95 \cdot 10^3$ atoms for the 30S and 50S subunits, but no timings were reported.) Certainly, the studies are not compareable, but the example should clearly show, that continuum electrostatic calculations are orders of magnitude less demanding than MD simulations on systems of similar size even looking for cutting-edge applications.

By continuum electrostatics, free energies are obtained directly (assuming no dramatic changes in the dynamics of the protein upon binding of the ligand), while MD simulations require ensemble averaged free energy calculations, which depend strongly on how well the conformational space is sampled (*i.e.,* how long the trajectory is and how easily the energy barriers can be overcome at the simulated temperature). However, while MD includes structural flexibility as part of the method, continuum electrostatics only allows for discrete structural changes. In this work, I distinguish conformational changes, which are global changes of the whole molecule (*i.e.,* independently determined structures) and rotameric changes, which are local changes of a part of the molecule (*i.e.,* different occupancies of a sidechain). While conformational changes lead to an exponential increase of computational time, rotamers can be added at only linear cost. The continuum electrostatic approach is a coarser approximation than MD, only including flexibility by the value of the dielectric constant and by a number of discrete conformers and rotamers, but it has not to be worse than MD, if electrostatic interactions are crucial and changes in the dynamics of the system are not.

1.2 Programs for Ligand Binding Studies Based on Continuum Electrostatic Calculations

A number of implementations of solvers for the linearized Poisson-Boltzmann (PB) equation have been written in a number of groups. Examples are the PB solver in UHBD of McCammon *et al.* [23, 24], DelPhi of Honig *et al.* [25–29], APBS of Baker and Holst [22] and MEAD of Bashford *et al.* [30, 31]. APBS and MEAD are freely available under the GNU Public License (GPL), while the other programs have proprietary licenses. APBS is probably the most advanced PB solver at the moment, providing a solver for the non-linearized and the linearized equation. It uses multi-grid-level methods and is fine-grained parallelized scaling up to thousands of CPUs [22].

The PB solvers calculate solvation energies for pre-generated or selected sites. The process of generating or selecting sites and combining the energies appropriately is usually a discipline of another set of programs. Multiflex is a program provided together with the MEAD library, which allows to calculate the shift, when a model compound with experimentally measured $\mathrm{p}K_a$ value is transferred into the protein environment. For UHBD and APBS scripts exist, which perform a similar function. MCCE of Gunner *et al.* is based on DelPhiand has many powerful features [32, 33]. In my work, Perl Molecule was written for the structure preparation step and QMPB for the calculation of energies.

Analytical calculation of titration curves is only possible for very small systems. For larger systems Monte Carlo programs are often used. Also here, each group tends to use their own program. Donald Bashford originally used Paul Beroza's program MCTI for Multiflex. In the group of Ernst-Walter Knapp Karlsberg is developed and used [34]. In our group, Matthias

Ullmann's program CMCT is used in combination with Multiflex and GMCT in combination with QMPB.

1.3 Aim of this Work

Intense research in several groups has shown the power of a continuum electrostatic approach in calculating protonation and reduction probabilities in many biological systems [13, 35–44]. However, we found a number of shortcomings in the available programs, which are based on a too focussed view in the underlying theory. The most striking problem for $\mathrm{p}K_a$ calculations with Multiflex is, that each site can only be protonated or not. This binary description works surprisingly well for most acidic and basic amino acids, *e.g.,* placing the bound proton in the middle between the carboxylate oxygens, however, the limitation becomes obvious for the amino acid histidine, for which the two tautomers can not be well described by a single structure. The "histidine titration problem" could be solved specifically by some external helper program, yet it remained for all other sites, which should be better described by a number of protonated forms. Another problem was, that not only protonation of sites was of interest to study, but also the reduction of sites, especially when reduction and protonation were coupled. Multiflex was never intended to be used for studies with different ligand types, *i.e.,* protons and electrons, and therefore such studies were not straight forward.

In the last years it became more and more obvious, that the completely static approximation of protein structures, only including some flexibility implicitly by the choice of the dielectric constant, is not fully satisfactioning. In some proteins the protonation probability of important residues could be adjusted by the choice of a certain hydrogen bond network, *i.e.,* rotating protons of hydroxyl groups appropriately. For Multiflex the hydrogen placement step is required previous to the calculation of the protonation probability introducing a significant bias [45]. Also the orientation of histidine rings and asparagine and glutamine head groups were found to be important to include properly since they can often not be assigned with certainty by x-ray crystallography [46, 47]. Others have found, that the best agreement with experimental $\mathrm{p}K_a$ values in some small proteins could be obtained, when setting the dielectric constant of the protein as high as 20 [48]. Including a discrete set of sidechain rotamers, however, even better agreement could be obtained using a realistic dielectric constant like 4 [33].

With the increasing computational power available, it is feasible to study larger systems by PB calculations. While in the beginning the methods were mainly tested on small proteins like lysozyme or BPTI [27, 32, 33, 35, 46, 48–52], nowadays the aim is to study big proteins, *e.g.,* involved in photosynthesis [53–57] or the respiratory chain [58–60]. Unlike for small proteins, many cofactors and metal centers need to be included in the calculations, which before must be parameterized by quantum chemical calculations. Multiflex is not able to include such sites in a physically consistent way and the "histidine titration problem" became even more pressing for many such sites. Not only more than one proton can bind to many cofactors, but also different ligand types and different rotameric forms need to be considered.

It was found, that it is time to remove all these problems and limitations by a theory, which is general enough to handle all present and forseeable future applications. Therefore, in this thesis a theoretical model should be build, which can handle sites binding any number of

ligands. The ligands can be of the same type or of any number of different types. It should be possible to describe sites by any number of rotamer forms, allowing *e.g.,* hydrogen rotamers of hydroxyl groups or different rotamers of a complete sidechain. Any number of ligand types should be possible to include in a calculation, allowing not only to study protonation and reduction reactions, but also the binding of ions or other ligand molecules. It should be possible to study sites parameterized by quantum chemical calculations, in combination with sites parameterized by experimental data on model compounds.

The generalized theory should be implemented into a program, which does not impose unnecessary constraints and the computational effort should remain reasonable even for large systems. The new program should be tested and applied to a number of systems of scientific interest.

1.4 Outline of the Thesis

Chapter 1 aims to position the field of computational biochemistry in the scientific scenery, first, and then briefly reviews the strength and weaknesses of the most important methods. Next some of the available programs for ligand binding studies by PB electrostatics are mentioned. Discussing their limitations leads to the main aim of the work, developing a theory and appropriate programs to remove the encountered technical problems for future studies.

Chapter 2 gives an introduction to the physical chemistry of ligand binding. The linearized Poisson-Boltzmann equation is derived, which is the basis of the continuum electrostatics calculations in this work, and it is outlined how it can be solved numerically. Also the physical background of quantum chemistry, in particular DFT, and classical methods as molecular mechanics is given.

Chapter 3 gives a statistical mechanical description of ligand binding. A general energy function is described, which is specified in detail for sites parameterized by quantum chemical calculations and sites parameterized by experiments on model compounds. Finally, it is attempted to compare the new approach with those used in earlier works. Describing the generalized theory, this chapter forms the core of the work.

Chapter 4 concentrates on the informatics aspects. After an introductory section, algorithms are discussed, which were developed aside from the main project. They are described for two purposes: First they give the reader a pleasant access to algorithmic thinking and secondly they turned out to be valueable for programs building up on my main work. In particular, the adaptive mesh refinement algorithm is considered as an important conceptional approach, which will help to limit the computational cost related to studies with many ligand types. The program QMPB is described, which implements the core of the theory described in the previous chapter. Since QMPB concentrates on the energy calculations, it depends on extensive input preparation by an additional program. For this purpose, Perl Molecule was written, which is also described in this chapter.

Chapter 5 demonstrates the abilities of the generalized theory and its implementation in QMPB and Perl Molecule on a number of examples. First, it is tested, if the new set of programs is able to reproduce results on lysozyme obtained with the previously used

programs. Then a number of projects are described, which were mostly done in collaboration with other group members, but also external partners. The effects of protonation and phosphorylation on the regulatory protein HPr were studied. The investigations on factors influencing the reduction potential of ferredoxin were driving the developments and many features were tested on this system. The application to the Cu_B center of Cytochrome *c* oxidase is an example for a site with many microstates. The survey of quantum chemical methods for calculating solvation energies of quinones highlights problems, which can occur in parameterizing sites theoretically. Finally, the long range proton transfer through gramicidin A is an example for kinetic studies, which can build on the thermodynamic data calculated based on the approach described in this work.

Chapter 6 draws the conclusions of my work and points to present and future applications.

The reference section of the thesis contains a nomenclature to explain the meaning of terms as they are used in the text and the Perl Molecule ontology. However, no general definitions are attempted. A collection of abbreviations and acronyms as well as a list of equation symbols are given. A bibliography closes this reference part of the work. Appendix A documents the file formats used by the described programs and appendix B includes the manual pages to the written programs.

CHAPTER 2

CONCEPTS AND THEORETICAL METHODS FOR CALCULATIONS OF LIGAND BINDING ENERGETICS

In this work, methods and programs for calculating energies of ligand binding reactions are presented. This chapter gives a survey of the physico-chemical concepts and theoretical methods used in this work.

Section 2.1 gives an introduction into the thermodynamic description of chemical reactions. For many binding reactions, electrostatic energies are the most important contribution to the microstate energy. Electrostatic energies are calculated using continuum electrostatic methods, which are introduced in section 2.2. To introduce the underlying physical picture, I derive the Linearized Poisson-Boltzmann Equation (LPBE) from the first Maxwell equation. A numerical method used for solving the LPBE is discussed.

The continuum electrostatic approach allows to calculate the energy for transferring a set of atoms from one dielectric environment into another one. In spite of being very powerful for the calculations described here, the continuum electrostatic computations rely on parameterization. In part, parameters can be obtained from experiment, *e.g.,* measurements of binding constants of model reactions and structure determination by x-ray crystallography or NMR spectroscopy. In part, parameters have to be calculated by quantum chemistry or classical mechanics, *e.g.,* energies of formation and vibration, partial charges or conformational and rotameric energies. Therefore, these two theoretical approaches are introduced in section 2.3 and section 2.4, respectively.

For many quantum chemical applications Density Functional Theory (DFT), was found to be superior to many *ab initio* methods in both accuracy and computation time. Therefore, DFT is introduced in section 2.3.1. It is primarily used for calculating energies of formation, vibrational energies and electrostatic potentials for fitting partial charges. The energetics of model reactions in vacuum and rotamer energies can be derived by quantum mechanical (QM) calculations.

Also for larger structural changes, conformational and rotameric energies can be calculated by molecular mechanics (MM) force fields (section 2.4.1). Having a good force field parameterization, calculation of energies by MM is computationally less demanding and not necessarily

less accurate than calculation of energies by QM. However, the simulation of chemical reactions is not possible with standard MM methods.

For specific tasks even coarser methods can be used. The sidechains of proteins were found to adopt specific rotamers most of the time. Therefore, additional sidechain rotamers can be generated by using rotamer databases (section 2.4.2) instead of running molecular dynamics (MD) simulations based on a MM force field. Another common task is to place hydrogen atoms, which are usually missing in structures obtained from x-ray crystallography. Most hydrogen positions are well defined based on geometric criteria and others can be placed taking possible hydrogen bonds into account (section 2.4.3).

2.1 Concepts of Ligand Binding Reactions

In this section equilibrium constants (section 2.1.1) and chemical potentials (section 2.1.2) are introduced in general. Emphasis is put on the difference of microscopic properties (based on a microstate description used in this work) and macroscopic properties usually obtained from experiment. The difference between (standard) reaction free energy and (standard) binding free energy is pointed out (section 2.1.3). The cases of proton and electron binding (section 2.1.4 and section 2.1.5, respectively) are discussed in detail, because of their fundamental importance in biochemistry.

2.1.1 Microscopic and Macroscopic Equilibrium Constants

In this work, chemical reactions like

$$M + \nu_\lambda \lambda \rightleftharpoons M\lambda_{\nu_\lambda}. \tag{2.1}$$

are analyzed. A molecule M reacts with ν_λ ligand molecules of type λ to form a complex $M\lambda_{\nu_\lambda}$. M and λ are reactants and $M\lambda_{\nu_\lambda}$ is the product of the reaction. ν_λ is the stoichiometric factor. It is not necessary to distinguish reactions forming a covalent bond between the molecule and the ligand and binding reactions, where the complex is formed by a non-covalent interaction.

If the reaction eq. 2.1 is in thermodynamic equilibrium, it can be described by an equilibrium constant K:

$$K = \frac{[M\lambda_{\nu_\lambda}]}{[M][\lambda]^{\nu_\lambda}} \tag{2.2}$$

Here $[M]$ and $[\lambda]$ denote the activity of the reactants and $[M\lambda_{\nu_\lambda}]$ denotes the activity of the product. Only activities instead of concentrations will be used in this work. The activity $[\Lambda] = \frac{\gamma_\Lambda c_\Lambda}{{c_\Lambda}^\circ}$ of chemical species Λ (of molecule M as well as any ligand type λ) is given by the concentration c_Λ times the activity coefficient γ_Λ relative to the standard concentration ${c_\Lambda}^\circ$. ${c_\Lambda}^\circ$ has the same units as c_Λ, so that $[\Lambda]$ is a unit-less quantity. Also the equilibrium constant K is unit-less.

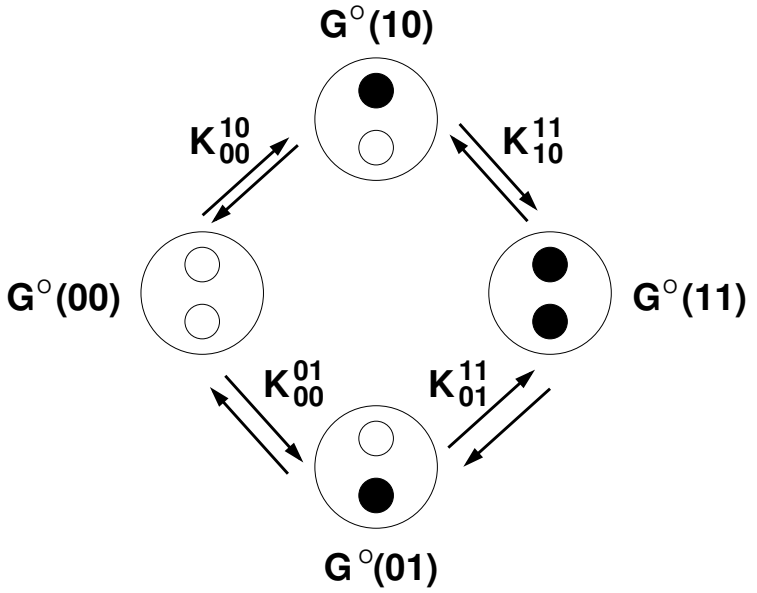


Figure 2.1. Schematic representation of the equilibria between different microstates of the system. In this example, a molecule has two distinct, interacting binding sites for a ligand. Each site can be in an unbound form (empty circle) or a bound form (black circle), which results into four microstates of the system. At standard conditions the system is fully described by four standard energies ($G^\circ(00)$, $G^\circ(10)$, $G^\circ(01)$ and $G^\circ(11)$) or four microscopic equilibrium constants (K^{10}_{00}, K^{01}_{00}, K^{11}_{10} and K^{11}_{01}). Many experimental techniques, however, would only measure two macroscopic equilibrium constants, *i.e.,* for the binding of the first and the second ligand molecule.
If the sites would not interact, the equilibrium constants the K^{10}_{00} and K^{11}_{01} as well as K^{01}_{00} and K^{11}_{10} would be the same, because the binding energy to the first site is independent of the form of the second site.

The free energy of the reaction ΔG depends on the standard reaction free energy ΔG° and the activity of reactants and products:

$$\Delta G = \Delta G^\circ + RT \ln \frac{[M\lambda_{\nu_\lambda}]}{[M][\lambda]^{\nu_\lambda}} \tag{2.3}$$

R is the universal gas constant and T is the absolute temperature.

In equilibrium ($\Delta G = 0$), the standard reaction free energy ΔG° can be directly obtained from the equilibrium constant K:

$$\Delta G^\circ = -RT \ln K \tag{2.4}$$

The equations above are general for a macroscopic system. For a molecule M with more than one ligand binding site (for the same or different types of ligands λ), the stoichiometry of the product $M\lambda_{\nu_\lambda}$ usually does not describe an unique microstate. Fig. 2.1 shows the stepwise binding of ligands to a molecule with two interacting ligand binding sites. In the fully unbound form $\vec{x}_1$ = (00), both binding sites are empty (empty circles). If a ligand (black circle) binds, it can bind in two distinct, tautomeric forms $\vec{x}_2$ = (10) and $\vec{x}_3$ = (01). The fully bound form $\vec{x}_4$ = (11) has both binding sites filled with ligand molecules. At standard conditions, *i.e.,* the

ligands having an activity of one, each of the four microstates has a standard energy ($G^\circ(00)$, $G^\circ(10)$, $G^\circ(01)$ and $G^\circ(11)$, respectively). $\vec{x}$ is the state vector of the respective microstate, which will be introduced formally in section 3.1.1. Equilibrium constants can be defined for each of the four microscopic reactions:

$$\begin{aligned} K_{00}^{10} &= \frac{[(10)]}{[(00)][\lambda]} \qquad K_{10}^{11} = \frac{[(11)]}{[(10)][\lambda]} \\ K_{00}^{01} &= \frac{[(01)]}{[(00)][\lambda]} \qquad K_{01}^{11} = \frac{[(11)]}{[(01)][\lambda]} \end{aligned} \tag{2.5}$$

or generally

$$K_{\lozenge}^{\blacklozenge} = \frac{[\blacklozenge]}{[\lozenge][\lambda]}. \tag{2.6}$$

Here $K_{\lozenge}^{\blacklozenge}$ is a particular microscopic equilibrium constant, were $\lozenge$ is a particular reactant microstate and $\blacklozenge$ the product microstate of a molecule M. The notation $[\lozenge]$, $[\blacklozenge]$ and $[\lambda]$ refer to the activity of the reactant state, the product state and the ligand λ. The standard reaction free energies of the microscopic reactions are:

$$\begin{aligned} \Delta G_{(00)}^{(10)\circ} &= G^\circ(10) - G^\circ(00) = -RT \ln K_{00}^{10} \\ \Delta G_{(10)}^{(11)\circ} &= G^\circ(11) - G^\circ(10) = -RT \ln K_{10}^{11} \\ \Delta G_{(00)}^{(01)\circ} &= G^\circ(01) - G^\circ(00) = -RT \ln K_{00}^{01} \\ \Delta G_{(01)}^{(11)\circ} &= G^\circ(11) - G^\circ(01) = -RT \ln K_{01}^{11} \end{aligned} \tag{2.7}$$

or generally

$$\Delta G_{\lozenge}^{\blacklozenge\circ} = G^\circ(\blacklozenge) - G^\circ(\lozenge) = -RT \ln K_{\lozenge}^{\blacklozenge}. \tag{2.8}$$

Selecting one of the four microstates as reference state and setting its energy to a fixed value (*e.g.,* $G^\circ(00) = 0$), the remaining three microscopic reaction free energies can be determined from the given microscopic equilibrium constants. The energies of tautomeric microstates (*e.g.,* $G^\circ(10)$ and $G^\circ(01)$) are usually different, except *e.g.,* if the tautomers can be interconverted by a symmetry operation.

If the binding of ligand λ is measured by techniques like potentiometry of calorimetry, two macroscopic equilibrium constants $\bar{K}_1$ and $\bar{K}_2$ would be obtained. The macroscopic equilibrium constants can be expressed in terms of microscopic equilibrium constants or, using eq. 2.7 and $\beta = \frac{1}{RT}$, in terms of microscopic standard free energies:

$$\bar{K}_1 = \frac{[(10)] + [(01)]}{[00][\lambda]} = K_{00}^{10} + K_{00}^{01} = \frac{e^{-\beta G^\circ(10)} + e^{-\beta G^\circ(01)}}{e^{-\beta G^\circ(00)}} \tag{2.9}$$

$$\bar{K}_2 = \frac{[(11)]}{[(10)][\lambda] + [(01)][\lambda]} = \frac{K_{10}^{11} K_{01}^{11}}{K_{10}^{11} + K_{01}^{11}} = \frac{e^{-\beta G^\circ(11)}}{e^{-\beta G^\circ(10)} + e^{-\beta G^\circ(01)}} \tag{2.10}$$

Macroscopic constants describe the equilibrium between the macrostate with (N_λ-1) and N_λ ligands bound, not the equilibrium for individual sites or between microstates of the molecule.

It is obvious, that the two macroscopic equilibrium constants are not sufficient, to describe the microscopic energetics of a system of two interacting binding sites, which has four microstates. However, for a system with one binding site, microscopic and macroscopic equilibrium constant coincide and the energetics can be fully described by a macroscopic equilibrium constant.

The formalism introduced here, can easily be extended to any number of sites. However, it can be shown, that the number of parameters, which can be extracted from the titration[1] curves of all $N_{\mathrm{site},i}$ individual sites (in conformer i) is ${N_{\mathrm{site},i}}^2 - N_{\mathrm{site},i} + 1$, but the number of independent microscopic constants is $2^{N_{\mathrm{site},i}} - 1$ [61]. However the system has $2^{N_{\mathrm{site},i}}$ microstates, assuming only two forms (bound or unbound) *per* site. For $N_{\mathrm{site},i} > 3$ sites, it follows that the energetics of the system can not be measured anymore, even not by methods monitoring the binding of ligands to individual sites (as NMR or IR). Macromolecules of biological interest usually have many more sites and the description of a binding site with only two forms is not sufficient in many cases. The computational approach, described in this work, calculates microstate energies (section 3.1.2), from which microscopic and macroscopic equilibrium constants can be obtained (section 3.1.3).

2.1.2 Chemical Potential and the Progress of a Chemical Reaction

In cases of biological interest, reactions typically occur in a mixed solvent, usually an aqueous electrolyte. Each chemical species Λ (including biomolecules M and N_{ligand} ligand types λ) has a chemical potential μ_Λ. At equilibrium conditions reaction 2.1 can be written as

$$\mu_M + \nu_\lambda \mu_\lambda = \mu_{M\lambda_{\nu_\lambda}} \tag{2.11}$$

The chemical potential μ_Λ can be calculated from the standard chemical potential μ°_Λ and the activity $[\Lambda]$:

$$\mu_\Lambda = \mu^\circ_\Lambda + RT \ln[\Lambda] \tag{2.12}$$

The standard chemical potential μ°_Λ is the chemical potential at standard conditions, *i.e.*, activity $[\Lambda] = 1$. The chemical potential is related to the Gibbs free energy of the system as can be seen from the total differential:

$$dG = VdP - SdT + \sum_\Lambda \mu_\Lambda dn_\Lambda \tag{2.13}$$

[1] Titration is a procedure to determine the amount of some unknown substance by quantitative reaction with a measured volume of a solution of precisely known concentration. Usually, a known number of ligand molecules λ is added to a known number of molecules M (the number of molecules is usually known as product of volume and concentration of solutions of M and λ). The population of ligand molecules in the bulk solvent (macroscopic techniques as calorimetry or potentiometry) or in a certain binding site (microscopic techniques as NMR or IR) is measured as function of the logarithm of ligand concentration. For systems with a single binding site, these curves are sigmoidal and the inflection point can be used to determine the equilibrium constant and standard free energy of binding. For systems with more than one binding site, these curves can be non-monotonic and the inflection points can not be associated with individual physical binding sites.

At constant temperature T and pressure P the change in Gibbs free energy ∂G with changing number of particles ∂n_Λ is the chemical potential:

$$\mu_\Lambda = \left(\frac{\partial G}{\partial n_\Lambda}\right)_{T,P,n_{\iota\neq\Lambda}} \tag{2.14}$$

During chemical reactions reactants are consumed and products are formed leading to a changing number of particles n_Λ. The change in energy is proportional to the stoichiometric factor ν_Λ of component Λ. The number of particles n_Λ and the stoichiometric factor ν_Λ are related by the progress variable (or extent of reaction) ξ: $dn_\Lambda = \nu_\Lambda d\xi$. The reactant state ($\xi = 0$) is marked by $\lozenge$ and the product state ($\xi = 1$) by $\blacklozenge$. The total differential, eq. 2.13, can be written for the reactant and product state at constant temperature and pressure as:

$$dG^\lozenge = -\sum_\Lambda \mu_\Lambda^\lozenge(\nu_\Lambda^\lozenge d\xi) \tag{2.15}$$

$$dG^\blacklozenge = \sum_\Lambda \mu_\Lambda^\blacklozenge(\nu_\Lambda^\blacklozenge d\xi) \tag{2.16}$$

The negative sign in the reactant state is due to the fact, that $dG^\lozenge$ decreases, if the reaction progresses and $d\xi$ increases. The free energy during the reaction can be described as sum of the energies of the reactant state and the product state as function of the common progress variable ξ:

$$\Delta G_{\text{reac}} = \frac{dG^\blacklozenge + dG^\lozenge}{d\xi} = \sum_\Lambda \nu_\Lambda^\blacklozenge \mu_\Lambda^\blacklozenge - \sum_\Lambda \nu_\Lambda^\lozenge \mu_\Lambda^\lozenge \tag{2.17}$$

Substituting with eq. 2.12, the reaction free energy can be written as:

$$\begin{aligned}\Delta G_{\text{reac}} &= \sum_\Lambda \nu_\Lambda^\blacklozenge \mu_\Lambda^{\circ\blacklozenge} - \sum_\Lambda \nu_\Lambda^\lozenge \mu_\Lambda^{\circ\lozenge} + RT\sum_\Lambda \nu_\Lambda^\blacklozenge \ln[\Lambda]^\blacklozenge - RT\sum_\Lambda \nu_\Lambda^\lozenge \ln[\Lambda]^\lozenge \\ &= \sum_\Lambda \nu_\Lambda^\blacklozenge \mu_\Lambda^{\circ\blacklozenge} - \sum_\Lambda \nu_\Lambda^\lozenge \mu_\Lambda^{\circ\lozenge} + RT\ln\frac{\prod_\Lambda [\Lambda]^{\blacklozenge\nu_\Lambda^\blacklozenge}}{\prod_\Lambda [\Lambda]^{\lozenge\nu_\Lambda^\lozenge}}\end{aligned} \tag{2.18}$$

Since the standard free energy is

$$\Delta G^\circ = \sum_\Lambda \nu_\Lambda^\blacklozenge \mu_\Lambda^{\circ\blacklozenge} - \sum_\Lambda \nu_\Lambda^\lozenge \mu_\Lambda^{\circ\lozenge} \tag{2.19}$$

and using eq. 2.3, the equilibrium constant $K_\lozenge^\blacklozenge$ of the reaction can be written in a more general form than eq. 2.6:

$$K_\lozenge^\blacklozenge = \frac{\prod_\Lambda [\Lambda]^{\blacklozenge\nu_\Lambda^\blacklozenge}}{\prod_\Lambda [\Lambda]^{\lozenge\nu_\Lambda^\lozenge}} \tag{2.20}$$

2.1.3 Microstate Energy, Binding Free Energy and Reaction Free Energy

Generally binding reactions (as Fig. 2.1) are not only studied under standard conditions, but the chemical potential μ_λ of the ligands λ is varied to obtain titration curves. The energy of each microstate changes with changing thermodynamic variables, in particular the chemical

potential of the ligands $\{\mu_\lambda\}$. The nth microstate energy in conformation i can be written as:

$$G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\}) = G^\circ(\vec{x}_{\mathrm{i,n}}) + \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda(\vec{x}_{\mathrm{i,n}})\mu_\lambda \tag{2.21}$$

The microstate energy $G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})$ is identical to the standard free energy $G^\circ(\vec{x}_{\mathrm{i,n}})$ at standard conditions, *i.e.,* when the activity $[\lambda] = 1$ and $\mu_\lambda = 0$. At non-standard conditions each ligand type λ contributes with its chemical potential μ_λ times the stoichiometric factor $\nu_\lambda(\vec{x}_{\mathrm{i,n}})$ of free ligands λ to the energy. The stoichiometric factor can be chosen freely for the reference microstate and has to be given relative to that value for each other microstate.

The binding free energy is the difference in microstate energy between a particular unbound microstate and a particular bound microstate:

$$\begin{aligned}\Delta G_{\mathrm{bind}}(\vec{x}_{\mathrm{i,1}}, \vec{x}_{\mathrm{i,2}}, \{\mu_\lambda\}) &= G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,2}}, \{\mu_\lambda\}) - G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,1}}, \{\mu_\lambda\}) \\ &= G^\circ(\vec{x}_{\mathrm{i,2}}) - G^\circ(\vec{x}_{\mathrm{i,1}}) + \sum_{\lambda}^{N_{\mathrm{ligand}}} (\nu_\lambda(\vec{x}_{\mathrm{i,2}}) - \nu_\lambda(\vec{x}_{\mathrm{i,1}}))\mu_\lambda \\ &= \Delta G^\circ(\vec{x}_{\mathrm{i,1}}, \vec{x}_{\mathrm{i,2}}) - \sum_{\lambda}^{N_{\mathrm{ligand}}} N_\lambda \mu_\lambda\end{aligned} \tag{2.22}$$

N_λ is the number of ligands bound during the reaction, *i.e.,* it is the negative of the difference in number of free ligands (stoichiometric factors).

Using these definitions (and omitting the conformer index i), the microscopic reaction of a molecule M and ligands λ can be studied. The reaction passes from an reactant form $\lozenge$ to a product form $\blacklozenge$. The reaction free energy can be calculated from eq. 2.18:

$$\begin{aligned}\Delta G_{\mathrm{reac}} =& \nu_M \mu_M^{\circ\blacklozenge} + \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda{}^{\blacklozenge} \mu_\lambda^\circ - \nu_M \mu_M^{\circ\lozenge} - \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda{}^{\lozenge} \mu_\lambda^\circ \\ &+ RT \ln \frac{[M]^{\blacklozenge\nu_M}}{[M]^{\lozenge\nu_M}} + RT \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda{}^{\blacklozenge} \ln[\lambda]^{\blacklozenge} - RT \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda{}^{\lozenge} \ln[\lambda]^{\lozenge}\end{aligned} \tag{2.23}$$

Here, each sum over all chemical species Λ (in eq. 2.18) is replaced by a term for the molecule M and a sum over all ligand types λ. The standard chemical potential μ_λ° of ligand λ is independent of the progress of the reaction ξ, but the stoichiometric factor changes due to the different number of bound ligand molecules ($N_\lambda = \nu_\lambda{}^{\lozenge} - \nu_\lambda{}^{\blacklozenge}$). Instead, the standard chemical potential of molecule M is different in reactant form $\lozenge$ and product form $\blacklozenge$, because it contains the energy of the bound ligands. The stoichiometric factor ν_M of molecules M is constant during the progress of the reaction ξ (usually ν_M is chosen to be one).

The chemical potential of the ligand λ can be expressed by eq. 2.12 for the reactant and product form. The number of ligand molecules of type λ in the bulk solvent should be large compared to the number of ligand molecules bound or released upon the reaction. Therefore, the reaction causes a negligible change in the activity of λ, so that $[\lambda]^{\blacklozenge} \approx [\lambda]^{\lozenge} \equiv [\lambda]$. Thus, the

reaction free energy in eq. 2.23 simplifies to:

$$\Delta G_{\mathrm{reac}} = \nu_M(\mu_M^{\circ\blacklozenge} - \mu_M^{\circ\lozenge}) + RT \ln \frac{[M]^{\blacklozenge\nu_M}}{[M]^{\lozenge\nu_M}} - \sum_{\lambda}^{N_{\mathrm{ligand}}} N_\lambda \mu_\lambda \tag{2.24}$$

The difference in binding free energy of molecule M equals the difference in standard free binding energy and the change in free ligands of type λ as given in eq. 2.22:

$$\begin{aligned}\Delta G_{\mathrm{bind}}(M) =& \nu_M(\mu_M^{\circ\blacklozenge} - \mu_M^{\circ\lozenge}) + \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda{}^{\blacklozenge}\mu_\lambda{}^{\blacklozenge} - \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_\lambda{}^{\lozenge}\mu_\lambda{}^{\lozenge} \\ =& \Delta G^\circ(M) - \sum_{\lambda}^{N_{\mathrm{ligand}}} N_\lambda \mu_\lambda \end{aligned} \tag{2.25}$$

Comparing eq. 2.24 and eq. 2.25, the difference between the reaction free energy ΔG_{reac} of the system and the binding free energy $\Delta G_{\mathrm{bind}}(M)$ for molecule M becomes obvious:

$$\Delta G_{\mathrm{bind}}(M) = \Delta G_{\mathrm{reac}} - RT \ln \frac{[M]^{\blacklozenge\nu_M}}{[M]^{\lozenge\nu_M}} \tag{2.26}$$

The difference is due to different equilibrium conditions: The binding reaction is in equilibrium, $\Delta G_{\mathrm{bind}}(M) = 0$, if $\Delta G_{\mathrm{reac}} = RT \ln \frac{[M]^{\blacklozenge\nu_M}}{[M]^{\lozenge\nu_M}}$, which does not require ΔG_{reac} to be zero. This equation is used for titration calculations, *i.e.,* to study ligand binding to molecule M as function of the bulk concentration (or chemical potential) of all ligand types λ. The application section will focus on proton and electron binding reactions (*i.e.,* calculation of protonation and reduction probabilities). Therefore, these two ligands are discussed in detail in the next sections. However, the theory and programs presented here are also suited to study, *e.g.,* water binding or binding of biological important ions like Na^+, K^+ or Ca^{++}. The study of complex (organic) molecules as ligands (*e.g.,* drug binding to receptors) may be hampered by internal degrees of freedom of the ligand or the surface area dependent non-electrostatic part of the solvation energy, which are difficult to include.

2.1.4 The Protonation Reaction

For describing the chemical potential of protons two quantities are defined, the pH and the $\mathrm{p}K_a$ value. The pH is the negative decadic logarithm of the activity of protons in solution $[H^+]$:

$$\mathrm{pH} \equiv -\lg[H^+] \tag{2.27}$$

The $\mathrm{p}K_a$ value is defined as the negative decadic logarithm of the equilibrium constant

$$\mathrm{p}K_a \equiv -\lg K_a \tag{2.28}$$

of the deprotonation (dissociation) reaction of a protonated acid

$$HA \overset{K_a}{\rightleftharpoons} A^- + H^+ \tag{2.29}$$

the equilibrium constant is given by

$$K_a = \frac{[A^-][H^+]}{[HA]} \tag{2.30}$$

Combining the four equations for a single titrateable group leads to the Henderson-Hasselbalch equation:

$$\mathrm{pH} = \mathrm{p}K_a + \lg \frac{[A^-]}{[HA]} \tag{2.31}$$

Since all equations in this work are based on binding reactions, eq. 2.29 and eq. 2.30 have to be rewritten for the protonation of an acid, described by the equilibrium constant K_b:

$$A^- + H^+ \overset{K_b}{\rightleftharpoons} HA; \qquad K_b = \frac{[HA]}{[A^-][H^+]} = \frac{1}{K_a} \tag{2.32}$$

From the above definition of the $\mathrm{p}K_a$ value (and $\log_b x = \log_b a \cdot \log_a x$), the standard binding free energy (eq. 2.4) of protonating a group A^- is:

$$\Delta G^\circ(AH - A^-) = -RT \ln K_b = RT \ln K_a = -RT \ln 10 \ \mathrm{p}K_a \tag{2.33}$$

The chemical potential relative to standard chemical conditions ($[H^+]^\circ = 1$; $\mathrm{pH}^\circ = 0$; $\mu^\circ_{H^+} = RT \ln [H^+]^\circ = RT \ln 1 = 0$), can be obtained from eq. 2.12:

$$\mu_{\mathrm{H}^+} = \mu^\circ_{H^+} + RT \ln [H^+] = -RT \ln 10 \ \mathrm{pH} \tag{2.34}$$

If the reaction in eq. 2.1 is the binding of single proton to an acid, eq. 2.25 can be written as

$$\Delta G(AH - A^-) = \Delta G^\circ(AH - A^-) - \mu_{\mathrm{H}^+} = RT \ln 10(\mathrm{pH} - \mathrm{p}K_a) \tag{2.35}$$

The standard binding free energy $\Delta G^\circ(AH - A^-)$ can be expressed as a $\mathrm{p}K_a$ value (eq. 2.33). The chemical potential of protons μ_{H^+} is described by the pH (eq. 2.34).

2.1.5 The Reduction Reaction

In a redox reaction electrons are transferred from an electron donor B_{red} to an electron acceptor A^+_{ox}:

$$A^+_{\mathrm{ox}} + B_{\mathrm{red}} \overset{K_{\mathrm{redox}}}{\rightleftharpoons} A_{\mathrm{red}} + B^+_{\mathrm{ox}}; \qquad K_{\mathrm{redox}} = \frac{[A_{\mathrm{red}}][B^+_{\mathrm{ox}}]}{[A^+_{\mathrm{ox}}][B_{\mathrm{red}}]} \tag{2.36}$$

The reaction can be divided into two half-reactions, such as

$$A^+_{\mathrm{ox}} + e^- \rightleftharpoons A_{\mathrm{red}} \tag{2.37}$$

$$B_{\mathrm{red}} \rightleftharpoons B^+_{\mathrm{ox}} + e^- \tag{2.38}$$

One half-reaction oxidizes B_{red} and gains an electron, the other half-reaction reduces A^+_{ox} using an electron. Each half reaction can be physically separated as to form an electrochemical

cell, which allows in experiment to replace the redox partner of a protein by an electrode system, by convention the standard hydrogen electrode (or an electrode system which is calibrated relative to the standard hydrogen electrode). The standard hydrogen electrode (SHE) does the oxidation half-reaction

$$H_2(g) \rightleftharpoons 2H^+ + 2e^-, \tag{2.39}$$

in which H^+ in solution at $\mathrm{pH} = 0$, $25°C$, and 1 atm is in equilibrium with H_2 gas that is in contact with a platinized platin electrode.

The free energy in an electrochemical cell reaction is related to the electrical work done on the system $\Delta G(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = W_{\mathrm{elec}}$. The electrochemical cell does electrical work $-W_{\mathrm{elec}}$, which is equal to the product of the charge transferred reversible ($\mathcal{F}$) times the potential difference $\mathcal{E}$ of the electrodes:

$$\Delta G(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = -\mathcal{F}\Delta\mathcal{E} \text{ and } \Delta G°(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = -\mathcal{F}\Delta\mathcal{E}° \tag{2.40}$$

Here $\mathcal{F}$ is the Faraday constant, the electrical charge of 1 mol electrons ($1\mathcal{F} = 96485\frac{\mathrm{C}}{\mathrm{mol}} = 96.485\frac{\mathrm{kJ}}{\mathrm{mol}\cdot\mathrm{V}} = 23.045\frac{\mathrm{kcal}}{\mathrm{mol}\cdot\mathrm{V}}$);

The free energy of the reduction half-reaction, eq. 2.37, is according to eq. 2.3:

$$\Delta G(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = \Delta G°(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) + RT \ln \frac{[A_{\mathrm{red}}]}{[A^+_{\mathrm{ox}}][e^-]} \tag{2.41}$$

$$= \Delta G°(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) + RT \ln K_{\mathrm{red}} \tag{2.42}$$

With eq. 2.40 the Nernst equation for eq. 2.37 is obtained:

$$\Delta\mathcal{E} = \Delta\mathcal{E}° - \frac{RT}{\mathcal{F}} \ln K_{\mathrm{red}} \tag{2.43}$$

The reduction potential $\mathcal{E}$ is proportional to the activity of electrons. It should be measured under equilibrium conditions, *i.e.*, $\Delta G(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = 0$ and therefore (eq. 2.40) $\Delta\mathcal{E} = 0$.

$$\mathcal{E} = -\frac{RT}{\mathcal{F}} \ln[e^-] = \mathcal{E}° - \frac{RT}{\mathcal{F}} \ln \frac{[A_{\mathrm{red}}]}{[A^+_{\mathrm{ox}}]} \tag{2.44}$$

The reduction potential $\mathcal{E}$ is proportional to the negative natural logarithm of the activity of electrons in solution $[e^-]$ in analogy to the pH.

The standard reduction potential $\mathcal{E}°$ of the standard hydrogen electrode (SHE) is $\mathcal{E}°_{\mathrm{SHE}} = -\frac{RT}{\mathcal{F}} \ln[e^-]° = 0V$. The standard reduction potential is usually given relative to the SHE: $\Delta\mathcal{E}° = \mathcal{E}° - \mathcal{E}°_{\mathrm{SHE}} = \mathcal{E}°$. From eq. 2.40 and eq. 2.41 results:

$$\mathcal{E}° = -\frac{1}{\mathcal{F}}\Delta G°(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = \frac{RT}{\mathcal{F}} \ln K_{\mathrm{red}} \tag{2.45}$$

For a molecule with a single redox active site, the midpoint potential $\mathcal{E}°$ is proportional to the natural logarithm of the equilibrium constant for reducing the molecule. The midpoint potential is analogous to the $\mathrm{p}K_a$ value, which is the decadic logarithm of the equilibrium constant for protonating the molecule.

The change in chemical potential relative to standard chemical conditions ($[e^-]^\circ = 1$; $\mathrm{pH}^\circ = 0$; $\mu^\circ_{e^-} = 0$), can be obtained from eq. 2.12 to be proportional to the activity of electrons. Rewriting eq. 2.44, one sees that the chemical potential and the reduction potential are proportional as well:

$$\mu_{\mathrm{e}^-} = RT \ln[e^-] = -\mathcal{F}\mathcal{E} \tag{2.46}$$

If the reaction in eq. 2.1 is the binding of electrons to an oxidized molecule, eq. 2.25 can be written as

$$\Delta G(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) = \Delta G^\circ(A_{\mathrm{red}} - A^+_{\mathrm{ox}}) - \mu_{e^-} = -\mathcal{F}(\mathcal{E} - \mathcal{E}^\circ) \tag{2.47}$$

The standard reduction potential $\mathcal{E}^\circ$ is proportional to the standard binding free energy $\Delta G^\circ(A_{\mathrm{red}} - A^+_{\mathrm{ox}})$ (eq. 2.45) and the reduction potential $\mathcal{E}$ is proportional to the chemical potential of electrons (eq. 2.46).

2.2 Continuum Electrostatics

For the binding reactions studied in this work, the binding free energy is strongly dependend on the electrostatic energy of the ligand in the bulk solvent and in the binding site, since primarily charged, small ligands are studied. Electrostatic interactions are the most far reaching interactions in molecular systems, often leading to strong cooperativity or anti-cooperativity in binding of charged ligands.

To study these effects, a continuum electrostatic model is used, which gives a rather accurate description of the electrostatic interactions. This model partitions space into dielectric regions and assigns dielectric constants to them. It can be combined with an atomic picture (section 3.2), in the region of interest, *i.e.,* in the biomolecule. However, the bulk solvent can be treated as a continuum, leading to dramatic savings in the computational cost to determine the energy of microstates compared to other methods, which treat the solvent explicitly. The description of a system in the grand canonical ensemble (section 3.1.1) works seamlessly, while it can be problematic for other methods like MD.

This section derives the Linearized Poisson-Boltzmann Equation (LPBE), which is used in the computations, from the first Maxwell equation. The derivation is thought to be useful to understand the physical picture and the approximations included in the model. It is also discussed how the LPBE can be solved computationally. Here, SI units are used instead of the e.s.u.-c.g.s units commonly found in electrostatic literature.

2.2.1 The First Maxwell Equation

All continuum electrostatic equations can be derived from the first Maxwell equation (Gauss's law) under certain boundary conditions (*e.g.,* [62]).

Gauss's law in differential form is:

$$\vec{\nabla} \cdot \vec{D} = \rho \tag{2.48}$$

The divergence $\vec{\nabla}$ in cartesian coordinates at a point $\vec{r} = (r_x, r_y, r_z)$ in an Euclidean space is $\vec{\nabla}\vec{r} = \frac{\partial r_x}{\partial x} + \frac{\partial r_y}{\partial y} + \frac{\partial r_z}{\partial z}$. The divergence of the electric displacement field (electric flux density) $\vec{D}$ equals the free electric charge density ρ. This equation can be read: The source of an electric field is a charge.

Gauss's law in integral form is:

$$\phi = \oint_{\mathcal{S}} \vec{D} d\vec{A} = \int_V \rho dV \tag{2.49}$$

The electric displacement field $\vec{D}$ integrated over a surface $\mathcal{S}$ equals the free electric charge density ρ inside the volume V enclosed by the surface $\mathcal{S}$. $d\vec{A} = \vec{n}dA$ are infinitesimal surface area elements described by the surface normal vector $\vec{n}$ and planar elements of area A approximating the surface $\mathcal{S}$. The total charge in a volume V is determined by the net flow of electric flux ϕ across the surface $\mathcal{S}$.

In linear isotropic materials[2], the displacement field $\vec{D}$ is the sum of electric field $\vec{E}$ and polarization density $\vec{P} = \chi_e \varepsilon_0 \vec{E}$:

$$\vec{D} = \varepsilon_0 \vec{E} + \vec{P} = (1 + \chi_e)\varepsilon_0 \vec{E} = \varepsilon \vec{E} \tag{2.50}$$

χ_e is the electrical susceptibility of the material, ε_0 is the electrical permittivity of vacuum and ε is the electrical permittivity of the material. The permittivity (ε, ε_0) has units of farad *per* meter (F/m) or coulomb squared *per* joule meter (C^2/Jm). In contrast the relative permittivity or dielectric constant $\varepsilon_\mathrm{r} = \frac{\varepsilon}{\varepsilon_0}$ is unit-less. In non-dispersive, isotropic media, $\varepsilon = \varepsilon_0 \varepsilon_\mathrm{r}$ is a time-independent scalar, simplifying eq. 2.48 to:

$$\vec{\nabla} \cdot \varepsilon \vec{E} = \rho \tag{2.51}$$

The electrostatic potential $\varphi(\vec{r})$ at a point is the gradient of the electric field that occurs when the magnetic field is time invariant ($\vec{\nabla} \times \vec{E} = 0$):

$$\vec{E} = -\nabla \varphi(\vec{r}) \tag{2.52}$$

2.2.2 The Coulomb Equation

The description of charges by charge distributions ρ is very general. Originating from the 18^{th} and 19^{th} century, when the existence of atoms was still under discussion, it is also today a good description of electron density inside matter. However, for many applications a description of charges as point charges q is conceptionally easier, *i.e.,* it is closer to the classical chemical view of atoms as charged particles building up molecules. To merge both descriptions, point charges q can be seen as Gaussian charge distributions ρ with a volume V small compared to the system of interest and the integral over the volume of the charge distribution

[2]Linear materials are those which show a proportionality between $\vec{D}$ and $\vec{E}$. In isotropic media the electrical permittivity ε is a scalar constant. In anisotropic media it is a second rank tensor causing birefringence, which can be measured by polarimetry. Applications of this property are, *e.g.,* measurements of sugar concentrations or to modify the optical properties of polymers in liquid crystal displays by an external electric field.

equals the point charge:

$$\int_V \rho dV = q \tag{2.53}$$

The electrostatic field in distance $\vec{r}$ of a point charge q is determined by measuring the electrostatic potential $\varphi(\vec{r})$ (eq. 2.52). The dielectric medium is assumed to be infinite and homogeneous with a permittivity of ε. Due to the spherical symmetry of the electrostatic field $\vec{E}$ the strength only depends on the distance $\vec{r}$. Therefore a spherical surface A at radius $\vec{r}$ around the central point charge is used for integration. Combining eq. 2.49 and eq. 2.50, the total flux through the surface is the electrical field integrated over the surface. Since the surface vector is $d\vec{A} = \vec{n} \cdot dA$, the electrical field $\vec{E}$ can be replaced by its radial (orthogonal) component $E_r = \vec{E} \cdot \vec{n}$, which is constant and can be taken out of the integral:

$$\phi = -\oint_{\mathcal{S}} \varepsilon \vec{E} d\vec{A} = \oint_{\mathcal{S}} \varepsilon \vec{E} \cdot \vec{n} dA = \oint_{\mathcal{S}} \varepsilon E_r dA = \varepsilon E_r \oint_{\mathcal{S}} dA \tag{2.54}$$

The integral over the surface A of a sphere is $\oint_{\mathcal{S}} dA = 4\pi\vec{r}^2$. Since the only charge inside the sphere is the point charge, Gauss's law (eq. 2.49) and eq. 2.53 gives

$$E_r \cdot 4\pi\vec{r}^2 = \frac{q}{\varepsilon} \text{ and } E_r = \frac{1}{4\pi\vec{r}^2}\frac{q}{\varepsilon}. \tag{2.55}$$

Thus, the Coulomb law was derived from Gauss's law. By integrating eq. 2.55 the Coulomb potential can be obtained

$$\varphi(\vec{r}) = \frac{1}{4\pi|\vec{r}|}\frac{q}{\varepsilon} \tag{2.56}$$

The electrostatic energy W_{elec} is the work needed to bring a point charge q_2 from infinity into the distance $\vec{r}$ of another charge q_1:

$$W_{\text{elec},2} = q_2\varphi(\vec{r}) \tag{2.57}$$

Inserting the Coulomb potential eq. 2.56, the Coulomb energy is

$$W_{\text{elec},2} = \frac{q_1 q_2}{4\pi\varepsilon|\vec{r}|}. \tag{2.58}$$

The Coulomb energy is for example used in molecular mechanics force fields (section 2.4.1).

2.2.3 The Poisson Equation

The Coulomb potential eq. 2.56 is only valid in a homogeneous dielectric and for point charges. A more general equation for the electrostatic potential, Poisson's equation, can be obtained combining eq. 2.51 and eq. 2.52. It is also valid if space is partitioned in several dielectric regions. For each region there is a electrical permittivity ε, so that $\varepsilon(\vec{r})$ is a function in space returning the electrical permittivity at point $\vec{r}$.

$$\vec{\nabla}\varepsilon(\vec{r})\vec{\nabla}\varphi(\vec{r}) = -\rho \tag{2.59}$$

In vacuum, the equation simplifies to $\vec{\nabla}^2\varphi(\vec{r}) = \Delta\varphi(\vec{r}) = -\frac{\rho}{\varepsilon_0}$, in a homogeneous dielectric it simplifies to $\vec{\nabla}^2\varphi(\vec{r}) = -\frac{\rho}{\varepsilon}$. The Poisson equation is often used to include a solvent description into quantum mechanical calculations (section 2.3), *e.g.,* the Polarizable Continuum Model (PCM) [18–20].

2.2.4 The Poisson-Boltzmann Equation

The Poisson equation (eq. 2.59) assumes a fixed charge distribution. However, a biomolecule is usually solvated by an ionic solution. The mobile ions are assumed to arrange around the molecule according to a Boltzmann distribution and the electrostatic field of the molecule. Thereby the ions in turn modify the electrostatic field, the molecule is exposed to. Such a system is described by the Poisson-Boltzmann equation (PBE).

The charge distribution of the solution $\rho(\vec{r})$ consists of two parts, the fixed charges of the molecule $\rho_{\mathrm{f}}(\vec{r})$ and the mobile charges of the ions $\rho_{\mathrm{m}}(\vec{r})$

$$\rho(\vec{r}) = \rho_{\mathrm{f}}(\vec{r}) + \rho_{\mathrm{m}}(\vec{r}) \tag{2.60}$$

The charge distribution of the molecule $\rho_{\mathrm{f}}(\vec{r})$ is normally represented by point charges q_{a} at the nuclei of the atoms. The mobile charges of the ions $\rho_{\mathrm{m}}(\vec{r})$ are distributed according to Boltzmann statistics. The mean concentration $\langle c_\lambda(\vec{r})\rangle$ of ions of type λ at the position $\vec{r}$ is

$$\langle c_\lambda(\vec{r})\rangle = c_{\mathrm{bulk},\lambda}\exp\left(-\frac{\mathcal{W}_\lambda(\vec{r})}{k_B T}\right), \tag{2.61}$$

where $c_{\mathrm{bulk},\lambda}$ is the bulk concentration of the ionic species λ and $\mathcal{W}_\lambda(\vec{r})$ is the potential of mean force experienced by the ion of type λ at position $\vec{r}$. For a dilute solution, the potential of mean force is

$$\mathcal{W}_\lambda(\vec{r}) = \mathrm{z}_\lambda \mathrm{e}_0 \varphi(\vec{r}), \tag{2.62}$$

where z_λ is the charge number (or valency) of the ion of type λ and e_0 is the elementary charge. The charge density $\rho_{\mathrm{m}}(\vec{r})$ is given by a sum over the charge density of all N_{iontype} ionic species in the solution

$$\rho_{\mathrm{m}}(\vec{r}) = \sum_{\lambda}^{N_{\mathrm{iontype}}} c_{\mathrm{bulk},\lambda} N_A \mathrm{z}_\lambda \mathrm{e}_0 \exp\left(-\frac{\mathrm{z}_\lambda \mathrm{e}_0 \varphi(\vec{r})}{k_B T}\right). \tag{2.63}$$

Combining eq. 2.59, eq. 2.60 and eq. 2.63 leads to the Poisson-Boltzmann equation (PBE):

$$\vec{\nabla}\varepsilon(\vec{r})\vec{\nabla}\varphi(\vec{r}) = -\rho_{\mathrm{f}}(\vec{r}) - \sum_{\lambda}^{N_{\mathrm{iontype}}} c_{\mathrm{bulk},\lambda} N_A \mathrm{z}_\lambda \mathrm{e}_0 \exp\left(-\frac{\mathrm{z}_\lambda \mathrm{e}_0 \varphi(\vec{r})}{k_B T}\right) \tag{2.64}$$

PBE solvers exist for molecular systems (*e.g.,* as option in the program APBS [63]), however the complicated exponential dependence of electrostatic potential and charge distribution does not allow an efficient calculation of ligand binding energetics (section 3).

2.2.5 The Linearized Poisson-Boltzmann Equation

The PBE can be linearized for small electrostatic potentials, which fulfill $\frac{e_0\varphi(\vec{r})}{k_BT} < 1$. The exponential can be approximated by an expansion to the linear term:

$$\sum_{\lambda}^{N_{\text{iontype}}} c_{\text{bulk},\lambda} N_A z_\lambda e_0 \exp\left(-\frac{z_\lambda e_0 \varphi(\vec{r})}{k_BT}\right) \approx \sum_{\lambda}^{N_{\text{iontype}}} c_{\text{bulk},\lambda} N_A z_\lambda e_0 - \sum_{\lambda}^{N_{\text{iontype}}} c_{\text{bulk},\lambda} {z_\lambda}^2 \frac{N_A {e_0}^2}{k_BT}\varphi(\vec{r}) \quad (2.65)$$

The first term of the expansion vanishes because of the electroneutrality of the solution

$$\sum_{\lambda}^{N_{\text{iontype}}} c_{\text{bulk},\lambda} z_\lambda e_0 = 0. \quad (2.66)$$

Experimental conditions are usually described by a ionic strength I instead of the exact concentration of all N_{iontype} types of ions:

$$I = \frac{1}{2} \sum_{\lambda}^{N_{\text{iontype}}} c_{\text{bulk},\lambda} {z_\lambda}^2 \quad (2.67)$$

By introducing the inverse Debye length

$$\kappa(\vec{r}) = \sqrt{2I \frac{N_A {e_0}^2}{\varepsilon(\vec{r}) k_BT}}, \quad (2.68)$$

and writing $\bar{\kappa}^2(\vec{r}) = \varepsilon(\vec{r})\kappa^2(\vec{r})$ the linearized Poisson-Boltzmann equation (LPBE) is obtained:

$$\vec{\nabla}\varepsilon(\vec{r})\vec{\nabla}\varphi(\vec{r}) = -\rho_f(\vec{r}) + \bar{\kappa}^2(\vec{r})\varphi(\vec{r}) \quad (2.69)$$

2.2.6 Solving the LPBE Numerically

For a few simple geometries analytical solutions of the LPBE, eq. 2.69, exist [64, 65]. For irregular geometries, the LPBE can be solved by numerical methods. Most often, finite difference methods [26] are used for solving the PBE or LPBE, but also boundary element methods [66] or multi-grid-level based methods [22, 63, 67–69] can be used. The principle idea of finite difference methods is to replace a differential operator

$$\nabla f(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \quad (2.70)$$

by a quotient of finite differences, assigning h a fixed (non-zero) value. The approximation approaches the exact result, as h goes to zero. An equation of N independent variables is therefore mapped onto a N-dimensional lattice where the difference between two consecutive grid points is h. By finite differences every linear differential equation becomes a system of linear equations, which can be solved by numeric algorithms.

To solve the PBE or LPBE for molecular systems, the flowchart in Fig. 2.2 is followed. The description focusses on the implementation in MEAD, so different Poisson-Boltzmann solvers may vary in details.

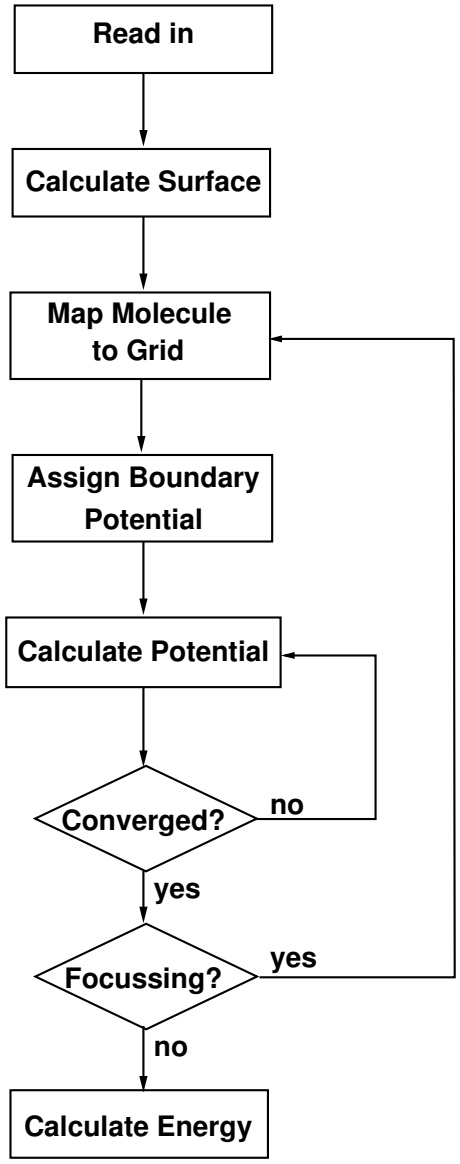


Figure 2.2. Solving the LPBE by finite difference methods. First, all parameters for the calculation are read (*e.g.,* coordinates, charges, radii, dielectric constants and grid definitions). Then the surface of the molecule is calculated as dielectric boundary. According to the boundaries, dielectric constants are mapped onto the grid. The atomic charges are interpolated onto neighboring grid points. Electrostatic potentials are assigned to the grid points based on an initial guess. It is important, that the initial potentials at the boundaries of the grid are a good guess, because they are constant during the calculation. The finite difference formulation of the LPBE (eq. 2.76) is solved by an iterative scheme, until the electrostatic potential does not change significantly anymore between two subsequent iterations. The computations require on one hand a large initial grid to minimize the error due to the approximated boundary potential, on the other hand a fine final grid is required to minimize the error due to the finite difference approximation. Both requirements can usually not be fulfilled directly due to a limited amount of memory. Therefore, consecutively smaller and finer grids are calculated using the previous grid to define the potentials at the boundaries. This method is called focussing. Finally, electrostatic energies are calculated as product of charge and electrostatic potential.

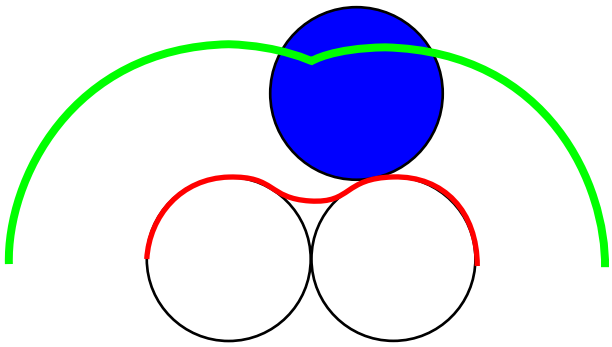


Figure 2.3. Surface calculation in MEAD. The solvent accessible surface (red line) of the two atoms (white circle) is calculated by a 'rolling ball' (blue circle). The surface of the Stern layer is shown by the green line.

Read in: The coordinates of the atoms, charges and radii of the molecule are read in (pqr-file(s)). The grid definitions are read consisting of the center of the grid (in the coordinate system of the molecule), the number of grid points *per* dimension and the grid spacing. Consecutively finer grids may be specified for focussing (see below). Dielectric constants ε_{r} for each dielectric region and other parameters of the LPBE like temperature T and ionic strength I are given as input.

Calculate Surface: Based on the coordinates and radii of atoms surfaces are calculated for each dielectric region. MEAD calculates an analytical surface description, which is independent of a particular grid definition [70]. The surface can be stored and loaded in different runs (section 4.4.1).

The boundary between the protein dielectrics and the solvent dielectrics is defined by the solvent accessible surface of the protein [71]. The solvent accessible surface is typically calculated using the 'rolling ball' algorithm using a sphere (of solvent) of a particular radius to 'probe' the surface of the molecule (Fig. 2.3). The choice of the 'probe radius' does have an effect on the observed surface area, as using a smaller probe radius detects more surface details and therefore reports a larger surface. The probe radius is an input parameter, which by default approximates the radius of a water molecule by a value of 1.4 Å.

Map Molecule to Grid: All physical properties of the system, *i.e.,* charge, electrostatic potential, electrical permittivity and ion accessability, are mapped onto a grid. The charges are mapped to the grid by an interpolation scheme, *e.g.,* the fraction of the charge q_{a} ($\vec{r}_{\mathrm{a}}$) assigned to the nearest grid point at $\vec{r}_{\mathrm{p}}$ is given by $q_{\mathrm{p}} = q_{\mathrm{a}}(1-\frac{r_{\mathrm{ax}}-r_{\mathrm{px}}}{h})(1-\frac{r_{\mathrm{ay}}-r_{\mathrm{py}}}{h})(1-\frac{r_{\mathrm{az}}-r_{\mathrm{pz}}}{h})$, where r_{ax} is the x-component of the vector $\vec{r}_{\mathrm{a}}$, *etc.* and h the grid spacing (mesh size). Analogous other neighboring grid points get the remaining fraction of the charge assigned according to their distance.

The spatial dependent electrical permittivity $\varepsilon(\vec{r}_{\mathrm{p}}) = \varepsilon_{\mathrm{p}}\varepsilon_0$ is defined on a grid, which is shifted by half a grid unit compared to the charge grid ($\varepsilon_1 ... \varepsilon_6$ in Fig. 2.4). Therefore, each analytical surface is used to assign the dielectric constant of the region, if the point is

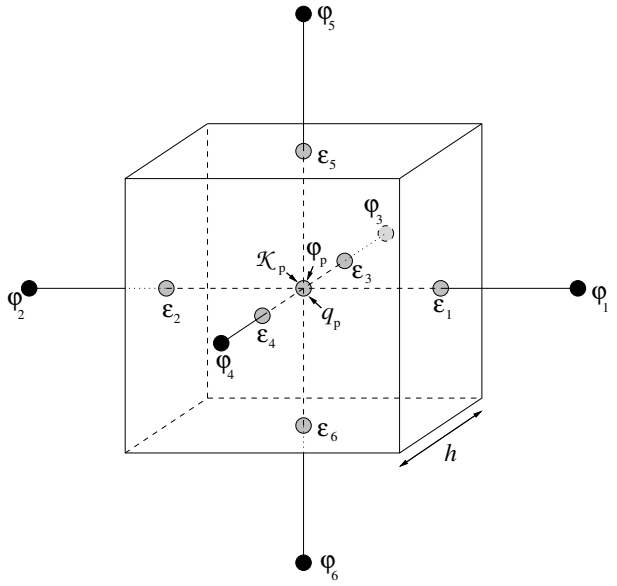


Figure 2.4. Solving the LPBE by a finite difference method. The charge distribution $\rho(\vec{r})$ is represented by point charges, which are mapped onto a grid by fractional charges (here fractional charge q_P at grid point p). The ion accessibility $\mathcal{K}_p$ decides if the modified and squared inverse Debye length $\bar{\kappa}^2(\vec{r})$ has to be calculated for the point $\vec{r}_\mathrm{P}$. The dielectric constant $\varepsilon_\mathrm{r}(\vec{r})$ is discretized on a grid, which is shifted by half a grid spacing h with respect to the grid of point charges. The electrostatic potential φ_p is calculated on the charge grid with one grid spacing distance to q_P.

inside the surface. Points, which are outside of all surfaces, get the dielectric constant of the solvent assigned.

A ion accessibility $\mathcal{K}_p$ is defined on a grid, which marks if the modified and squared inverse Debye length $\bar{\kappa}^2(\vec{r})$ needs to be calculated for the particular grid point p. The value of $\bar{\kappa}^2(\vec{r})$ changes throughout the calculation due to the changing distribution of mobile ions. However, the spacial region, in which the ions can move, is fixed during the calculation. $\mathcal{K}_p$ is zero within the molecule and within a solvent exclusion layer (Stern layer), which is by default a 2 Å thick skin around the protein. It is calculated from the atomic radii of the molecule increased by, *e.g.,* 2 Å and not from the analytical surface description. The Stern layer ensures that mobile ions keep a layer of solvent molecules, *i.e.,* that they do not desolvate comming close to the protein, which would require an additional energy contribution. In the volume elements that are partially occupied by the protein, the ion accessibility $\mathcal{K}_p$ at the center grid point is the average of the ion accessibilities $\mathcal{K}_1 \ldots \mathcal{K}_6$ of the associated grid lines. This averaging smooths the step between the area where $\bar{\kappa}^2(\vec{r})$ is calculated and where it is not calculated.

Assign Boundary Potential: In order to perform a calculation, the electrostatic potential φ_p on the grid needs to be initialized. A reasonable starting point is the Debye-Hückel

expression

$$\varphi_p = \frac{\sum_{\mathrm{a}}^{N_{\mathrm{atom}}} q_{\mathrm{a}} \exp\left(-\bar{\kappa}_a |\vec{r}_{\mathrm{p}} - \vec{r}_{\mathrm{a}}|\right)}{\varepsilon_{\mathrm{w}} \varepsilon_0 |\vec{r}_{\mathrm{p}} - \vec{r}_{\mathrm{a}}|}, \tag{2.71}$$

where φ_p is the potential at the grid point p with coordinate $\vec{r}_{\mathrm{p}}$; q_{a} is the fixed partial charge of atom a; $\bar{\kappa}_a$ is the inverse Debye length (defined as the squareroot of $\bar{\kappa}_a^2$ in eq. 2.73 at the position of $\vec{r}_{\mathrm{a}}$ of atom a), and ε_{w} is the dielectric constant of the solution.

It is important that the potential at the outer boundaries of the grid is given accurate enough by the initial approximation since these potential values will not change during the calculation. For that reason, it is important that the distance between the protein and the outer boundary of the grid is large enough (at least 10 Å).

Calculate Potential: Each value of φ_p on the grid represents an average of the continuous function over the volume surrounding the given point $\varphi(\vec{r})$. For example, the potential φ_p is given by $\int \varphi(\vec{r}) d\vec{r}$ over those grid points which lie closer to the grid point p than any other grid point, *i.e.,* the integral is taken over a cube of side h centered at the point p. The integration of the LPBE (eq. 2.69) over the volume gives

$$\int \vec{\nabla}\varepsilon(\vec{r})\vec{\nabla}\varphi(\vec{r}) d\vec{r} - \int \bar{\kappa}^2(\vec{r})\varphi(\vec{r}) d\vec{r} + \int \rho(\vec{r}) d\vec{r} = 0 \tag{2.72}$$

The second integral $\int \bar{\kappa}^2(\vec{r})\varphi(\vec{r}) d\vec{r}$ is approximated by $\bar{\kappa}_p^2 \varphi_p h^3$, where

$$\bar{\kappa}_p^2 = \mathcal{K}_p \bar{\kappa}^2(\vec{r}_p) \tag{2.73}$$

is the Debye-Hückel parameter $\bar{\kappa}^2(\vec{r}_p)$ (eq. 2.68) times the ion accessibility function $\mathcal{K}_p$ associated with grid point p at position $\vec{r}_{\mathrm{p}}$. The third term is the charge inside the volume element around grid point p, q_{p}. The first term can be transformed to a surface integral (Gauss's theorem). Thus, finally

$$\oint_{\mathcal{S}} \varepsilon(\vec{r})\vec{\nabla}\varphi(\vec{r}) d\vec{A} - \bar{\kappa}_p^2 \varphi_p h^3 + q_{\mathrm{p}} = 0 \tag{2.74}$$

is obtained. $d\vec{A}$ is the surface normal representing the area of the cube. A finite difference expression for the first term gives

$$\oint_{\mathcal{S}} \varepsilon(\vec{r})\vec{\nabla}\varphi(\vec{r}) d\vec{A} = \sum_{i=1}^{6} \frac{\varepsilon_{\mathrm{i}} \varepsilon_0 (\varphi_i - \varphi_p) h^2}{h} = \sum_{i=1}^{6} \varepsilon_{\mathrm{i}} \varepsilon_0 (\varphi_i - \varphi_p) h, \tag{2.75}$$

where φ_p is the potential at the grid point p and φ_i ($i = 1 \ldots 6$) is the potential at the six neighboring grid points; ε_{p} is the associated dielectric constant (see Fig. 2.4). The finite difference expression for φ_p can be obtained from eq. 2.74 and eq. 2.75:

$$\varphi_p = \frac{\left(\sum_{i=1}^{6} \varepsilon_{\mathrm{i}} \varepsilon_0 \varphi_i\right) + \frac{q_{\mathrm{p}}}{h}}{\left(\sum_{i=1}^{6} \varepsilon_{\mathrm{i}} \varepsilon_0\right) + \bar{\kappa}_p^2 h^2} \tag{2.76}$$

Thus, the potential at a given position depends on the potential at the neighboring positions of the grid.

Converged: The finite difference expression (eq. 2.76) defines a linear set of equations for the electrostatic potential at each grid point φ_p as a function of the electrostatic potential of the neighboring grid points and initial parameters. A number of numerical algorithms exist to solve a set of linear equations. The most simple are the Jacobian and Gauss-Seidel algorithms, but the Successive Over-Relaxation (SOR) algorithm can show a much faster convergence if the relaxation parameter is know or can be estimated well [28, 72, 73].

Equation 2.76 is iterated as long as the potential between two subsequent iteration steps does not change significantly anymore (convergence criterion).

Focussing: In order to obtain a sufficient precision and numerical stability of the calculated potentials, the grids used in the calculation should have a resolution of at least 0.25 Å. Even for relatively small proteins the number of grid points needed at this resolution to cover the protein and an adequate part of the solvent would require a prohibitive amount of memory for practical calculations. Therefore, the grid is refined in several steps. One starts with a grid that is large enough to hold the whole protein and has a distance of at least 10 Å between the protein and the outer boundary of the grid. The center of the grid is usually chosen as the geometric center of the protein. This grid will have a crude resolution (*e.g.,* 2.0 Å). Once the potential of this grid is converged, it can be used to initialize a finer grid (*e.g.,* 1.0 Å), which is embedded in the cruder grid. This procedure, called focussing [29], is repeated until a sufficiently fine grid (less than 0.25 Å) can be used. The finer grids are centered on the center of interest, *i.e.,* the set of atoms that form the site. The computational cost of focussing is usually less than solving the LPBE once *per* focussing step, because the convergence is improved in the ith iteration by better initial values on the electrostatic potential grid obtained by the $(i-1)$th iteration.

Calculate Energy: Once the potential at the finest grid is converged, electrostatic energies (eq. 2.57) can be calculated by multiplying the charges with the potential at the point of the charges. Since the potential is only known at the grid points, which do usually not coincide with the location of the charges of interest, the potential at the position of the charges is interpolated from the grid. It is important, that all the charges used for calculating the electrostatic potential, are situated within the finest grid. Details on the calculation of Born and background energies are given in section 3.2.3 and section 3.2.4, respectively. Born energies (as discussed in section 3.2.3) can not be calculated by the finite difference approach, because point charges are mapped to fractional charges on the nearest grid points by an interpolation scheme (see **Map Molecule to Grid**). The electrostatic energy, calculated by a set of point charges times the electrostatic potential on the grid, includes also an interaction energies between the point charges and their fractional charges. These artificial grid energies cancel exactly, when calculating differences in Born energy in two different environments with the same grid definitions. Therefore, the homogeneous and heterogeneous transfer energies (section 3.2.5 and section 3.2.6) are free of grid artefacts.

2.3 Quantum Chemistry

Quantum chemistry (here synonymously used with quantum mechanics, QM) is the foundation of most chemical structure based theoretical studies. Molecular mechanics (MM) heavily depends on QM for the parameterization of force fields. Electrostatic calculations are based on either force field charges (which were determined by QM) or charges directly calculated by QM.

In this work, QM is used to calculate energies of formation, $H_{\mathrm{vac},i}(j_k)$ and vibrational energies, $G_{\mathrm{vib},i}(j_k)$. Other common uses of QM, like the calculation of transition state energies (*i.e.,* kinetic studies of chemical reactions, reaction pathways) or the calculation of excited states of molecules (*i.e.,* to simulate or interpret spectroscopic data) were not done and are therefore not discussed.

For a fixed set of atomic coordinates the energy can be calculated by adopting the wave functions of the electrons or the electron density in an iterative procedure until self-consistency is reached. This procedure is called single point or self consistent field (SCF) calculation.

A second run type is geometry optimization: Based on single point calculations the Jacobian (matrix of first derivatives) is calculated and the atomic coordinates are changed. This procedure is iterated to make the gradient vanish, *i.e.,* to reach a (local) minimum of atomic coordinates and electron distribution. However, this procedure might lead to oscillations (which usually can be overcome changing some parameters) or a structure which is a saddle point in the energy landscape. Negative frequencies in a subsequent frequency calculation indicate a saddle point, while a structure with only positive frequencies is in an energy minimum. However, usually one aims for a global minimum, but there is no method to ensure that the global minimum is reached. The best approximation is to start from several different structures and take the lowest energy minimum.

However, the parts of biomolecular systems, one wants to treat quantum mechanically, are often so large, that an exhaustive conformational study is not affordable. Often even the frequency calculations to exclude saddle points are not feasible. Fortunately, good structural data of biomolecules is often available, which is used as starting point. If the structure only changes within the experimental error during the geometry optimization, it is an indication for a reasonable result. At best the results can be verified using multiple independently determined experimental structures.

The third run type, discussed here, is a frequency or normal mode calculation (calculation of the Hessian, the matrix of second derivatives). Here two variants are common: If an analytical expression for the second derivative of the energy function is available, frequencies can be calculated analytically. However, if the analytical second derivative is not available (which is the case for the exchange correlation functions I use), the Hessian has to be approximated numerically. For a harmonic approximation, the cost is 6 N_{atom} times the cost of a single point calculation, where N_{atom} is the number of atoms of the molecule. Therefore, harmonic frequency calculations of 20-30 atoms are feasible, however over 100 atoms, which are often used in geometry optimization for a reasonable model, are out of reach for frequency calculations. In this work, frequency calculations are mostly used for calculating vibrational energy changes, $\Delta G_{\mathrm{vib},i}(j_k)$ due to ligand binding on small model systems and it is assumed that these energy differences are transferable to larger systems.

Except the energy, an outcome of any single point calculation or geometry optimization is an electronic distribution, which is equivalent to a charge distribution (of known total charge). This charge distribution is mapped to atom centered point charges, usually used in MM and electrostatics calculations. This mapping is not at all unique, thus different approaches (outlined in section 2.3.2) are possible. In the following, an introduction into density functional theory and charge fitting methods will be given [74, 75].

2.3.1 Introduction to Density Functional Theory

Conventional *ab initio* methodology, involves solving Schrödinger's equation [76–78]

$$H\Psi = E\Psi \tag{2.77}$$

by a variational ansatz. Assuming the energy function, described by the Hamiltonian H is known, the wavefunction Ψ is searched, for which the energy E is minimal. Therefore, one has to work with a 3 N_e-dimensional wavefunction Ψ for an N_e-electron system (*e.g.,* in Hartree-Fock (HF) and post-HF methods), which is conceptionally complicated and computationally costly.

In density functional theory (DFT), instead of the 3 N_e-dimensional wavefunction, a simple three-dimensional electron density $\rho(\vec{r})$ is used. The Hohenberg-Kohn theorem [79] states, that the total energy of a system in its ground state is a functional of the system's electronic density, $\rho(\vec{r})$, and any density, $\rho'(\vec{r})$, other than the true density will necessarily lead to a higher energy. DFT requires only minimizing the energy functional $E[\rho(\vec{r})]$. Unfortunately, the exact nature of the energy functional is not known. The Kohn-Sham approach [80] partitioned the functional in the following manner:

$$E[\rho(\vec{r})] = U[\rho(\vec{r})] + T[\rho(\vec{r})] + E_{XC}[\rho(\vec{r})] \tag{2.78}$$

Here, $U[\rho(\vec{r})]$ is the classical electrostatic energy, $T[\rho(\vec{r})]$ the kinetic energy of a system of noninteracting electrons and $E_{XC}[\rho(\vec{r})]$ the exchange and correlation (XC) term, which is the difference between $T[\rho(\vec{r})]$ and the true electronic kinetic energy of the system.

The classical electrostatic energy, $U[\rho(\vec{r})]$, is the sum of electron-nucleus attraction and electron-electron repulsions:

$$U[\rho(\vec{r})] = \left(\sum_{\mathrm{a}}^{N_{\mathrm{atom}}} \int \frac{-Z_{\mathrm{a}}\rho(\vec{r})}{|\vec{r} - \vec{r}_{\mathrm{a}}|} d\vec{r} \right) + \frac{1}{2} \int\int \frac{\rho(\vec{r})\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} d\vec{r} d\vec{r}', \tag{2.79}$$

where $\rho(\vec{r})$ and $\rho(\vec{r}')$ are single-electron densities at the position $\vec{r}$ and $\vec{r}'$, respectively. The nucleus a at position $\vec{r}_{\mathrm{a}}$ has a charge of Z_{a}.

Kohn and Sham expressed the charge distribution $\rho(\vec{r})$ of an N_e-electron system (with N_α spin up electrons and N_β spin down electrons) as the sum of the square moduli of singly occupied, orthonormal Kohn-Sham (KS) molecular orbitals,

$$\rho(\vec{r}) = \rho_\alpha(\vec{r}) + \rho_\beta(\vec{r}) = \sum_{\vartheta=\alpha,\beta} \rho_\vartheta(\vec{r}) = \sum_{\vartheta=\alpha,\beta} \sum_{i}^{N_\vartheta} |\Psi_{\vartheta,i}(\vec{r})|^2. \tag{2.80}$$

The kinetic energy, $T[\rho(\vec{r})]$, can be defined as

$$T[\rho(\vec{r})] = \sum_{\vartheta=\alpha,\beta} \sum_{i}^{N_\vartheta} \int \Psi_{\vartheta,i}(\vec{r}) \frac{-\nabla^2}{2} \Psi_{\vartheta,i}(\vec{r}) d\vec{r}. \tag{2.81}$$

Following the Hohenberg-Kohn theorem, the energy functional is minimized by the true ground state density, $\rho(\vec{r})$. The energy functional $E[\rho(\vec{r})]$ must be stationary with respect to any arbitrary variation in either of the spin densities, *i.e.*,

$$\frac{\delta E[\rho(\vec{r})]}{\delta \rho_\alpha(\vec{r})} = \frac{\delta E[\rho(\vec{r})]}{\delta \rho_\beta(\vec{r})} = 0 \tag{2.82}$$

This condition yields the one-electron KS equations,

$$\left\{ \frac{-\nabla^2}{2} - \left(\sum_{\mathrm{a}}^{N_{\mathrm{atom}}} \frac{Z_{\mathrm{a}}}{|\vec{r} - \vec{r}_{\mathrm{a}}|} \right) + \int \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} d\vec{r}' + \frac{\delta E_{XC}[\rho(\vec{r})]}{\delta \rho_\vartheta(\vec{r})} \right\} \Psi_{\vartheta,i}(\vec{r}) = \epsilon_i \Psi_{\vartheta,i}(\vec{r}), \ \vartheta = \alpha, \beta \tag{2.83}$$

For practical DFT calculations, with an initial guess at the total spin densities, $\rho_\alpha(\vec{r})$ and $\rho_\beta(\vec{r})$ the KS equations are constructed and solved. The resulting set of KS spin-orbitals, $\Psi_{\vartheta,i}(\vec{r})$, are then used to generate new guesses at $\rho_\alpha(\vec{r})$ and $\rho_\beta(\vec{r})$. This procedure is repeated until self-consistency is achieved, meaning the same densities and KS orbitals are regenerated.

The KS treatment shifts the problem of an unknown total energy functional, $E[\rho(\vec{r})]$, to an unknown XC energy functional $E_{XC}[\rho(\vec{r})]$, but surprisingly simple approximations of $E_{XC}[\rho(\vec{r})]$ can give fairly accurate results.

The Local Spin Density Approximation (LSDA) [81] approximates $E_{XC}[\rho(\vec{r})]$ by

$$E_{XC}[\rho(\vec{r})] = \int \rho(\vec{r}) \epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r})] d\vec{r}, \tag{2.84}$$

where $\epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r})]$ is the XC energy density at a point $\vec{r}$ in space. The XC potentials, $v_{XC,\vartheta}(\vec{r})$, are given by

$$v_{XC,\vartheta}(\vec{r}) = \frac{\delta E_{XC}[\rho(\vec{r})]}{\delta \rho_\sigma(\vec{r})} = \rho(\vec{r}) \frac{d\epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r})]}{d\rho_\sigma(\vec{r})} + \epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r})]. \tag{2.85}$$

The heart of the LSDA is the approximation that a particular point in space of an inhomogeneous distribution of electrons with densities $\rho_\alpha(\vec{r})$ and $\rho_\beta(\vec{r})$ has the same values of $\epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r})]$, $v_{XC,\alpha}(\vec{r})$ and $v_{XC,\beta}(\vec{r})$ as any point in a homogeneous distribution of electrons of the exact same densities $\rho_\alpha(\vec{r})$ and $\rho_\beta(\vec{r})$. The value of $\epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r})]$ has been determined for a large number of homogeneous gases of interacting electrons by means of quantum Monte Carlo methods [82, 83]. I use the parameterization of Vosko, Wilk and Nusair (VWN) [81].

Even with the crude approximation of LSDA, DFT is able to provide quantitatively accurate geometries, charge distributions and vibrational spectra on a wide variety of systems, surpassing HF and challenging post-HF methods. However, LSDA systematically overestimates binding energies, which limits its use for thermochemical applications such as computation of atomization energies or heats of reactions. Also hydrogen-bond strength are overestimated leading to distorted geometries for systems, where such interactions are important, like in biomolecules.

For such applications, gradient-corrected XC energy functionals are available, trying to correct for the shortcomings of LSDA:

$$E_{XC}[\rho(\vec{r})] = \int \rho(\vec{r})\epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r}), \nabla\rho_\alpha(\vec{r}), \nabla\rho_\beta(\vec{r})]d\vec{r} \tag{2.86}$$

The XC energy density $\epsilon_{XC}[\rho_\alpha(\vec{r}), \rho_\beta(\vec{r}), \nabla\rho_\alpha(\vec{r}), \nabla\rho_\beta(\vec{r})]$, now not only contains the charge densities $\rho_\alpha(\vec{r})$ and $\rho_\beta(\vec{r})$, but also their gradients. By this, the conceptual simplicity of LSDA is lost, leading to various models, which have been proposed. Gradient-corrected XC functionals are computationally more expensive since the derivatives of $\rho_\alpha(\vec{r})$ and $\rho_\beta(\vec{r})$ need to be evaluated. Fortunately, it is often not necessary to include gradient corrections in the SCF procedure, but only include them to calculate energies and forces. Gradient corrections are thus simply treated in a perturbative fashion, rendering gradient-corrected DFT calculations as inexpensive as their LSDA counterparts. I use the gradient-corrected exchange and correlation energy functional of Perdew and Wang [84, 85].

The KS equation can be solved numerically in a basis-set-free approach [86], but for practical applications the KS orbitals, $\{\Psi_i\}$, are usually expressed as a linear combination of atom-centered basis functions,

$$\Psi_i(\vec{r}) = \sum_{\mu}^{N_b} C_{\mu i}\chi_\mu(\vec{r}), \tag{2.87}$$

where $\{\chi_\mu\}$ forms a linear combination of atomic orbital (LCAO) basis consisting of N_b functions. $\{C_{\mu i}\}$ are the expansion coefficients for the ith KS orbital. In the Amsterdam Density Functional (ADF, [87–89]) package Slater-type orbitals (STOs) are used instead of the common Gaussian-type orbitals (GTOs). STOs describe better the shape of atomic and molecular orbitals and can only be approximated as linear combination of GTOs. Therefore, a lower number of basis functions and a faster convergence is expected.

Each atomic orbital is usually represented by a number of basis functions, given by the ζ value. The relatively high basis sets TZP and TZ2P(+) of ADF are used. Both are core double-ζ, valence triple-ζ basis sets, meaning that the core atomic orbitals are repesented by two STOs and the valence orbitals are represented by three STOs. TZP adds one polaization function and TZ2P is a doubly polarized basis set. The TZ2P+ basis sets for the transition metals Sc-Zn are nearly identical to TZ2P except for a better description of the d-space (4 d-functions instead of 3). In agreement with Szilagyi and Winslow [90] I found that consistent results could only be obtained with the TZ2P(+) basis set for iron-sulfur clusters.

For Gaussian calculations the 6-31G [91] and 6-311G [92] basis sets are used. According to the nomenclature of Pople and co-workers, both basis sets represent the core orbitals by a contraction of six primitive GTOs (PGTOs) and the inner part of the valence orbitals is a contraction of three PGTOs. In case of the 6-31G basis set, the outer part of the valence orbitals is represented by one PGTO. In case of the 6-311G basis set, the valence basis is split into three functions represented by three, one and one PGTOs, respectively. To each basis set diffuse and/or polarization functions can be added. Diffuse functions denoted by + add diffuse s- and p-functions on heavy atoms only, while ++ indicates that also diffuse s-functions are added on hydrogen atoms. A single d-type polarization function is marked by *

(6-31G(d) or 6-31G*). A single d-type and a single p-type polarization function is marked by ** (6-31G(d,p) or 6-31G**).

2.3.2 Charge Fitting

The electron density $\rho(\vec{r})$ is equivalent to a charge distribution for a known total charge, but this charge distribution does not suit the atomic model of a force field. Also for electrostatic calculations usually atom centered point charges are used. In principle, the QM charge distribution could be used directly in electrostatic calculations, but the QM grid is usually much larger than the molecule and would therefore cross dielectric boundaries leading to artefacts.

It is sufficient to find a set of point charges, which reproduces the electrostatic potential of the charge distribution. Most applications use a single partial charge at the atomic center, but also a few partial charges *per* atom can be used [13]. Unlike the charge distribution, atomic point charges are not an observable. The mapping between the charge distribution and a set of point charges is not unique. In this process an under determined set of equations has to be solved. Additional constraints, like conservation of monopole, dipole and higher multipoles increases the number of equations even further.

In the CHELPG (CHarges from ELectrostatic Potentials using a Grid based method) scheme by Breneman and Wiberg [93], atomic charges are fitted to reproduce the molecular electrostatic potential (MEP) at a number of points around the molecule. As a first step of the fitting procedure, the MEP is calculated at a number of grid points spaced 0.3 Å apart and distributed regularly in a cube. The dimensions of the cube are chosen so that the molecule is located at the center of the cube, adding 2.8 Å between the molecule and the end of the box in all three dimensions. All points falling inside the van-der-Waals radius of the molecule are discarded from the fitting procedure. After evaluating the MEP at all valid grid points, atomic charges are derived that reproduce the MEP by a least-square method using Lagrange multipliers. The only additional constraint in the fitting procedure is that the sum of all atomic charges equals that of the overall charge of the system. CHELPG charges are frequently considered superior to Mulliken charges as they depend much less on the underlying theoretical method used to compute the wavefunction (and thus the MEP). Unlike the CHELP scheme, CHELPG is less dependent on the orientation of the molecule.

The required matrix inversion is performed by a Gauss-Jordan elimination procedure in the original implementation. However, this procedure gives little indication of how close the matrix to be inverted is to being singular. Mouesca *et al.* [37] used a singular value decomposition (SVD) analysis instead to understand how unique and well-defined the point charge fit is and what are the expected uncertainties. This implementation is used, if not stated otherwise.

Behera and Ullmann [94] implemented the algorithm alternatively using the Lagrange multiplier technique or the technique of generalized inverse. The singular value decomposition (SVD) is used for solving the set of linear equations. The program, by now part of ADF, provides many options to constrain multipoles or charges of atoms and charge groups. Multiple conformations can be taken into account. Atomic charges may depend strongly on the conformation of the molecule. Representative atomic charges for flexible molecules should therefore be derived as average values over several conformers.

One of the weak points of CHELPG (and other approaches based on fitting the MEP) is the treatment of larger systems, in which some of the innermost atoms are located far away from the points at which the MEP is evaluated. In such a situation, variations of the innermost atomic charges will not lead to significant changes of the MEP outside of the molecule and fitting of these atomic charges will therefore not result in meaningful results.

Swart [95] *et al.* developed a method to fit charges based on an atomic multipole expansion. For that purpose, the total density is written as a sum of atomic densities, from the atomic densities a set of atomic multipoles is defined and the atomic multipoles are reconstructed by distributing charges over all atoms. Their multipole derived charges (MDC) reproduce both the atomic and molecular multipole moments. This method was extended by Thomas Ullmann (unpublished) to allow additional constraints, like setting charges or groups of charges to certain values. Additional to the RMSD values, Hodgkin indices were introduced as quality measure.

2.4 Molecular Modeling

The methods described in this section use a coarser view on molecular systems. Instead of calculating molecular structures and properties based on very little knowledge, *e.g.,* parameters for the XC functions in DFT, the methods discussed here heavily depend on parameterization. To keep the number of parameters reasonable, the methods take sever approximations. Only positions of nuclei are studied, ignoring all electronic degrees of freedom. Atoms or even complete residues are the building blocks of the model.

Modern molecular mechanics force fields (*i.e.,* for biopolymers) parameterizes each atom type differently dependend on its hybridization and surrounding atoms. Therefore, despite the simple physical model, a huge number of parameters are required. Database derived force fields try to exploit the information contained in the over 45000 structures of biomolecules stored in the PDB database to derive energetic information based on statistics. For some purposes like hydrogen placement, even some geometric and first-principle chemical considerations as well as coarse upper and lower boundaries of parameters are sufficient to obtain a limited number of structures.

2.4.1 Molecular Mechanics

For molecular dynamics (MD), force fields were derived, which describe molecular energies by classical potentials. Atoms are represented as spheres with fixed radii and (partial) charges. Electrons are not represented explicitly, so that chemical reactions can usually not be simulated. Bonds, angles, torsions and improper torsions are represented by simple potentials with an equilibrium distance and a stiffness. The non-bonded terms, *i.e.,* the Lennard-Jones and Coulomb potential are only calculated between all atoms that do not share a bond or bond angle. The interaction between atoms that share a torsion is sometimes scaled by a factor. The Lennard-Jones potential models the van-der-Waals attraction and the Pauli repulsion. The electrostatic energy is calculated by a Coulomb potential (eq. 2.56).

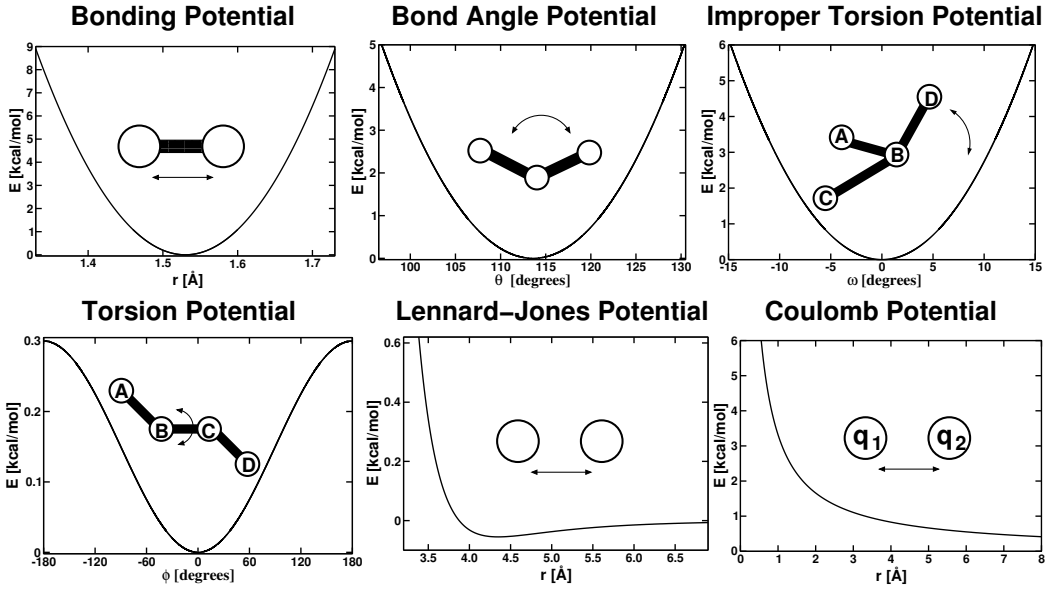


Figure 2.5. Energy terms in a force field. The dependence of the force field energy as function of the bond length, bond angle, improper torsion angle, torsion angle or distance in the Lennard-Jones and Coulomb term are shown.

For many model building and minimization jobs, CHARMM [96–99] was used, which has an energy function like:

$$\begin{aligned} E = &\sum^{N_b} k_b(r-r_0)^2 + \sum^{N_\theta} k_\theta(\theta-\theta_0)^2 + \sum^{N_\omega} k_\omega(\omega-\omega_0)^2 \\ &+ \sum^{N_\phi} k_\phi(1-\cos(n\phi-\delta)) + \sum_{i<j} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right] + \sum_{i<j}\frac{q_i q_j}{\varepsilon r} \end{aligned} \tag{2.88}$$

The bond length r, bond angle θ and improper torsion angle ω have an equilibrium distance or equilibrium angle r_0, θ_0 and ω_0, respectively. The potentials are associated with a force constant k_b, k_θ and k_ω, respectively (Fig. 2.5). The improper torsion angle is given for four atoms A, B, C, and D, where A-B-C define a plane and atom D is bound to atom B. The angle between the bond B-D and the plane is the improper torsion angle, which is used to fix stereo isomers and ensure ring planarity. The dihedral (or torsion) angle is defined by four sequentially bonded atoms A-B-C-D, described by a cosine dependence in eq. 2.88. The periodicity of the angle is n, the phase shift δ and the barrier height k_ϕ. The next term is the Lennard-Jones potential, which is combined of the Pauli repulsion (term to the 12^{th} power) and the van-der-Waals attraction (term to the 6^{th} power). $\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j)$ is the zero point distance (*i.e.*, $E_{LJ} = 0$ for $r = \sigma_{ij}$) and $\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}$ is the depth of the potential (energy minimum at $r \approx 1.122\sigma$). The last term is the Coulomb potential, which is the product of the charge of the atom pair divided by the distance and the electrical permittivity of a homogeneous medium.

All information needed in eq. 2.88 is given in so-called parameter and topology files. Atomtypes are defined, which represent an atom in a particular environment, *e.g.*, a carbonyl carbon in a peptide bond. For each atomtype interacting with another atomtype, the parame-

terfile tabulates, *e.g.,* equilibrium distances and angles or stiffnesses of the harmonic potential. A topology file defines the bond topology of molecules and assigns charges to atoms. In CHARMM, sets of atoms are grouped to so called "charge groups" with an integer total charge. These charge groups can be used as building blocks, maintaining an integer charge of the system.

The art of designing a force field is to combine experimental (*e.g.,* data from vibrational spectroscopy, NMR data or crystallographic data for the equilibrium distances and angles) and theoretical data (from QM geometry optimization, normal mode analysis and charge fitting) to obtain a consistent set of parameters, which are transferable and able to reproduce experimental results within their regime of applicability. Unfortunately, the derivation of force field parameters is often not straight forward, nevertheless attempts are made to automate this task [100].

This indirect and iterative scheme of force field generation requires also special care, when using force field data for other applications, like electrostatics. Often only the charges are used, while radii are taken from other sources. The reason is, that (polar) hydrogen radii are often reduced. For the definition of dielectric boundaries in the electrostatic calculations, more realistic radii are preferred. When rotamers are included to incorporate a discrete flexibility into the electrostatic calculations one has to notice that the energy function is changed compared to the original force field due to the different radii used. Attention has to be paid to define the rotamers correctly, so that all energy terms are constant, which are not included. For example, in a molecule consisting of the sequentially bonded atoms A-B-C-D-E, a torsion angle change between atoms B and C (rotating atom A) requires, that atoms B, C and D are fixed. Bonded interactions between atom A and atom E are constant, so that atom E can be rotated independent of atom A. If the molecule is branched, one has to follow all branches and include the atoms in the same way, *i.e.,* atoms bound to B (except C) have to rotate in the same way as A, atoms bound to C have to be fixed and atoms bound to D can rotate concertedly with E. In simple cases, *i.e.,* hydrogen rotamers, the bonded energy terms can be pre-calculated and taken from a lookup table as function of the dihedral angle (section 5.3). Non-bonded interactions have to be calculated separately and included by an interaction energy (section 3.2.7). Also at the boundary of parts explicitly treated by QM and classical parts, artefacts from non-bonded energies of neighboring atoms have to be avoided (section 3.3.1).

2.4.2 Database Derived Force Fields or Statistical Potentials

Molecular mechanics and also DFT are based on a physical model describing the system in atomic or even electronic detail. The physical model is described in terms of mathematical equations, which contain parameters, which can be adapted to fit to experimental results. Larger systems are constructed from small building blocks, which were parameterized in detail. Database derived force fields assume that a structural database contains all information needed to parameterize a force field. The structural information is broken into building blocks (usually on a *per*-amino-acid basis) and analyzed statistically. For a sufficiently large ensemble of microstates, in which the probability of microstates is Boltzmann distributed, the relative energy of each microstate can be directly obtained from its probability:

$$\Delta G = -RT \ln\langle x \rangle \tag{2.89}$$

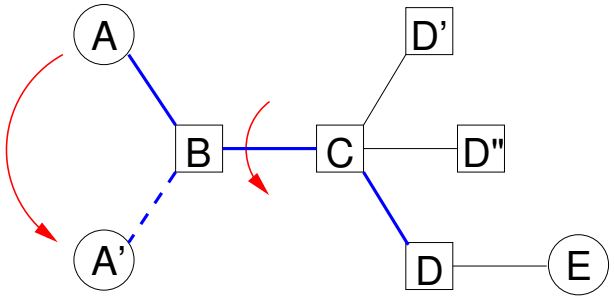


Figure 2.6. Rotamer of A in a molecule A-B-C-D-E. Atom A is rotated to position A' by changing the torsion angle between atom B and C. Atoms marked by a box (B, C and D as well as D' and D") must have fixed coordinates. Atom E can be rotated independent of the bonded terms of atom A.

For $\langle x \rangle$ being the probability of a certain configuration, the energy ΔG can be directly obtained inverting the Boltzmann distribution. The advantage of a statistical potential is that it converts a wealth of experimental data directly into an energy function. However, it is always questionable how well a Boltzmann distribution is represented by a database, which might be biased. Also, the lack of a detailed physical model hampers a further analysis to improve the understanding of the system.

For a statistical analysis it is necessary to cluster the experimentally found rotamers into a limited number of discrete rotamers. To include rotamers of protein side-chains into continuum electrostatic calculations, such a discretization is also necessary. A continuous, time dependent treatment of protein flexibility as in molecular dynamics is much more costly. For the sampling of a complex conformational space a statistical treatment based on a discrete conformational space is much more efficient, since no time has to be spend to overcome intermediate barriers. Indeed, the main advantage of a dynamics simulation, to obtain time-dependent information on the progress of the system, is lost. In this work, the focus is on thermodynamics, finding a low energy ensemble, not on kinetics, looking at the time evolution of the system. Therefore, rotamers from a rotamer database are well suited for the purpose of this work. The internal energies of the rotamers might be taken from the rotamer database, but also could be pre-calculated by molecular mechanics.

Many rotamer libraries have been parameterized over the last decades [101]. They differ in resolution and the number of proteins included in the analysis as well as details of the statistical treatment. The rotamer library of Dunbrack and Cohen [102] is one of the larges and is widely used. The authors provide a backbone independent and a backbone dependent library, *i.e.*, a library where the probability of the sidechain rotamers is given dependent on the ϕ- and ψ-angles of the backbone. The database (in ASCII format) is freely available from the webpage of the authors and can be used to generate sidechain rotamers within Perl Molecule (section 4.6).

2.4.3 Geometrical considerations

The coordinates of non-hydrogen atoms of a biomolecule are often well defined by x-ray crystallography. However, the position of hydrogen atoms can usually not be determined, even for the very high resolution structures of ferredoxin (section 5.3). For many hydrogen, *e.g.,* aliphatic or aromatic hydrogen, the position is well defined (assuming a rather constant bond length) from the position of the non-hydrogen atoms. However, especially for the hydrogens of hydroxy-, carboxyl- and amino-groups, usually more than one hydrogen position (rotamer or tautomer) is possible. Multiple hydrogen bond networks can be constructed, depending *e.g.,* if a hydroxyl group acts as donor or acceptor in a hydrogen bond. Unfortunately, the outcome of electrostatic calculations may critically depend on the hydrogen bond network, which is given as input [45].

A lot of special programs are available to add hydrogens to a given structure (*e.g.,* HBplus [103, 104], HBexplore [105, 106], Whatif [107, 108]) and also molecular mechanics programs can be used (*e.g.,* HBUILD in CHARMM [96, 99]). Most of the programs choose hydrogen positions based on some energy function. Often the algorithms is very naive (*e.g.,* the result may depend on the residue number) and can not ensure to find a global minimum. Also the energy function used for hydrogen placement, may not be in agreement with the energy function based for later energy calculations. Some programs work well for amino acids, but cofactors cause problems. The validity of the found hydrogen bond network can be questioned, when involved cofactors have to be removed for adding hydrogens to sidestep the limits of the program.

During this work, Hwire [40] was found particularly useful. It generates hydrogen positions based on geometric considerations only, excluding doubts connected to inconsistent force fields. For all according to the geometric criteria possible positions in which a hydrogen could participate in a hydrogen bond network an atom is placed. These alternative hydrogen positions are treated as rotamers in this work. Therefore, the decision on a certain hydrogen rotamer is made at the same time as the protonation and redox state of the system is determined. The hydrogen bond network changes with the state variables.

Hwire reads a topology file, which is flexible enough to be adopted even for unusual cofactors (*i.e.,* complex metal centers or organic molecules like quinone).

The hydrogen placement is done based on the geometric considerations in Fig. 2.7. Angles and distances have to be in a “permitted” range for accepting a certain hydrogen position. The default values of Hwire (Tab. 2.1) are probably good for $O \cdots H$ hydrogen bonds, but even Calimet and Ullmann used slightly larger values in their work [40]. For the much longer $S \cdots H$ hydrogen bonds in iron-sulfur clusters the parameters given in Tab. 2.1 were used. Unfortunately, Hwire does not allow to distinguish between $O \cdots H$ and $S \cdots H$ hydrogen bonds. The applied strategy avoids overlooking possible hydrogen bonds in favor of having too long and weak hydrogen bonds. These hydrogen bonds are expected to be disfavored in the final energy evaluation due to their high electrostatic energies in the Poisson-Boltzmann calculations.

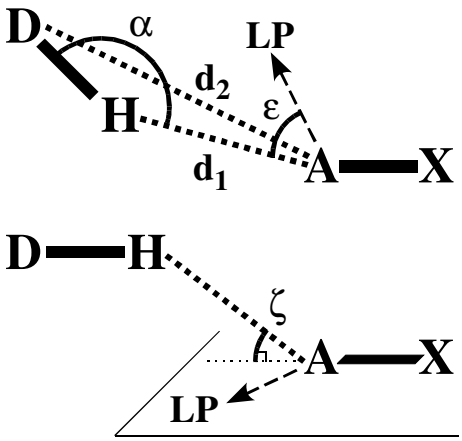


Figure 2.7. Hydrogen bond criteria of Hwire. Donor atom (D), acceptor atom (A), hydrogen atom (H) and lone pair (LP) geometries are described by two distances and three angles. The distance d_1 is between the proton (H) and the acceptor (A), the distance d_2 is between the donor (D) and the acceptor (A). α is the angle between donor (D), hydrogen (H) and acceptor (A). ε is the angle between the hydrogen (H), the acceptor (A) and any of the acceptor lone pairs (LP), if they exist. The angle ζ is defined for hydrogen bonds involving a sp2 acceptor or donor. ζ is defined as angle of the hydrogen bond with respect to the plain LP, A, X (or X', D, H for a donor).

	Default value	Value used
$0 \leq d_1 \leq$	2.5	3.8
$0 \leq d_2 \leq$	3.9	4.5
$\alpha \geq$	120.0	110.0
$\varepsilon \leq$	45.0	180.0
$\zeta \leq$	45.0	180.0

Table 2.1. Hwire parameters given as default and used in this work. Distances d_1 and d_2 are given in Å, angles α, ε and ζ are given in degree.

2.5 Summary

In section 2.1, chemical reactions in terms of equilibrium constants and chemical potentials were introduced. Microscopic and macroscopic binding constants were distinguished and it was found that both microstates and macrostates can be described by microscopic energies. However, it was also shown that a macroscopic description is not sufficient to describe the energetics of a system with more than one ligand binding site.

Section 2.1.2 introduced chemical potentials as a useful concept to describe chemical reactions. The common description of acid-base equilibria in terms of solution pH and dissociation constant of the acid ($\mathrm{p}K_a$ value, section 2.1.4) were introduced. It was shown, that an analogous description of redox equilibria in terms of reduction potential of the solution and standard reduction potential of the redox active group (section 2.1.5) can be given. For molecules with a single proton or electron binding site the titration behavior can be described by the commonly known Henderson-Hasselbalch and Nernst equations, respectively. For molecules with more binding sites a statistical mechanics description in terms of chemical potentials and binding free energies will be introduced in section 3.1.

An important contribution to the energy function is the electrostatic energy term, which is calculated using a continuum electrostatics approach. Therefore, the linearized Poisson-Boltzmann equation (LPBE) was derived from the first Maxwell equation (section 2.2) and it was outlined, how the LPBE can be solved numerically by finite difference methods (section 2.2.6). More details will be given in section 3.2.

Quantum chemistry (QM), especially Density Functional Theory (DFT), was introduced (section 2.3), because it is required for optimizing structures, calculating energies of formation and vibration of reactants and products and as basis for the fitting of charges. QM derived energy terms will be used as part of the energy function in particular in section 3.3. The energetics of small structural changes like rotamer energies may be described by QM, but they can be calculated faster and not necessarily less accurate by molecular mechanics (MM, section 2.4.1 and section 3.4). Instead to obtain conformational energies describing larger structural changes, either available experimental values or energies calculated by MM can be used, while QM calculations are usually too costly. Other classical mechanic models, *i.e.,* the description of rotamers by a library (section 2.4.2) or the placement of hydrogens based on geometric criteria (section 2.4.3), were briefly discussed, because they are used for structure preparation (section 4.6) for subsequent ligand binding energy calculations.

Chapter 3

A Generalized Theory for Calculations of Ligand Binding Energetics

In the previous chapter, a rigorous statistical mechanics description of ligand binding was avoided, because the aim was to give only a flavor of the problems to be studied. Now, the ligand binding will be described in the grand canonical ensemble (section 3.1.1) and it will be shown how microscopic and macroscopic properties can be calculated from the grand canonical partition function (section 3.1.3). In practice, the calculation of the grand canonical partition function is often not affordable and therefore an approximate treatment based on Monte Carlo calculations (section 3.1.4) is introduced. An energy function for the microstate energy composed of contributions for each instances of each sites is given (section 3.1.2). Since this energy function is dominated by electrostatic energy contributions, the continuum electrostatic model is described in more detail, *i.e.,* it is formulated to become applicable for molecular calculations (section 3.2). Energy contributions are defined for a concise description of energies of instances of sites (in section 3.3 and section 3.4).

I will refer to sites as sets of atoms described by different coordinates (rotamers) or directly involved in binding of one or more ligands (of potentially different types). Electrostatic interactions between sites are described in the model, but not electronic interactions. For practical reasons, sites should be as small as possible, but the partial charges of the atoms should add up to an integer charge and π-bonds should not be divided. Due to binding of ligands, the charges and coordinates of sites change. The different combinations of charge forms and rotamer forms are called instances. Larger structural changes, which can not be well described by rotamers, can be included as conformers. Usually, conformers are based on different structures obtained at potentially different chemical or physical conditions. A more detailed discussion can be found in section 3.2.2.

Two possible descriptions of sites will be introduced. One description is based on experimentally determined binding energies of model compounds in solution. The effect of transferring the site from the model compound environment into the protein environment is calculated (section 3.4). The other description is based on quantum chemical calculations (section 3.3). It can be used, if no experimental data is available or if the site is too complex for an accurate experimental description. Such experimental problems determining energies for model compounds can arise *e.g.,* from strong electronic coupling between the ligands binding to the site hindering a decomposition of the microstates (as it is the case for benzoquinone, section 5.5),

or from instability of the site outside of the protein hindering synthesis of model compounds (as it is the case for the iron-sulfur center of ferredoxin, section 5.3).

The physical description of the system is general in respect to the number of sites, the number of instances *per* site, the number of ligands in the system and the number of ligands binding to each single site. Sites described by model compounds or by quantum mechanical calculations can be mixed. This physical model allows for titration calculations in a high-dimensional chemical potential space. It is implemented in the program QMPB (section 4.3), which scales linearly with the number of instances and number of CPUs due to a suitable formulation of the energy function in this chapter.

At the end of this chapter, the energy function is compared with the functions used in previous works. It is shown that the previous formulations only cover the special case of two instances *per* site (section 3.5). The limitations of the previous model are underlined, *e.g.,* leading to serious approximations for including rotamers. The previous treatment of sites by quantum chemistry was hampered by the lack of appropriate programs resulting in an exponential growth of computational cost with the number of instances and the inability to combine the quantum chemical treatment with the model compound based treatment of sites (*i.e.,* calculation of interaction energies with acceptable computational cost).

3.1 Towards an Energy Function for Ligand Binding

Section 2.1 introduced chemical reactions in terms of equilibrium constants and chemical potentials. Microscopic and macroscopic equilibrium constants can be calculated from microstate energies. The connection between the microstate and the macrostate is formed by the the grand canonical partition function (section 3.1.1). A suitable microstate energy function is described in section 3.1.2. How to calculate probabilities and equilibrium constants from the grand canonical partition function is discussed in section 3.1.3.

3.1.1 The Grand Canonical Partition Function

To describe ligand binding energetics the grand canonical ensemble was chosen, which allows ligand molecules to enter and leave the system during simulation. The macroscopic state of the system is defined by a set of thermodynamic variables $\{\mu_\lambda\}$:

$$\{\mu_\lambda\} = \left\{T, V, \mu_1, \ldots, \mu_\lambda, \ldots, \mu_{N_{\text{ligand}}}\right\}, \tag{3.1}$$

where T is the absolute temperature, V the volume and μ_λ the chemical potential of ligand type λ ($\lambda = 1 \ldots N_{\text{ligand}}$). The temperature T is a simple variable in the calculations. Moderate changes of the temperature do not affect the calculations significantly, since the temperature enters the calculation inverse in the thermodynamic β ($\beta = \frac{1}{RT}$) or the inverse Debye length $\kappa(\vec{r})$ (eq. 2.68). A larger effect is the dependence of the electrical permittivity $\varepsilon(\vec{r})$ on the temperature, *i.e.,* the dielectric constants have to be adopted accordingly. The volume V has not to be further defined, since the electrostatic model usually considers a single macromolecular complex (molecule) in an infinite implicit solvent. This model describes well a highly dilute

solute. Since the chemical potentials μ_λ (section 2.1.2) of all ligand types λ are the most important in the set of thermodynamic variables, the symbol $\{\mu_\lambda\}$ was chosen.

The grand canonical partition function Ξ is the sum of the partition functions of all conformations i, Ξ_i, for a given set of thermodynamic variables $\{\mu_\lambda\}$:

$$\Xi(\{\mu_\lambda\}) = \sum_{i}^{N_{\text{conf}}} \Xi_i(\{\mu_\lambda\}) \tag{3.2}$$

The grand canonical partition function Ξ_i of conformation i is a Boltzmann weighted sum of $N_{\text{micro},i}$ energies of the ensemble of microstates $G_{\text{micro}}(\vec{x}_{\text{i,n}}, \{\mu_\lambda\})$ (characterized by their state vector $\vec{x}_{\text{i,n}}$ and a set of thermodynamic variables $\{\mu_\lambda\}$):

$$\Xi_i(\{\mu_\lambda\}) = \sum_{n}^{N_{\text{micro},i}} \sigma(\vec{x}_{\text{i,n}}, \{\mu_\lambda\}) \exp\left(-\beta G_{\text{micro}}(\vec{x}_{\text{i,n}}, \{\mu_\lambda\})\right) \tag{3.3}$$

Here, $\sigma(\vec{x}_{\text{i,n}}, \{\mu_\lambda\})$ is the symmetry number (degeneracy factor).

There are $N_{\text{micro},i} = \prod_j^{N_{\text{site},i}} N_{\text{instance},i,j}$ microstates, where $N_{\text{site},i}$ is the number of sites in conformer i and $N_{\text{instance},i,j}$ is the number of instances in site j of conformer i ($N_{\text{instance},i,j}$ is generally any natural number, but in the titration theory with a binary state vector, section 3.5, it is two).

The symmetry number $\sigma(\vec{x}_{\text{i,n}}, \{\mu_\lambda\})$ for the microstate with energy $G_{\text{micro}}(\vec{x}_{\text{i,n}}, \{\mu_\lambda\})$ is equal to one as long as all microstates are represented explicitly, what is assumed in the following. For achiral molecules, there might be several isoenergetic microstates, which can be transferred into each other by symmetry operations. In this cases, the symmetry number has to be taken into account. Proteins have many chirality centers and therefore impose a chiral environment onto ligand molecules. In calculations of ligand molecules in solution (*e.g.,* quinone in solution, section 5.5) the symmetry number might be important [109]. Also a symmetric ligand might have multiple isoenergetic microstates in the unbound form, which become non-isoenergetic upon binding to the protein (section 3.3).

Each microstate n of conformation i is characterized by an individual state vector $\vec{x}_{\text{i,n}}$

$$\vec{x}_{\text{i,n}} = \left(x_{i,n}^1, \ldots, x_{i,n}^j, \ldots, x_{i,n}^l, \ldots, x_{i,n}^{N_{\text{site},i}}\right) \tag{3.4}$$

Instances $k \equiv x_{i,n}^j$ and $m \equiv x_{i,n}^l$ are the values of the state vector $\vec{x}_{\text{i,n}}$ at the position j and l, respectively. I will refer to instance k and m of site j and l as j_k and l_m, respectively.

3.1.2 The Microstate Energy Function

The microstate energy $G_{\text{micro}}(\vec{x}_{\text{i,n}}, \{\mu_\lambda\})$ was introduced in section 2.1.3 as sum of the standard energy of the microstate $G^\circ(\vec{x}_{\text{i,n}})$ and the stoichiometric factor ν_λ times the chemical potential μ_λ summed over the N_{ligand} ligand types λ (eq. 2.21). It was pointed out, that the stoichiometric factor is relative to a freely chosen reference state. Here, I set the reference state as the (hypothetic) fully unbound state, *i.e.,* that $N_\lambda(\vec{x}_{\text{i,n}})$ is the number of bound ligands of type λ to

the molecule. The conformational energy $G_{\mathrm{conf},i}$ is the energy of conformer i.

$$G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\}) = G_{\mathrm{conf},i} + G^\circ(\vec{x}_{\mathrm{i,n}}) - \sum_{\lambda}^{N_{\mathrm{ligand}}} N_\lambda(\vec{x}_{\mathrm{i,n}})\mu_\lambda \tag{3.5}$$

For practical calculations it is not feasible to calculate all possible microstates separately, *i.e.,* the standard energy of each microstate $G^\circ(\vec{x}_{\mathrm{i,n}})$. I assume, that the standard energy can be split into a sum of contributions of each instance k of site j, the intrinsic energy $G_{\mathrm{intr},i}(j_k)$, and a contributions due to the interaction with each instance m of each other site l, the interaction energy $G_{\mathrm{inter},i}(j_k, l_m)$:

$$G^\circ(\vec{x}_{\mathrm{i,n}}) = \sum_{j}^{N_{\mathrm{site},i}} G_{\mathrm{intr},i}(j_k) + \frac{1}{2}\sum_{j}^{N_{\mathrm{site},i}}\sum_{l\neq j}^{N_{\mathrm{site},i}} G_{\mathrm{inter},i}(j_k, l_m). \tag{3.6}$$

This assumption of additivity is true, if a MM force field or electrostatic energies calculated by the Poisson equation or LPBE (section 3.2) are used. The number of bound ligands $N_\lambda(\vec{x}_{\mathrm{i,n}}) = \sum_j^{N_{\mathrm{site},i}} n_{\lambda,i}(j_k)$ can also be split into contributions of each site, so that $n_{\lambda,i}(j_k)$ is the number of ligands of type λ bound to instance j_k. Therefore, the microstate energy can be written as

$$G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\}) = G_{\mathrm{conf},i} + \sum_{j}^{N_{\mathrm{site},i}} \left(G_{\mathrm{intr},i}(j_k) - \sum_{\lambda}^{N_{\mathrm{ligand}}} n_{\lambda,i}(j_k)\mu_\lambda \right) + \frac{1}{2}\sum_{j}^{N_{\mathrm{site},i}}\sum_{l\neq j}^{N_{\mathrm{site},i}} G_{\mathrm{inter},i}(j_k, l_m). \tag{3.7}$$

By exploiting the assumed additivity as in eq. 3.7, the computational cost reduces from $O\left(\sum_i^{N_{\mathrm{conf}}} \prod_j^{N_{\mathrm{site},i}} N_{\mathrm{instance},i,j}\right)$ (for eq. 3.5) to $O\left(\sum_i^{N_{\mathrm{conf}}} \sum_j^{N_{\mathrm{site},i}} N_{\mathrm{instance},i,j}\right)$ (for eq. 3.7). For example, a system with two conformers and 11 sites with 10 instances each, has $2 \cdot 10^{11}$ microstates, but the formulation in eq. 3.7 reduces the computational cost to calculating $2 \cdot 11 \cdot 10 = 220$ intrinsic energies, which is assumed to be the time determining step (*i.e.,* solving the LPBE). The double sum of the interaction energy does not contribute significantly to the computational cost, since it is a simple multiplication, as will be shown in section 3.2.7. Both, the matrix of $G_{\mathrm{intr},i}(j_k)$ values and the tensor of $G_{\mathrm{inter},i}(j_k, l_m)$ values can be stored and looked up upon calculating the microstate energy $G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})$ in a separate step.

Each conformation i has a user defined conformational energy $G_{\mathrm{conf},i}$. The conformational energy can be calculated by a MM force field (section 2.4) or it can be used to weight different experimental structures according to experimental results. When calculating the conformational energy, it is important to exclude all energy contributions, which enter the calculation of the microstate energy by other terms, *i.e.,* electrostatic interactions between sites or rotamer energies.

The intrinsic energy $G_{\mathrm{intr},i}(j_k)$ describes the energy of instance k of site j in a different environment (*e.g.,* the environment of a model compound or in vacuum) and the energy for transferring the instance of the site from the different environment into the protein environment (section 3.2.5). It contains the energy due to interaction of instance j_k with the background (charge) set of atoms, *i.e.,* all atoms not belonging to any site. The intrinsic energy $G_{\mathrm{intr},i}(j_k)$ is defined differently for sites which are parameterized based on QM calculations (section 3.3) or based on experimental results (section 3.4).

The interaction energy $G_{\mathrm{inter},i}(j_k, l_m)$ describes the interaction of the current instance k of site j with instance $m \equiv x^l_{i,n}$ of any other site l according to the state vector $\vec{x}_{i,n}$ of microstate n. The factor $\frac{1}{2}$ is to correct for double counting the interaction of instance j_k with instance l_m and instance l_m with instance j_k, because the double sum runs twice over all $N_{\mathrm{site},i}$ sites.

It would be possible to use microstate energies $G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})$ as in eq. 3.3 directly. However, numerical errors might result, if the exponential of the energy approaches the precision of the number representation in a computer. Therefore, it is useful to define a reference state $G_{\mathrm{micro}}(\mathrm{ref}, \{\mu_\lambda\})$ such, that $G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})$ has small values (*i.e.,* for the most contributing species in the statistical average, eq. 3.10):

$$\Delta G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\}) = G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\}) - G_{\mathrm{micro}}(\mathrm{ref}, \{\mu_\lambda\}) \tag{3.8}$$

This shift of the zero point of the microstate energies does not change the relative population of the microstates to each other. It only reduces the numerical error significantly.

3.1.3 Calculation of Properties Based on the Partition Function

Microstate n in conformer i has for a given set of thermodynamic variables $\{\mu_\lambda\}$ a population (or probability) of

$$\langle \vec{x}_{\mathrm{i,n}}(\{\mu_\lambda\})\rangle = \frac{\sigma(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})}{\Xi_i(\{\mu_\lambda\})} \exp\Big(-\beta G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})\Big) \tag{3.9}$$

The symmetry number $\sigma(\vec{x}_{\mathrm{i,n}}, \{\mu_\lambda\})$ was discussed in section 3.1.1. The probability $\langle x(j_k, \{\mu_\lambda\})\rangle$ of site j being in instance k can be calculated using a delta function $\delta_{i,n}(j_k)$, which is one, if $x^j_{i,n} = k$ and zero otherwise:

$$\langle x(j_k, \{\mu_\lambda\})\rangle = \sum_i^{N_{\mathrm{conf}}} \sum_n^{N_{\mathrm{micro},i}} \delta_{i,n}(j_k) \langle \vec{x}_{\mathrm{i,n}}(\{\mu_\lambda\})\rangle \tag{3.10}$$

or for a particular conformation i

$$\langle x_i(j_k, \{\mu_\lambda\})\rangle = \sum_n^{N_{\mathrm{micro},i}} \delta_{i,n}(j_k) \langle \vec{x}_{\mathrm{i,n}}(\{\mu_\lambda\})\rangle \tag{3.11}$$

By changing the chemical potential μ_λ of ligand type λ (in the set of thermodynamic variables $\{\mu_\lambda\}$), the population of instance k of site j varies, leading to probability curves (or titration curves) for this instance.

The uptake of ligand λ is the total average number $\langle X_\lambda\rangle$ of ligands of type λ bound to the molecule. It is given by the sum of the probabilities of each microstate $\langle \vec{x}_{\mathrm{i,n}}\rangle$ multiplied by the total number of ligands of type λ $N_\lambda(\vec{x}_{\mathrm{i,n}})$, bound in the particular microstate:

$$\langle X_\lambda\rangle = \sum_i^{N_{\mathrm{conf}}} \sum_n^{N_{\mathrm{micro},i}} N_\lambda(\vec{x}_{\mathrm{i,n}}) \langle \vec{x}_{\mathrm{i,n}}\rangle \tag{3.12}$$

In general, as stated in section 2.1.1, microscopic and macroscopic equilibrium constants can be determined from microscopic free energies. The microscopic equilibrium constant for the

reaction, where microstate $n = 1$ is the reactant state and microstate $n = 2$ is the product state, is given by:

$$K^{\vec{x}_{\mathrm{i},2}}_{\vec{x}_{\mathrm{i},1}} = \exp\Big(-\beta(G^\circ(\vec{x}_{\mathrm{i},2}) - G^\circ(\vec{x}_{\mathrm{i},1}))\Big) \tag{3.13}$$

The N_λth macroscopic equilibrium constant of ligand type λ is given by:

$$\bar{K}_{N_\lambda} = \frac{\sum_i^{N_{\mathrm{conf}}} \sum_n^{N_{\mathrm{micro},i}} \delta\Big(N_\lambda(\vec{x}_{\mathrm{i,n}}) - 1\Big) \exp\Big(-\beta(G_{\mathrm{conf},i} + G^\circ(\vec{x}_{\mathrm{i,n}}))\Big)}{\sum_i^{N_{\mathrm{conf}}} \sum_n^{N_{\mathrm{micro},i}} \delta\Big(N_\lambda(\vec{x}_{\mathrm{i,n}})\Big) \exp\Big(-\beta(G_{\mathrm{conf},i} + G^\circ(\vec{x}_{\mathrm{i,n}}))\Big)} \tag{3.14}$$

The delta function $\delta(N_\lambda(\vec{x}_{\mathrm{i,n}}))$ is defined to be one, if the microstate with state vector $\vec{x}_{\mathrm{i,n}}$ has N_λ bound ligands of the type λ and zero otherwise.

3.1.4 Approximating Probabilities of Microstates

Most of the equations in section 3.1.3 depend on the knowledge of all $N_{\mathrm{micro},i}$ microstate energies in each of the N_{conf} conformers to calculate the partition function Ξ:

$$N_{\mathrm{conf}} \cdot N_{\mathrm{micro},i} = \sum_i^{N_{\mathrm{conf}}} \prod_j^{N_{\mathrm{site},i}} N_{\mathrm{instance},i,j} \tag{3.15}$$

An example in section 3.1.2 showed, that this number is very large already for moderate systems. Except for very small systems, the partition function can not be calculated explicitly (by programs as SMT, implemented by Matthias Ullmann). A number of schemes exist [27, 36, 110–113] to approximate the probability $\langle \vec{x}_{\mathrm{i,n}}(\{\mu_\lambda\})\rangle$ of a microstate without evaluating the partition function. In this work, the Metropolis Monte Carlo (MMC) method is used. The Adaptive Mesh Refinement (AMR, section 4.2.3) method can focus the MMC calculations on regions in chemical potential space, which are of interest since the probabilities are not constant. AMR can also reduce the number of instances, which need to be sampled, and finally the state vector length by removing instances, which are known not to be populated in a certain part of chemical potential space.

A property of the Boltzmann distribution in the grand canonical partition function is, that microstates with low energy contribute exponentially more than microstates with high energy. Therefore, a sampling method, which finds an ensemble of low energy microstates, is able to obtain a good approximation of the probability of microstates significantly contributing in the grand canonical partition function. A Monte Carlo (MC) method with importance sampling (as implemented by Matthias Ullmann in the program GMCT) is suitable for this purpose.

For a given set of thermodynamic variables $\{\mu_\lambda\}$ and for a random conformation i an initial state vector $\vec{x}_{\mathrm{i,n}}$ is generated randomly. In a MC move, the instance m of a randomly-chosen site j is changed to a randomly-chosen instance k. The energy change between instance m

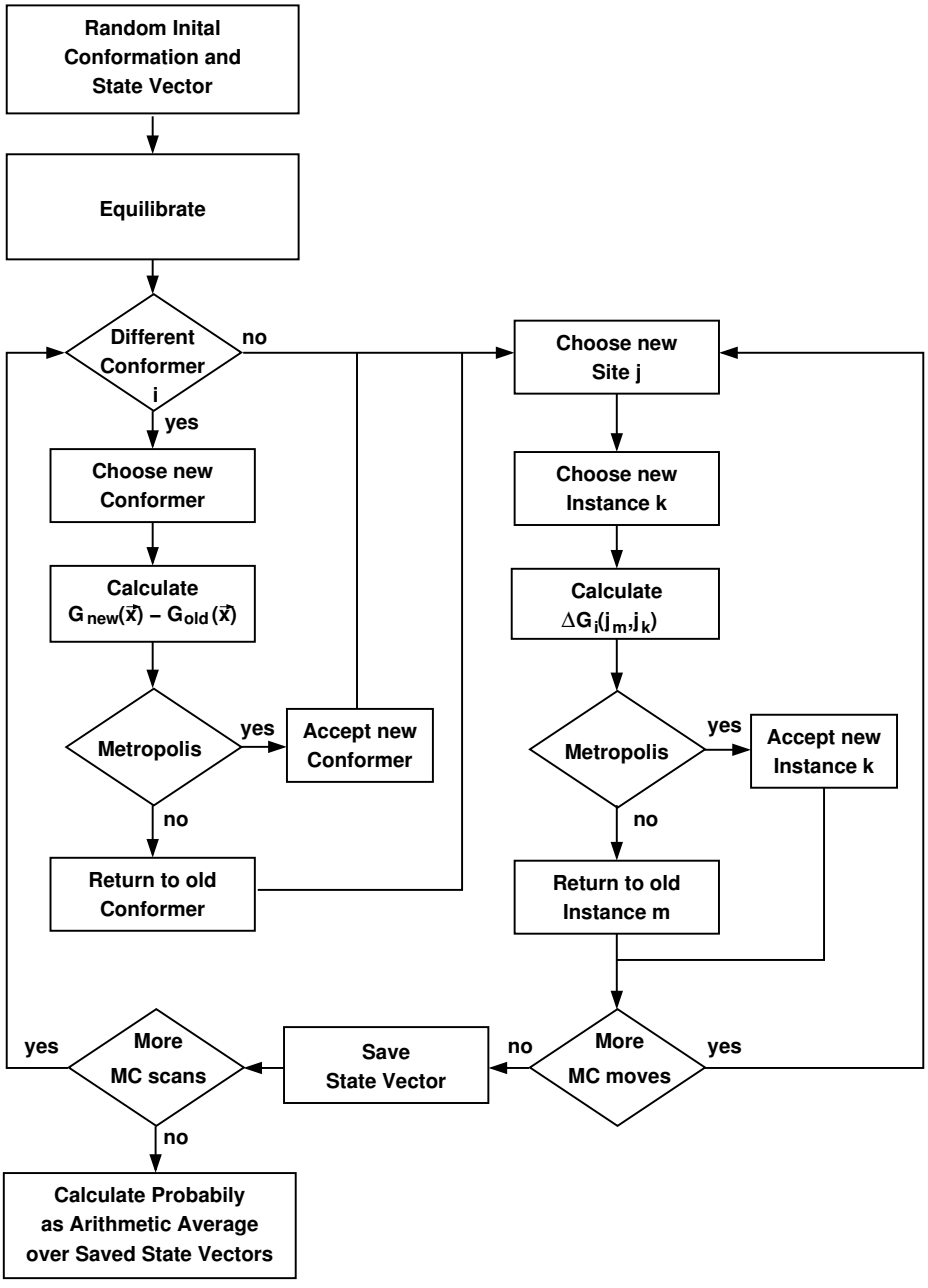


Figure 3.1. Flowchart of the Metropolis Monte Carlo procedure to approximate the probability of microstates. The probability of all instances is calculated from a pre-defined number of low energy microstates instead of calculating the statistical average from the ensemble of all possible microstates (eq. 3.10). The procedure is described in the text. The equilibration step is equivalent to the loop over the number of MC scans, except that the protonation state vectors are not saved.

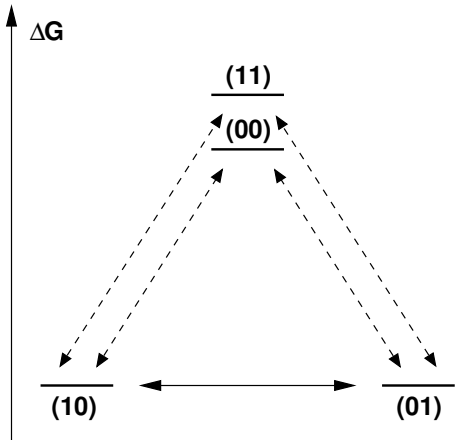


Figure 3.2. Treatment of two strongly coupled sites in Monte Carlo (MC) titration calculation. If the barrier for the transition from microstate (10) to microstate (01) via the microstates (11) or (00) is to large (dashed arrow) a MC step is performed that simultaneously switches from (10) to (01) (solid arrows).

and k is calculated:

$$\Delta G_i(j_m, j_k) = \left(G_{\mathrm{intr},i}(j_m) - \sum_{\lambda}^{N_{\mathrm{ligand}}} n_{\lambda,i}(j_m)\mu_\lambda \right) - \left(G_{\mathrm{intr},i}(j_k) - \sum_{\lambda}^{N_{\mathrm{ligand}}} n_{\lambda,i}(j_k)\mu_\lambda \right) + \frac{1}{2} \sum_{l \neq j}^{N_{\mathrm{site},i}} \left(G_{\mathrm{inter},i}(j_m, l_w) - G_{\mathrm{inter},i}(j_k, l_w) \right) \tag{3.16}$$

The energy $\Delta G_i(j_m, j_k)$ is the difference between the microstate energies (eq. 3.7) of the state vector before and after the change of the state vector. Since the contribution of all other sites $l \neq j$ and the conformation energy $G_{\mathrm{conf},i}$ cancels, because it is constant, the energy function gets a much simpler form. The new microstate is accepted according to the Metropolis criterion [114] (Metropolis Monte Carlo, MMC), *i.e.,* if $\Delta G_i(j_m, j_k) \leq 0$, the microstate is always accepted, if $\Delta G_i(j_m, j_k) > 0$ the new microstate is accepted with a probability of $\exp(-\beta \Delta G_i(j_m, j_k))$. A MC scan is finished after $N_{\mathrm{site},i}$ moves, *i.e.,* after $N_{\mathrm{site},i}$ attempts to change a randomly-chosen site. During the MC scan the conformation i is attempted to change to another conformation with identical state vector several times and the new conformation is accepted according to the Metropolis criterion. After a few hundred MC scans to reach equilibrium, the state vectors describing the microstate are accumulated after each MC scan. The probability of each instance $\langle x(j_k, \{\mu_\lambda\}) \rangle$ is calculated by arithmetic averaging. It is not required to calculate a statistical average (eq. 3.10), because the microstates are already Boltzmann distributed due to the Metropolis criterion. A flowchart of the MC procedure is given in Fig. 3.1. For the calculation of titration curves, the chemical potential of the ligands μ_λ is usually changed iteratively in small increments within pre-defined ranges.

If ligand binding to two sites is strongly coupled, there may be two microstates with low energy. For example, two sites may share a ligand, *i.e.,* the microstate energy is low, if the ligand is bound to one of the two sites. However, the energy for both sites without the ligand

bound or both with a ligand bound may be significantly higher. The energy difference may be large enough to make the transfer of the ligand from one site to the other unlikely, if one of the intermediate states has to be passed through (Fig. 3.2). The problem can be avoided by changing the instance of both sites simultaneously - a so-called double-move. The move is also accepted according to the Metropolis criterion. In analogy, triple-moves are possible for the case that three sites couple strongly.

3.2 Continuum Electrostatics at Atomic Detail

In section 2.2 continuum electrostatics was introduced by deriving the Poisson, Poisson-Boltzmann (PBE) and linearized Poisson-Bolzmann (LPBE) equation from Gauss's law. It was also discussed, how the PBE and LPBE can be solved numerically. The treatment was usually based on the general concept of a charge distribution $\rho(\vec{r})$, but in the context of the Coulomb equation (section 2.2.2) also point charges q were introduced.

This section concentrates on the atomic description within the continuum electrostatic model, focussed to be useful for the calculation of contributions to the microstate energy function (section 3.1.2). It is explained, why additivity of electrostatic potentials is required (section 3.2.1), how boundaries are defined for rotamers and conformers and what the consequences of the two descriptions of molecular flexibility are (section 3.2.2). The concepts of Born and background energies are introduced (section 3.2.3 and section 3.2.4) and it is shown how they can be used to calculate transfer energies from homogeneous and heterogeneous environments into the protein (section 3.2.5 and section 3.2.6). Finally, the electrostatic interaction energy, as it is used in eq. 3.7, is introduced (section 3.2.7).

3.2.1 Point Charges

For practical calculations on molecular systems, usually a point charge q_a is associated with the coordinate of each atom a:

$$\rho(\vec{r}) = \sum_{\mathrm{a}}^{N_\mathrm{atom}} q_\mathrm{a}(\vec{r}_\mathrm{a}) \tag{3.17}$$

The values of point charges are often taken from MM force fields (section 2.4.1) and have non-integer values (partial charges) to describe correctly dipoles and higher multipoles. However, sets of atoms - so-called charge groups - have an integer charge. Since systems with non-integer charge are usually physically not meaningful, care has to be taken when dividing the system into smaller sub-systems (*i.e.,* sites) ensuring that each fraction always has an integer charge (*i.e.,* each instance). Integer charges can be assured, when always only complete charge groups (or several charge groups at once) are transferred from one sub-system to another. If charges are calculated quantum chemically (section 2.3), attention has to be payed for including only complete charge groups into the calculation. Else the quantum chemically derived charges are not compatible to the force field charge group approach.

Electrostatic energies (as charge times electrostatic potential, eq. 2.57) are additive when the electrostatic potential $\varphi(\vec{r}_\mathrm{a})$ is calculated by the Poisson equation or LPBE. The non-linearized PBE, however, is not additive due to the exponential term of the Boltzmann distribution,

but this additivity is retrieved by linearization in the LPBE. The calculation of the microstate energy, eq. 3.7, contains electrostatic energy terms for each instance of each site, thus it is of crucial importance to evaluate the potentials ensuring additivity. Due to this additivity, the computational cost for calculating the electrostatic energy contributions is drastically reduced (section 3.1.2). Hence, the computational cost using the PBE is prohibitive for ligand binding studies on usual proteins or a probably small error is made assuming the same additivity for the PBE.

3.2.2 Dielectric Boundaries

In section 2.2.6, it was pointed out that the result of electrostatic calculations depend critically on the dielectric boundaries. The method to calculate solvent accessible surfaces based on atom coordinates and radii was described. For the interior of the surfaces a different dielectric constant is assigned compared to the solvent.

In section 3.1.2, I introduced conformers and assigned them a conformer energy, $G_{\mathrm{conf},i}$. A different conformation has a significantly different protein structure. Therefore, it must have different dielectric boundaries in the continuum electrostatics calculations. All instances of all sites have to be calculated separately for each set of dielectric boundaries, thus the computational cost multiplies by the number of conformations. Instead, a common set of dielectric boundaries is defined by the coordinates of all atoms of the protein in all their instances (*i.e.,* all rotamer forms and with all ligands bound) so that an additional (rotamer) instance adds only a single calculation of the LPBE. However, rotamers are a coarser approximation due to the common set of dielectric boundaries. Therefore, the choice between conformers and rotamers is sometimes a trade-off. For small ligands, small changes between the rotamers (*i.e.,* hydrogen rotamers) and buried sites with the same dielectric constant as the background set, the approximation by the common set of dielectric boundaries is good. In other cases it depends on the system.

3.2.3 Born Energy

The Coulomb energy eq. 2.58 became infinity if the distance between the central point of the charge distribution and the point of the potential measurement became zero in a homogeneous dielectric. In case of the Poisson equation (eq. 2.59) the dielectric media is generally not homogeneous, but the electrical permittivity ε depends on the point in space $\vec{r}$. The dielectric boundaries $\vec{\nabla}\varepsilon(\vec{r})$ shield the electrostatic field (eq. 2.52), so that the electrostatic energy $\rho\varphi(\vec{r};\rho)$ is finite. The electrostatic field $\varphi(\vec{r};\rho)$ acting back on the charge distribution ρ is called reaction field. The electrostatic energy due to the interaction of a charge distribution ρ with its reaction field is called Born energy. Usually for molecular systems, the charge distribution ρ is described by (partial) point charges q_{a} at positions $\vec{r}_{\mathrm{a}}$ of atoms a: The Born energy is therefore written as:

$$G_{\mathrm{Born}} = \frac{1}{2}\sum_{\mathrm{a}}^{N_{\mathrm{atom}}} q_{\mathrm{a}}\varphi(\vec{r}_{\mathrm{a}};\rho) \tag{3.18}$$

The electrostatic potential $\varphi(\vec{r}_\mathrm{a};\rho)$ at the center of an atom a, depends on one hand on the position $\vec{r}_\mathrm{a}$ of the atom, on the other hand on the charge distribution ρ of this and all other atoms (given by their partial charge and position).

The factor $\frac{1}{2}$ originates from the linear response approximation: The Born energy is related to the work charging a cavity from $q_\mathrm{i} = 0$ to q_f within a uncharged macroscopic dielectric body. The reaction field acting back at the inducing charge is assumed to be proportional to q and of opposite sign:

$$\rho = Cq;\ C < 0 \tag{3.19}$$

The work of charging from $q_\mathrm{i} = 0$ to q_f is then

$$\begin{aligned} W_\mathrm{elec} &= \int_0^{q_\mathrm{f}} \rho dq \\ &= \int_0^{q_\mathrm{f}} Cq dq \\ &= \frac{C}{2} {q_\mathrm{f}}^2 \\ &= \frac{1}{2}\varphi_f q_\mathrm{f} < 0, \end{aligned} \tag{3.20}$$

where φ_f is the reaction field potential (or back potential) at the end of the charging process. The effective charging work does not equal the charge times the potential, but only half of it. The missing work is the energy cost of polarizing the polarizable media [115].

3.2.4 Background Energy

Additionally to the charge distribution ρ of the atom set of interest, there might be fixed charges Q_a in the system leading to the so-called background energy:

$$G_\mathrm{back} = \sum_\mathrm{a}^{N_\mathrm{atom}} Q_\mathrm{a}\varphi(\vec{r}_\mathrm{a};\rho) \tag{3.21}$$

The electrostatic potential $\varphi(\vec{r}_\mathrm{a};\rho)$ at the point of a background charge Q_a, depends on one hand on the position $\vec{r}_\mathrm{a}$ of the background charge, on the other hand on the charge distribution ρ of all atoms of the atom set of interest.

An electrostatic energy is generally the sum of Born and background energy:

$$W_\mathrm{elec} = G_\mathrm{Born} + G_\mathrm{back} \tag{3.22}$$

A very important property of the background energy is that it is symmetric: With ρ_q being the charge distribution of charges q_a at the positions $\vec{r}_\mathrm{a}$ and ρ_Q being the charge distribution of charges Q_b at the positions $\vec{r}_\mathrm{b}$ the following is valid:

$$\sum_\mathrm{a}^{N_\mathrm{atom}} q_\mathrm{a}\varphi(\vec{r}_\mathrm{a},\rho_\mathrm{Q}) = \sum_b^{N_\mathrm{atom}} Q_\mathrm{b}\varphi(\vec{r}_\mathrm{b};\rho_\mathrm{q}) \tag{3.23}$$

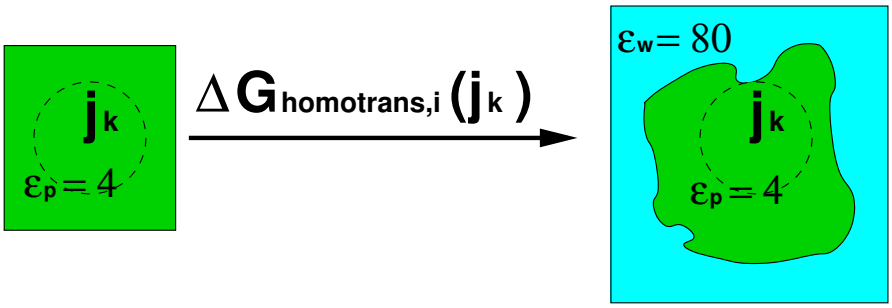


Figure 3.3. Calculation of a homogeneous transfer energy $\Delta G_{\mathrm{homotrans,i}}(j_k)$. ε_{p} is the dielectric constant of the protein and ε_{w} is the dielectric constant of the solvent, *e.g.*, water.

This property is independent of the background energy being calculated from the Poisson, Poisson-Boltzmann or the linearized Poisson-Boltzmann equation.

The linear response approximation relates the Born energy to the work of charging a cavity (section 3.2.3). However, for isolated systems (*i.e.*, the thermodynamic cycles described below) the charge has to remain constant (conservation of charge). Instead, the difference in electrostatic energy due to transferring the charge of instance k of site j in conformer i from one point in space to another is the objective of the calculations. In this work, the destination of all charge transfers is the heterogeneous protein system, which consists of at least two dielectric regions (the protein and the aqueous solvent), but two types of transfer energies can be distinguished, dependent on the source of the charge transfer: The source can be a homogeneous environment with same dielectric constant as the site or a heterogeneous environment, where the site is part of a model compound embedded in a different dielectric region (*e.g.*, aqueous solvent).

3.2.5 Homogeneous Transfer Energy

The source of the charge transfer is a homogeneous dielectric environment (Fig. 3.3). The heterogeneous transfer energy is the sum of Born and background energy for transferring instance j_k of conformer i:

$$\Delta G_{\mathrm{homotrans,i}}(j_k) = \Delta G_{\mathrm{Born,homotrans,i}}(j_k) + G_{\mathrm{back,homotrans,i}}(j_k) \tag{3.24}$$

The difference in Born energy is calculated as:

$$\begin{aligned}
\Delta G_{\mathrm{Born,homotrans,i}}(j_k) &= G_{\mathrm{Born,protein,i}}(j_k) - G_{\mathrm{Born,homo,i}}(j_k) \\
&= \frac{1}{2} \sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k) \varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i}}(j_k)) - \frac{1}{2} \sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k) \varphi_{\mathrm{homo}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i}}(j_k)) \\
&= \frac{1}{2} \sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k) [\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i}}(j_k)) - \varphi_{\mathrm{homo}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i}}(j_k))]
\end{aligned} \tag{3.25}$$

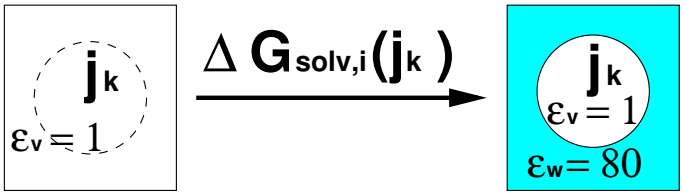


Figure 3.4. Calculation of a solvation energy $\Delta G_{\mathrm{solv,i}}(j_k)$. ε_{v} is the dielectric constant of vacuum and ε_{w} is the dielectric constant of the solvent, *e.g.*, water.

There are $N_{\mathrm{atom,i,j_k}}$ partial charges $q_{\mathrm{a,i}}(j_k)$ of instance k of site j, which interact with the electrostatic potential $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ in the protein environment and with the electrostatic potential $\varphi_{\mathrm{homo}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ in the homogeneous dielectric. The electrostatic potentials are due to the charge distribution $\rho_{\mathrm{i}}(j_k) = \sum_{\mathrm{a}}^{N_{\mathrm{atom},i,j_k}} q_{\mathrm{a,i}}(j_k)$.

In the calculations described here, there are no background charges present in the homogeneous dielectric, but only in the protein environment. Therefore, the background energy of instance k of site j in conformer i is:

$$\begin{aligned} G_{\mathrm{back,homotrans,i}}(j_k) &= G_{\mathrm{back,protein,i}}(j_k) \\ &= \sum_{a}^{N_{\mathrm{back,i}}} Q_{\mathrm{protein,a,i}}\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k)) \end{aligned} \tag{3.26}$$

The background (charge) set of the protein has $N_{\mathrm{back,i}}$ charges $Q_{\mathrm{protein,a,i}}$, which interact with the electrostatic potential $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ of instance j_k, identical to the potential calculated for the Born energy. The background charge set is constant for all sites of a particular conformer i. Therefore, calculating the background energy only requires additional multiplications.

For the special case, where the homogeneous dielectric is vacuum, the electrostatic energy is called solvation energy (Fig. 3.4):

$$\Delta G_{\mathrm{solv,i}}(j_k) = \Delta G_{\mathrm{Born,homotrans,i}}(j_k) + G_{\mathrm{back,homotrans,i}}(j_k) \tag{3.27}$$

The solvation energy is the difference in energy of a molecule in vacuum and in a given solvent (described by a dielectric constant ε_{r}). Born calculated the solvation energy of spherical ions in a homogeneous solvent analytically [116]. Therefore the energy difference due to the difference in the electric field in different media is called Born solvation energy.

In fact, the solvation energy discussed here is only the electrostatic part. A full solvation energy would include the work for creating a cavity for the atoms in the solute. This work is largely entropy dependent and can only be estimated from the surface area and some empirical factor [117, 118]. In the titration calculations only the difference in solvation energy between charge forms is calculated. The boundaries are kept constant so that the cavity does not change and therefore there is no associated work to be done.

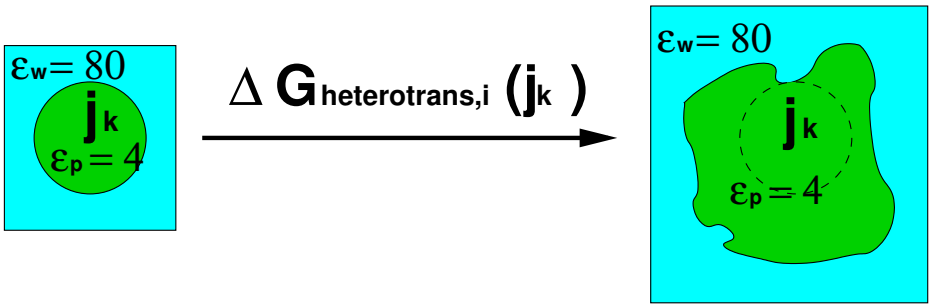


Figure 3.5. Calculation of a heterogeneous transfer energy. ε_{p} is the dielectric constant of the protein and ε_{w} is the dielectric constant of the solvent, *e.g.*, water.

3.2.6 Heterogeneous Transfer Energy

The source of the charge transfer is a heterogeneous dielectric environment (Fig. 3.5). Usually a model compound in solution is used as reference point. The transfer of a subset of charges (those belonging to instance k of site j) of the model compound into the protein environment is calculated. The coordinates (rotamer form) remain the same. The heterogeneous transfer energy is the sum of Born and background energy:

$$\Delta G_{\mathrm{heterotrans,i}}(j_k) = \Delta G_{\mathrm{Born,heterotrans,i}}(j_k) + \Delta G_{\mathrm{back,heterotrans,i}}(j_k) \tag{3.28}$$

The difference in Born energy is calculated as:

$$\begin{aligned}\Delta G_{\mathrm{Born,heterotrans,i}}(j_k) &= G_{\mathrm{Born,protein,i}}(j_k) - G_{\mathrm{Born,model,i}}(j_k) \\ &= \frac{1}{2}\sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k)\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k)) - \frac{1}{2}\sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k)\varphi_{\mathrm{model}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k)) \\ &= \frac{1}{2}\sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k)[\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k)) - \varphi_{\mathrm{model}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))]\end{aligned} \tag{3.29}$$

There are $N_{\mathrm{atom,i,j_k}}$ partial charges $q_{\mathrm{a,i}}(j_k)$ of instance k of site j, which interact with the electrostatic potential $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ in the protein environment and with the electrostatic potential $\varphi_{\mathrm{model}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ in the heterogeneous dielectric environment of the model compound. The electrostatic potential is due to the charge distribution $\rho_{\mathrm{i}}(j_k) = \sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} q_{\mathrm{a,i}}(j_k)$ of the atoms a in instance k of site j in conformer i.

Usually there are background charges present in the model compound, *i.e.*, charges which are not transferred into the protein environment. The protein has a different set of background charges, which are identical to those discussed for the homogeneous transfer energy. The

difference in background energy is calculated as:

$$\begin{aligned}\Delta G_{\mathrm{back,heterotrans,i}}(j_k) &= G_{\mathrm{back,protein,i}}(j_k) - G_{\mathrm{back,model,i}}(j_k) \\ &= \sum_{a}^{N_{\mathrm{back,i}}} Q_{\mathrm{protein,a,i}}\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k)) - \sum_{a}^{N_{\mathrm{back,model,i,j}}} Q_{\mathrm{model,a,i}}(j)\varphi_{\mathrm{model}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))\end{aligned} \tag{3.30}$$

The background (charge) set of the protein has $N_{\mathrm{back,i}}$ charges $Q_{\mathrm{protein,a,i}}$, which interact with the electrostatic potential $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ of instance j_k in the protein. The background charge set of the model compound has $N_{\mathrm{back,model,i,j}}$ charges $Q_{\mathrm{model,a,i}}\ (j)$, which interact with the electrostatic potential $\varphi_{\mathrm{model}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ of instance j_k in the model compound. The model compound can be different for each site j, but it has to be constant for each instance k of site j. Again, the electrostatic potentials $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ and $\varphi_{\mathrm{model}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ are identical to those calculated for the Born energy and calculation of the background energy only requires additional multiplications.

3.2.7 Interaction Energy

The background energy term of the transfer energies only contains the interaction of atoms a of instance k of site j with the background charge set, *i.e.,* those atoms, not belonging to any site. The interaction energy $G_{\mathrm{inter},i}(j_k,l_m)$ (eq. 3.7) between instance j_k and instance l_m is given by:

$$G_{\mathrm{inter},i}(j_k,l_m) = \sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,l_m} Q_{\mathrm{a,i}}(l_m)\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k)) = \sum_{\mathrm{a}}^{N_{\mathrm{atom}},i,j_k} Q_{\mathrm{a,i}}(j_k)\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(l_m)) \tag{3.31}$$

Due to the symmetry (eq. 3.23) the interaction energy is independent of the order of the sites, *i.e.,* $G_{\mathrm{inter},i}(j_k,l_m) = G_{\mathrm{inter,i}}(l_m,j_k)$. Because the electrostatic potential $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}};\rho_{\mathrm{i}}(j_k))$ is calculated anyway for each instance k of each site j, the calculation of interaction energies only requires additional multiplications. The theoretical symmetry can be used to detect numerical errors in the calculations.

The intrinsic energies, discussed in the following, are focussed on electrostatic energy contributions. Therefore, also the interaction energy contains only an electrostatic term. One might include other interactions (like Lennard-Jones potentials, section 2.4.1) into the energy function as long as they are additive and can be splitted into an intrinsic part and an interaction part.

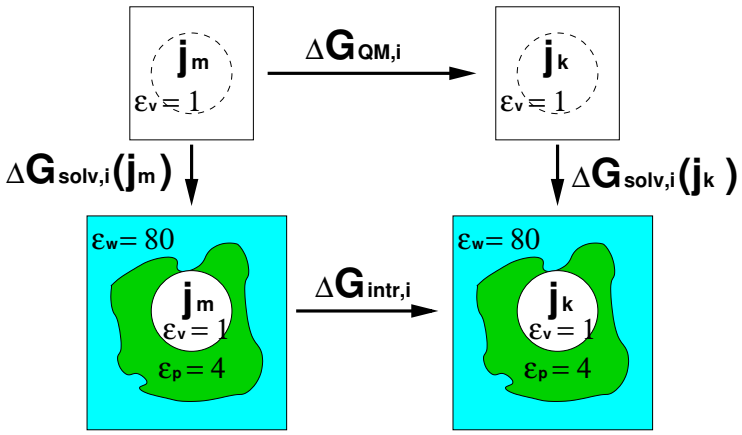


Figure 3.6. Intrinsic energy calculation based on QM calculations. The energy difference between two instances k and m of a quantum center j in vacuum $\Delta G_{\mathrm{QM,i}} = (H_{\mathrm{vac},i}(j_k) + G_{\mathrm{vib},i}(j_k) + G_{\mathrm{free},i}(j_k)) - (H_{\mathrm{vac},i}(j_m) + G_{\mathrm{vib},i}(j_m) + G_{\mathrm{free},i}(j_m))$ can be calculated by QM (section 2.3). The solvation energies $\Delta G_{\mathrm{solv,i}}(j_m) = \Delta G_{\mathrm{Born,homotrans,i}}(j_m) + G_{\mathrm{back,homotrans,i}}(j_m) - G_{\mathrm{corr},i}(j_m)$ and $\Delta G_{\mathrm{solv,i}}(j_k) = \Delta G_{\mathrm{Born,homotrans,i}}(j_k) + G_{\mathrm{back,homotrans,i}}(j_k) - G_{\mathrm{corr},i}(j_k)$ can be computed by continuum electrostatics (section 3.2.5). The energy difference between the two instances in the protein is the difference of their intrinsic energy $\Delta G_{\mathrm{intr,i}} = G_{\mathrm{intr},i}(j_k) - G_{\mathrm{intr},i}(j_m)$ (eq. 3.32).

3.3 Intrinsic Energies based on Quantum Chemical Data

The method of calculating so-called absolute[1] intrinsic energies, described here, allows to obtain microstate energies of sites for which no specific experimental values are available. Instead quantum mechanical (QM) calculations are used to parameterize the site. The procedure follows the thermodynamic cycle in Fig. 3.6.

Often it is difficult to synthesize model compounds with the same geometry as in the protein. Especially metal clusters are often instable, when they are extracted from the protein environment. The organic scaffold synthesized to stabilize the cluster and adjust its geometry is often so complex and non-protein like (very aromatic and hydrophobic) that it is questionable how well it is suited as model compound. Experiments with these model compounds are often made in complex organic solvents and not in water, which may lead to additional complications for the transferability of the results. In other cases as heme proteins, the puckering of the porphyrin or the out-of-plane position of the iron due to the fifth and sixth ligand may play an important role, which is hard to reproduce in synthetic model compounds. Further, a site may bind several ligands and there is a strong electronic coupling between the ligand binding sites. An example is quinone (section 5.5), which binds two electrons and two protons. Despite remarkable experimental attempts to characterize this biological important

[1] Absolute is used here in contrast to the relative intrinsic energy, where the shift in energy of an instance is calculated relative to the energy of the instance in a model compound in solution. Certainly, the underlying quantum mechanical calculations are not based on first principles, but the basis sets (section 2.3.1) and exchange correlation functions (section 2.3.1) are tuned to obtain chemically reasonable results. Furthermore, each standard reduction potential is (by definition) relative to the potential of the standard hydrogen electrode and each $\mathrm{p}K_a$ value is relative to the solvation energy of the proton. Both (closely related) values have to be measured and calculations are relative to these values.

system, there is still no complete picture. In other cases, the sites are only stable for a short time (as phosphoserine and phosphohistidine, section 5.2) or have many tautomers (*e.g.,* nucleic acids), that it is impossible to obtain pure microstates in experiment. As discussed in section 2.1, there are also sever limitations on the number of microstate energies which are experimentally obtainable from microscopic equilibrium constants.

For the method described here, only the (experimental) standard chemical potential of each ligand type in solution is needed, *i.e.,* the standard chemical potential of electrons and protons to characterize all proton and electron binding sites. Each instance of the site is parameterized by quantum chemical and continuum electrostatic calculations. The clear benefit is the independence of specific experimental data, but the disadvantage is that errors in the QM methods or the charge fitting enter the calculations, often leading to less accurate results as relative calculations, when appropriate experimental data is available. Rotameric or tautomeric instances of sites can be parameterized in exactly the same way.

The absolute intrinsic energy $G_{\mathrm{intr},i}(j_k)$ (in eq. 3.7) for an instance k of site j in conformer i is the sum of a homogeneous transfer energy (section 3.2.5) and energy terms, which can mostly be calculated by QM (section 2.3), *i.e.,* Born energy $\Delta G_{\mathrm{Born,homotrans,i}}(j_k)$ (eq. 3.25) and background energy $G_{\mathrm{back,homotrans,i}}(j_k)$ (eq. 3.26), the QM energy of formation $H_{\mathrm{vac},i}(j_k)$, the vibrational energy $G_{\mathrm{vib},i}(j_k)$, an energy correction $G_{\mathrm{corr},i}(j_k)$ and an energy of unbound ligand molecules $G_{\mathrm{free},i}(j_k)$:

$$G_{\mathrm{intr},i}(j_k) = \Delta G_{\mathrm{Born,homotrans,i}}(j_k) + G_{\mathrm{back,homotrans,i}}(j_k) + H_{\mathrm{vac},i}(j_k) + G_{\mathrm{vib},i}(j_k) - G_{\mathrm{corr},i}(j_k) + G_{\mathrm{free},i}(j_k) \tag{3.32}$$

The energy of formation (total bonding energy) $H_{\mathrm{vac},i}(j_k)$ is the energy of the site and all bound ligands given by the QM program after geometry optimization. The vibrational energy $G_{\mathrm{vib},i}(j_k)$ can be obtained by QM frequency calculations.

3.3.1 Energy Correction

If the quantum region is covalently bound to the protein, there are electrostatic interactions in the Poisson-Boltzmann energies between bonded atoms or those sharing a bond angle. These interactions would be excluded in normal force field calculations (section 2.4.1) because they depend on the electronic structure, *i.e.,* orbital geometries and can not be described properly by the Coulomb equation. Instead force fields include extra bonded and angle terms, which coarsely model these interactions. In case of relative ligand binding energy calculations, model compounds are defined, which include bonded atoms and those sharing a bond angle. Therefore, these critical electrostatic energies are canceled by subtracting the background energy of the site in the model compound and in the protein. For absolute ligand binding energy calculations this interaction energy can be calculated and the results corrected by

$$G_{\mathrm{corr},i}(j_k) = \sum_{a}^{N_{\mathrm{atom}}} \sum_{A}^{N_{\mathrm{bond\ a,angle\ a,i}}} Q_{\mathrm{protein,A,i}} \varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}}; q_{\mathrm{a,i}}(j_k)), \tag{3.33}$$

where A are those $N_{\mathrm{bond\ a,angle\ a,i}}$ atoms (with charge $Q_{\mathrm{protein,A,i}}$) of the background set, which are in bond or in angle relationship with atom a of the quantum region (Fig. 3.7). The cor-

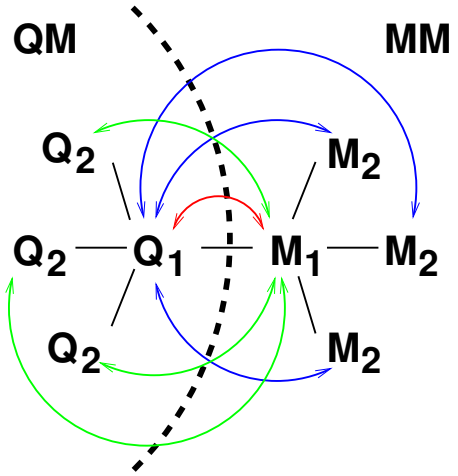


Figure 3.7. Correction for interactions between the quantum region (QM) and the region described by force field charges (MM). The atoms marked Q_1 and M_1 are covalently bound in the protein, but belong to the QM and MM region, respectively. The atoms Q_2 are covalently bound to Q_1 and the atoms M_2 are covalently bound to M_1. The electrostatic interaction between the bonded atoms Q_1 and M_1 (red), as well as the electrostatic interactions between the atoms having a bond angle, *i.e.*, $Q_2 - M_1$ (green) and $M_2 - Q_1$ (blue), would be excluded in a force field. The additional electrostatic interactions in the intrinsic energy $G_{\mathrm{intr},i}(j_k)$ can be canceled by a correction term $G_{\mathrm{corr},i}(j_k)$, which is calculated by additional electrostatic calculations only containing the appropriate atoms.

rection energy can also be given as input to avoid repetitive calculation or to include also other energy terms like the Lennard-Jones term, which is assumed to be constant between instances.

3.3.2 Energy of Free Ligands

The bulk solvent in the calculations is a mixture of water, ions and ligands. The model combines calculations of transfer energies in an implicit continuum model with explicit ligand molecules considered upon ligand binding. If the intrinsic energy of each instance $G_{\mathrm{intr},i}(j_k)$ (eq. 3.32) contains also the energy of formation and vibrational terms of the ligands bound to the instance ($H_{\mathrm{vac},i}(j_k)$ and $G_{\mathrm{vib},i}(j_k)$, respectively) as well as electrostatic energy terms ($\Delta G_{\mathrm{Born,homotrans,i}}(j_k)$, $G_{\mathrm{back,homotrans,i}}(j_k)$ and $G_{\mathrm{corr},i}(j_k)$), than the electrostatic energy terms, the energy of formation, as well as vibrational, translational and entropic terms of the free ligands need to be considered, too. The energy of unbound ligand molecules, $G_{\mathrm{free},i}(j_k)$ in eq. 3.32, accounts for those energy terms:

$$G_{\mathrm{free},i}(j_k) = \sum_{\lambda}^{N_{\mathrm{ligand}}} \nu_{\lambda,i}(j_k)\mu_{\lambda}^{\circ} = \sum_{\lambda}^{N_{\mathrm{ligand}}} \Big(\nu_{\lambda,i}(j) - n_{\lambda,i}(j_k)\Big)\mu_{\lambda}^{\circ} \tag{3.34}$$

The stoichiometric factor $\nu_{\lambda,i}(j_k)$, *i.e.*, the number of explicit unbound ligands, can be defined by the number of ligands bound to a particular instance j_k, $n_{\lambda,i}(j_k)$, and a reference number of ligands $\nu_{\lambda,i}(j)$. The number of bound ligands is identical with $n_{\lambda,i}(j_k)$ in eq. 3.7.

The reference number of ligands $\nu_{\lambda,i}(j)$ can be freely chosen shifting the reference point in energy, however only a few choices have a physical meaning associated. $\nu_{\lambda,i}(j)$ may be the maximal number of ligands λ site j can bind, so that $n_{\lambda,i}(j_k) = \nu_{\lambda,i}(j)$ in the fully bound form. An alternative useful definition could be that $\nu_{\lambda,i}(j)$ equals the maximum number of ligands λ, which can bind to the protein at the same time or that $\nu_{\lambda,i}(j)$ equals the total number of ligands λ in the simulation. The advantage of the first definition is, that the number of binding sites of the molecule for a certain ligand λ is not required for having a pool of ligands large enough to fill all binding sites. Also the total number of ligands of type λ in the system is problematic, because it depends on the chemical potential μ_λ of ligand type λ in the bulk. By the first definition only the number of ligands $n_{\lambda,i}(j_k)$ of type λ bound to a certain instance j_k is necessary. To parameterize the instance j_k with $n_{\lambda,i}(j_k)$ ligands bound by QM calculations, the composition of ligands and their orientation is required anyway. This definition makes the input for instances of sites easier transferable between different systems.

The standard chemical potential μ°_λ of a single unbound ligand ($\nu_\lambda = 1$) of type λ is given as input $\mu^\circ_\lambda = G^\circ_\lambda - RT \ln \sigma(\lambda)$. Here, $\sigma(\lambda)$ is the symmetry factor of ligand λ in bulk solution (see discussion in section 3.1.1). For electrons, protons and spherical ions $\sigma(\lambda) = 1$. For other molecules $\sigma(\lambda)$ may have different values, *i.e.*, $\sigma(H_2O) = 2$, $\sigma(H_3O^+) = 3$, $\sigma(HPO_4^{2-}) = 12$ *etc.*. The last example takes the four oxygens the proton can be bound to into account as well as the three isoenergetic rotamers the proton can have in each of the four bound forms.

The standard chemical potential μ°_λ has to be calculated in analogy to the site (eq. 3.32). In particular, the terms H_{vac} and G_{vib} have to be calculated by the same QM method. The solvation energy may be calculated by continuum electrostatic methods or experimentally obtained. The background energy term, correction energy term and interaction energy term are usually zero for a ligand.

The standard chemical potential $\mu^\circ_{H^+}$ of a single free proton can be calculated as:

$$\mu^\circ_{H^+} = H_{\mathrm{vac}}(H^+) + \Delta G_{\mathrm{Born}}(H^+) + G_{\mathrm{trans}}(H^+) + \Delta(PV) - T[S(H^+)] \tag{3.35}$$

$H_{\mathrm{vac}}(H^+)$ is the energy of formation of the proton. Dependent on the definition of the zero-point in energy of the QM program it might be zero (Gaussian) or not (ADF). The Born solvation energy of the proton $\Delta G_{\mathrm{Born}}(H^+) = -260.5 \frac{\mathrm{kcal}}{\mathrm{mol}}$ can be calculated from the experimentally measured potential of the standard hydrogen electrode [119]. The translational energy of a proton $G_{\mathrm{trans}}(H^+) = \frac{3}{2} k_B T$ and the energy change due to the volume change in the gas phase reaction $\Delta(PV) = k_B T$ are estimated from the ideal gas approximation. The entropic portion of the gas-phase free energy of the proton $T[S(H^+)] = 7.8 \frac{\mathrm{kcal}}{\mathrm{mol}}$ is derived from the Sackur-Tetrode equation.

The standard chemical potential $\mu^\circ_{e^-}$ of a single free electron can be calculated from the experimental potential of the standard hydrogen electrode $\Delta\mathrm{SHE} \approx -4.43V$ [120]:

$$\mu^\circ_{e^-} = \mathcal{F} \Delta\mathrm{SHE} \tag{3.36}$$

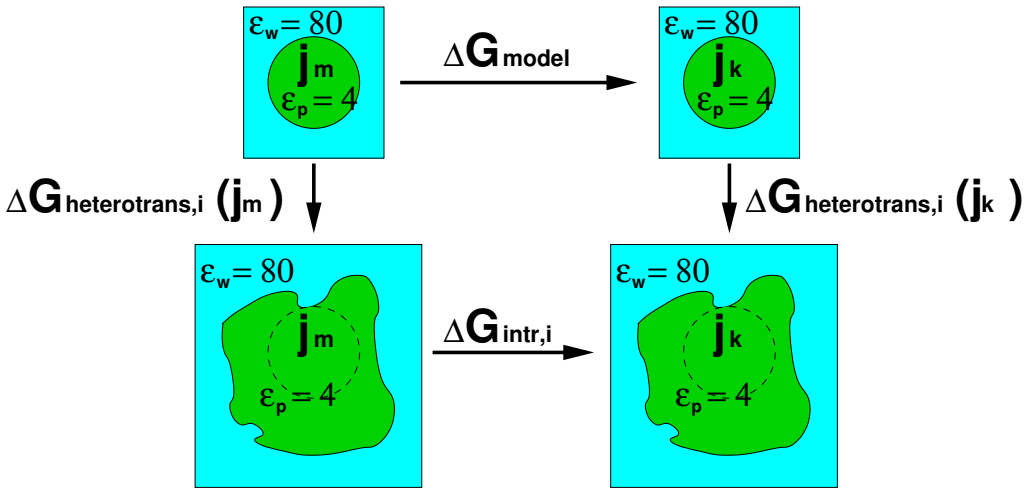


Figure 3.8. Intrinsic energy calculation relative to a model compound (without rotamers). The energy difference $\Delta G_{\mathrm{model}}$ between two charge forms j_m and j_k of a model compound is known in solution. The heterogeneous transfer energies $\Delta G_{\mathrm{heterotrans,i}}(j_k)$ and $\Delta G_{\mathrm{heterotrans,i}}(j_m)$ can be computed by continuum electrostatics (section 3.2.6). The energy difference between the two charge forms in the protein is the difference of their intrinsic energy $G_{\mathrm{intr},i}(j_k)$ (eq. 3.37).

3.4 Intrinsic Energies based on Experimental and Molecular Mechanics Data

Using the thermodynamic cycle shown in Fig. 3.8, relative intrinsic energy calculations allow to obtain microstate energies for experimentally well characterized sites. Model compounds have been synthesized, which are geometrically and chemically similar to the site in the protein environment. Equilibrium constants have been measured, which allow to calculate microstate energies of the model compound in solution. For example, the sidechain and backbone $\mathrm{p}K_a$ values of all amino acids are measured in tripeptides and dipeptides, respectively. These results can be used to calculate the $\mathrm{p}K_a$ shift for the amino acids in the protein environment by electrostatic methods. Such calculations generally show good agreement with measurements of apparent $\mathrm{p}K_a$ values in the protein. Also for other ligands such calculations can be done. Unlike the absolute intrinsic energy calculations, relative intrinsic energy calculations do not depend on computationally expensive quantum mechanical calculations and errors in the charge model seem to cancel out quite well.

Relative intrinsic energies can be calculated for sites with more than one rotamer form by molecular (or quantum) mechanics. Rotamers allow to include sidechain flexibility and different hydrogen bond networks (due to rotation of hydroxyl groups) into continuum electrostatic calculations. Sites may bind ligands and have multiple rotamer forms.

The relative intrinsic energy $G_{\mathrm{intr},i}(j_k)$ for an instance k of site j is the sum of a transfer energy (section 3.2.5 or section 3.2.6), the energy of a model compound in the same instance (charge

and rotamer form) $\Delta G_{\mathrm{model}}(j_k)$ and the internal energy of the rotamer form $\Delta G_{\mathrm{rotamer}}(j_k)$:

$$G_{\mathrm{intr},i}(j_k) = \Delta G_{\mathrm{Born},i}(j_k) + \Delta G_{\mathrm{back},i}(j_k) + \Delta G_{\mathrm{model}}(j_k) + \Delta G_{\mathrm{rotamer}}(j_k) \tag{3.37}$$

The model energy $\Delta G_{\mathrm{model}}(j_k)$ is the ligand binding energy of the model compound relative to a reference charge form. For a site binding a single proton, the model energy of the protonated form is $\Delta G_{\mathrm{model}}(\mathrm{protonated}) = -RT \ln 10 \mathrm{p}K_{a,\mathrm{model}}$ and the model energy of the deprotonated form is $\Delta G_{\mathrm{model}}(\mathrm{deprotonated}) = 0$ (or $\Delta G_{\mathrm{model}}(\mathrm{protonated}) = 0$ and $\Delta G_{\mathrm{model}}(\mathrm{deprotonated}) = RT \ln 10 \mathrm{p}K_{a,\mathrm{model}}$, respectively). The energy of formation, vibrational and entropic terms as well as the energy associated with the unbound ligand are all hidden in the experimental model energy.

The differences in Born energy $\Delta G_{\mathrm{Born},i}(j_k)$ and background energy $\Delta G_{\mathrm{back},i}(j_k)$ are the difference in electrostatic energy of the site in the model compound environment and in the protein environment. For sites having different charge forms heterogeneous transfer energies (section 3.2.6) and for sites having different rotamer forms homogeneous transfer energies (section 3.2.5) are calculated, respectively. For sites having both, different charge and rotamer forms, both transfer energies are computed.

In theory, each rotamer form would have a slightly different model energy. Usually, the rotamer form is not determined together with the model energy, but the model energy is measured for an ensemble of rotamers in solution. Therefore, it is probably the best approximation to attribute the model energy to the rotamer with the lowest internal energy (which is therefore the most populated in the experimental ensemble) and calculate the intrinsic energies of all instances relative to this rotamer form. This reference rotamer has a rotamer energy $\Delta G_{\mathrm{rotamer}}(j_k) = 0$ and all non-reference rotamers have a rotamer energy relative to the reference rotamer.

3.4.1 Non-Ligand Binding Reference Rotamer Form

Site j has several rotamer forms, but only one charge form, and instance k is chosen to be the reference rotamer form. Therefore, the intrinsic energy is set to zero

$$G_{\mathrm{intr},i}(j_k) = 0 \tag{3.38}$$

and the energy of all other rotamers is given relative to this instance.

In Fig. 3.9, instance $j_k = q_1 r_1$ would be a reference rotamer form without model compound, if only the red thermodynamic cycle would be calculated.

3.4.2 Ligand Binding Reference Rotamer Form

Site j has several charge forms and one or more rotamer forms. The rotamer form of instance k is chosen to be the reference rotamer for the site j. There has to be one reference instance *per* charge form, but all reference instances have to belong to the same rotamer form. The intrinsic energy

$$G_{\mathrm{intr},i}(j_k) = \Delta G_{\mathrm{Born,heterotrans,i}}(j_k) + \Delta G_{\mathrm{back,heterotrans,i}}(j_k) + \Delta G_{\mathrm{model}}(j_k) \tag{3.39}$$

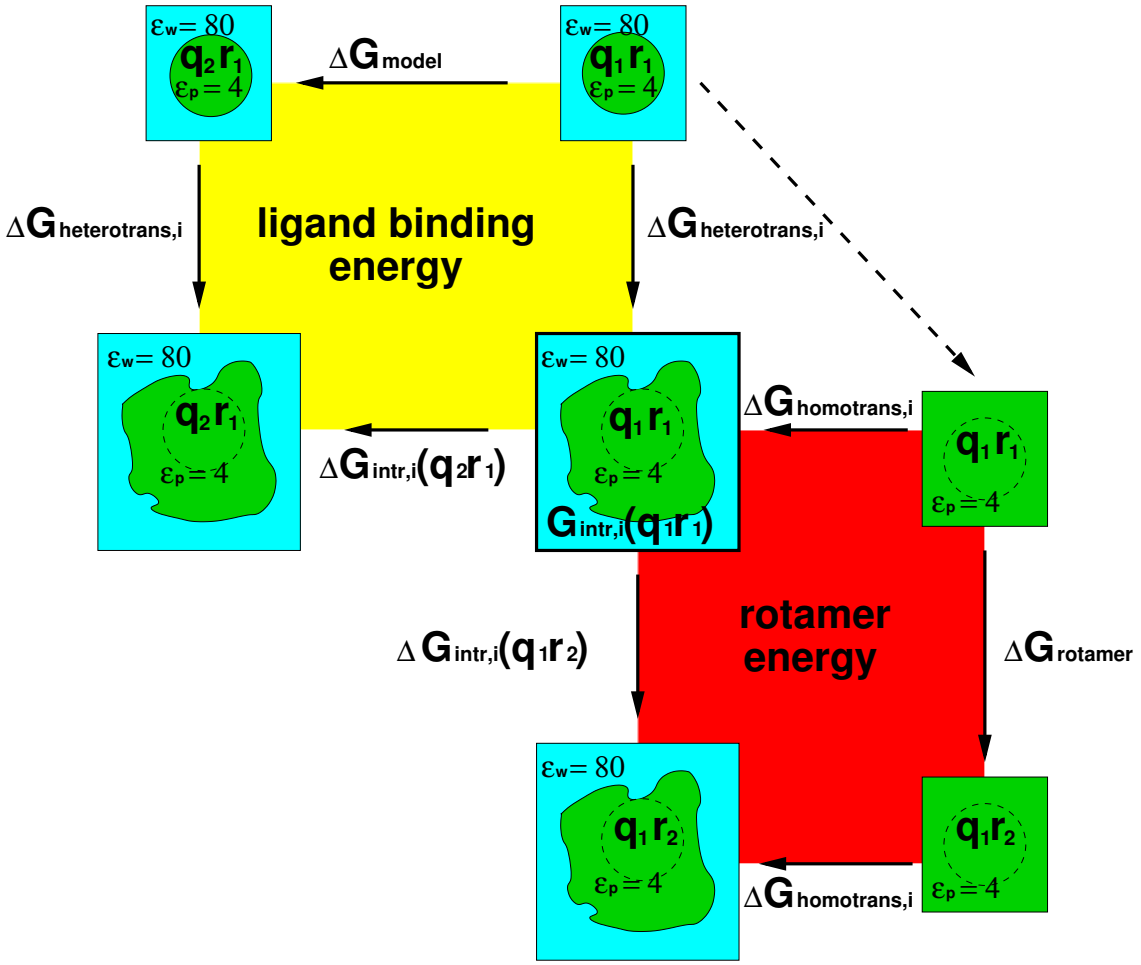


Figure 3.9. Thermodynamic cycles for relative energy calculations. The red cycle describes the calculation of rotamer energies, the yellow cycle describes the calculation of ligand binding energies. The site j with chargeset q_1 and coordinate set r_1 is the reference instance for both charges and rotamers (thick frame). The intrinsic energy of the instance is $G_{\mathrm{intr},i}(j_{\mathrm{ref}}) = G_{\mathrm{intr,i}}(q_1r_1) = 0$, if all other instances differ only in the coordinate set (*e.g.,* q_1r_2, section 3.4.1). If also other chargeforms exist (*e.g.,* q_2r_1), the intrinsic energy of the instance is $G_{\mathrm{intr},i}(j_{\mathrm{ref}}) = G_{\mathrm{intr,i}}(q_1r_1) = \Delta G_{\mathrm{heterotrans,i}}(q_1r_1)$. If the instance is in the reference rotamer form, but not the reference chargeform, the intrinsic energy is $G_{\mathrm{intr,i}}(q_2r_1) = \Delta G_{\mathrm{heterotrans,i}}(q_2r_1) + \Delta G_{\mathrm{model}}$ (section 3.4.2). If the instance is not the reference rotamer, but no other chargeforms exist, the intrinsic energy is $G_{\mathrm{intr,i}}(q_1r_2) = \Delta G_{\mathrm{homotrans,i}}(q_1r_2) - \Delta G_{\mathrm{homotrans,i}}(q_1r_1) + \Delta G_{\mathrm{rotamer}}(j_k)$ (section 3.4.3). If the instance is not the reference rotamer and other chargeforms exist, the intrinsic energy is $G_{\mathrm{intr,i}}(q_1r_2) = \Delta G_{\mathrm{homotrans,i}}(q_1r_2) - \Delta G_{\mathrm{homotrans,i}}(q_1r_1) + \Delta G_{\mathrm{rotamer}}(j_k) + \Delta G_{\mathrm{heterotrans,i}}(q_1r_1)$. The equation can be simplified following the dashed arrow (section 3.4.4).

contains a heterogeneous transfer energy (section 3.2.6) and a model energy $\Delta G_{\mathrm{model}}(j_k)$, which is zero for the reference charge form. The rotamer energy $\Delta G_{\mathrm{rotamer}}(j_k)$ is always zero in a reference rotamer form.

In Fig. 3.9, instance $j_k = q_1 r_1$ would be a reference rotamer form with model compound, if the yellow (and maybe the red) thermodynamic cycle would be calculated.

3.4.3 Non-Ligand Binding Non-Reference Rotamer Form

Site j has several rotamer forms, but only one charge form, and instance k is not chosen to be the reference rotamer form.

A homogeneous transfer energy (section 3.2.5) of instance j_k relative to the reference rotamer j_{ref} is calculated. The model energy $\Delta G_{\mathrm{model}}(j_k)$ is zero. The rotamer energy $\Delta G_{\mathrm{rotamer}}(j_k)$ is the difference in internal energy of instance k relative to the reference rotameric form. Therefore, the intrinsic energy is given by

$$\begin{aligned} G_{\mathrm{intr},i}(j_k) =& \Delta G_{\mathrm{Born,homotrans,i}}(j_k) - \Delta G_{\mathrm{Born,homotrans,i}}(j_{\mathrm{ref}}) + G_{\mathrm{back,homotrans,i}}(j_k) \\ & - G_{\mathrm{back,homotrans,i}}(j_{\mathrm{ref}}) + G_{\mathrm{rotamer}}(j_k) - G_{\mathrm{rotamer}}(j_{\mathrm{ref}}). \end{aligned} \tag{3.40}$$

In Fig. 3.9, instance $j_k = q_1 r_2$ would be a non-reference rotamer form without model compound, if only the red thermodynamic cycle would be calculated. The difference in homogeneous transfer energy between instance $q_1 r_2$ and $j_{\mathrm{ref}} = q_1 r_1$ needs to be calculated.

3.4.4 Ligand Binding Non-Reference Rotamer Form

Site j has several rotamer forms and several charge forms. Instance k is not chosen to be the reference rotamer form.

The difference in homogeneous transfer energy (section 3.2.5) of instance j_k relative to the reference rotamer j_{ref} plus the heterogeneous transfer energy (section 3.2.6) of instance j_{ref} has to be calculated:

$$\Delta G_{\mathrm{Born},i}(j_k) = G_{\mathrm{Born,protein,i}}(j_k) - G_{\mathrm{Born,homo,i}}(j_k) + G_{\mathrm{Born,homo,i}}(j_{\mathrm{ref}}) - G_{\mathrm{Born,model,i}}(j_{\mathrm{ref}}) \tag{3.41}$$

$$\Delta G_{\mathrm{back},i}(j_k) = G_{\mathrm{back,protein,i}}(j_k) - G_{\mathrm{back,model,i}}(j_{\mathrm{ref}}) \tag{3.42}$$

The energies $G_{\mathrm{Born,protein,i}}(j_{\mathrm{ref}})$ and $G_{\mathrm{back,protein,i}}(j_{\mathrm{ref}})$ would occur for the homogeneous transfer energy with a negative sign and for the heterogeneous transfer with a positive sign and therefore cancel out. The model energy $\Delta G_{\mathrm{model}}(j_{\mathrm{ref}})$ is specified for the the model compound in the reference rotameric form. The rotamer energy $\Delta G_{\mathrm{rotamer}}(j_k)$ is the difference in internal energy of instance k relative to the reference rotameric form. Hence, the intrinsic energy can be written as

$$G_{\mathrm{intr},i}(j_k) = \Delta G_{\mathrm{Born},i}(j_k) + \Delta G_{\mathrm{back},i}(j_k) + \Delta G_{\mathrm{model}}(j_{\mathrm{ref}}) + G_{\mathrm{rotamer}}(j_k) - G_{\mathrm{rotamer}}(j_{\mathrm{ref}}). \tag{3.43}$$

In Fig. 3.9, instance $j_k = q_1 r_2$ would be a non-reference rotamer form with model compound, if both the red and the yellow thermodynamic cycle would be calculated. The difference in homogeneous transfer energy between instance $q_1 r_2$ and $j_{\mathrm{ref}} = q_1 r_1$ and the heterogeneous transfer energy of instance $q_1 r_1$ needs to be calculated.

3.5 Comparison to Previously Used Energy Functions

3.5.1 Calculations based on Experimental Data

In this work, a state vector $\vec{x}_{\mathrm{i,n}}$ (eq. 3.4) was introduced, which allows a varying number of instances $k \equiv x^j_{i,n}$ for each site j. Many binding events, however, can be described by two forms, a bound form and an unbound form only. Previous works used a binary state vector, so that $k = 0$ in the unbound form and $k = 1$ in the bound form. For a protein with $N_{\mathrm{site},i}$ binding sites and two forms each, the total number of microstates is $N_{\mathrm{micro},i} = 2^{N_{\mathrm{site},i}}$. The contribution of a site to the microstate energy was described as an energy difference between bound and unbound form instead of energies associated with each individual instance. Instead of the general description by chemical potentials and energies, specific formulations were used as pH and $\mathrm{p}K_a$ or $\mathcal{E}$ and $\mathcal{E}^\circ$ [121, 122]. In fact, the program used before (Multiflex), is only thought to be used for $\mathrm{p}K_a$ calculations. Standard reduction potential calculations have to be done in $\mathrm{p}K_a$-units. The user has to keep track of the different ligand types and recalculate energies for subsequent titration calculations. Therefore, only $\mathrm{p}K_a$ calculations will be discussed here.

Multiflex allows only relative $\mathrm{p}K_a$ calculations (relative to a model compound, section 3.4) in a two dielectric environment (protein and solvent). Instead of the intrinsic energy $G_{\mathrm{intr},i}(j_k)$ (eq. 3.37) an intrinsic $\mathrm{p}K_a$ value is calculated:

$$\mathrm{p}K_{a,\mathrm{intr},i}(j) = \mathrm{p}K_{a,\mathrm{model}}(j) - \frac{1}{RT \ln 10}\left(\Delta\Delta G_{\mathrm{Born},i}(j) + \Delta\Delta G_{\mathrm{back},i}(j)\right) \tag{3.44}$$

The $\mathrm{p}K_a$ value describes the deprotonation equilibrium of an acid (section 2.1.4) at site j. The double differences in Born and background energy, describes first the difference between the model compound environment and the protein environment in the heterogeneous transfer (section 3.2.6) and second the difference between the protonated $j^{\blacklozenge}$ and deprotonated $j^{\diamondsuit}$ charge form.

$$\Delta\Delta G_{\mathrm{Born},i}(j) = \Delta G_{\mathrm{Born,heterotrans,i}}(j^{\blacklozenge}) - \Delta G_{\mathrm{Born,heterotrans,i}}(j^{\diamondsuit}) \tag{3.45}$$

$$\Delta\Delta G_{\mathrm{back},i}(j) = \Delta G_{\mathrm{back,heterotrans,i}}(j^{\blacklozenge}) - \Delta G_{\mathrm{back,heterotrans,i}}(j^{\diamondsuit}) \tag{3.46}$$

The model $\mathrm{p}K_a$ value, $\mathrm{p}K_{a,\mathrm{model}}(j)$, is the experimental $\mathrm{p}K_a$ value of a model compound of site j in aqueous solution. The intrinsic $\mathrm{p}K_a$ value, $\mathrm{p}K_{a,\mathrm{intr},i}(j)$ is here directly the $\mathrm{p}K_a$ value of site j if all other sites are in their reference state.

Microstate energies are calculated relative to a reference state, which is chosen to be the neutral protonation form. The protein is not divided into a background charge set and the interaction energy of instances of sites, but the background energy includes all interactions of site j with all other sites in their reference charge form.

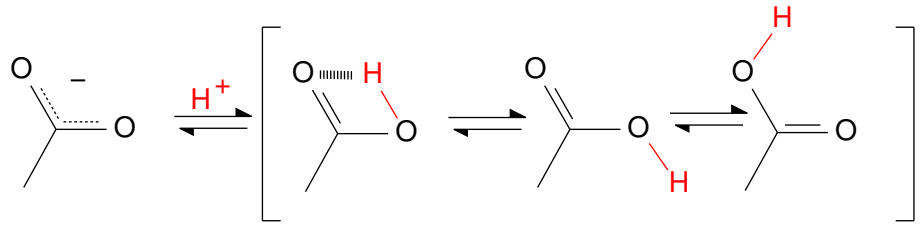


Figure 3.10. A carboxylic acid has different tautomers and rotamers in the protonated form.

The interaction energy $\Delta\Delta G_{\mathrm{inter,i}}(j,l)$ between sites j and l can be written as double difference of interaction energies $G_{\mathrm{inter},i}(j_k,l_m)$ (eq. 3.31):

$$\Delta\Delta G_{\mathrm{inter,i}}(j,l) = \Big(G_{\mathrm{inter},i}(j^{\blacklozenge},l^{\blacklozenge}) - G_{\mathrm{inter},i}(j^{\diamondsuit},l^{\blacklozenge})\Big) - \Big(G_{\mathrm{inter},i}(j^{\blacklozenge},l^{\diamondsuit}) - G_{\mathrm{inter},i}(j^{\diamondsuit},l^{\diamondsuit})\Big) \tag{3.47}$$

The microstate energy can be written (using equations from section 2.1.4) as:

$$\begin{aligned} G_{\mathrm{micro}}(\vec{x}_{\mathrm{i,n}},\mathrm{pH}) = G_{\mathrm{conf},i} &+ \sum_{j}^{N_{\mathrm{site},i}} [x^j_{i,n} - x^j_{i,\mathrm{ref}}]\Big[RT\ln 10\Big(\mathrm{pH} - \mathrm{p}K_{a,\mathrm{intr},i}(j)\Big)\Big] \\ &+ \frac{1}{2}\sum_{j}^{N_{\mathrm{site},i}}\sum_{l\neq j}^{N_{\mathrm{site},i}} [x^j_{i,n} - x^j_{i,\mathrm{ref}}][x^l_{i,n} - x^l_{i,\mathrm{ref}}]\Delta\Delta G_{\mathrm{inter,i}}(j,l) \end{aligned} \tag{3.48}$$

As in eq. 3.7, $G_{\mathrm{conf},i}$ is the conformational energy. The first sum runs over all sites to calculate the difference of intrinsic $\mathrm{p}K_a$ value (intrinsic energy) and pH (chemical potential of protons in the bulk solvent) and the second sum adds the interaction energy with all other sites. Sites only contribute to the microstate energy, if they are in a different protonation form than in the reference state and interaction energy is only added if both sites are not in their reference form. The sign of the contribution is positive if the site in the reference form ($x^j_{i,\mathrm{ref}}$ and $x^l_{i,\mathrm{ref}}$, respectively) is deprotonated and the site in the form of interest is deprotonated, which is the case for bases. Acids have a protonated reference form and the contribution is negative. In the generalized formulation this complexity arising from the filter function ($[x^j_{i,n} - x^j_{i,\mathrm{ref}}]$ and $[x^l_{i,n} - x^l_{i,\mathrm{ref}}]$, respectively) is avoided, because the energies are not calculated relative to a physical reference state. Instead an artificial state is chosen as reference state, where all atoms belonging to instances of sites are uncharged.

Treatment of Carboxyl-, Amine-, Hydroxyl- and Thiol-Groups

In the last two decades, the approach described above was successfully used for many applications [13, 35–44], even though the calculation of $\mathrm{p}K_a$ values with a single protonated form is strictly only valid for sites with no tautomers or rotamers. Since no amino acid strictly fulfills this requirement for proton binding (Fig. 3.10 and Fig. 3.11), a single average charge distribution (set of partial charges) describes the bound form and another the unbound form of the site. The carboxyl-group of acidic amino acid sidechains (*i.e.*, aspartate and glutamate) and

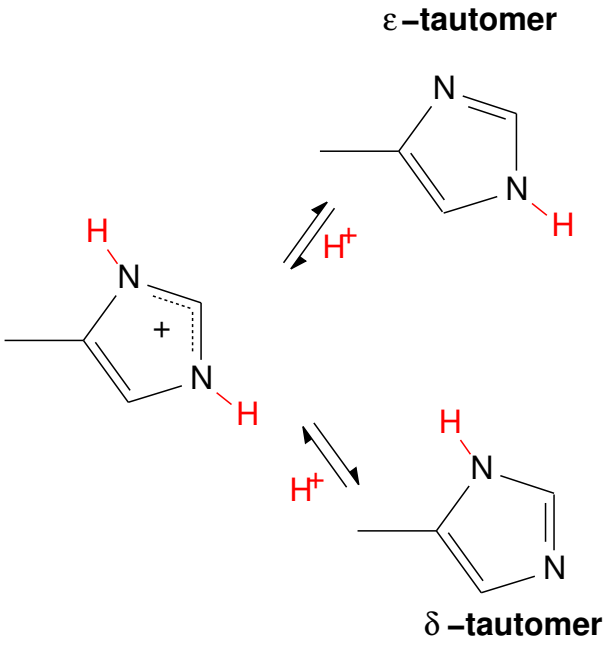


Figure 3.11. Histidine has two tautomers in the singly protonated form (protonated on $N\delta 1$ or $N\epsilon 2$ with $\mathrm{p}K_a$ values of 7.0 and 6.6 in solution, respectively). The doubly protonated form is shown as well. The doubly deprotonated form has a $\mathrm{p}K_a$ value of about 14.0 in solution and can therefore be excluded for most calculations.

the C-terminus can be described by a single protonated form, where the proton has a position between the two oxygens (as the first tautomer in Fig. 3.10). The average charge distribution avoids the problem of the two tautomers and the rotameric forms, where the protonated acid is a hydrogen bond donor. Also the basic amino acids, *i.e.*, lysine and the N-terminus can be described by a amine-group with or without positive charge and arginine can be described by a charged or uncharged guanidinium group. For hydroxyl- and thiol-groups (of cystein, tyrosine, serine and threonine), however, this method is problematic, since the calculated $\mathrm{p}K_a$ values of neighboring titrateable groups depends strongly on the presence of hydrogen bonds. The orientation of the hydrogen atom has to be assigned before the $\mathrm{p}K_a$ calculations introducing a bias if different hydrogen bond networks are possible [45]. Certainly, different hydrogen bonds would also affect, the $\mathrm{p}K_a$ value of the hydroxyl- and thiol-groups, but usually these groups were not considered as titrateable due to their high $\mathrm{p}K_a$ value. However, to study proton-transfer (section 5.6) this approximation is not sustainable.

Treatment of Histidine

The amino acid, for which the shortcomings of the approach using a binary state vector becomes most obvious is histidine (Fig. 3.11). The imidazole ring of histidine has two tautomers in the singly protonated form, which have the proton located on one or the other side of the imidazole ring. Therefore, the hydrogen bonding pattern is clearly different and also the tautomers have different $\mathrm{p}K_a$ values. The doubly deprotonated form has a $\mathrm{p}K_a$ value, which is too high to be populated under physiological conditions. The solution of Bashford and

coworkers [123] treats the tautomers as conformers, *i.e.,* performing a continuum electrostatic calculations of the complete protein for the δ- and ϵ-tautomer separately. Once the protonation of the δ-tautomer at $N\epsilon 2$ is permitted to yield the doubly protonated form and once the protonation and of the ϵ-tautomer at $N\delta 1$ is permitted to yield the doubly protonated form. Postprocessing of the calculated energies (by an additional program) is done to obtain a binary state vector with two bits for histidine. The fully deprotonated form, which was never calculated, but exists in the state vector, was assigned an arbitrary high energy to avoid population in the titration calculations. All interaction energies with histidine have to be modified to be correct for the modified physical model. For N histidines $2 + N$ Multiflex calculations (titrating all residues) are necessary. The results are correct in the physiological pH range, but the methodology is computational expensive for many histidines and non-intuitive. It is not transferable with acceptable effort to other sites like quinones (section 5.5).

Treatment of Rotamers

The insufficiency of the approach doing calculations with a binary state vector became even more evident, when You and Bashford included rotamer forms into the $\mathrm{p}K_a$ value calculations [51]. Effectively, the You and Bashford treatment of rotamers, calculates a thermodynamic average $\mathrm{p}K_{a,\mathrm{intr},i}$ for a site, when all other sites are in their reference form. It would be correct to include the rotamer forms equivalently to the charge forms as independent microstates. This would allow to adopt the rotamer population to the charge forms of surrounding sites. However, such a treatment is not possible with a binary state vector and requires a more general physical model, as it is described in this work.

For inclusion of rotamers into $\mathrm{p}K_a$ calculations, eq. 3.44 is extended by You and Bashford [51] ($\beta = \frac{1}{RT}$):

$$\begin{aligned}\mathrm{p}K_{a,\mathrm{intr},i}(j) = \mathrm{p}K_{a,\mathrm{model}}(j) - \frac{1}{RT\ln 10}\ln \sum_{r}^{N_{\mathrm{rotamer},i,j}} \Bigg[&\exp\Big(-\beta\Delta G_{\mathrm{env},i}(j^{\diamondsuit}_{r,\mathrm{protein}})\Big) \\ &- \exp\Big(-\beta\Delta G_{\mathrm{env},i}(j^{\blacklozenge}_{r,\mathrm{protein}})\Big) \\ &- \exp\Big(-\beta\Delta G_{\mathrm{env},i}(j^{\diamondsuit}_{r,\mathrm{model}})\Big) \\ &+ \exp\Big(-\beta\Delta G_{\mathrm{env},i}(j^{\blacklozenge}_{r,\mathrm{model}})\Big)\Bigg]\end{aligned} \tag{3.49}$$

The protonated $j^{\blacklozenge}$ and unprotonated $j^{\diamondsuit}$ form of site j in rotamer r are calculated in the protein environment j_{protein} and model environment j_{model}. The free energy $\Delta G_{\mathrm{env},i}$ is given as the sum of Born $\Delta G_{\mathrm{Born},i}$, background $\Delta G_{\mathrm{back},i}$ and a non-electrostatic, charge form independent energy $\Delta G_{\mathrm{rotamer}}$:

$$\Delta G_{\mathrm{env},i} = \Delta G_{\mathrm{Born},i} + \Delta G_{\mathrm{back},i} + \Delta G_{\mathrm{rotamer}} \tag{3.50}$$

The Born and background energies are calculated as homogeneous transfer energies (section 3.2.5):

$$\Delta G_{\mathrm{Born},i}(j^{\diamondsuit}_{r,\mathrm{protein}}) = \Delta G_{\mathrm{Born,homotrans,i}}(j^{\diamondsuit}_{r,\mathrm{protein}}) = G_{\mathrm{Born,protein,i}}(j^{\diamondsuit}_{r}) - G_{\mathrm{Born,homo,i}}(j^{\diamondsuit}_{r})$$
$$\Delta G_{\mathrm{Born},i}(j^{\blacklozenge}_{r,\mathrm{protein}}) = \Delta G_{\mathrm{Born,homotrans,i}}(j^{\blacklozenge}_{r,\mathrm{protein}}) = G_{\mathrm{Born,protein,i}}(j^{\blacklozenge}_{r}) - G_{\mathrm{Born,homo,i}}(j^{\blacklozenge}_{r})$$
$$\Delta G_{\mathrm{Born},i}(j^{\diamondsuit}_{r,\mathrm{model}}) = \Delta G_{\mathrm{Born,homotrans,i}}(j^{\diamondsuit}_{r,\mathrm{model}}) = G_{\mathrm{Born,model,i}}(j^{\diamondsuit}_{r}) - G_{\mathrm{Born,homo,i}}(j^{\diamondsuit}_{r})$$
$$\Delta G_{\mathrm{Born},i}(j^{\blacklozenge}_{r,\mathrm{model}}) = \Delta G_{\mathrm{Born,homotrans,i}}(j^{\blacklozenge}_{r,\mathrm{model}}) = G_{\mathrm{Born,model,i}}(j^{\blacklozenge}_{r}) - G_{\mathrm{Born,homo,i}}(j^{\blacklozenge}_{r})$$

$$\Delta G_{\mathrm{back},i}(j^{\diamondsuit}_{r,\mathrm{protein}}) = G_{\mathrm{back,homotrans,i}}(j^{\diamondsuit}_{r,\mathrm{protein}}) = G_{\mathrm{back,protein,i}}(j^{\diamondsuit}_{r})$$
$$\Delta G_{\mathrm{back},i}(j^{\blacklozenge}_{r,\mathrm{protein}}) = G_{\mathrm{back,homotrans,i}}(j^{\blacklozenge}_{r,\mathrm{protein}}) = G_{\mathrm{back,protein,i}}(j^{\blacklozenge}_{r})$$
$$\Delta G_{\mathrm{back},i}(j^{\diamondsuit}_{r,\mathrm{model}}) = G_{\mathrm{back,homotrans,i}}(j^{\diamondsuit}_{r,\mathrm{model}}) = G_{\mathrm{back,model,i}}(j^{\diamondsuit}_{r})$$
$$\Delta G_{\mathrm{back},i}(j^{\blacklozenge}_{r,\mathrm{model}}) = G_{\mathrm{back,homotrans,i}}(j^{\blacklozenge}_{r,\mathrm{model}}) = G_{\mathrm{back,model,i}}(j^{\blacklozenge}_{r})$$

The Born energy needs the additional subtraction of the potential of the site in a homogeneous dielectric $\varphi_{\mathrm{homo}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i}}(j_r))$ with dielectric constant of the interior of the site. This term is needed to remove grid artefacts, which were canceled in the single-rotamer calculation by the difference of protein and model compound energy.

Additionally to the terms in eq. 3.50, one might need to consider an additional term $\Delta G_{\mathrm{selfback},i}$, which is the homogeneous transfer energy of the background charges of the protein

$$\rho_{\mathrm{i,j,r,protein}} = \sum_{a}^{N_{i,j,r,\mathrm{protein}}} Q_{\mathrm{a,i}}(j_{r,\mathrm{protein}}) \tag{3.51}$$

or model compound

$$\rho_{\mathrm{i,j,r,model}} = \sum_{a}^{N_{i,j,r,\mathrm{model}}} Q_{\mathrm{a,i}}(j_{r,\mathrm{model}}), \tag{3.52}$$

which changes due to the changing boundary with rotamer r of site j. If the boundary changes significantly, one might even need to include the surface area dependent non-electrostatic part of the solvation energy.

$\Delta G_{\mathrm{selfback},i}$ is computed as:

$$\begin{aligned}\Delta G_{\mathrm{selfback},i}(j^{\diamondsuit}_{r,\mathrm{protein}}) &= \Delta G_{\mathrm{selfback},i}(j^{\blacklozenge}_{r,\mathrm{protein}}) \\ &= \frac{1}{2}\sum_{a}^{N_{i,j,r,\mathrm{protein}}} Q_{\mathrm{a,i}}(j_{r,\mathrm{protein}})[\ \varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i,j,r,protein}}) \\ &\qquad -\varphi_{\mathrm{homo}}(\vec{r}_{\mathrm{a,i}}; \rho_{\mathrm{i,j,r,protein}})]\end{aligned} \tag{3.53}$$

$$\begin{aligned}\Delta G_{\text{selfback},i}(j^{\diamondsuit}_{r,\text{model}}) &= \Delta G_{\text{selfback},i}(j^{\blacklozenge}_{r,\text{model}}) \\ &= \frac{1}{2}\sum_{a}^{N_{i,j,r,\text{model}}} Q_{\text{a,i}}(j_{r,\text{model}})[\ \varphi_{\text{model}}(\vec{r}_{\text{a,i}};\rho_{\text{i,j,r,model}}) \\ &\qquad -\varphi_{\text{homo}}(\vec{r}_{\text{a,i}};\rho_{\text{i,j,r,model}})]\end{aligned} \tag{3.54}$$

The electrostatic potentials have to be calculated on a grid, which has to be larger than the protein, but should be as fine as the grid of the finest focussing step of the site. Practical calculations on such a grid are not feasible due to memory requirements and computational cost. For small changes of the dielectric boundary, this term is small, that I excluded it from my calculation scheme assuming a combined dielectric boundary. In fact, in many cases, the groups are buried and the dielectric boundary is unchanged. In cases, where the change is significant, conformers can be used.

Despite the problematic inclusion of the rotamer energy into the intrinsic $\mathrm{p}K_a$ value, the calculation scheme including a molecular mechanics energy $\Delta G_{\text{rotamer}}$ and a thermodynamic cycle using homogeneous transfers is very similar to the approach I describe in this work. However, my distinction between reference and non-reference rotamers avoids computation of $G_{\text{Born,model,i}}(j_r)$ and $G_{\text{back,model,i}}(j_r)$ for each non-reference rotamer in each charge form. For calculations with many rotamer forms and charge forms *per* site this optimization can lead to a significant saving in processor time.

Other Works to Include Protein Flexibility into Electrostatic Calculations

While methods for calculating protonation probabilities in proteins by continuum electrostatic methods exist in a number of groups (Honig, Gunner, McCammon, Bashford, Wade, Vriend, Knapp, Ullmann and other), there are relatively few attempts to include protein flexibility.

Nielsen *et al.* [46] developed a method to calculate $\mathrm{p}K_a$ values combining Whatif [107], DelPhi [28] and the package of Yang *et al.* [27]. Hydrogen atoms were placed and optimized by different protocols and His, Asn and Gln were allowed to "flip", *i.e.,* rotate by 180° around the $\chi 2$, $\chi 2$ and $\chi 3$ torsion angle, respectively. A threshold-accepting algorithm combined with tree search methods to optimize the global hydrogen bond network [108] was found to be the preferable hydrogen placement method. The hydrogen placement problem was treated as precursory step independent of the outcome of the $\mathrm{p}K_a$ calculations. In another work [47] the authors used two structures (for strongly interacting groups four structures) *per* site with globally optimized hydrogen bond networks to compute Born and background energies. Therefore, the hydrogen bond network was always optimal to stabilize a certain protonation state, however the energy to convert the system from a hydrogen bond network optimal for one protonation state to a hydrogen bond network optimal for another protonation state was not included into the microstate energy.

Maybe the most advanced method, multiconformational continuum electrostatics (MCCE), was developed in the group of Marilyn Gunner [32, 33]. Their energy function does not only include electrostatic terms, *i.e.,* Born, background and interaction energy, but also a non-electrostatic contribution to the intrinsic energy and interaction energy and an entropy correction. Their method allows to include hydrogen rotamers and tautomers. Singly-protonated histidine is described by two tautomers with the same model $\mathrm{p}K_a$ value [32]. The method was

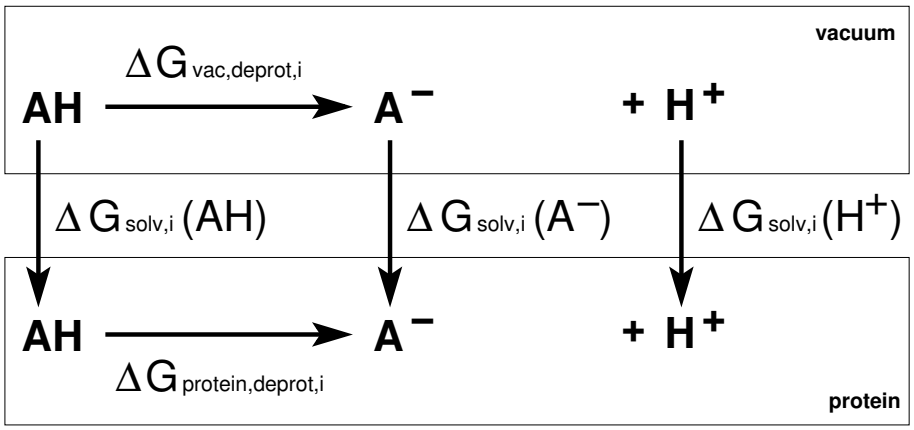


Figure 3.12. Thermodynamic cycle to calculate absolute $\mathrm{p}K_a$ values. The free energy of dissociation of a proton from an acid in the protein ($\Delta G_{\mathrm{protein,deprot,i}}$) is calculated from the free energy of dissociation in vacuum ($\Delta G_{\mathrm{vac,deprot}}$) and the solvation energy of the reactants and products. (Adapted from [126])

extended to include sidechain rotamers [33] and reduction potential calculations [124]. Most likely, from the description in Zhu *et al.* [124] and the fact that histidine is modeled with a single $\mathrm{p}K_a$ value for both tautomers, more than a single reduction potential or $\mathrm{p}K_a$ value *per* site is not possible. The physical basis of the method is, however, not described in sufficient detail. Also the source code of their program is not available to be analyzed. Therefore, further speculations are avoided which would be necessary for a comparison with the method presented here.

Very recently the group of Ernst-Walter Knapp the programs TAPBS [125] and Karlsberg 2.0 [34] were written, which seem to be similar to QMPB and GMCT, respectively. TAPBS builds on APBS instead of MEAD. Multiple steps of reduction and protonation as well as rotamers and conformers seem to be possible. However, the approach chosen here to allow arbitrary ligands, not only electrons and protons, is broader and more flexible. Since details about the underlying physical model are not published at the time of writing, a detailed comparison is not possible as well.

3.5.2 Calculations based on Quantum Chemical Data

Also absolute $\mathrm{p}K_a$ value and standard reduction potential $\mathcal{E}^\circ$ calculation have been done [126] very analogous to the calculation of absolute intrinsic energies (section 3.3).

The $\mathrm{p}K_a$ value of conformer i can be calculated according to the thermodynamic cycle in Fig. 3.12:

$$\begin{aligned} \mathrm{p}K_{a,i} &= \frac{1}{RT\ln 10}\Delta G_{\mathrm{protein,deprot,i}} \\ &= \frac{1}{RT\ln 10}\Big(\Delta G_{\mathrm{vac,deprot,i}} + \Delta G_{\mathrm{solv,i}}(A^-) + \Delta G_{\mathrm{solv,i}}(H^+) - \Delta G_{\mathrm{solv,i}}(AH)\Big) \end{aligned} \tag{3.55}$$

The solvation energies (section 3.2.5) $\Delta G_{\mathrm{solv,i}}(A^-)$ and $\Delta G_{\mathrm{solv,i}}(AH)$ can be calculated by continuum electrostatics methods (*e.g.,* with Solvate or Solinprot). The solvation energy of a proton $\Delta G_{\mathrm{solv,i}}(H^+)$ (identical with $\Delta G_{\mathrm{Born}}(H^+) = -260.5\frac{\mathrm{kcal}}{\mathrm{mol}}$ in eq. 3.35) can be calculated from the experimentally measured potential of the standard hydrogen electrode. The gas phase deprotonation energy $\Delta G_{\mathrm{vac,deprot,i}}$ is calculated from:

$$\Delta G_{\mathrm{vac,deprot,i}} = \Delta H_{\mathrm{vac,deprot,i}} + \Delta G_{\mathrm{vib,deprot,i}} + G_{\mathrm{trans}}(H^+) + \Delta(PV) - T[S(H^+)] \tag{3.56}$$

$\Delta H_{\mathrm{vac,deprot,i}} = H_{\mathrm{vac,i}}(A^-) + H_{\mathrm{vac,i}}(H^+) - H_{\mathrm{vac,i}}(AH)$ is the difference in energy of formation in vacuum of the dissociated (deprotonated and hydrogen ion) and associated (protonated) system. $\Delta G_{\mathrm{vib,deprot,i}}$ is the change in vibrational energy between the protonated and deprotonated form. Both terms can be calculated as by QM methods (section 2.3). The translational energy of a proton $G_{\mathrm{trans}}(H^+)$, the energy change due to the volume change in the gas phase reaction $\Delta(PV)$ and the entropic portion of the gas-phase free energy of the proton $T[S(H^+)]$ can be estimated as in eq. 3.35.

The standard reduction potential $\mathcal{E}^\circ$ is calculated by a similar approach:

$$\mathcal{E}^\circ = \frac{1}{\mathcal{F}}\left(\Delta G_{\mathrm{vac,redox,i}} + \Delta G_{\mathrm{solv,i}}(A_{\mathrm{ox}}) + \Delta G_{\mathrm{solv,i}}(e^-) - \Delta G_{\mathrm{solv,i}}(A^-_{\mathrm{red}})\right) \tag{3.57}$$

In the calculation of the free energy difference of the molecule in vacuum $\Delta G_{\mathrm{vac,redox,i}}$ the entropic (or vibrational) energy contributions are usually neglected and assumed to be identical with $\Delta H_{\mathrm{vac,redox,i}}$, which can be calculated by QM methods as before. The solvation energy of the oxidized and reduced form, $\Delta G_{\mathrm{solv,i}}(A_{\mathrm{ox}})$ and $\Delta G_{\mathrm{solv,i}}(A^-_{\mathrm{red}})$, respectively, can be calculated by continuum electrostatics methods as before. The solvation energy of the electron $\Delta G_{\mathrm{solv,i}}(e^-) = \mu^\circ_{e^-}$ is given in eq. 3.36.

Despite the well established theory, there was no program before QMPB to solve the multiple site titration problem, *i.e.,* calculate interaction energies of multiple sites. First approaches were ignoring the protein (solvation of the center in water with Solvate) or treating the protein as independent of pH and $\mathcal{E}$ (with Solinprot). A protonation state of the protein was determined by normal Multiflex calculations assuming a certain charge form of the quantum center. Even if the calculation was repeated for all charge forms of the quantum center and a consensus protonation state of the protein could be found, the calculations used an inconsistent physical picture, because the solvation energies for the quantum center were calculated in a dielectric of vacuum, but the Multiflex calculations were done in a dielectric environment of the protein. The additional dielectric boundary could not be included due to the limitation of Multiflex to two dielectric regions (the protein and the solvent). The effect on the calculated $\mathrm{p}K_a$ values within the protein might be small since the same mistake is done in both charge forms, if the protein sites are sufficiently far away from the quantum center and not interacting strongly.

I extended Multiflex to deal with three dielectric regions (Multiflex3D, section 4.4.5), to overcome this problem. Multiflex3D allows to compute intrinsic energies and interaction energies according to the lower part of the thermodynamic cycle in Fig. 3.13. The upper part of the thermodynamic cycle shows the vacuum calculation of two protonation forms by QM methods. The solvation energy is calculated for transferring the quantum center in a vacuum dielectric region into water. A model $\mathrm{p}K_a$ value is calculated *via* the upper thermodynamic cycle, which can be used as input for Multiflex3D together with appropriate boundary definitions.

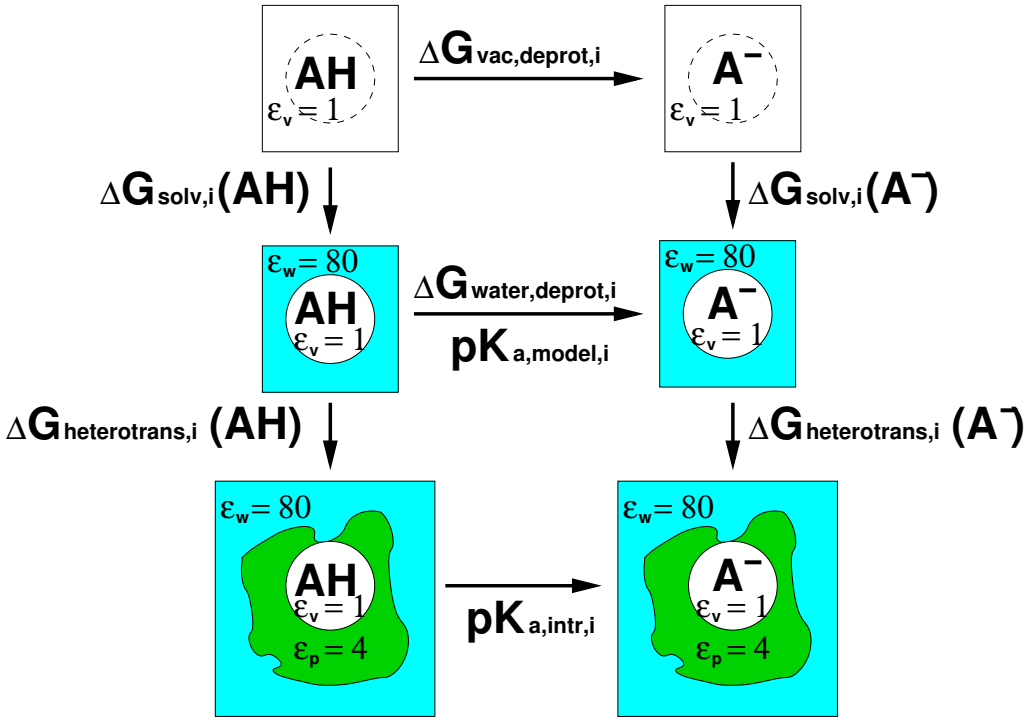


Figure 3.13. Thermodynamic cycles to calculate intrinsic $\mathrm{p}K_a$ values and interaction energies with Multiflex3D. The upper cycle calculates deprotonation energies in water ($\Delta G_{\mathrm{water,deprot,i}}$) from dissociation energies in vacuum ($\Delta G_{\mathrm{vac,deprot,i}}$) and solvation energies (*e.g.*, obtained with Solvate). The model $\mathrm{p}K_a$ value is obtained as $\mathrm{p}K_{a,\mathrm{model},i} = \frac{1}{RT \ln 10} \Delta G_{\mathrm{water,deprot,i}}$. It can be used together with the QM derived charges to calculate the intrinsic $\mathrm{p}K_a$ values, $\mathrm{p}K_{a,\mathrm{intr},i}$, and the interaction energy matrix $\Delta\Delta G_{\mathrm{inter,i}}(j,l)$, eq. 3.47.

In principle, treatment of rotamers (or tautomers) is possible by the same method [41], if they are stable within the QM geometry optimization. The QM energies include the torsional energy differences of the rotamers and solvation energy calculations are carried out for each rotamer. The quantum center has to be chosen such, that the rotamer differences do not affect the boundary between the quantum region and the protein (see section 3.3).

However, no matter if Multiflex or Multiflex3D is used, the framework of the binary state vector only allows for two rotamer or charge forms, which is generally not enough. If only a single site is used (to avoid the multiple site titration problem) more rotamer or charge forms can be included in the calculation of the grand canonical partition function, but the calculation has to be done manually.

3.6 Summary

This chapter introduced an energy function for computing ligand binding energetics. The energy function is more general than previous formulations allowing for

- any number of sites

- any number of instances, *i.e.,* the combinations of charge forms and rotamer forms
- any number of ligands in the system
- any type of ligands in the system
- any number of ligands binding to a particular site
- any type of ligands binding to a particular site.

Sites can be either parameterized by experimental data or by quantum mechanical (QM) calculations. The system is described by a grand canonical ensemble of microstates (section 3.1.1), which are a function of a set of thermodynamic variables and a state vector. The thermodynamic variables of highest importance are the chemical potentials of all the ligand types in the bulk solvent. The state vector is a vector of integer values of a length equal to the number of sites. The integer values identify a particular instance of a site.

The microstate energy is written in a form, which is convenient for computation and parallelization exploiting a required additivity of the energy terms. The microstate energy contains three terms (section 3.1.2): The first term accounts for conformational, *i.e.,* global structural changes. The second term is the energy of each site summed over all sites, splitted into a chemical potential dependent energy term and an intrinsic energy. The intrinsic energy of the site is the energy of the particular instance selected by the state vector for the site, if the chemical potential of all ligands is zero and if all atoms belonging other sites have a charge of zero. The definition of this artificial state as reference state leads to a significant simplification of the equations compared to previous formulations. The third term is a double sum calculating the pairwise interaction energy of a particular instance of a site with the instances of all other sites according to the state vector. This formulation of the microstate energy is convenient, because the intrinsic energies and interaction energies can be tabulated once and looked up many times for all combinations of instances of sites in any state vector and for any set of chemical potentials. The calculation of intrinsic energies and interaction energies is independent of the calculation of instances of the same site and all other sites. Therefore, the calculations can be easily performed in parallel.

The probability of a particular microstate (and therefore a particular state vector) can in principle be calculated for a given set of thermodynamic variables from the partition function (section 3.1.3). By varying the thermodynamic variables, microscopic titration curves can be calculated and microscopic as well as macroscopic equilibrium constants can be obtained. In practice, it is computationally not feasible to calculate the partition function for larger systems, but the probability of instances of sites have to be approximated, *e.g.,* by a Metropolis Monte Carlo method (section 3.1.4).

Currently, the effect of the protein (and other sites in the protein) on a particular site is described purely by continuum electrostatics. Already section 2.2 described how generally electrostatic potentials and electrostatic energies can be calculated from a given charge distribution and sets of dielectric boundaries. However, section 3.2 is oriented more towards practical calculations in molecular systems. The Born energy is defined as the interaction of a point charge with its reaction field and the background energy is due to the interaction of fixed charges of the protein with the electrostatic potential due to partial charges of atoms in the site. The interaction energy is due to the interaction of pairs of point charges of different

sites. In practice, always energies for transferring a set of charges from a simpler dielectric environment into the protein environment are calculated. I differentiated homogeneous and heterogeneous transfer energies, dependent on the initial dielectric environment being homogeneous, *i.e.,* described by a single dielectric constant, or being a two dielectric environment, *i.e.,* described by a dielectric constant for the molecule and a dielectric constant for the solvent.

Intrinsic energies can be calculated by two methods, either by QM calculations or by using experimental data. The second method calculates the shift of the binding constant in the protein relative to experimental conditions (relative intrinsic energy), while the first method aims to obtain the binding constant with little prior knowledge (in contrast named absolute intrinsic energy). The calculation of absolute intrinsic energies (section 3.3) involves calculating the energy of formation and the energy of vibration of the instance of the site in vacuum by QM methods. Then, the site is transferred into the protein, but thereby energy contributions are calculated which are normally excluded in force fields. However, this error can be corrected exactly by a correction term. The energy of free ligands needs to be included, which is governed by the standard chemical potential of each ligand. For electrons, it is the experimental value of the potential of the standard hydrogen electrode, for larger ligands it may include an experimental solvation energy and additional terms, which may be calculated by QM methods and estimated by thermodynamic considerations. The calculation of relative intrinsic energies (section 3.4) is governed by the experimentally determined binding energy and the energy shift due to transferring the instance of the site from a system describing the experimental conditions into the protein environment. It is assumed, that the experimental binding energy can be described by a single rotamer with lowest rotamer energy in solution. All other rotamers of the site are calculated relative to this reference rotamer. Also sites, which do not titrate may have multiple rotamers.

Previous energy functions described the state vector only by two instances, bound and unbound. This required approximations to describe titrateable groups in proteins and fails for more complicated cases as histidine or if rotamers should be included. In case of histidine a solution was found, which is correct in the physiological pH-range, but computationally costly. For rotamers thermodynamic average intrinsic $\mathrm{p}K_{a,\mathrm{intr},i}$ were calculated for the site, when all other sites are in their reference form. Even more complicated systems, which bind more than one ligand (*e.g.,* electrons and protons) could not be treated. The choice of a better, however artificial, reference state simplified the equations. Previous absolute energy calculations required to calculate the contributions by separate programs and do the energy calculation manually. For systems with more than one ligand binding site, the multiple site titration problem could not be solved properly. The required software was not available and a manual calculation was much too time consuming. In part, the problem was solved by developing Multiflex3D. However it inherited the same limits of a binary state vector from Multiflex. Since often sites, which are calculated by QM methods are those, which are too complex to obtain reliable experimental data because of too many possible instances, the limits of a binary state vector became even more pressing. Additionally, the programs available were thought to be used only for a single ligand type, a proton. Including more ligand types is possible, but gets increasingly difficult for the user. A program based on the more generalized ligand binding theory, which is described in this chapter, should free from such limits.

CHAPTER 4

DEVELOPMENT OF SOFTWARE FOR LIGAND BINDING STUDIES

The ability of a site in a biomolecule to bind a particular ligand is strongly influenced by its local environment. Therefore, most theoretical ligand binding studies are based on the detailed knowledge of a three dimensional structure of the biomolecule, as it can be obtained by x-ray crystallography or NMR spectroscopy. The data is stored in the PDB database [127, 128] in pdb-files (appendix A.1.1). Besides other information these files contain the names of atoms and their cartesian coordinates. Some information required for electrostatic calculations is however missing, *i.e.*, the charges and radii of the atoms. Crystal structures of biomolecules usually lack hydrogen atoms, which need to be added by some theoretical method (section 2.4.3). NMR structures contain different molecular models, which need to be separated and certain models be selected for the calculation. Several other steps are usually required to prepare a structure as stored in a pdb-file, before it can be used for continuum electrostatic calculations. All these steps are usually performed by sets of shell scripts and small helper programs for very specialized tasks. For modeling purposes, larger programs like CHARMM might be used, which often raise additional barriers due to their requirements on preceding structure preparation steps. Finally, the pqr-file (section A.1.2) and other input required for the electrostatic calculations can be generated. In our group, Multiflex by Donald Bashford was usually used to calculate intrinsic $\mathrm{p}K_a$ values and interaction energies, stored in `*.pkint` and `*.W` files, respectively. These files were read by a Monte Carlo program, CMCT, to calculate protonation probability curves.

All these programs were replaced during my work. Multiflex can be replaced by QMPB (section 4.3), which implements the ligand binding theory described in the previous chapter. Due to its more general concept it offers many more possibilities to the user than the previously used program. However, QMPB was designed to be very simplistic to allow understanding each step of the calculation and therefore depends even more on pre-processing the input data as Multiflex does. To give new users the possibility to become familiar with QMPB, the program Multiflex2qmpb was written (section 4.5), which performs all necessary tasks to convert the input for Multiflex into input suited for QMPB. Despite useful as a drop-in replacement for migrating older projects to use QMPB and to verify that old results can be reproduced with QMPB, using Multiflex2qmpb limits the user largely to the features of Multiflex. To unleash the full power of QMPB a object library, called Perl Molecule (section 4.6), was written, which allows to write simple but powerful scripts to perform all pre-processing tasks from a pdb-file

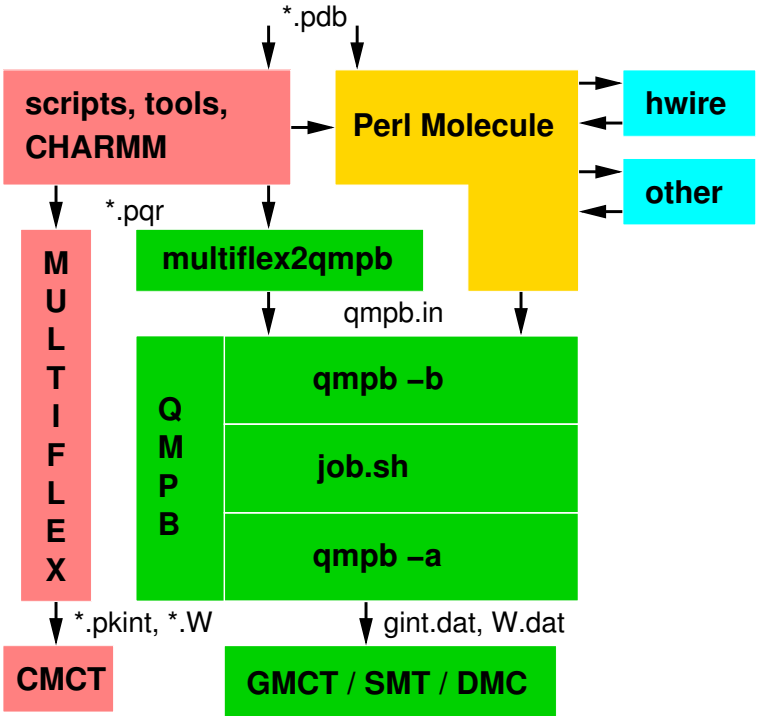


Figure 4.1. Data flow between different programs used in this work. The old chain of programs (marked in red) uses shell scripts to convert a pdb-file into an input suitable for calculations with Multiflex. Titration curves are calculated by the Monte Carlo program CMCT. QMPB (in green) implements a more general titration theory replacing Multiflex. GMCT replaces CMCT for titration calculations. The input for Multiflex can be converted by Multiflex2qmpb or Perl Molecule into an input suitable for QMPB. Perl Molecule can also be used to convert the structural data directly into an input suitable for calculations with QMPB. External programs (cyan) can be used as plug-ins for Perl Molecule.

to the input files required by QMPB. Perl Molecule has a plug-in interface to some external programs like Hwire for specialized tasks. The Monte Carlo program CMCT, was replaced by GMCT by Matthias Ullmann, capable of calculating ligand binding probabilities of sites with more than two instances. If the total number of instances is small enough, an analytical evaluation of the statistical average is also possible with the program SMT by Matthias Ullmann. To simulate transfer reactions, the program DMC by Mirco Till and Torsten Becker can use energies calculated by QMPB. For an overview, Fig. 4.1 compares the old (in red) and new (in yellow and green) chain of programs.

In this chapter, section 4.1 gives an introduction into scientific software development including object oriented programming terminology and concepts of parallelization. Section 4.2 discusses algorithms, I devised for programs developed by others in the group, *i.e.*, a state vector representation as a mixed radix number (as it is suitable for iterators as in SMT, section 4.2.1) and a state vector representation as a tree structure (as it is suitable for a cache in DMC, section 4.2.2). The adaptive mesh refinement (AMR, section 4.2.3) algorithm is an idea about speeding up titration calculations, especially for large numbers of ligand types, which only became possible by the generalized theory introduced in this work.

In section 4.3, my program QMPB is discussed implementing the core of the theory. It uses a set of programs based on the MEAD object library by Donald Bashford [31] for the electrostatic calculations. These programs and some modifications to the MEAD library are discussed in section 4.4. The pre-processor programs Multiflex2qmpb and Perl Molecule are described in section 4.5 and section 4.6.

4.1 General Considerations on Scientific Software Development

The previous chapters derive a conceptional model of ligand binding based on thermodynamic considerations (Fig. 4.2). This model describes the system in terms of a continuum electrostatic approach as a set of dielectric regions with fixed and mobile charges (section 2.2). Mathematically the system is described by the linearized Poisson-Boltzmann equation (eq. 2.69), which can be solved numerically by discretization on a grid (section 2.2.6). Additionally, energy contributions are identified, which can in part be computed by quantum chemical or classical force field calculations (section 2.3 and section 2.4, respectively). In part the energy contributions rely on experimental results on small model compounds. All these terms I combined into a consistent physical picture (described in chapter 3). For practical applications, it is necessary to develop computer programs, which calculate the energy terms and combine them properly according to the physical picture. In this chapter these computer science aspects of my work are discussed. In particular, the next section formulates my aims during software development and the following sections discuss in more detail, how some of these aims can be achieved.

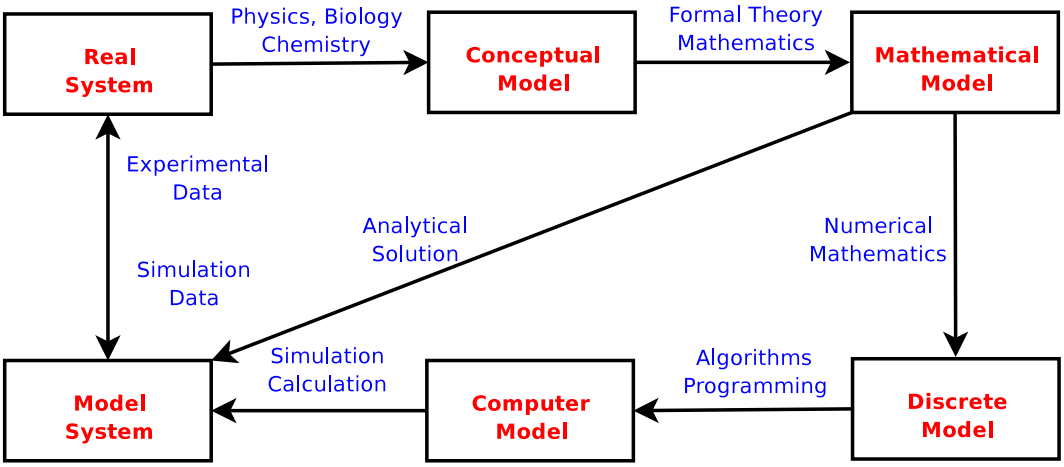


Figure 4.2. Building scientific models. The real system is casted into a conceptional model based on scientific knowledge. Using mathematics a mathematical model is derived, which can be either solved analytically or discretize to be solved numerically. The discrete model is implemented in programs, which can be run on a computer. By numerical calculations or from the analytical solution of the mathematical model results are obtained for the model system. These results can be compared with experimental data on the real system to obtain a deeper understanding.

4.1.1 Aims of Scientific Software Development

Three pairs of aims are crucial in scientific software development (in order of decreasing importance):

Usefullness and Correctness

First, a program has to fulfill its purpose, else it is not usefull. Therefore, the underlying (physical) theory has to be implemented correctly. One can formulate a requirements catalog or usage scenarios to define the purposes of the software and later validate if they are fulfilled. To check the physical theory and the correct implementation, one can design test systems, which either are simple enough, that they can be calculated (at least in parts) manually or there are other programs available which provide a subset of the functionality and can be used for testing. A problem in particular with scientific software is that it is a tool for ongoing research. Often the later use is different from initial intention, because applications become interesting, which could not be forseen at the time of development. Therefore, introduction of unnecessary limits should be avoided.

Maintainability and Modularization

Second, since the programs should be usefull for many years of research and almost certainly will be modified over time, the program has to be written in a maintainable way from the beginning. As for any larger piece of software, a modular design has several advantages. Primarily, breaking a large program into modules allows to manage them fairly independent. For new programmers it is easier to understand the software by starting with the most important parts and leaving those aside, which are currently not of interest to them. In particular, the fast understandability for new programmers is important for scientific software, since usually there is no maintainer, which does changes on the code according to the demands of the users, but the users have to extend the software themselves if the demands for their current project exceed the current capabilities of the program.

Performance and Scalability

The third aim of particular importance for scientific software is performance and scalability of the code. Many scientific computational projects are limited by the available CPU time. Often a single computation requires days, weeks or sometimes even CPU-years to finish. Instead, ordinary software mostly waits for user input. Often approximations can not be used, which would be closer to the physical picture of the system, because the computational time would increase dramatically. Therefore, development of scientific software is always a trade-off between the accuracy of the model and the necessary CPU time.

4.1.2 Modularization and Object-Oriented Programming

A modularization of the code should lead to separate data structures and code acting on the data. The module should provide an interface to the rest of the program (called application programming interface, API), which should remain constant over time. The reorganization of

the data structures in the module as well as the algorithms acting on the data may change over time, but such changes remain local to the module and do not require changes on the rest of the program as long as the API remains the same. Ideally, new functionality can be added by extending or replacing a single module instead of changing the whole program. One should avoid large data structures which contain information about all aspects of the system, which are passed to most subroutines (resembling global variables). It should be clear, where data structures are generated, extended or modified and were they are deleted. Otherwise it is very hard for a new programmer to understand the program. A large number of subroutines modifying the data complicates debugging. Also necessary changes on the data structures at a later point in time would require changes in all subroutines acting on the data.

Most modern programming languages provide concepts of modularization. However, the organization of the modules depends largely on the developer, *i.e.,* many modules tend to be collections of subroutines grouped by some aspect. Not necessarily a module is defining a data structure and providing all subroutines acting on this data structure. It depends on the discipline of the programmer, if the data structures are only manipulated by routines from the module or if the data is also modified by other parts of the code. Since the 1990s, object-oriented programming (OOP) is widely accepted as a programming paradigm, which facilitates and emphasizes the development of modular, manageable and reuseable software. OOP shifts the view on a program from a list of instructions given to the computer to a set of cooperating objects. Each object has its own data structure (called attributes) and subroutines (called methods) acting on the internal data. The internal data should not be modified by other objects directly, but only by accessor methods provided by the object (encapsulation). However, how much encapsulation is enforced depends on the programming language. Each object is defined by its class - it is a particular instance of a class. For example, each instance of a titrateable site in QMPB is described by an instance of a `site` class. There is a `site` object for each combination of charge form and rotamer form of the site. Subclasses can be derived from classes to define more specialized versions. The subclass inherits all attributes and methods (collectively called members) from the parent class, but can add additional members or redefine them. In QMPB for example, instances can be objects of the `MMsite`, `QMsite` or `Modelsite` class. The three classes inherit everything common from a parent class `site`, but for example the method computing the intrinsic energy is implemented differently (overwritten). The feature of analogous methods[1] with the same name, but different behavior is called polymorphism. The `site` class is called abstract, because no object can be instantiated directly. However, due to the class hierarchy each instance, which is an object of the `MMsite` or `QMsite` class, is at the same time an object of the `site` class.

The decision how to separate a domain of interest into classes and the way objects of these classes interact is to a certain degree arbitrary. It should reflect the view of the programmer (and possibly also the user) on the domain of interest. The degree of modularization also depends on the size of the overall software project. For a rather simple program, the number of classes should be small to avoid the introduction of unnecessary overhead and complexity due to object interactions. For a larger project, however, more classes need to be introduced, to keep each class small and manageable.

[1] In most OOP languages methods have to be virtual to be overwritten. Since in Perl all methods are by default virtual, I do not differentiate between virtual and non-virtual methods.

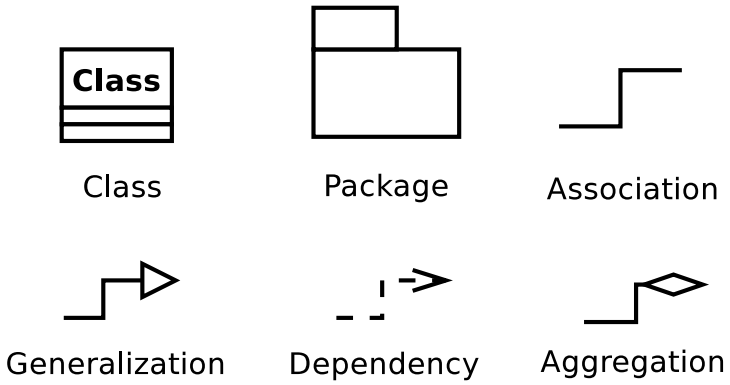


Figure 4.3. Graphical symbols used in UML class diagrams.

4.1.3 Unified Modeling Language

An abstract model of an object oriented software system can be described graphically by the unified modeling language (UML). UML is a standardized specification language for object modeling. UML is officially defined by the Object Management Group (OMG) in the UML metamodel [129]. UML 2.0 specifies 13 types of diagrams and a large number of concepts, with associated graphical symbols (see [130]). Some of the diagrams in this chapter follow the UML standard more closely and others are more individual. Here, a short summary is given for graphical elements taken from UML.

Class diagrams show the static relationship between classes or objects. They use the following concepts and graphical representations (Fig. 4.3):

Class: Box with class name in the first line in a bold font. Abstract classes have a class name in a bold-italic font. The box may be splitted into three parts with the class name in the first part, the attributes in the second part and the methods in the third part. Abstract methods are also given in a bold-italic font (Fig. 4.13 and Fig. 4.20).

Generalization: The generalization ("is a" relationship) between two classes is symbolized by an arrow with an empty head. The head is pointing towards the parent class. For example a `QMsite` "is a" `site` in QMPB (Fig. 4.13).

Package: A package (or module, here usually identical with a class) is depicted by a box with a tab (Fig. 4.21).

Dependency: The dependency of a package (or class) is symbolized by a dashed arrow with stick head. The head is pointing towards the package (Fig. 4.21).

Association: The association between instances of two classes (objects) is indicated by a line. The order of the association can be noted by numbers near the ends of the line. The objects "know" (*e.g.,* store references) about each other (Fig. 4.19).

Aggregation: The aggregation or composition of objects ("has a" relationship) can be symbolized by an arrow with a lozenge as head. The head points towards the aggregation class, the tail towards the component class. The order of the association can be noted by numbers near the ends of the arrow (Fig. 4.19). A component in a composition has zero to one (0..1) or exactly one (1) aggregation class, while a component in an aggregation can have any number (*) of aggregation classes.

Sequence diagrams (Fig. 4.14) show the sequential interaction between different objects in the coarse of time. A sequence diagram allows the specification of simple runtime scenarios in a graphical manner. The diagram shows the name of the participating processes (or objects by their names or class names) in the top row. Parallel vertical lines originate from each process. The line is dashed if the process is not active and a slim box if it is active. Horizontal arrows indicate the messages exchanged between the processes in the order in which they occur. Solid arrows indicate method calls (method name above the arrow) and dashed arrows mark the return of a method call. Boxes with `loop` in the upper left corner indicate loop structures (an action is performed a given number of times). Dog-eared boxes indicate notes for further explanation, which are linked by dotted (non-vertical) lines to the points in the diagram to which they belong.

4.1.4 Optimization, Scaleability and Parallelization

Time consuming parts of the program have to be optimized by the developer to run as fast as possible. Optimizing the code for speed often includes explicitly storing partial results in separate variables to look them up instead of recalculating them. Easily the number of variables used in such a subroutine exceeds the number of variables a reader of the code can remember and keep track of (*e.g.,* the `sor` routine of MEAD). Such subroutines tend to get long, because breaking into several pieces would require function calls, which are time consuming for the CPU. Also cascaded loops to work on high-dimensional data structures are much faster than method calls to objects manageing parts of the data. So readability and manageability of the code is sacrificed for a gain in execution time. Fortunately, only a few percent of the code of any program are such time-critical routines. Therefore, one usually can write most of the program in a manageable style as described before and only the parts which are known to be the time-critical (or were found to be time-critical by profiling a prototype) have to be written in a style optimized for performance. Furthermore, it can be considered - as it is done for QMPB - to write the non-time-critical part of the program in a scripting language as Perl or Python, which is easy to program and maintain and only write the time-critical parts in a lower level, but faster language like FORTRAN, C or C++. Using different languages, one can find an optimal trade-off between development time and runtime. Ideally, an interface for the scripting language to the lower level routines is written (section 4.4.4), but also helper programs can be called from the scripting language to speed up the calculations.

Often the gain from a performance oriented style of programming and a programming language, which produces faster executing binary code, is small compared to the performance impact of a better algorithm. Every non-trivial problem can be solved in different ways - by different algorithms - and often the intuitive approach is not the fastest. For example, many problems naturally imply a recursive algorithm (*i.e.,* processing a tree-like data structure), however an iterative algorithm might be faster. Often even experts fail to predict, which al-

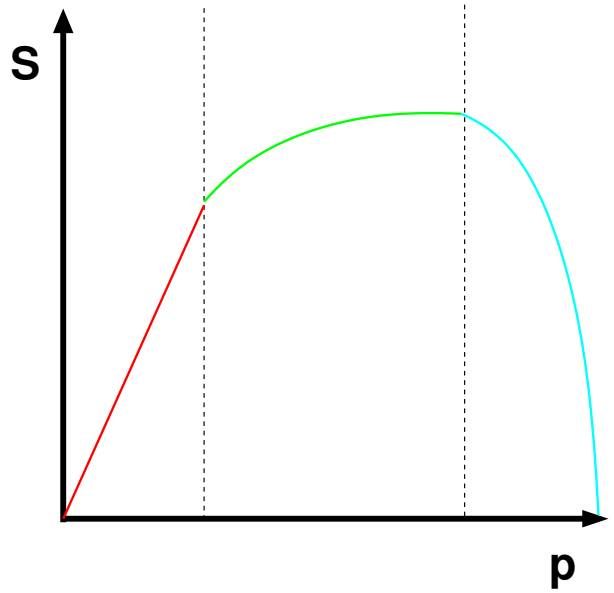


Figure 4.4. Speedup of a parallel algorithm dependent on the number of processors used. The speedup is defined as $S(p) = \frac{T_1}{T_p}$, where p is the number of processors used, T_1 is the execution time of the sequential algorithm and T_p is the execution time of the parallel algorithm. In the first phase (red) the speedup is linear with the number of processors, *e.g.*, when doubling the number of processors, the execution time halves. The second phase (green) shows a logarithmic speedup, which asymptotically approaches a minimum execution time. The third phase (cyan) shows a speedup decreasing with number of processors. Adding more processors increases the execution time.

gorithm performs better. Therefore, it is necessary to implement and benchmark the most promissing candidates. Section 4.2.1 gives an example, where the reformulation of an algorithm gave a performance increase.

Most scientific programs are tested with a small system, but for the research work they will run with increasingly larger systems. The behavior of the code with increasing number of elements in its data structures (*e.g.*, number of atoms or number of grid points) is described as its scalability or complexity of the time-limiting algorithm. The complexity is characterized by the Landau or "Big O" notation. The symbol O is used to describe an asymptotic upper bound for the magnitude of a function in terms of another, usually simpler, function. One usually aims for a linear scaling, *i.e.*, the CPU-time spent is proportional to the number of elements ($O(N)$), but unfortunately many algorithms show a behavior of $O(N^2)$ or worse, *e.g.*, $O(\exp(N))$ or $O(N!)$. For example using the Poisson-Boltzmann equation for ligand binding calculations would show an exponential scaling behavior with the number of sites since all microstates need to be calculated explicitly. The approximation by the linearized Poisson-Boltzmann equation (section 2.2.5) leads to a linear scaling behavior with the total number of instances. Section 4.2.2 shows an example of a cache, where the time for a cache lookup could be converted from an exponential dependence on the cache size into a constant value (independent of the cache size).

Additionally to increasing the performance and scalability of a program by better algorithms, parallelization is a technique widely used in scientific computing to reduce the run-time of programs. Unlike the previous approaches, which reduce the CPU time directly, here only the wallclock time, the user has to wait, is reduced by dividing the problem into sub-problems, which can be solved at the same time on different CPUs. In an ideal case the wallclock time reduces to the run time on a single CPU divided by the number of CPUs, *i.e.,* doubling the number of processors doubles the execution speed (linear speedup, Fig. 4.4). In exceptional cases the speedup may even be better due to the distribution of the data over the cache of several processors (super linear speedup). However, usually the CPUs need to communicate partial results leading to a logarithmic scaling behavior (from a certain number of CPUs on) or even worse, increasing the number of CPUs might increase the execution time, because the slowdown by the required communication dominates over the theoretically increased speed of solving the problem.

Generally one can distinguish fine grained and coarse grained parallelizations. For example the Poisson-Boltzmann solver APBS [22, 63, 67–69] shows a fine grained parallelization. Each point of the grid, on which the (L)PBE is solved, can in principle be associated with a single CPU. The fine grained parallelization allows to solve the PBE on thousands of CPUs [22] at once, allowing for grid sizes too large to fit into the memory of any available computer. However, each CPU has to communicate the electrostatic potential at the grid points at the boundaries of its subgrid with the CPUs responsible for the neighboring subgrids at each iteration step until consistency is reached. Therefore, such a fine grained parallelization has a significant communication overhead. Multiflex and QMPB instead are examples for a coarse grained parallelization, allowing to solve the LPBE for each instance on a separate CPU. For larger biological systems, the total number of instances is in the order of several hundred or thousand, usually exceeding the number of available CPUs. Unlike for the fine grained parallelization, the calculations are completely independent and do not require any communication during the calculation. Only at the end the partial results have to be gathered and combined by QMPB. Therefore, calculations of ligand binding energetics belong to the class of “embarrassingly parallel” [131] applications.

There are different approaches to realize the parallelization of programs, also dependent on the underlying computer hardware. At the time of the mainframe computers, symmetric multiprocessing (SMP, Fig. 4.5 A) was common, using several CPUs on the same mainboard sharing the memory. Mostly due to the high cost, such systems were replaced by non-uniform memory architectures (NUMA, Fig. 4.5 B), which had processor boards with SMP processors and memory. Direct access to the memory of other processor boards was possible, however with a much higher latency. Nowadays, most supercomputers are computer clusters, often using “standard” computer hardware and rather slow networks connecting the compute nodes. However, the other techniques are still used. For example the NUMA architecture is realized in modern AMD processors, where each processor has an own main memory on the mainboard and the processors can access the memory of the other processors. SMP machines with more than two CPUs usually were very expensive, but recently they became much cheaper (and therefore more wide-spread) with the modern multi-core CPUs or the Cell processors with multiple synergistic processing elements (SPE). At the same time, single computers and computer clusters are connected to form a global computational “grid”. The “grid” provides a much larger number of CPUs than any available cluster, however connected over a network

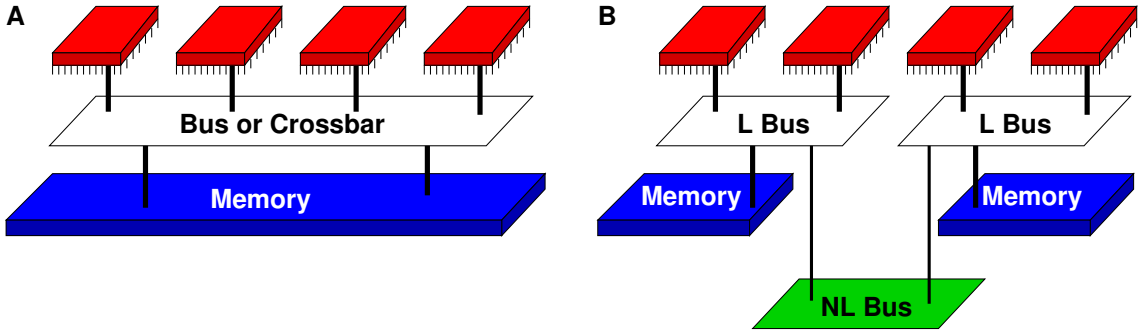


Figure 4.5. Multiprocessor computer architectures. A: Symmetric multiprocessing (SMP) architecture. All processors (red) are connected by a bus or crossbar switch with each other and the shared memory (blue). B: Non-uniform memory architecture (NUMA). Each processor or a set of SMP processors are connected to their own memory by a local bus (L Bus). The processors can also access the memory of other processors *via* an non-local bus (NL Bus), which usually has higher latency and lower bandwidth than the local bus. Compute clusters replace the non-local bus by network hardware and require a message passing system, which is implemented in software.

(the internet) with a much higher latency and lower bandwith than the already relatively slow networks of compute clusters.

For SMP machines, parallelization with threads and shared memory is most efficient, however it is generally non-applicable for non-SMP machines or clusters of SMP machines. For parallel computing on different computers (as in a cluster) a network socket has to be opened and a messaging system has to be established to perform interprocess communication (IPC). Some programs rely on their own message passing layer (*e.g.*, GAMESS [132, 133], Gaussian [134]), which can be optimized for the special IPC pattern of the application. However, most parallel programs either use PVM (parallel virtual machine) or MPI (message passing interface) for their IPC. At least for MPI there is a grid-enabled version integrating components of the globus toolkit. The current version of QMPB casts off the problem of parallelization to a helper program and therefore allows the user to adopt for the local computing environment. QMPB generates a shell script with calls to programs based on the MEAD library, which can be executed in parallel. The distribution of the jobs (as well as distribution of input data if no network file system can be used) is the task for the helper program. Currently, the helper program splits the shell script generated by QMPB into chunks dependent on the number of processors and submits the parts as independent jobs to a queuing system managing the resources of a compute cluster. The same approach could be taken to submit to a queuing system managing grid resources.

4.2 Algorithms Contributed to Other Projects

Before discussing the larger programs QMPB and Perl Molecule, I will discuss three algorithms I contributed to software projects of others in the group. The first two algorithms deal with the same entity, the state vector, but use two fundamentally different data structure. Both are optimized for their purpose. They are a nice example how a better algorithm (including

the right choice of data structure) can lead to a significant improvement in performance and scaleability of a program. The third algorithm contains a new idea about the way titration calculations are performed, which offers the potential of significant time saving. The potential is especially large for high-dimensional titration calculations, which just became available by the generalized titration theory as it is implemented in QMPB.

The first algorithm describes the state vector as an array (or vector) of integers (section 4.2.1). The algorithm offers an efficient way to increment the state vector, as it is done in programs calculating the partition function, eq. 3.3, explicitly (*i.e.,* SMT).

The second algorithm describes the state vector as a tree data structure (section 4.2.2). The algorithm is very efficient in searching for a particular state vector, *i.e.,* for looking up pre-calculated microstate energies, which form a cache in DMC to speed up the application.

The third algorithm optimizes the computation of titration curves especially for calculations with many ligand types (section 4.2.3). Usually, the titration curves are calculated by scanning the chemical potentials of all ligand types over a pre-defined range. Usually, sites only titrate in a relatively small part of the chemical potential space. The basic idea of the algorithm is to focus the computational effort on these interesting parts.

4.2.1 A State Vector Iterator for SMT

An iterator is a common pattern in OOP [135], but similar constructions can also be written in procedural languages [136]. The core of an iterator is a function (or method) which returns the next value, *e.g.,* of a list, each time the function is called. A simple example is the `foreach` statement in Perl:

```
1 foreach $val (@list) {
2 ...
3 }
```

Foreach element of the list, the code block enclosed by the `foreach` loop is processed. Each iteration the variable `$val` is pointing to another element of the list (in the order given by the list). The iterator requires two components: First, it needs an access method for the current element (in the example by setting the variable) and second a element traversal method, which moves the iterator from the current element to the next element of the list. Therefore, the iterator needs an internal variable, which points to the current element of the list. Usually, there is also a method to check if the iterator is exhausted, *i.e.,* if the iterator is beyond the last element of the list and the loop has to be ended.

An advantage of an iterator compared to a normal iteration (*e.g.,* using a `for` loop) is that the complete list does not need to exist at any time, if the next element of the list can be calculated from the current element. In case of state vectors, systems often have so many microstates, that it is impossible to store all possible state vectors in a list in memory. Using an iterator, only the memory for storing the current state vector and a vector containing the largest instances number for each site (called maximum-state-vector) is needed. The next state vector can be calculated by the iterator based on this information. The iterator is exhausted, if the current state vector equals the maximum-state-vector.

Figure 4.6. The clock shown in the picture uses a roll counter (totalisator) for presenting the date. Each wheel of the mechanical counter has a different number of figures to present. There are 31 days of a month, 12 months in a year and 7 days in a week. Therefore, it presents a mixed radix system. Similar counters found in mileage counters in cars or electricity meters can be used as a thought model for the incrementing scheme discussed here.

An OOP implementation of a state vector iterator, would probably have a constructor defining the first and the last state vector. The first state vector can be omitted to start at a state vector, where instance zero is chosen for all sites. The last state vector is the maximum-state-vector. The length (number of sites) of the first state vector has to be the same as for the maximum-state-vector. An access method has to be written, returning the current state vector. A traversal method iterates from the current to the next state vector until the maximum-state-vector is reached and the iterator is therefore exhausted. The current state vector would be a private variable of the object.

Perl allows to create local variables within functions, which persist multiple calls to the function. Such a variable allows to store the state of the iterator (the current state vector) also in a procedural context, not only in an object oriented context [136].

Since the iterator should also be implemented in C, the algorithm does not use this Perl specific feature. C would require to store the state of the iterator in a global variable, which is generally considered as bad style of programming for good reasons [137]. Instead, it was chosen to pass the current state vector as a reference to each call of the iterator function. The iterator function increments the current state vector (by changing its value) and returns TRUE on success or FALSE on error, especially if the iterator is exhausted. The maximum-state-vector is also passed as reference to the function each time, but is never changed.

Previous works were describing the system by a binary state vector. Therefore, the natural representation of a state vector in a computer is a bit vector with the length equal to the number of sites. Incrementing the state vector was equal to counting in a binary numeral system from zero to the maximum-state-vector. The generalized titration theory allows an arbitrary (positive integer) number of instances for each site. The number of instances is usually different between two sites. To describe the state vector as number anyway, instead of a standard positional numeral system (as the binary, decimal or hexadecimal system), a

mixed radix numeral system has to be used. A standard numeral system has a constant base for all positions (*e.g.,* 1 for the binary system, 10 for the decimal system or 16 for the hexadecimal system). A mixed radix system allows to have a different base for each position (*e.g.,* 7 days *per* week, 24 hours *per* day, 60 minutes *per* hour or Fig. 4.6). Here, the maximum-state-vector gives the basis for each position (site).

The algorithm of binary counting and incrementing the state vector as mixed radix number is basically the same: Start at the lowest power position (conventionally printed the right-most) and increment the value of the state vector by one. If the value is larger than the base, set it (and the values at all lower power positions) to zero and increment the next higher power position by one.

```
1  sub inc_x {
2    my $x = shift;
3    my $max = shift;
4    my $l = $#{$max}; # length of state vector
5    die "State vector with largest instance number for each site is undefined!" if ($l < 0);
6    if ($#{$x} < 0) { # init state vector
7      $$x[$_] = 0 foreach (0.. $l);
8      return 1; # null state vector
9    } else {
10     foreach my $site (0..$l) {
11       if ($$x[$site] < $$max[$site]) {
12         $$x[$site]++;
13         return 1; # valid state vector
14       } else {
15         $$x[$site] = 0;
16       }
17     } #foreach
18     return 0; # no more state vectors
19   }
20 }
21 ...
22 while (inc_x (\@x, \@max)) {
23 ...
24 }
25 ...
```

The algorithm is shown here in the Perl implementation as function `inc_x`. The function might be called in a `while` loop giving a reference to an array for the current state vector `x` and a reference to an array for the maximum-state-vector `max` as parameters. `max` has to be predefined, otherwise the program terminates (*via* `die`) in the function `inc_x`. If `x` is not defined, the state vector is initialized with as many array elements set to zero as the maximum-state-vector has and returned as valid state vector (`return` value of function `1`). If `x` is defined, the `foreach` loop starts to run over all sites. If the instance of a given site is lower than the maximum number of instances (given in the maximum-state-vector), the position of the state vector (instance number) is incremented and the state vector returned (`return` value of function `1`), otherwise it is set to zero. If the `foreach` loop runs through all positions of the state vector without being able to increment a position (and returning with `1`), the iterator is

exhausted and zero is returned. This leads to a termination of the `while` loop in the calling routine.

For a maximum-state-vector of `(1 2 3)` the algorithm gives:

```
(0 0 0), (0 0 1), (0 0 2), (0 0 3),
(0 1 0), (0 1 1), (0 1 2), (0 1 3),
(0 2 0), (0 2 1), (0 2 2), (0 2 3),
(1 0 0), (1 0 1), (1 0 2), (1 0 3),
(1 1 0), (1 1 1), (1 1 2), (1 1 3),
(1 2 0), (1 2 1), (1 2 2), (1 2 3)
```

The view on the state vector as a mixed radix number also allows to convert each state vector into a unique decimal number: The vector `(1 1 3)` for example, consists of the instances 3 (*i.e.,* `(0 0 3)`), 4 (*i.e.,* `(0 1 0)`) and 12 (*i.e.,* `(1 0 0)`), therefore it has the decimal value 19. As another example, the vector `(1 2 2)` consists of the the instances 2 (*i.e.,* `(0 0 2)`), 8 = 2 · 4 (*i.e.,* `(0 2 0)`) and 12 (*i.e.,* `(1 0 0)`), therefore it has the decimal value 22. The number of microstates (and associated state vectors) is $N_{\mathrm{micro},i} = \prod_j^{N_{\mathrm{site},i}} N_{\mathrm{instance},i,j}$, so the above example has 2 · 3 · 4 = 24 microstates with decimal state vector numbers of 0 to 23.

In programs calculating the partition function (eq. 3.3) explicitly (*i.e.,* SMT) this algorithm is used. In Perl this algorithm is about 10 times faster than an algorithm previously used. In C the advantage reduces to a factor of 2, probably due to automatic optimizations of the compiler. In SMT, however, generation of the state vectors is not the time limiting step, such that the main advantage lies in simplicity and readability of the code.

4.2.2 A State Vector Cache for DMC

Caches are ubiquitous in computer science. If slow operations have to be done multiple times, it is usually faster to store the results and look them up the second time they are needed. All modern CPUs have an integrated cache memory which stores temporary results to avoid transfer to the compareably slow main memory (RAM). Harddisks have a cache of RAM integrated to avoid reading data multiple times from the disk. SCSI controllers (especially RAID controllers) often have a substantial amount of memory for the same purpose. Caches are also common in software. For example, most web browsers cache the last visited webpages on the local harddisk to avoid downloading them again from the internet. It is also common to have caching proxy servers in the local network storing *e.g.,* webpages downloaded by one user and providing them to all users on the network. Thereby, network latency and costs for the internet connection are reduced.

In DMC the calculation of sets of rates need to be redone very often (for fixed thermodynamic variables) to find the most probable next reaction. Since evaluation of the rates leading away from a current microstate is the time limiting step in the DMC calculations, the program can be accelerated if evaluation of the rates is replaced by a lookup of a set of rates for a previously calculated microstate. However, this lookup needs to be fast compared to the evaluation of the rates. The obvious approach would be to define a hash, where the state vector is the key and the set of rates is the value. In C, lacking a hash as a part of the core language, one would probably construct a chained list containing the state vector and the set of rates

in each element of the list. In the worst case, a program using this data structure would require to pass through all elements of the list, comparing the state vector stored with the current state vector, to decide that the current state vector is not cached and therefore the set of rates for the current microstate must be evaluated. The cost for this procedure would increase linearly with the number of elements cached up to the number of microstates of the system ($O(N_{\mathrm{micro},i})$) and quadratically with the length of the state vector ($O(N_{\mathrm{site},i} \cdot N_{\mathrm{site},i})$) With some optimizations, *i.e.,* progressing to the next element of the list as soon as the first instance in the stored microstate vector and the current microstate vector does not match, one could yield a scaling behavior similar to $O(N_{\mathrm{site},i} \ln N_{\mathrm{site},i})$. Usually, the number of microstates, which need to be cached, is much larger than the number of sites (even squared), so that the $O(N_{\mathrm{micro},i})$ term is the most critical. Converting each state vector into a single number as shown before, would reduce the number of comparisons to one *per* microstate, but still $O(N_{\mathrm{micro},i})$ comparisons would be needed. An advantage of this approach is its linearity, *i.e.,* it would be easy to store the state vector number (or the state vector as a string, if string comparison is sufficiently fast) and the energy in a database as cache. Such a database can be significantly larger than a cache in memory. However it would require a lot of disk operations, which are orders of magnitude slower than operations in memory.

A tree representation solves the previously discussed problems and thus was implemented by Mirco Till. A tree in computer science is an acyclic graph. It has a single root node at the top, which is connected to a number of inner nodes below, called its child nodes. Each child node has a single parent node (superior node). Inner nodes are organized in levels, where the nodes of the bottommost level are called leaf nodes. An example of a tree structure can be the file system of any computer. The root node `/` has a number of inner nodes, which are directories in case of the file system (*e.g.,* `/usr`, `/bin`, `/lib`). Directories can contain directories (*e.g.,* `/usr/bin`, `/usr/lib`, `/usr/share/doc`). Files are leaf nodes (*e.g.,* `/usr/bin/tcsh`). In case of the state vector representation as tree, the number of levels of the tree equals the number of sites plus a level of leaf nodes storing the set of rates of each microstate. Each node can have as many child nodes as this site can have instances. If a child node with a certain instance number is not present, it means that the current state vector was not used before and the set of rates needs to be computed and the appropriate branch needs to be added to the tree.

Fig. 4.7 shows the example from the last section in tree representation. The maximum-state-vector is again `(1 2 3)`, so the first site can have two, the second three and the third four child nodes. In practice, one would allocate memory for an array of references (pointers) for each potential child node. To find the set of rates $\{k\}$ associated with a microstate energy $G_{\mathrm{micro}}(000)$, one follows the reference of array element zero at each level ending up at a leaf node containing the set of rates leading away from this microstate. For the set of rates $\{k\}(G_{\mathrm{micro}}(113))$, one follows the reference of array element one on the first two levels, on the third level one follows the reference of array element three to the leaf node containing the set of rates. In any case one needs to follow as many references as there are sites to get to a previously computed set of rates. In case one is looking for a set of rates, which is not stored, a reference is not defined at the position the state vector deviates from an already known state vector, *e.g.,* looking for the state vector `101` would fail at the second level. Instead of scaling with $O(N_{\mathrm{micro},i})$, the algorithm scales with $O(N_{\mathrm{site},i})$. As result, the computational effort remains small independent of the size of the cache.

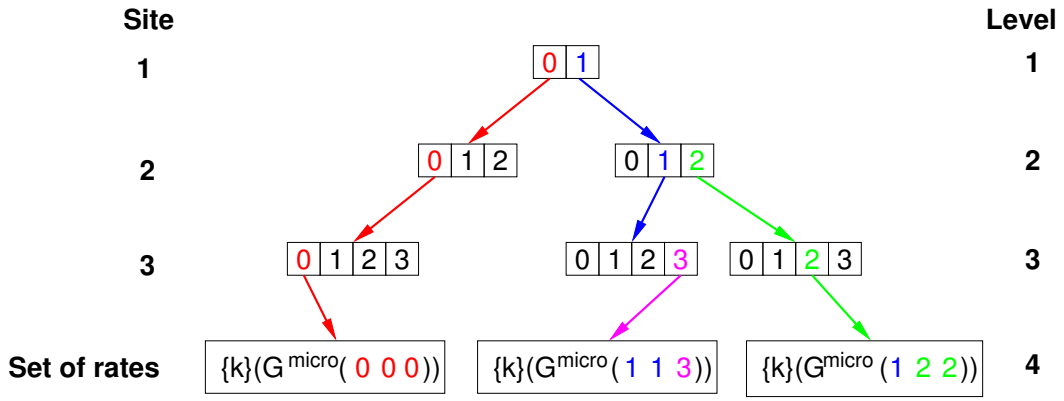


Figure 4.7. Three state vectors in tree representation of a system with maximum-state-vector of `(1 2 3)`. Each of the three sites is represented by an array with as many elements as the site has instances. The array elements may contain references to arrays representing the next site. The last site contains references to already computed sets of rates $\{k\}$.

For biological systems the number of microstates is so large, however, that only a small fraction can be stored in the cache (even if *e.g.,* 4GB of RAM are reserved for the cache). An important part of the caching algorithm therefore is not only the part looking up cached sets of rates for a particular microstate and extending the data structure, but also to remove microstates which are less likely to be looked up in near future. In case of *e.g.,* proton transfer calculations, reactions take primarily place in the neighborhood of the site with the extra proton. Microstates which are the result of a proton transfer from the current site of the proton to an acceptor site in hydrogen bond distance are more likely than a sudden protonation of a site far away. However, such an algorithm would need to include knowledge of the spacial organization of the sites and probably would be quite complex (*e.g.,* some clustering or graph algorithm). Instead, at the moment we only use the fact that the majority of the moves in DMC are forward and backward reactions and rotations. Therefore, the current implementation uses a cache, which works as FIFO (first in, first out), meaning that the first microstate added is the first to be removed. An alternative would be, to increment a counter each time a set of rates for a particular microstate is looked up and remove the least often looked up microstate first. However, this would need to be combined with a FIFO like behavior, because a just added microstate would have a counter of zero and therefore be removed very fast. Instead, a microstate, which was looked up many times in the beginning of the simulation, but was not needed for a long time, would remain in the cache. As result, both, the number of lookups and the time in cache, would need to be considered for such a caching scheme.

In case of the gramicidin A (gA) simulations in section 5.6, the simple FIFO cache was very efficient with hit rates of 80-90% and speeding up the simulation time by a factor of 13. The simulations discussed in the application section were only possible due to the caching algorithm. However, for bigger systems, *i.e.,* the bacterial reaction center, the cache efficiency is low and is one of the major obstacles to overcome for future applications.

It should be pointed out, that for the same entity (here the state vector) often different representations are possible. Often moving from one representation to another leads to algorithms,

which are much better suited for a particular purpose. In the example given here, the dependence of the algorithm with the number of microstates could be completely removed. In the application to gA, the system was modeled by 11 sites with 10 instances each (section 5.6). Instead of a search in 10^{11} microstates for an identical state vector, only 10 pointers need to be dereferenced. Instead of - in the worst case - $10^{11} \cdot 10$ comparisons of elements of the current state vector with the elements of all possible state vectors in a complete cache, only 10 comparisons are necessary, no matter how large the cache is.

4.2.3 Accelerating Titration Calculations using Adaptive Mesh Refinement

I developed an algorithm to speed up titration calculations, only afterwards noticing the previous work by Berger, Oliger and Coella [138, 139], who developed a similar algorithm to trace particle movement in time. Their adaptive mesh refinement (AMR) algorithm became important in the astrophysics, fluid dynamics and high-energy physics communities.

In section 3.1.3, the calculation of probabilities and equilibrium constants was described. It was pointed out, that an analytical evaluation of the partition function is only possible for small systems with not much more than 20 sites. Probabilities of such systems can be calculated, *e.g.,* with the program SMT. For larger systems the probabilities have to be approximated, *e.g.,* by a Monte Carlo procedure (section 3.1.4) as implemented in the program GMCT. Usually, the probabilities for a known set of thermodynamic variables (*i.e.,* chemical potentials) are not the aim of the calculations, but the dependence of the probability on varying chemical potentials. Therefore, the chemical potentials have to be scanned within a reasonable range and the probabilities calculated at each point. For a fine resolution of the titration curves, especially in the region were a particular site titrates, one needs to sample the chemical potential range in fine steps. The number of points, at which the probabilities need to be calculated increases exponentially with the number of chemical potentials taken into account (number of ligand types in the calculation). Already for 1-dimensional protonation probability calculations on a small protein in a range of pH 0 to 14 with a step size of 0.1 pH units and standard values for the number of MC scans and MC moves (section 3.1.4), the Monte Carlo titration is slower than the electrostatic calculations with QMPB. The discrepancy increases exponentially with the number of ligand types (QMPB scales linear and GMCT scales exponential).

The computational time of Perl Molecule and QMPB is in the order of seconds. The computational effort of solving the PBE with the MEAD based helper programs scales with the total number of instances of the system for a particular conformer i, where the total number of instances is the number of instances of each site $N_{\mathrm{instance},i,l}$ summed over all $N_{\mathrm{site},i}$ sites:

$$O\left(\sum_{l}^{N_{\mathrm{site},i}} N_{\mathrm{instance},i,l}\right) \tag{4.1}$$

Each computation takes in the order of minutes *per* instance (but may be run in parallel). The computational effort of the Monte Carlo titrations scales instead

$$O\left(\prod_{\lambda}^{N_{\mathrm{ligand}}} \left(N_{\mathrm{site},i} \cdot N_{\mathrm{scans}} \cdot N_{\mathrm{steps},\lambda}\right)\right), \tag{4.2}$$

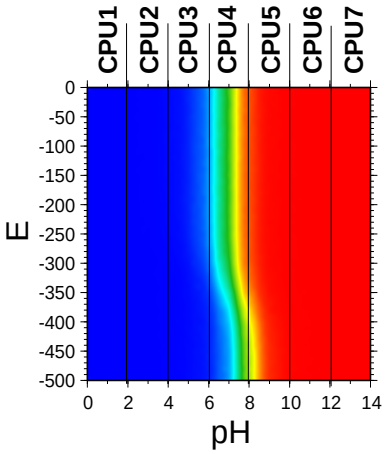


Figure 4.8. Speeding up titration calculations by splitting up chemical potential space. Each subrange of chemical potentials can be calculated on a different processor. In this example, the interesting part of the titration curve is only calculated by CPU4, while the other processors only calculate constant values.

so that each computation is in the order of 10 minutes for one ligand type λ (N_{ligand}= 1). The effort however increases exponentially with the number of ligand types N_{ligand}, that computations with more than two or three ligand types become unfeasible.

The number of MC scans N_{scans} can not be reduced without reducing sampling quality and the number of increments in chemical potential $N_{\mathrm{steps},\lambda}$ is defined by chemical potential range and step size. The step size needs to be sufficiently fine in regions where a site titrates. The number of sites $N_{\mathrm{site},i}$ in a conformer i depends on the system and equals the number of MC moves N_{moves} (section 3.1.4). Certainly, since the result for a given set of chemical potentials is calculated completely independent of the calculation for another set of chemical potentials, the calculations are embarrassingly parallel and can be easily split over many processors (Fig. 4.8). However, by this method one can only expect a linear speedup, so that the wallclock time is divided by the number of processors used.

The adaptive mesh refinement (AMR) approach, however, tries to reduce the execution time due to a better algorithm. The primary idea, is that most sites only titrate in a relatively small region of chemical potential space and they can be described by a single instance in large regions of chemical potential space (Fig. 4.9).

If it would be known *a priori*, that a site only populates a single instance in a given region of chemical potential space, all other instances could be excluded from the titration calculations. Most sites bind only one (or a few) ligand types and do not interact strongly with sites binding other ligands (*i.e.*, a protein with a redox site may have some protonateable residues surrounding the redox site and they interact by electrostatics strongly, but protonateable residues far away from the redox site are independent of the reduction potential and may only depend on the pH). Therefore, these sites titrate only in one dimension and are constant in the other dimensions. Taking this fact into account could help reducing the number of points in chemical potential space, which are needed to describe the titration behavior of a particular

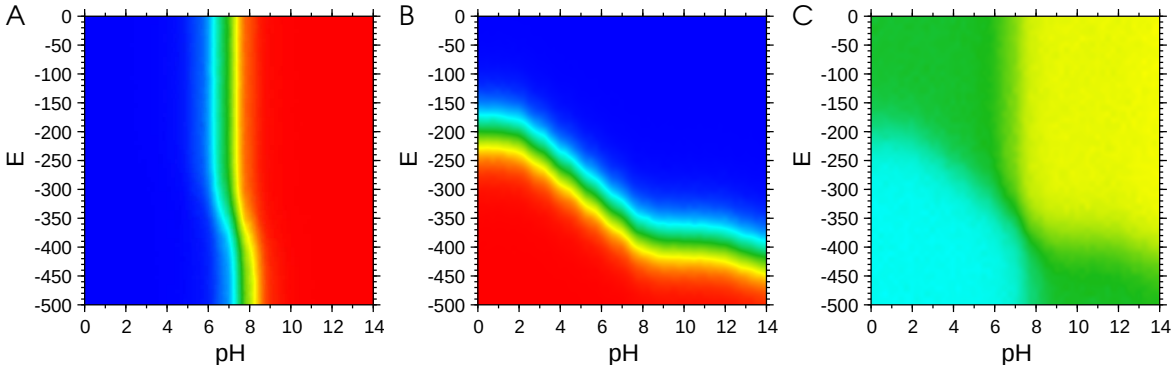


Figure 4.9. Examples for titration curves. A: A protonateable site with only little dependence on the reduction potential. B: A redox site with a strong dependence on the pH. C: A site with different rotamer forms, which strongly depend on both the pH and reduction potential. At low pH and reduction potential the instance plotted here is populated to about 75% (cyan), at high pH and reduction potential it is only populated about 25% (yellow). In the other areas it is populated about 50%.

site. Therefore, such an algorithm could reduce the exponential scaling with number of ligand types to a scaling with lower power for most sites.

To introduce the AMR algorithm, Fig. 4.10 shows how an arbitrary step function of two variables can be approximated. Given a computational method, which allows to calculate $f(x, y) = z$, where z is either zero (red) or one (blue), the function can be successively better described by partitioning space. Here, I chose to half the range of both variables in each step, so that squares (or rectangles) of decreasing sizes are obtained. In higher dimensions, space would be partitioned in cubes (or cuboids). Other geometrical forms (*i.e.,* triangles) are possible as well, but rectangles (or cuboids) are particularly easy to compute in cartesian space. Each time the step function has the same value at the four corner points of the rectangle (generally $2^{N_{\text{ligand}}}$ vertices of the hypercube), the interior of the rectangle is assumed to be constant with this value. The rectangle is excluded from further partitioning. As can be seen in the figure, the algorithm approaches relatively fast to a mesh with fine resolution in the vicinity of the step. It is obvious from the figure, that the number of points of an equally spaced grid with the same resolution as the finest mesh is much larger. Calculating all points on the equally spaced grid would therefore require many more evaluations of the function $f(x, y)$, which is assumed to be time consuming.

However, for a step function, AMR may lead to wrong results as shown in Fig. 4.10 H, when the function has critical points, which are closer than the coarsest grid. In most cases, such mistakes can however be seen, when critically inspecting the plots. For the application discussed here, the calculation of titration curves, the function is not a step function, but a continuous function. The protonation probability is a real number between zero and one at each point and the transition between the extrema has a finite width. One defines a tolerance (unimportant difference in probability), which is accepted for the vertices of the cube to vary and still considering the region as constant. For a given tolerance and assumed sigmoidal titration behavior between the extrema, a chemical potential range can be estimated, which is considered safe (not to miss critical points) for the coarsest grid.

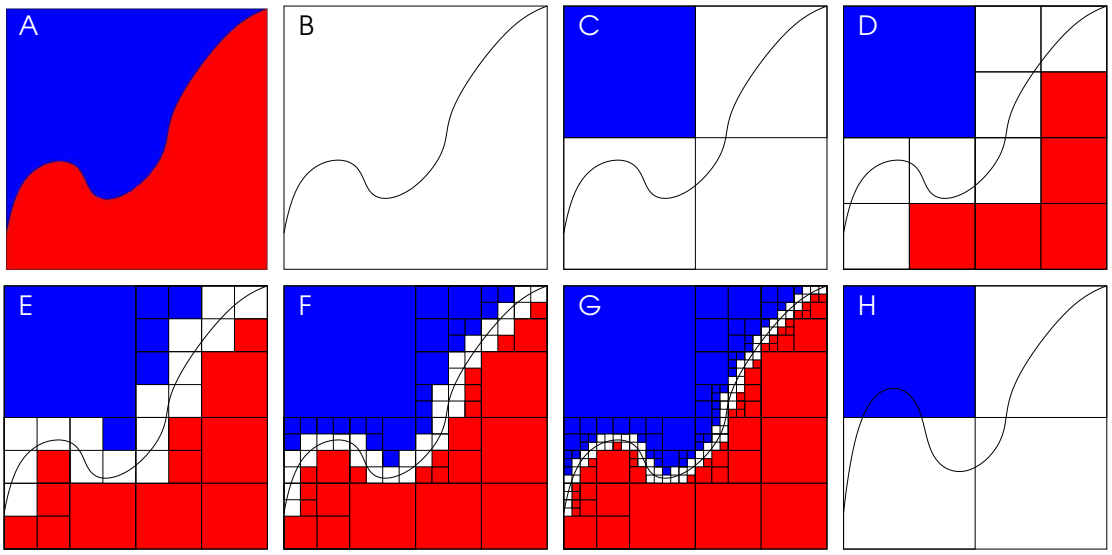


Figure 4.10. Approximation of a step function using adaptive mesh refinement (AMR). A: Arbitrary step function of two variables $f(x, y) = z$, where z is either zero (red) or one (blue). B: First step of AMR. Four corner points of the square are calculated. C: Second step of AMR. Four smaller squares are calculated by halving the edges of the outer square. Four points in the middle of each edge and the central point are calculated. For the upper left square all four corner points are identical (one). Therefore the region is marked as known and excluded from further calculations. D-G: Further steps of AMR. The squares from the previous step are divided into four sub-squares and the necessary corner points calculated. If the value at each corner point of a sub-square is identical it is marked as known and excluded from further calculations. After a few iterations the function is already very well approximated. Significantly less points need to be calculated compared to a homogeneous mesh of the finest grid size. H: AMR may fail if the function has critical points, which are closer than the coarsest grid.

My implementation allows application to an arbitrary number of chemical potentials (or ligand types) spanning a high dimensional cartesian coordinate system. For simplicity, space is partitioned in a binary fashion, *i.e.,* dividing each chemical potential range into halves at each step of the calculation. The implementation is recursive, because it was found more intuitive and I estimated, that a recursion depth of about 6 is sufficient for current applications. However, an alternative iterative algorithm may be easier to parallelize, which was not intended so far.

The application of AMR to titration calculations is however complicated by the fact, that a calculation of probabilities always acts on all sites and instances at the same time. Probabilities for a single site, where all other sites are kept fixed to a certain probability independent of the chemical potentials, are usually not desired (however, the algorithm would perform extremely well in such cases). Hence, minimizing $N_{\mathrm{steps},\lambda}$ of eq. 4.2 is not particularly efficient in dimensions of chemical potential space, where lots of sites titrate, forming nearly a continuum (*i.e.,* a protein has so many protonateable sites, that AMR can barely act on the chemical potential space of protons between pH 0 and 14). However, most systems have only one such dimension (*i.e.,* there are only a few redox active sites or ion binding sites which might have distinct titration ranges). Therefore, an implementation, which does not divide space in all dimensions equally, but only divides those dimensions, in which the chemical potential difference exceeds the tolerance, would lead to (hyper-) cuboids which have longer edges in those

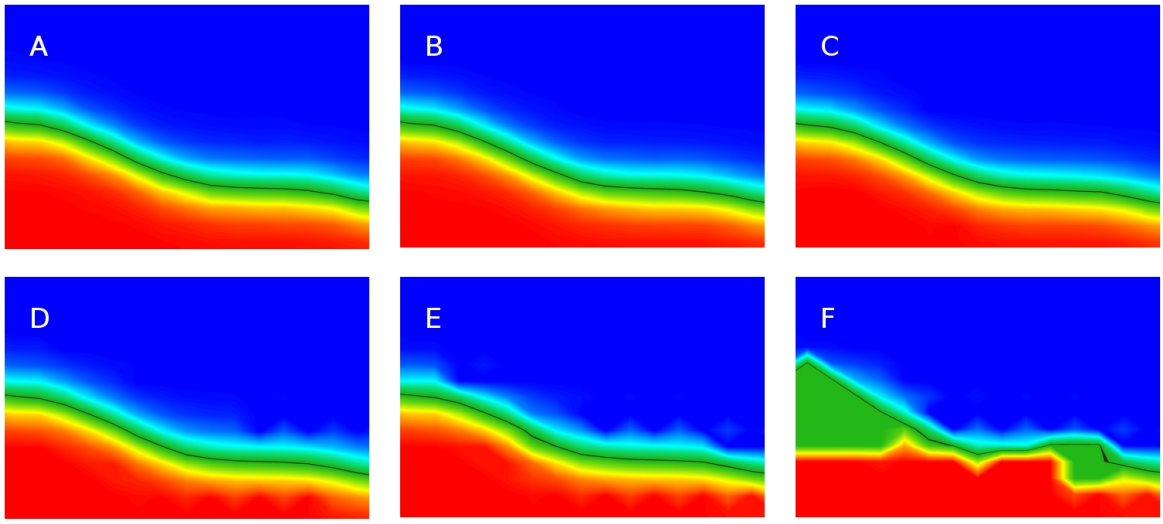


Figure 4.11. Benchmarks of AMR with different tolerance values. The quality of the titration curves is not much effected up to a tolerance of 30%, however the run-time can be significantly reduced. A: full Monte Carlo, 100% run-time B: AMR with tolerance of 2%, 80% run-time C: AMR with tolerance of 5%, 72% run-time D: AMR with tolerance of 10%, 70% run-time E: AMR with tolerance of 30%, 54% run-time F: AMR with tolerance of 50%, 49% run-time

dimensions, which are less crowded. Probably, other n-dimensional space filling tessellations (honeycombs) can be found, which faster approximate titration curves, however it has to be seen, if the computational effort to compute these polytopes does not slow down the AMR procedure too much.

Another approach reducing the basis of eq. 4.2, is minimizing other factors, in particular $N_{\mathrm{site},i}$. The computational effort can be reduced by removing instances which are not populated in the next coarser grid (therefore enhancing the sampling of the other instances). In particular for sites with many instances, it is often the case that only very few instances are in fact populated, but the other instances could not be excluded *a priori*. If a site is only represented by a single instance in a certain region of chemical potential space, the site can be removed (by modifying the intrinsic energies of all other instances of all other sites by the interaction energy with this instance) for finer grid calculations necessary for other sites. Removing sites reduces the length of the state vector (*i.e.*, $N_{\mathrm{site},i}$). In particular for calculations including rotamers it is common, that (rotameric) instances adopt a fixed ratio over large ranges of chemical potential space (Fig. 4.9 C). The state vector length could be reduced by such sites using a mean-field approximation (calculating mean interaction energies as described in section 5.2). Furthermore, for short state vectors a statistical evaluation of the partition function (by SMT) is more efficient than the Monte Carlo approximation scheme (in GMCT). The AMR algorithm can easily switch between the methods as soon as the state vector is shorter than a pre-defined threshold value.

Due to the multi-faced dependence of the performance of the algorithm on the system it was not possible to derive an analytical expression for the complexity of the algorithm. Two border cases can be formulated: First, all instances are equally populated through space. Without mean-field approximation, this would lead to a complexity identical to eq. 4.2. Due to the overhead of AMR some additional penalty would be added to the linear scanning of chemical potential space. Second, no instance is populated through space. It is sufficient to calculate the vertices of the coarsest grid, *i.e.,*

$$O\left(N_{\mathrm{site},i} \cdot N_{\mathrm{scans}} \cdot 2^{N_{\mathrm{ligand}}}\right). \tag{4.3}$$

More realistic scenarios perform between these two border cases. Very preliminary benchmarks of a test-implementation have shown, that a factor of two can easily be obtained for two dimensional titration calculations compared to sequential scanning of chemical potential space. In this test, space was partitioned in squares, not in probably more efficient rectangles. One dimension was the pH range of 0 to 14, which was covered to a large degree by protonateable sites, so that AMR could not perform very well in this dimension. The second dimension was the reduction potential of a single site, however AMR could not take much advantage out of this fact due to the strictly squared (not rectangular) partitioning scheme. Therefore, it is very likely, that AMR will perform much better with a rectangular partitioning scheme and more, but less populated dimensions. The current implementation introduces a large overhead by using GMCT (and SMT) as external helper programs and generating new input files for each square. A significant performance impact can probably be gained by implementing the algorithm together with the probability calculations in a single program (or using SWIG, section 4.4.4). However, interestingly the calculations have shown, that the tolerance can be chosen quite high (as 30%) without changing the titration curves significantly (Fig. 4.11), but reducing the run-time exponentially. The exponential dependence is probably due to the fact, that the ranges in which a site is considered to titrate get smaller. Therefore, larger squares can be used to approximate the titration curves and less smaller cubes are needed.

Another limitation of the current implementation is, that an evenly spaced grid is used, appropriate for the finest mesh resolution. This choice was made to ease the implementation, which was focussed on saving processor time. However, one of the advantages of AMR, the reduced memory consumption, is not exploited. For three dimensions, 64 points of double (8 byte) *per* dimension and 500 instances (in total) already 1GB of memory would be consumed. Since such a system is rather small, it can be estimated, that not unrealistic systems (five dimensions, 128 points, 5000 instances) could easily require petabytes of memory - three to four orders of magnitude beyond what is currently commonly available as disk space and five to six orders of magnitude beyond what is currently available as memory. Therefore, exploiting the conceptionally inherent memory saving capabilities of AMR using a sparse matrix (sparse tensor) or graph algorithm for storing probability values, may be as required for future applications as the processor time saving capabilities.

A project, which Eva-Maria Krammer already started, is the competitive binding of different inhibitors instead of Coenzyme Q to the bacterial reaction center in addition to her studies on electron and proton binding to the protein and its cofactors. This application will require not only to include the chemical potentials of protons and electrons, but also of Coenzyme Q and its inhibitors into the ligand binding calculations. Titrations in such a four or higher

dimensional system will be very time-consuming and barely doable without AMR. Here, AMR is expected to perform much better, because only a few proton and electron binding sites in the vicinity will depend on the cofactor or inhibitor being bound. Similarly, the binding of inhibitors will only depend on the protonation or redox state of a few sites of the protein.

4.3 QMPB - A Program for Calculating Binding Energetics of Multiple Ligands

QMPB allows to calculate intrinsic energies and interaction energies of sites with multiple instances and multiple ligand types. It performs energy calculations based on the theory discussed in chapter 3. Calculations for a particular site can be based on experimental results on model compounds or on results of quantum chemical calculations. The flexibility of QMPB even allows to combine sites of both types in a single calculation. Model compounds can be freely defined to optimally fit the experimental setup by the calculation. The current version is able to deal with three dielectric regions, unlike Multiflex, which had only two. More dielectric regions are planned for the future. QMPB uses external helper programs (discussed in section 4.4) to solve the LPBE to calculate electrostatic energies.

4.3.1 Aims and General Concepts for the Development of QMPB

My primary aim when designing QMPB was to make it as transparent to the user as possible. Each step should be understandable and repeatable in small parts by the user. This transparency was felt to be necessary after having a more monolithic program, Multiflex, which was often used as a black box, sometimes leading to mistakes. Also the theory, introduced in chapter 3, needed to be tested carefully. The transparent and modular design allowed Thomas Ullmann for example to replace the helper programs for the electrostatic calculations by versions, which included a new membrane model and membrane potentials.

An important aim was also to provide linear scalability with the total number of instances, *i.e.,* the number of independent PBE calculations, which need to be done to solve the problem. Since these calculations are independent, the parallel execution on different processors should be possible. Because MEAD is the PBE solver primarily used in the group, it was chosen as basis to develop the helper programs discussed in section 4.4. Other PBE solvers as APBS could be used, if someone desires and develops analogous helper programs. In fact, most of the processor time is spend by these helper programs, so that the runtime of QMPB itself is negligible. Therefore, to obtain a linear scaling with the number of processors, it was sufficient to allow running the helper programs in parallel. Also it was not required to write QMPB completely in C++ (as MEAD is written in), but to use the - especially for string processing, which is a major part of QMPB - more convenient and flexible language Perl.

QMPB requests an input file in a rather general, human readable format. However, the input file will usually not be written by the user, because it can be very long for proteins with many sites. Instead pre-processors (section 4.5 and section 4.6) should generate the input file, which should only be controlled and in exceptional cases modified or extended by the user. The input file includes a large number of other files, mostly pqr-files containing the coordinates and charges of instances. Also these files are generated by the pre-processor

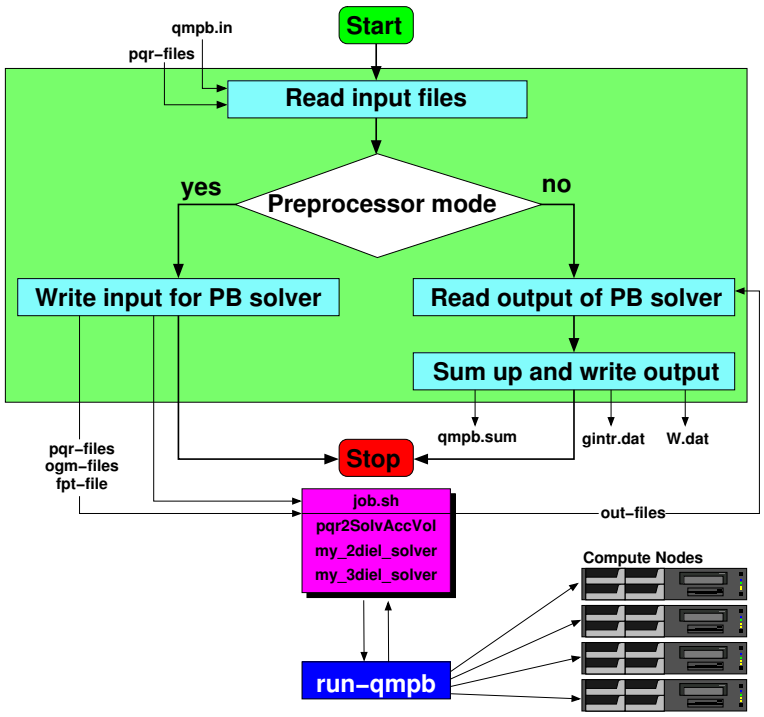


Figure 4.12. Conceptional flowchart of QMPB. QMPB (green box) is called twice, once in a pre-processor mode to set up the calculations by the PB solver programs (left half of the figure) and once to post-process the output files of the PB solver programs and write `gintr.dat` and `W.dat` (right half of the figure). The PB solver calculations can be done in parallel. `run-qmpb` splits up the `job.sh` shell script (generated by QMPB) and runs the parts on different computers. The figures in this section are color coded (main program - green, `site_master` - cyan, `site` - yellow, `job.sh` - pink).

programs. Therefore, a large part of the knowledge how to set up QMPB calculations properly became part of the pre-processors. This division leaves QMPB only with its core function to read and validate the input, prepare the helper programs to run, collect the output of the helper programs and sum up the intrinsic energies according to the thermodynamic cycles in section 3. The intrinsic energies and the interaction energies are each written to a file, which can be read by other programs (*e.g.*, GMCT, SMT, DMC) to perform the titration or (electron- or proton-) transfer calculations.

4.3.2 Overview of a Program Run

The considerations in section 4.3.1 lead to a three-step design: First, QMPB is run in a pre-processor mode (left half of Fig. 4.12). It reads the input file (`qmpb.in` and pqr-files) and generates the input for the helper programs solving the PBE (*i.e.*, pqr-files, ogm-files for the grid definition and a extended fpt-file to calculate the interaction energy). QMPB writes a shell script (`job.sh`), doing calls to the helper programs (section 4.4).

Second, this shell script (`job.sh`) is executed. Optionally, the shell script can be split into parts and submitted to different computers (*i.e.*, *via* a queuing system of a compute cluster).

The approach chosen here completely separates off the questions concerning parallelization and distributed computing. In QMPB no assumptions need to be made about the computing environment. Instead an external program can be adopted for the particularities of the site. Currently, an additional Perl program (`run-qmpb`) fulfills this function.

Third, QMPB is run in a post-processor mode (right half of Fig. 4.12). It reads the input file (`qmpb.in` and pqr-files) and the output files (`out-files`) generated by the helper programs. Intrinsic energies are computed according to the thermodynamic cycles in section 3. Three files are written: one for the intrinsic energies (`gintr.dat`), one for the interaction energies (`W.dat`) and a summary file (`qmpb.sum`). The summary file contains all the contributions to the intrinsic energies and is usefull for the user to understand the calculations as they are performed by QMPB.

4.3.3 The Input File

This section outlines the input file of QMPB placing emphasis on the general structure, closely reflecting the object structure discussed in the next section, and leaving a full explanation for each possible input option to appendix A.6. The aim of defining the input file format was to allow access to the full functionality of QMPB. Since it was thought to be written by pre-processors, it was not important to be very compact. Instead it should be well readable by the user to check and modify input options. The input is structured into a block with general options and blocks for each instance of each site. The options are mostly given as `key=value` pairs, where the value can sometimes be a space separated list of values. The order of the parameters in a block and the order of blocks is arbitrary. I refer to line numbers of the given examples in the following, however the line numbers are generally not important for the program. All energies are given in units of $\frac{\text{kcal}}{\text{mol}}$.

The General Block

The general block may contain options similar to the following example:

```
1  meadpath = /home/essigke/bin
2  T = 300
3  I = 0.1
4  backfile = background.pqr
5  MGMcenter = ON_CENT_OF_INTR
6  MGMpoints = 131
7  MGMspace = 0.2
8  OGMcenter = ON_GEOM_CENT ON_CENT_OF_INTR
9  OGMpoints = 131 131
10 OGMspace = 1 0.2
11 epsin1 = 1
12 epsin2 = 4
13 Ligand_Labels = proton electron
```

Line 1 in the example defines the path to the helper programs (section 4.4), then some parameters for the Poisson-Boltzmann solver are given (absolute temperature T and ionic strength I). Line 4 gives the name of the background pqr-file. It contains the coordinates, charges and

radii of all atoms of the molecule not belonging to any site. The next six lines specify parameters for the grid, on which the PBE is solved numerically. The options starting with `MGM` define parameters for the grid of model compounds (or `Modelsite`), the other define parameters for the grid of instances of sites (the terminology is adopted from MEAD). It is shown, that either a single or multiple values can be given. As many focussing steps are done when solving the PBE as there are values. In a similar way, grids can also be specified for instances (in the blocks below, however only once *per* site) overwriting the default values given here. The option to use different grid definitions allows using a rather small grid (which is fast to solve) for most sites, but larger grids for sites, which require it.

The centering type allows three options (as in Multiflex). `ON_ORIGIN` defines the center of the grid to be at the origin $(0,0,0)$. `ON_GEOM_CENT` defines the center of the grid to be at the geometrical center of the instance pqr-file. Both options are translated into coordinates by the MEAD library. The third option `ON_CENT_OF_INTR` (on center of interest) defines the center of the grid to be at the geometrical center of all instances of a particular site. This center is therefore identical for all instances of the site as it is required to cancel grid artefacts. It is at an optimal position to allow small grids with high resolution. Since calculation of this center requires knowledge of the coordinates of all atoms in all instances, it is done by QMPB. The keyword is replaced by the calculated coordinates in the grid files written by QMPB for all instances.

The keys `epsin1`, `epsin2` and (optionally) `epsext` define the dielectric constant of the three dielectric regions currently available in QMPB. In the current version of QMPB they are mandatory, but in future versions they probably will be removed in favor of the `eps` key in the instance blocks. A list of names for the ligand types are assigned to the key `Ligand_Labels` in line 13. The names have to be given in the same order as the values for the keys `N` and `Gfree` discussed below.

An Instance of a QMsite

The next example shows the definition of an instance of a site, for which absolute binding energies should be calculated (section 3.3). The instance can be parameterized by quantum mechanical (QM) calculations, explaining the name `QMsite`.

```
1  QMsite site_C_PHE_39_A instance_0_ox
2    file=site_C_PHE_39_Ainstance_0_ox.pqr
3    sid=21
4    iid=0
5    eps= 1
6    Hqm=-14998.1
7    Gvib=98.693
8    N= 0 0
9    Gfree= 0  -102.1558
10   QM_corr C_PHE_39_A.pqr int_C_PHE_39_A.pqr
11   QM_corr O_PHE_39_A.pqr int_O_PHE_39_A.pqr
12 ...
```

Line 1 specifies the type of site, a unique label for the site and a unique label for the instance. These labels should be intuitive for the user, facilitating a quick understanding of the input and output files, where these labels are reused. Line 2 sets the name of the pqr-file, which is defining the coordinates, charges and radii of atoms belonging to the instance of the site. Line 5 defines the dielectric constant of the region of the site. The optional keys `sid` and `iid` (line 3 and 4) associate the site and instance with unique numbers. They are used for sorting the output, which is usefull for post-processing scripts, but also comparison of different calculations. If the keys `sid` and `iid` are omitted, they are automatically generated by QMPB. Line 6 and 7 define the quantum chemical energy of the instance (`Hqm`, total bonding energy or energy of formation, $H_{\mathrm{vac},i}(j_k)$ in eq. 3.32) and the vibrational energy (`Gvib`, which is $G_{\mathrm{vib},i}(j_k)$ in eq. 3.32), respectively. Line 8 and 9 give the number of bound ligands of each type (`N`, which is $n_{\lambda,i}(j_k)$ in eq. 3.7 and eq. 3.34) and the standard chemical potentials for all ligand types (`Gfree`, which is μ°_{λ} in eq. 3.34). Line 10 and the following lines are for calculating the correction energy ($G_{\mathrm{corr},i}(j_k)$ in eq. 3.32). Section 3.5.2 explains, why this correction is necessary. The first pqr-file in each `QM_corr` line specifies an atom in the `QMsite` (named Q1 or Q2 in Fig. 3.7) and the second pqr-file specifies the atoms outside of this site, with which it artificially interacts (M1 and M2 in Fig. 3.7). Alternatively to the `QM_corr` lines, a line with the key `Gcorr` and the correction energy as value, can be given. Calculating the correction energy is very expensive and usually redundant in repetitive calculations.

An Instance of a MMsite - Non-Ligand Binding Reference Rotamer Form

Section 3.4 describes the calculation of energies of instances of sites relative to a reference. Four types of such instances were distinguished. The energy of the non-ligand binding reference rotamer form is defined to be zero. Nevertheless, the electrostatic energy in the heterogeneous environment of the molecule compared to a homogeneous dielectric is required by non-reference rotamer forms. Therefore the energy is calculated.

```
1 MMsite site_HT1_ALA_1_A instance_0_C
2   file=site_HT1_ALA_1_Ainstance_0_C.pqr
3   sid=0
4   iid=0
5   ref=self
6   eps= 4
7   N= 0 0
8  Gmm=0.3299
```

As for the `QMsite`, line 1 specifies the type of site, a unique label for the site and a unique label for the instance. The name `MMsite` was chosen in contrast to `QMsite`. Often the energies of rotamers of this type of site (given in line 8) are obtained from a molecular mechanics (MM) force field. Line 2-4 and 6-7 are analogous to the `QMsite`. The key `ref` in line 5 has the value `self`, indicating that the reference instance of this instance is the instance itself, neither another `MMsite` nor a `Modelsite`.

An Instance of a MMsite - Non-Ligand Binding Non-Reference Rotamer Form

The next example shows the input for an instance, which takes the previous example as reference rotamer:

```
1 MMsite site_HT1_ALA_1_A instance_1_A
2   file=site_HT1_ALA_1_Ainstance_1_A.pqr
3   sid=0
4   iid=1
5   ref=instance_0_C
6  Gmm=0.3355
```

The key `ref` in line 5 now has the value `instance_0_C`, which is the instance label of the previous example instance. By that the program knows, that it should use `instance_0_C` as reference instance for `instance_1_A`. The rotamer energy in line 6 is different from the reference instance. Note, that certain keys can only be given to the reference instance, not to non-reference instances.

An Instance of a MMsite - Ligand Binding Reference Rotamer Form

In the previous examples, instances of type `MMsite` only differed in their coordinates (rotamer form), not in the number of ligands bound (charge form). In this case, the ligand binding is described by a model compound in solution, for which ligand binding energies are known (*e.g.,* from experiment).

```
1 MMsite site_CE_LYS_4_A instance_0_p_ROT_-5
2   file=site_CE_LYS_4_Ainstance_0_p_ROT_-5.pqr
3   sid=1
4   iid=0
5   ref=model_instance_0_p_ROT_-5
6  Gmm=0.0316
```

The key `ref` in line 5 now has the value `model_instance_0_p_ROT_-5`, which is the instance label of the instance of a `Modelsite`. If the site only exists in a single rotamer form, the option `nohomo` should be given for the `MMsite`. This option avoids the calculation of the instance in a homogeneous dielectric, which is only needed for rotamers.

An Instance of a Modelsite

The next example shows how to specify a model compound:

```
1 Modelsite site_CE_LYS_4_A model_instance_0_p_ROT_-5
2   file=model_site_CE_LYS_4_Ainstance_0_p_ROT_-5.pqr
3   sid=1
4   iid=0
5   ref=instance_0_p_ROT_-5
6   Gmodel=-14.2665506152
7   eps= 4
8   N= 1 0
```

The input is very similar to a non-ligand binding reference rotamer form of a `MMsite`. The type of site is now `Modelsite`. A rotamer energy `Gmm` is not specified, because it is specified for the associated `MMsite` given by the key `ref`. Instead, the energy of the model compound in solution is specified with the key `Gmodel`. It is important, that all `Modelsite` instances of a site (due to different charge forms) are in the same rotamer form. The file specifying the `Modelsite` has to contain all atoms (with identical coordinates, charges and radii) as the associated `MMsite`, but usually contains additional atoms, which were present when determining the model energy in solution. The dielectric boundaries are calculated from all atoms in this file. The additional atoms contribute with their charges as background charge set to the electrostatic energy.

An Instance of a MMsite - Ligand Binding Non-Reference Rotamer Form

Finally, there might be rotamer form, which takes a ligand binding rotamer form as reference:

```
1  MMsite site_CE_LYS_4_A instance_1_p_ROT_24
2    file=site_CE_LYS_4_Ainstance_1_p_ROT_24.pqr
3    sid=1
4    iid=1
5    ref=instance_0_p_ROT_-5
6   Gmm=0.6172
```

The input is analogous to a non-ligand binding non-reference rotamer form, but the key `ref` now points to instance `instance_0_p_ROT_-5`. The distinction, necessary for the different calculation schemes can only be made by analyzing the `ref` value of the reference instance. If it is `self` it is a non-ligand binding non-reference rotamer form, instead if a `Modelsite` instance is specified, it is a ligand binding non-reference rotamer form.

4.3.4 The `job.sh` Script

From the input file QMPB produces a file `job.sh`, when it is run in pre-processor mode. Here are the lines defined by the input blocks given as examples above (splitted lines are marked by a backslash):

```
1  #!/bin/sh
2  /home/essigke/bin/pqr2SolvAccVol epsin1_region > pqr2SolvAccVol_epsin1_region.out
3  /home/essigke/bin/pqr2SolvAccVol epsin2_region > pqr2SolvAccVol_epsin2_region.out
4  /home/essigke/bin/my_3diel_solver -epslset epsin1_region -eps2set epsin2_region -T 300 \
5    -ionicstr 0.1 -epsin1 1 -epsin2 4 -fpt site -epshomo 1 site_C_PHE_39_A_instance_0_ox \
6     background > site_C_PHE_39_A_instance_0_ox.out
7  /home/essigke/bin/my_3diel_solver -epslset epsin1_region -eps2set epsin2_region -T 300 \
8    -ionicstr 0.1 -epsin1 1 -epsin2 4 C_PHE_39_A int_C_PHE_39_A > C_PHE_39_A.out
9  /home/essigke/bin/my_3diel_solver -epslset epsin1_region -eps2set epsin2_region -T 300 \
10   -ionicstr 0.1 -epsin1 1 -epsin2 4 O_PHE_39_A int_O_PHE_39_A > O_PHE_39_A.out
11 ...
12 /home/essigke/bin/my_3diel_solver -epslset epsin1_region -eps2set epsin2_region -T 300 \
13   -ionicstr 0.1 -epsin1 1 -epsin2 4 -fpt site -epshomo 4 site_HT1_ALA_1_A_instance_0_C \
14    background > site_HT1_ALA_1_A_instance_0_C.out
```

```
15 /home/essigke/bin/my_3diel_solver -eps1set epsin1_region -eps2set epsin2_region -T 300 \
16   -ionicstr 0.1 -epsin1 1 -epsin2 4 -fpt site -epshomo 4 site_HT1_ALA_1_A_instance_1_A \
17     background > site_HT1_ALA_1_A_instance_1_A.out
18 ...
19 /home/essigke/bin/my_3diel_solver -eps1set epsin1_region -eps2set epsin2_region -T 300 \
20   -ionicstr 0.1 -epsin1 1 -epsin2 4 -fpt site -epshomo 4 \
21     site_CE_LYS_4_A_instance_0_p_MAX_ROT_-5 background > \
22     site_CE_LYS_4_A_instance_0_p_MAX_ROT_-5.out
23 /home/essigke/bin/my_2diel_solver -epsin 4 -T 300 -ionicstr 0.1 \
24      site_CE_LYS_4_A_instance_0_p_MAX_ROT_-5 \
25      site_CE_LYS_4_A_model_instance_0_p_MAX_ROT_-5_back2d > \
26      site_CE_LYS_4_A_instance_0_p_MAX_ROT_s-5_2d.out
27 /home/essigke/bin/my_3diel_solver -eps1set epsin1_region \
28   -eps2set epsin2_region -T 300 -ionicstr 0.1 -epsin1 1 -epsin2 4 -fpt site \
29   -epshomo 4 site_CE_LYS_4_A_instance_1_p_MAX_ROT_24 background > \
30     site_CE_LYS_4_A_instance_1_p_MAX_ROT_24.out
31 ...
```

The first two lines calculate the analytical surface representation for the two dielectric regions of the `QMsite` instances (`epsin1_region`) and protein (`epsin2_region`) including the `background.pqr` file and all `QMsite` and `MMsite` instances using the helper program Pqr2SolvAccVol. More detailed information on the calculation of analytical surfaces and proper boundary definitions is given in section 4.4.1.

Lines 4-10 show the calculation steps invoked by the given input fragment for an instance of a `QMsite`. The first call of My_3Diel_Solver (section 4.4.2) calculates the Born energy of the instance in the three dielectric environment defined by the two solute regions `epsin1_region` and `epsin2_region` and the solvent. Additionally, the Born energy is calculated in a homogeneous dielectric of one (vacuum), due to the `epshomo` parameter. The background energy is calculated as the interaction of the electrostatic potential of the instance with all charges in the `background.pqr` file. Additionally, interaction energies are calculated for all atoms of all instances of all other sites according to the extended fpt-file `site.fpt` generated by QMPB. The file format is described in appendix A.1.5. All energy contributions are stored in the output file of My_3Diel_Solver. To calculate the energy correction discussed in section 3.3.1, the next calls of My_3Diel_Solver are made. The same boundary definitions as before are used, but the two pqr-files are those given by the `QM_corr` option in the input file. Here, only the background energy due to the interaction of the electrostatic potential of the atom in the first pqr-file $\varphi_{\mathrm{protein}}(\vec{r}_{\mathrm{a,i}}; q_{\mathrm{a,i}}(j_k))$ with the charges of the atoms $Q_{\mathrm{protein,A,i}}$ in the second pqr-file is of interest. Therefore, it is neither necessary to do a vacuum calculation to cancel grid artefacts in the Born energy (by the option `epshomo` as in line5), nor interaction energy calculations according to an extended fpt-file (by the option `fpt` as in line 5). Lines 12-17 call My_3Diel_Solver twice to calculate the rotamer instances of a `MMsite` without model compound. Except that the calculation in the homogeneous dielectric is done with the dielectric constant of the protein, the program calls are analogous to the calculation of the `QMsite` instance. Lines 19-30 show the calculations for the `MMsite` with model compound. Again, the calls of My_3Diel_Solver are analogous. However, for the instance of the model compound My_2Diel_Solver (section 4.4.3) is called. Since for every site the model compound has different dielectric boundaries they are

not calculated with Pqr2SolvAccVol in advance. The first pqr-file is identical in the calculation of the associated `MMsite` and in the calculation on the model compound. The second pqr-file contains all atoms, which were present in the file specified in the QMPB input, but the charges of the atoms found in the pqr-file of the `MMsite` are set to zero to calculate the background energy.

4.3.5 The Output Files

In a third step, after generation and execution of the `job.sh` script, QMPB is run in post-processor mode. The input file and all the output files generated by the helper programs are read. The electrostatic energy terms are extracted from the output files and properly added with the energies specified in the QMPB input file according to the equations given in chapter 3. The file `qmpb.sum` contains all energy contributions to the intrinsic energy in a readable format. For the example instance of a `QMsite`, page 122, the following information is given:

```
1  site_C_PHE_39_A instance_0_ox
2
3  Born Energy:
4  Site in ThreeDielectric 15283.6
5  Site in Homogeneous Dielectric 15439.8
6  Difference: -156.199999999999
7  Background Energy:
8  Site in ThreeDielectric -21.6103
9
10 QM Energies:
11 Total Bonding Energy: -14998.1
12 Vibrational Energy: 98.693
13 Energy of free ligands: -102.1558 (2 ligand types)
14
15 Error correction for 1-2 and 1-3 interaction with protein: -47.5910176
16
17 Sum: -15131.7820824
```

The Born energy was calculated in a homogeneous and three dielectric environment (lines 5 and 4). The given energies contain grid artefacts, which are canceled by calculating the difference (line 6). The background energy in the three dielectric environment is given (line 8). This energy contains the artefacts due to bonded and angle interactions which should be excluded. The correction energy is given in line 15 and subtracted from the sum. The quantum mechanical energy contributions (bonding energy, line 11 and vibrational energy, line 12) as well as the energy of free ligands (line 13) were given in the input file. Finally, all contributions are summed up and the intrinsic energy is given (line 17).

Next, the calculation on two rotamer forms by instances of a `MMsite` was discussed on pages 123–124. The `qmpb.sum` file gives the following information:

```
1  site_HT1_ALA_1_A instance_0_C
2
3  Reference Rotameric Form without Modelsite!
```

```
4
5  Born Energy:
6  Site in ThreeDielectric 137.608
7
8  Background Energy:
9  Site in ThreeDielectric 0.653352
10
11 Intrinsic Model Energy: 0
12
13 Sum of contributions: 0
```

```
14 site_HT1_ALA_1_A instance_1_A
15
16 Born Energy:
17 Site in ThreeDielectric 129.958
18 Site in Homogeneous Dielectric 146.483
19 Site in ThreeDielectric in reference rotameric form 137.608
20 Site in reference rotameric form in homogeneous dielectric 154.142
21 Difference 0.00899999999998613
22
23 Background Energy:
24 Site in ThreeDielectric 0.505013
25 Site in ThreeDielectric in reference rotameric form0.653352
26 Difference −0.148339
27
28 Molecular Mechanics Energy:
29 Site 0.3355
30 Site in reference rotameric form 0.3299
31 Difference 0.00559999999999999
32
33 Sum of contributions: −0.133739000000014
```

Lines 1 to 13 describe the reference rotamer instance. By definition the intrinsic energy is zero. However, for the non-reference rotamer instance (lines 14 - 33) the calculated Born and background energies are given for both instances. Double differences in Born (line 21), differences in background (line 26) and differences in rotamer energies (line 31) are calculated and summed up. Line 33 gives the intrinsic energy of the non-reference instance.

Finally, here are the lines of the `qmpb.sum` file referring to the two instances of the `MMsite` with model compound (pages 124–125):

```
1  site_CE_LYS_4_A instance_0_p_MAX_ROT_−5
2
3  Reference Rotameric Form!
4
5  Born Energy:
6  Site in ThreeDielectric 129.152
7  Modelsite in TwoDielectric 128.427
8  Difference 0.724999999999994
```

```
9
10 Background Energy:
11 Site in ThreeDielectric -0.746759
12 Modelsite in TwoDielectric -0.774909
13 Difference 0.02815
14
15 Intrinsic Model Energy: -14.2665506152
16
17 Sum of contributions: -13.5134006152
```

```
18 site_CE_LYS_4_A instance_1_p_MAX_ROT_24
19
20 Born Energy:
21 Site in ThreeDielectric 216.191
22 Site in Homogeneous Dielectric 234.968
23 Modelsite in reference rotameric form 128.427
24 Site in reference rotameric form in homogeneous dielectric 148.139
25 Difference 0.935000000000031
26
27 Background Energy:
28 Site in ThreeDielectric -0.741824
29 Modelsite in TwoDielectric -0.774909
30 Difference 0.0330849999999999
31
32 Intrinsic Model Energy: -14.2665506152
33
34 Molecular Mechanics Energy:
35 Site 0.6172
36 Site in reference rotameric form 0.0316
37 Difference 0.5856
38
39 Sum of contributions: -12.7128656152
```

Line 1 to 17 and line 18 to 39 give the energy contributions of the two instances of the `MMsite`. The `Modelsite` is not mentioned in a separate section, but their energies are given in the sections of each instance, where they are needed for the calculation. For the reference `MMsite` (line 1 - 17) the intrinsic energy (line 17) is calculated as sum of the energy of the model compound (line 15) and the difference in Born and background energy (line 8 and 13, respectively) between the instance and the model instance. For the non-reference rotamer instance (line 18 - 39), the double difference in Born energy (line 25) is calculated according to eq. 3.41 and the difference in background energy (line 30) is calculated according to eq. 3.42, which can be derived form Fig. 3.9. The difference in rotamer energy (line 37) is calculated and the terms are summed up to obtain the intrinsic energy (line 39)

The intrinsic energies are tabulated in the file `gintr.dat`:

```
1 #   siteID     instID  sitelabel          instlabel          intrinsic_energy proton electron
2          1          1  site_HT1_ALA_1_A instance_0_C                0.000      0        0
3          1          2  site_HT1_ALA_1_A instance_1_A               -0.134      0        0
```

```
4          2          1  site_CE_LYS_4_A  instance_0_p_MAX_ROT_-5      -13.513    1    0
5          2          2  site_CE_LYS_4_A  instance_1_p_MAX_ROT_24     -12.713    1    0
6  ...
7         22          1  site_C_PHE_39_A  instance_0_ox              -15131.782    0    0
```

The first two columns give the site and instance number (`sid` and `iid`) incremented by one as it is required for SMT and GMCT. QMPB usually starts counting at zero (as in the input file) and gives these numbers also as default output format. The next two columns give the site and instance label as they were given in the input file. The fifth column contains the intrinsic energy and all following columns give the number of bound ligands for each ligand type.

The third output file `W.dat` contains all the interaction energies (each line had to be splitted over two lines):

```
1  # siteID1 instID1 sitelab1          instlab1       \
2     siteID2 instID2 sitelab2          instlab2      symmetrized_energy interaction_energy  error
3         1       1 site_HT1_ALA_1_A instance_0_C    \
4          1       1 site_HT1_ALA_1_A instance_0_C              0.000              0.000  0.000
5         1       1 site_HT1_ALA_1_A instance_0_C    \
6          1       2 site_HT1_ALA_1_A instance_1_A              0.000              0.000  0.000
7  ...
8         1       1 site_HT1_ALA_1_A instance_0_C    \
9           2       1 site_CE_LYS_4_A instance_0_p_MAX_ROT_-5    0.108              0.109  0.001
10        1       1 site_HT1_ALA_1_A instance_0_C    \
11          2       2 site_CE_LYS_4_A instance_1_p_MAX_ROT_24    0.110              0.110  0.000
12 ...
13        1       1 site_HT1_ALA_1_A instance_0_C     \
14         22       1 site_C_PHE_39_A instance_0_ox             -0.053             -0.053  0.000
15 ...
```

The `W.dat` file contains the pairwise interaction energies of each instance (specified by site and instance number and site and instance label) with each other instance (specified by the same four attributes). The calculated interaction energy is given in column 10. Since each interaction energy is calculated twice (as electrostatic potential of site 1 times charge of site 2 and electrostatic potential of site 2 times charge of site 1) this redundancy can be used for error estimation. The symmetrized interaction energy (column 9) is the arithmetic average of the two interaction energies and the error (column 11) is the difference between the interaction energy and the symmetrized interaction energy.

4.3.6 Hierarchy and Collabortation of Objects

QMPB uses the object-oriented programming (OOP) paradigm to structure the code internally. As mentioned in section 4.1, QMPB has an abstract `site` class. From this abstract class `QMsite`, `MMsite` and `Modelsite` classes are derived (Fig. 4.13). The advantage of the abstract definition of a `site` is that all common properties of the three derived classes can be implemented at a central place and be used and potentially modified by the derived classes. These classes are instantiated, generating an object for each instance of a site. To manage all the `site` objects I developed a class called `site_master`. It provides methods to the main

program for the different steps of the QMPB calculation. The methods usually delegate the work to appropriate methods of the `site` class. Due to the OOP approach, the `site_master` class does not need to differentiate between objects of type `QMsite` and `MMsite`, but can instead rely that the object will act according to its type, *e.g.,* if the method `print_G` is invoked, due to the polymorphism different methods are called and the energy is computed differently for `QMsite` and `MMsite` objects. The sequence of the different calculation steps is shown in Fig. 4.14. Additionally, there is a `pqr` class, which encapsulates the contense of a pqr-file in an object. It allows reading, writing and limited modification of pqr-files. The `pqr` objects are often helper objects for objects derived from one of the `site` classes.

The site_master class

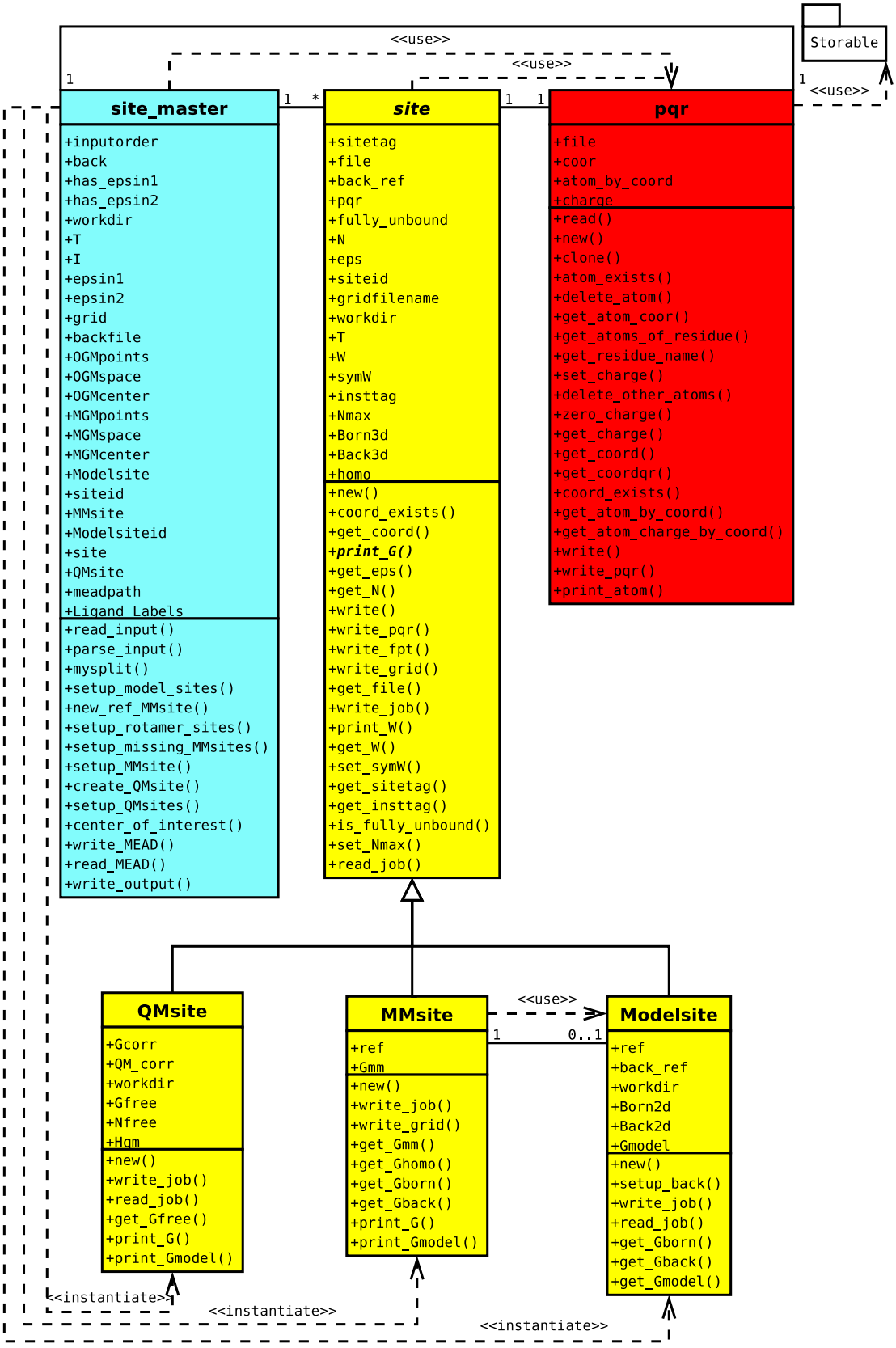
The `site_master` class represents a QMPB calculation on a single conformer. Objects are constructed by the method `read_input`, which takes a QMPB input file as most important parameter (Fig. 4.14). The input file is parsed and objects of the other classes are constructed according to the input file. A major part of the code in the `site_master` object is dedicated to these two steps. Other public methods, *i.e.,* methods accessed by the main program, are `write_MEAD` and `read_MEAD`, which generate the input for and reads the output file of the MEAD based helper programs (section 4.4). In fact, `site_master` only iterates over the generated `site` objects and triggers them to do the write and read operations for their particular instance. The method `write_output` generates the output files `gintr.dat`, `W.dat` and `qmpb.sum` also by initiating each instance to write its data into the files.

The site class

The site class is an abstract base class for the `QMsite`, `MMsite` or `Modelsite` classes (Fig. 4.13). It implements methods common to instances of all types of sites. The constructor `new` takes a hash with the options given for the instance as parameter. This hash is previously filled by the input file parsing routines of `site_master`. The constructor does the validation for the options, which are common to all sites. Each derived class overwrites the constructor to include validation for the options particular to its class. The constructor also generates a `pqr` object from the filename given in the input file, describing the coordinates, charges and radii of atoms belonging to this instance. The methods `write_job` and `read_job` are called by `site_master` `write_MEAD` and `read_MEAD` (Fig. 4.14). Besides a number of accessor and helper methods, the class implements the abstract method `print_G`, which is supposed to be overwritten implementing the calculation of the intrinsic energy and the generation of appropriate lines for the `qmpb.sum` and `gintr.dat` files for each derived class. The method `print_W` implements the generation of the instance specific part of the `W.dat` file.

The QMsite class

Objects of the `QMsite` class represent instances of a site, for which the intrinsic energy (section 3.3) should be calculated based on parameters obtained by quantum chemical methods. The constructor checks for the `QMsite` specific options, *i.e.,* the quantum mechanical energy (`Hqm`), the vibrational energy (`Gvib`) and an array of standard chemical potentials for all ligand types (`Gfree`). Optionally, a correction energy (`Gcorr`) can be defined or calculated from a


Storable
<<use>>
<<use>>
<<use>>
1
1
1 * 1 1
site_master
+inputorder
+back
+has_epsin1
+has_epsin2
+workdir
+T
+I
+epsin1
+epsin2
+grid
+backfile
+OGMpoints
+OGMspace
+OGMcenter
+MGMpoints
+MGMspace
+MGMcenter
+Modelsite
+siteid
+MMsite
+Modelsiteid
+site
+QMsite
+meadpath
+Ligand_Labels
+read_input()
+parse_input()
+mysplit()
+setup_model_sites()
+new_ref_MMsite()
+setup_rotamer_sites()
+setup_missing_MMsites()
+setup_MMsite()
+create_QMsite()
+setup_QMsites()
+center_of_interest()
+write_MEAD()
+read_MEAD()
+write_output()
site
+sitetag
+file
+back_ref
+pqr
+fully_unbound
+N
+eps
+siteid
+gridfilename
+workdir
+T
+W
+symW
+insttag
+Nmax
+Born3d
+Back3d
+homo
+new()
+coord_exists()
+get_coord()
+print_G()
+get_eps()
+get_N()
+write()
+write_pqr()
+write_fpt()
+write_grid()
+get_file()
+write_job()
+print_W()
+get_W()
+set_symW()
+get_sitetag()
+get_insttag()
+is_fully_unbound()
+set_Nmax()
+read_job()
pqr
+file
+coor
+atom_by_coord
+charge
+read()
+new()
+clone()
+atom_exists()
+delete_atom()
+get_atom_coor()
+get_atoms_of_residue()
+get_residue_name()
+set_charge()
+delete_other_atoms()
+zero_charge()
+get_charge()
+get_coord()
+get_coordqr()
+coord_exists()
+get_atom_by_coord()
+get_atom_charge_by_coord()
+write()
+write_pqr()
+print_atom()
QMsite
+Gcorr
+QM_corr
+workdir
+Gfree
+Nfree
+Hqm
+new()
+write_job()
+read_job()
+get_Gfree()
+print_G()
+print_Gmodel()
MMsite
+ref
+Gmm
+new()
+write_job()
+write_grid()
+get_Gmm()
+get_Ghomo()
+get_Gborn()
+get_Gback()
+print_G()
+print_Gmodel()
<<use>>
1
0..1
Modelsite
+ref
+back_ref
+workdir
+Born2d
+Back2d
+Gmodel
+new()
+setup_back()
+write_job()
+read_job()
+get_Gborn()
+get_Gback()
+get_Gmodel()
<<instantiate>>
<<instantiate>>
<<instantiate>>

set of pqr-files given in the input following the tag `QM_corr`. The method `write_job` calls the method of the `site` class and extends it with calls to the helper program My_3Diel_Solver for each pqr-file specified using the `QM_corr` tag. The method `read_job` similarly calls the method of the `site` class and extends it by reading all the output files generated by the additional calculations and by calculating the correction energy. The method `print_G` implements eq. 3.32 and eq. 3.34 to calculate the absolute intrinsic energy.

The MMsite class

Objects of the `MMsite` class represent instances of a site, for which the relative intrinsic energy should be calculated. The constructor checks for the rotamer energy (`Gmm`). Section 3.4 and section 4.3.3 distinguished four different types of instances, which are all described by this single class. The distinction between the four types is done according to the value of the `ref` key in the QMPB input file as discussed before. In the program, the instance labels are replaced by references (pointers) to the appropriate objects. During the calculation of the intrinsic energy by the method `print_G`, the references are used for calling the accessor methods of the reference site. Also the calculation scheme depends on the type of reference and sometimes on the type of the reference of the reference object. If the `MMsite` object represents a reference instance with model compound, it (mostly) manages its associated `Modelsite` object by itself after construction by the `site_master` object. Therefore some methods are extended compared to the `site` class.

The Modelsite class

Objects of the `Modelsite` class represent instances (in particular charge forms) of model compounds used by `MMsite` objects to calculate energies based on experimental model energies (`Gmodel`) and heterogeneous transfer energies. The electrostatic energy of a model compounds is calculated in a two dielectric environment using the helper program My_2Diel_Solver, not My_3Diel_Solver as it is used for `QMsite` and `MMsite` objects. Therefore the method `write_job` is overwritten. The background pqr-file as well as the dielectric boundaries are particular for each `Modelsite`. Due to the different program and the background energy term the method `read_job` is also overwritten. The method `print_G` is not implemented, since `Modelsite` objects do not contribute directly to the output files. Instead, `print_G` of the associated `MMsite`

Figure 4.13. Class diagram of QMPB including attributes and methods. The `site_master` class (cyan) is a container class, which manages objects derived from the `site` class ($1-*$ association with a `site` class, instantiate dependency with derived classes). The `site` class is an abstract base class, which contains the common attributes and methods of sites (generalization). `QMsite`, `MMsite` and `Modelsite` classes are derived from the `site` class. The `QMsite` class has some attributes specific for instances of quantum chemically parameterized sites, *e.g.,* `Hqm` stores $H_{\mathrm{vac},i}(j_k)$ (section 3.3). The intrinsic energy is computed in `print_G` according to eq. 3.32. Instead, the `MMsite` class implements `print_G` to compute eq. 3.37. Dependent, if ligands are bound or if it is a non-ligand binding site, the `MMsite` may use a `Modelsite` object to represent the instance of the site in the model compound environment ($1-0\ldots1$ association). Both, the `site_master` and the `site` class use the `pqr` class as helper class to represent pqr-files (use dependency). The figures in this section are color coded (main program - green, `site_master` - cyan, `site` - yellow, `job.sh` - pink).

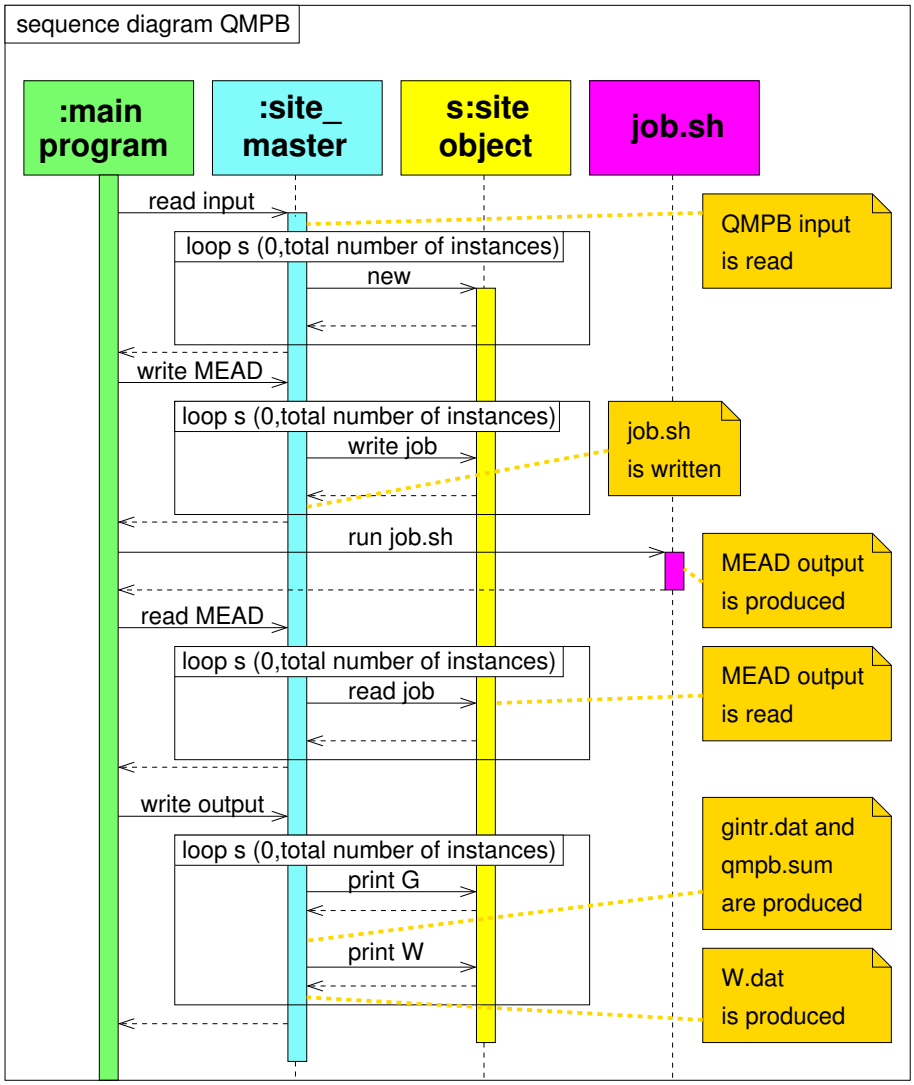


Figure 4.14. Sequence diagram of a QMPB run. The main program is slightly different here, since it runs in a single step the pre-processor part, the `job.sh` script and the post-processor part. Unlike the version discussed in the rest of the work, it is slightly simpler, but does not allow parallel execution of the `job.sh` script. The order of the method calls of the `site_master` object as well as the calls to the polymorphic methods of all the `QMsite` and `MMsite` objects are specified. The steps were input and output files are produced and processed are specified. The figures in this section are color coded (main program - green, `site_master` - cyan, `site` - yellow, `job.sh` - pink).

object accesses the energies *via* provided accessor methods and includes them into the calculation.

The pqr class

Objects of the `pqr` class are an object-oriented representation of a pqr-file. The objects are usually constructed by reading a pqr-file. The class provides a number of methods to access the data, for example `get_charge` returns the total charge of the pqr-file, `get_coord` returns all coordinates as (x, y, z)-triples or `coord_exists` checks if an atom with given coordinates is present in the pqr-file. The data can be manipulated (*e.g.,* `delete_atom` deletes a specified atom or `set_charge` sets the charge of a specified atom to a given value). Finally, the manipulated data can be written to a pqr-file again.

4.4 Extensions to the MEAD Library and Suite of Programs

Donald Bashford developed an object-oriented library for Molecular Electrostatics at Atomic Detail (MEAD [30]), *i.e.,* a Poisson-Boltzmann solver suited for biological applications. He provides a number of programs based on this library, *i.e.,* Solvate and Solinprot to calculate homogeneous transfer energies (section 3.2.5) into a two dielectric environment or a three dielectric environment, respectively, and Multiflex to do relative $\mathrm{p}K_a$ calculations (involving calculations of heterogeneous transfer energies, section 3.2.6). The theory underlying Multiflex in comparison to QMPB is discussed in section 3.5. For QMPB the available programs were not suited. Therefore, the programs Pqr2SolvAccVol, My_3Diel_Solver and My_2Diel_Solver were developed as helper programs for QMPB (section 4.4.1, section 4.4.2 and section 4.4.3, respectively). Calls of these programs with proper command line parameters and input files are written to the file `job.sh` in the pre-processor run of QMPB. The output files of the helper programs are parsed in the post-processor run of QMPB to obtain the electrostatic energies. Besides, these additions to MEAD a number of further extensions were made, which are summarized in section 4.4.5.

4.4.1 Dielectric Boundary Calculations with Pqr2SolvAccVol

To define the boundaries between dielectric regions, solvent accessible surfaces have to be calculated. This process is rather slow for complicated molecules with many atoms. MEAD implements an algorithm to calculate analytical surface representations [70]. From the analytical surface, a grid representation can be calculated according to the parameters given for the grid. The analytical representation is independent of grid parameters, so that the same analytical surface can be used to discretize grids of different center, size and resolution for focussing steps or for different sites. This discretization is very fast compared to the analytical surface calculation.

For each conformation in QMPB, all instances of all sites have to be calculated with identical dielectric boundaries. My_3Diel_Solver is run independently for each instance. If the analytical representation of the surface would be calculated at each run, it would noticeably contribute to the overall runtime. Therefore, Pqr2SolvAccVol was written, which takes the coordinates

of all atoms belonging to a dielectric region (in form of a pqr-file), calculates the analytical representation of the solvent accessible volume and writes it to disk. Runs of My_3Diel_Solver check if a file is present containing the analytical surface representation and read it instead of calculating the surface. The `SolvAccVol` class of MEAD provides methods to read and write the analytical surface in plain text or binary format, but so far no program was using these methods.

It is important for the discretization step, that the surfaces are calculated properly. If the coordinates used to define the analytical volume contain cavities, which are large enough for a water molecule, MEAD assigns the grid points in the volume of the cavity the dielectric constant of the solvent. This feature is required for water filled cavities in proteins. However, when molecules are described by more than two dielectric regions this feature can lead to mistakes. For example, having a `QMsite` defining a dielectric region, the protein region and the aqueous solution, one may calculate the region of the `QMsite` by the atoms belonging to instances of this site and the protein region by all other atoms of the protein. This would lead to a cavity in the protein volume, where the `QMsite` is located. Since the solvent accessible surface algorithm calculates in one case an outer surface of the `QMsite`, but in the other case an inner surface according to the protein atoms, the two surfaces are not identical. When combining the surfaces to discretize the dielectric constants on a grid, there may be grid points which are neither in the volume of the region of the `QMsite` nor in the volume of the protein. Therefore, MEAD will assign the dielectric constant of the solvent, nevertheless, the volume between the two surfaces is too small to fit a water molecule. This error significantly contributes to the calculated electrostatic energies. However, the correct way to perform such calculations requires to give the dielectric regions a hierarchy. The highest order dielectric region is calculated from the atoms in this region. The lower dielectric region include all atoms in its dielectric region, but also all atoms in higher order dielectric regions. QMPB, which is currently only able to deal with three dielectric regions including the solvent, takes the region with dielectric constant of `epsin1` as higher order region. The region with dielectric constant of `epsin2` is the lower order region, which gets all atoms of the higher order region also included for boundary definition.

Another important issue concerning boundary definitions is, that all instances of a site have to be in one dielectric region with constant boundaries. If the boundaries would change, the interaction with all instances of all other sites would need to be recalculated. Instead of the linear scaling with number of instances, one would obtain an exponential scaling behavior. This requirement leads to boundaries, which cover the site with the ligand bound in all possible ligand binding positions. For sites with rotamer instances, the boundaries also cover all rotamer forms. For sites, which are buried in a region of the protein with homogeneous dielectric constant, this approximation has no side effect. However, for sites next to the surface, an error is made by increasing the volume of the protein region. For small ligands like protons or hydroxyl group rotamers (Fig. 4.16), this error is not serious. However, for large ligands or sidechain rotamers, it has to be checked carefully, if the approximation is still valid. If the approximation is not acceptable, the instances have to be treated as different conformers, leading to a computational effort, which can only be afforded for very few instances. Especially, it grows exponentially, when different sites are treated by conformers.

QMPB defines the dielectric regions by including all instances of each site into the pqr-file used to calculate the analytical surface representation with Pqr2SolvAccVol.

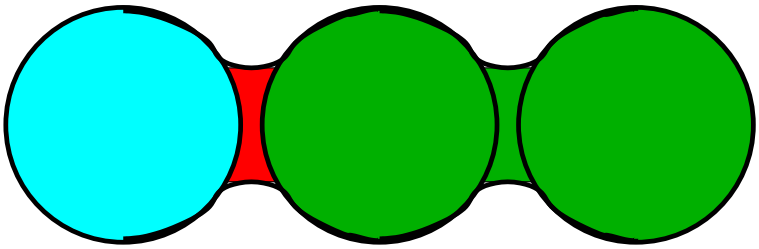


Figure 4.15. Simple example for the correct definition of dielectric boundaries. Two atoms (in green) have the same dielectric constant and a third atom (cyan) has a different dielectric constant. If MEAD calculates the dielectric regions separately, one obtains the cyan and green volumes. However, calculating one dielectric region for the three atoms, would result in the volume including the regions marked in cyan, red and green. The red volume, which should belong to the molecule would be assigned with the dielectric constant of the solvent in the separate calculation. This leads to errors in the calculation. Therefore, dielectric regions have to follow a hierarchy, *i.e.,* the cyan is the highest order dielectric region including only the cyan atom and the lower order (green) dielectric region has to include all three atoms.

4.4.2 Electrostatic Energy Calculations with My_3Diel_Solver

For `QMsite` and `MMsite` objects, QMPB uses My_3Diel_Solver to solve the LPBE. Two dielectric regions for the solute can be specified by a pqr-file (`eps1set` and `eps2set`). If an analytical surface file generated by Pqr2SolvAccVol exists, it is read instead of calculating the analytical surface. The three dielectric constants are specified by `epsin1`, `epsin2` and `epsext`. The pqr-file containing the atoms of the instance and the background pqr-file are read to calculate the Born and background energy in the heterogeneous environment. If the option `epshomo` is given, additionally the Born energy in a homogeneous environment with given dielectric constant is calculated.

The option `fpt` specifies an extended fpt-file (appendix A.1.4). It contains the site and instance number, the cartesian coordinates and the charge of each atom belonging to an instance of a site. My_3Diel_Solver uses this information to calculate the interaction energy with the current instance by multiplying its potential with the charge. The energies are given in the output file *per* instance summing up the energies of all atoms belonging to the same instance. The resulting interaction energies are tabulated as site and instance number followed by the interaction energy.

4.4.3 Electrostatic Energy Calculations with My_2Diel_Solver

For `Modelsite` objects, QMPB uses My_2Diel_Solver to solve the LPBE. One dielectric region for the solute (with dielectric constant `epsin`) and one dielectric region for the solvent (with dielectric constant `epsext`) are present. The dielectric boundaries are defined by the background pqr-file, while the instance pqr-file is identical to the file used for calculating the electrostatic energy in the three dielectric environment by My_3Diel_Solver.

Unlike for My_3Diel_Solver pre-calculating dielectric boundaries is not helpfull, since the boundaries are different for each site. Since the model compound only contains the isolated site, no

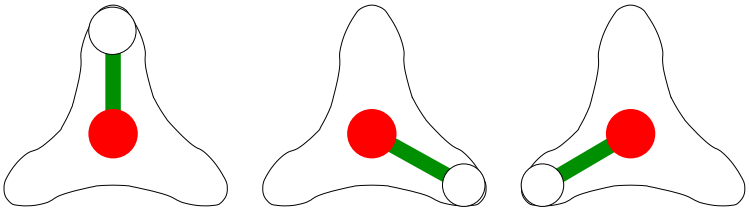


Figure 4.16. Boundary definitions for rotamers. Here, a hydroxyl group is shown (oxygen red, hydrogen white) with its dielectric boundary. The dielectric boundary is constant for all three rotamer instances. For clarity this figure exaggerates the error made by uniting the dielectric boundary compared to a single rotamer. In fact, the atom radius of oxygen is taken to be 1.5 Å, while the bond length is only 1.0 Å and the radius of hydrogen is 1.2 Å. Therefore, the surface is rather spherical with a bulge at the hydrogen position. Additionally, the hydroxyl groups are often not as isolated as shown here, but are small protrusions in the protein surface.

interaction energy needs to be calculated *via* an fpt file. However, the background energy is calculated as interaction of the site with charges in the model compound, which do not belong to the site and may only be present in the model compound.

In Multiflex, the model compounds were generated automatically by including all atoms in the residue of the site and the atoms forming the peptide bond with the previous and next residue. However, this automatic generation depends on the atom names, which are different in different force fields. For sites, which are not amino acids, this pattern matching failed. QMPB instead requires to specify a pqr-file for the model compound in the input. Therefore, it is possible to define arbitrary model compounds as they were used to determine the binding energy.

4.4.4 A Programming Interface to the MEAD Library using SWIG

After the current version of QMPB was finished, Thomas Weinmaier and I developed a programming interface between MEAD and Perl or Python [73]. The interface was generated using the program SWIG (Simplified Wrapper and Interface Generator [140]), which largely automates the generation of the interface module from the header files of a C or C++ program. However, due to the complexity of MEAD and language specifics some adoptions were necessary. Unlike the rudimentary interface to Python which existed before in MEAD, our interface is complete in the sense, that all classes and methods are available.

To demonstrate the abilities of the interface, Solvate, Solinprot, My_3Diel_Solver and My_2-Diel_Solver were implemented in Perl using the interface to the C++ classes of MEAD. They gave identical results as the programs completely written in C++. No performance difference could be measured between accessing methods of the MEAD library from C++ programs or Perl programs.

Besides its great value in writing tools using the MEAD library in more convenient high-level languages as Perl or Python, it is planned that future versions of QMPB will use this interface to avoid writing the `job.sh` script and the necessary input files. Instead, a single call to QMPB would perform all necessary calculations without additional IO. Such a design avoids the

redundant reading of the input file and the parsing of the output files of the helper programs. However, the parallelization can no longer be separated into an additional helper program, but becomes an integral part of QMPB. Also the replacement of the Poisson-Boltzmann solver by another one (like APBS) is not as simple anymore, but requires a plug-in interface structure.

4.4.5 Extensions to the MEAD Library and Additional Programs

Extended PQR Format

Earlier versions of MEAD could not deal with a chain label column in pqr-files. However, it is very common for larger proteins to have more than one chain. This lead to identical hash keys in the internal data structures of MEAD. A work-around was to increment the residue number by thousand for every chain. For calculations within the generalized titration theory (section 3.1.2) it was found usefull to store also conformer-ids, site-ids and instance-ids. Therefore, the pqr-file format (appendix A.1.2) was extended by four columns (integer) to accommodate these values (appendix A.1.3). The MEAD internal atom hashes use these values for key generation. Due to its object oriented design, all MEAD programs profit from this modification.

Multiflex3D - Multiflex Using a Three Dielectric Environment

Before QMPB was written, Multiflex3D was developed as an extended version of Multiflex, which uses the `ThreeValueDielectricByAtoms` class instead of the `TwoValueDielectricByAtoms` class of MEAD. Multiflex is written for protonation energy calculations in a protein in solvent. It does not allow to have a third dielectric region (*e.g.*, vacuum) for a center calculated by QM. Multiflex3D allows such a dielectric region. Therefore, it is possible to treat centers, which are calculated absolute (`QMsite` objects in QMPB) together with sites, which are calculated relative. For the relative calculations still only two instances are possible. For the absolute calculation only a single instance can be treated at once, but multiple runs of Multiflex3D and Solinprot and manual calculations also allow treatment of multiple instances in a physically correct way.

My_NDiel_Solver - A LPBE Solver for an Arbitrary Number of Dielectric Regions

The programs My_2Diel_Solver and My_3Diel_Solver are the helper programs for QMPB for calculations in a two and three dielectric environment. Future versions of QMPB should be able to operate with an arbitrary number of dielectric environments. To enable MEAD to perform such calculations the classes `NDielByFile` and `NElectrolyteByFile` were added. My_NDiel_Solver is a program using these classes, which can be a general replacement for My_2Diel_Solver and My_3Diel_Solver but also perform calculations for systems with a higher number of dielectric regions.

Instead of specifying the dielectric regions by `eps1set` and `eps2set` and the dielectric constants by `epsin1` and `epsin2` as parameters, a file with the same name as the background pqr-file, but with the ending `.diel` is read. It contains the number of dielectric regions in the first line and then a pqr-file, the dielectric constant and the solvent probe radius in each

following line. The pqr-files are used to calculate the analytical surface representation using the given probe radius. If a file containing the analytical surface representation already exists (*e.g.,* from a Pqr2SolvAccVol run) this file is read (and the probe radius ignored). When mapping the analytical surface onto a grid, the specified dielectric constant is used. The dielectric regions are given with descending priority, *i.e.,* if a grid point is contained in two or more dielectric regions, the dielectric constant of the first is assigned.

The following example defines a protein with a region treated quantum chemically, surface atoms with a higher dielectric constant than the core of the protein (due to higher mobility, *i.e.,* side chain rotamers on the surface, not treated explicitly), water cavities with a lower dielectric constant than the bulk solvent, a low dielectric membrane core and high dielectric membrane head groups:

```
6
qm_region          1 0.5
surface_atoms     20 1.4
whole_protein      4 1.4
water_cavities    10 2.0
inner_membrane     2 0.5
whole_membrane    20 1.4
```

The ion exclusion layer (Stern layer) can also be specified in a flexible way. A file with the same name as the background pqr-file, but with the ending `.ely` is read as well. It contains the number of electrolyte environments as first line and then a line for each electrolyte environment (pqr-file) and the ion exclusion layer thickness.

The following example defines a membrane protein. For the protein an ion exclusion layer of 2 Å is used. The membrane head groups are permeable for ions, but the core is not. Here the boundaries of the ion exclusion layer and the membrane core coincide.

```
2
whole_protein 2.0
inner_membrane 0.0
```

Write_Eps- Visualization of Dielectric Regions

As discussed for Pqr2SolvAccVol, MEAD implements methods to calculate the dielectric regions of the protein. It turned out, that for more than one solute dielectric region, special care needs to be taken to define the dielectric boundaries properly. For detailed analysis, Write_Eps was written, which writes the surfaces as they are computed by Pqr2SolvAccVol into an OpenDX file. Reading routines for this file format are available in various 3D viewer programs. For example, VMD [141] allows to visualize OpenDX volumetric data together with molecules. Unlike other surface calculation routines available in VMD or other programs, using Write_Eps the exact surfaces are represented as they are used in the electrostatic calculations (Fig. 4.17).

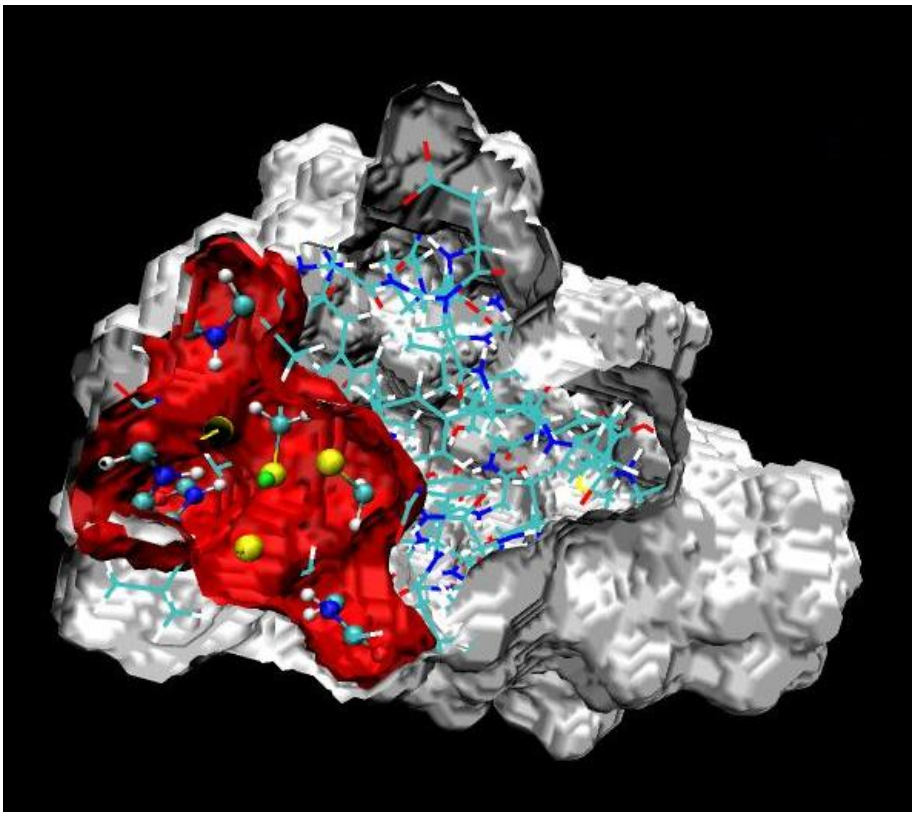


Figure 4.17. Dielectric regions in rubredoxin. The iron-sulfur cluster and some important residues in the surrounding are shown in ball-and-stick representation. These atoms were treated by QM calculations and assigned a dielectric constant of one in the electrostatic calculations. The surface of their dielectric region is shown in red. The rest of the protein was treated classically and was assigned a dielectric constant of four. The dielectric boundary to the solvent is shown in white.

4.5 Multiflex2qmpb - A Simple Generator for QMPB Input Files

The program Multiflex2qmpb converts the input files as they were needed to run Multiflex into an input file for QMPB. The major goal, when writing the program, was to ease repeating Multiflex calculations with QMPB. To replicate Multiflex calculations was thought to be useful for me to test the program QMPB and for Multiflex users to migrate their projects to the new program.

Multiflex2qmpb is essentially limited to the functionality of Multiflex only allowing for relative energy calculations using a model compound. The model compounds are defined by regular expressions as in Multiflex, even though they can be adopted to different force fields more easily. Multiflex2qmpb can read st-files (appendix A.2.1) in the same format as Multiflex containing a model $\mathrm{p}K_a$ value and charges for two instances.

Additionally, sites with more than two instances and different ligands are possible. Therefore, est-files (appendix A.2.2) were introduced, specifying model energies and charges of two or more instances. The est-files were useful to test, whether the "histidine titration problem"

(section 3.5.1) could be solved by QMPB. Multiflex required $2 + N$ runs for N histidine and an additional helper program to combine the separate files of intrinsic pK_a values (`*.pkint`) and interaction energies (`*.W`) into a single file of each type, which were suited for protonation probability calculations. QMPB instead allows a direct calculation of intrinsic energies and interaction energies of the three instances of any number of histidines in a single run. Analogously, other sites may contain any number of instances in QMPB adding only a computational cost proportional to the number of instances. With Multiflex treatment of such sites is not possible with an acceptable effort.

Since the beginning Multiflex2qmpb was only thought as an intermediate step until a program was available enabling the full power of QMPB. Now, the class library Perl Molecule can be used to easily write such programs (section 4.6). As a transitional solution, Multiflex2qmpb is kept rather simple. It takes the base of a filename, *e.g.*, `<molname>`, as only parameter and writes the QMPB input file to the standard output (appendix B.5). Analogously to Multiflex a `<molname>.pqr`, `<molname>.sites`, `<molname>.ogm` and `<molname>.mgm` are parsed and processed into several output files required for QMPB. The file formats are described in the appendix. The default behavior of Multiflex2qmpb can be influenced by modifying the header of the script:

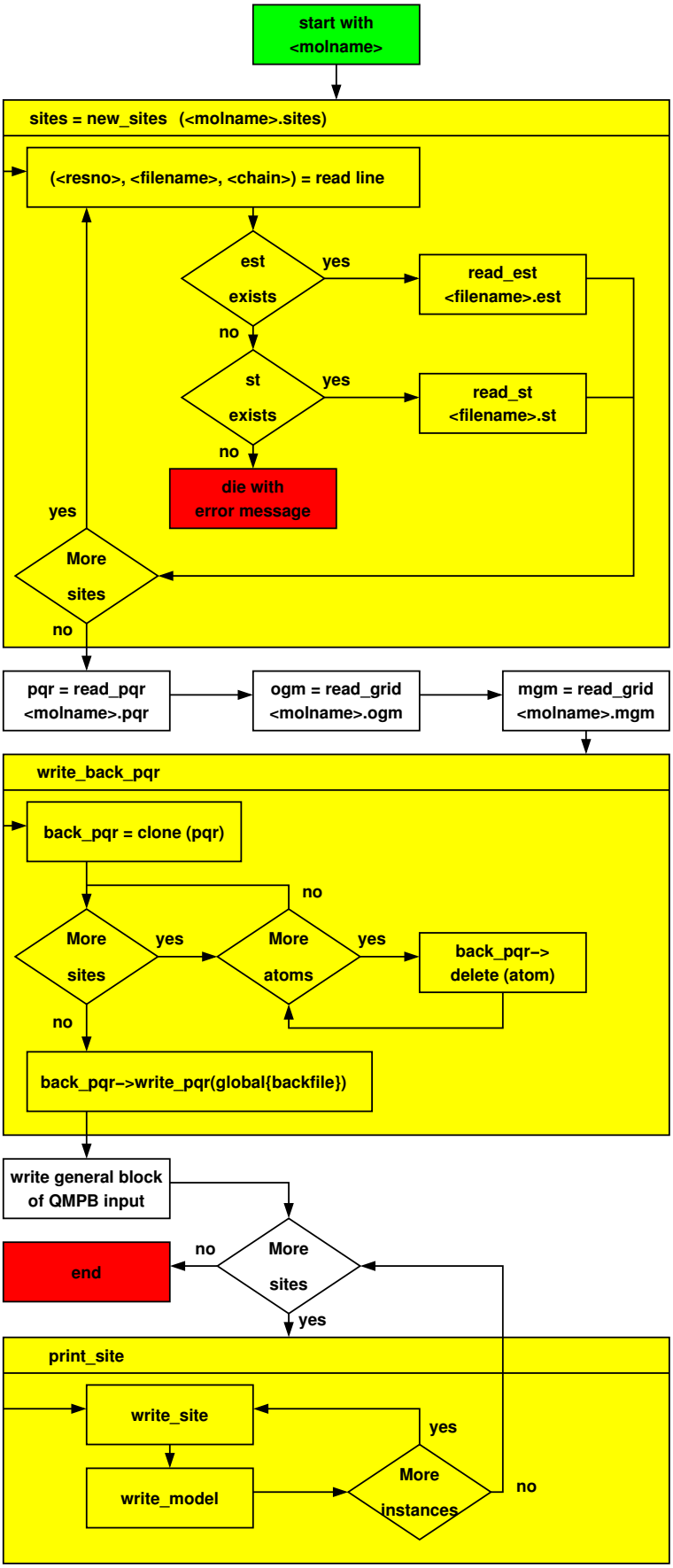
```
1  my %global = (
2                 T => "300",
3                 I => "0.1",
4                 backfile => "back.pqr",
5                 workdir => "qmpb",
6                 meadpath => "$ENV{'MEADPATH'}/bin",
7                 epsin1 => "1",
8                 epsin2 => "4"
9                );
10
11 # Atoms included in model compound of previous and next residue:
12 my @prev_res = qw(C O);
13 #my @next_res = qw(N HN CA); # CHARMM nomenclature, should include HA
14 #my @next_res = qw(N H CA); # for lysozym example
15 my @next_res = qw(N HN CA HA); # full CHARMM
16 my @next_PRO = qw(N CD HD1 HD2 CA HA);
17 my @next_GLY = qw(N HN CA HA1 HA2);
18 my @aminoacid = qw(ALA CYS ASP GLU PHE GLY HIS ILE LYS LEU MET ASN PRO GLN ARG SER THR VAL
19                    TRP TYR HSP HSE HSD HIE HID);
20 my $conv = - 1.371783713;
```

The hash `%global` contains parameters, which set key-value-pairs in the general block of the QMPB input file (section 4.3.3). Using Multiflex2qmpb sites must only contain atoms of a single residue (as in Multiflex). Model compounds are build by including all atoms belonging to the residue of the site and the atoms specified in `@prev_res` and `@next_res` of the previous and next residue (according to residue number), respectively. The definition can be adopted to the atom name nomenclature used in the `<molname>.pqr` file. In case of proline and glycine as next residues the CHARMM charge group includes a different set of atoms, which can be adjusted by the arrays `@next_PRO` and `@next_GLY`, respectively. For everything, which is not

an amino acid, *e.g.,* cofactors, the model compound is build by only including all atoms of the residue. No additional atoms are added to the model compound from the previous and next residue. Therefore, the array `@aminoacid` defines all residue names used for amino acids. The example contains a number of residue names for histidine which can occur dependent on the programs used for preparing the pqr-file. The variable `$conv` contains the factor used for converting from the model $\mathrm{p}K_a$ value found in st-files into the model energies required for the QMPB input file.

The programming of Multiflex2qmpb is rather straightforward. First, the `<molname>.sites` file is read by the function `read_sites`. The file has a column format with a residue number in the first column and the base name of a st-file or est-file in the second column. In a third column Multiflex2qmpb allows an optional chain label, which would be ignored by Multiflex. Using the base filename from the second column an est-file is searched and if that fails it is tried to find an st-file. Therefore, the new est-files are favored over st-files with the same name which remained in the project directory. If an est-file or st-file file is found it is read by the subroutines `read_est` or `read_st`, respectively. The function `read_sites` returns the contense of the st-files and est-files stored in a high dimensional hash together with the residue numbers of the sites and the chain labels. Analogously, the contense of the files `<molname>.pqr`, `<molname>.ogm` and `<molname>.mgm` are read into hashes by the functions `read_pqr` and `read_grid`. Next, in the procedure `write_back_pqr` a copy of the pqr-file data structure is made and all atoms present in a site are removed. The data is written into a pqr-file with a filename given as value to the `backfile` key of the `%global` hash. In the following of the main program, all key-value-pairs of the `%global` hash as well as the QMPB input parameters for specifying the grid dimensions and the `Ligand_Labels` are written to the standard output. The `Ligand_Labels` default to "`proton`" for st-files and must be given in est-files. Finally, for each site the procedure `print_site` is called. This procedure writes the QMPB input for each instance of the site to the standard output. It generates pqr-files for each instance and the model compound of each instance using the procedures `write_site` and `write_model`. The procedure `write_model` generates the model compounds according to the rules discussed above using the arrays `@prev_res`, `@next_res`, `@next_PRO`, `@next_GLY` and `@aminoacids`. The program flow is summarized in Fig. 4.18.

Figure 4.18. Conceptional flowchart of Multiflex2qmpb. The subroutines `new_sites`, `write_back` and `print_site` are shown in greater detail than the rest of the program to reflect the description in the text. The program reads the files `<molname>.pqr`, `<molname>.sites`, `<molname>.ogm` and `<molname>.mgm`. The file `<molname>.sites` contains a line for each site, specifying the residue number, a filename and optionally a chain label. The filenames correspond to est-file or st-file, which are read by Multiflex2qmpb. Then, the program writes a background pqr-file containing all atoms of the `<molname>.pqr` file not present in any site. The general block of the QMPB input file is written followed by a block for each instance of a `MMsite` and a `Modelsite` generated by the subroutines `write_site` and `write_model` called form the procedure `print_site`.


start with
<molname>
sites = new_sites (<molname>.sites)
(<resno>, <filename>, <chain>) = read line
est
exists
yes
read_est
<filename>.est
no
st
exists
yes
read_st
<filename>.st
no
die with
error message
yes
More
sites
no
pqr = read_pqr
<molname>.pqr
ogm = read_grid
<molname>.ogm
mgm = read_grid
<molname>.mgm
write_back_pqr
back_pqr = clone (pqr)
no
More
sites
yes
More
atoms
yes
back_pqr->
delete (atom)
no
back_pqr->write_pqr(global{backfile})
write general block
of QMPB input
end
no
More
sites
yes
print_site
write_site
yes
write_model
More
instances
no

4.6 Perl Molecule - A Class Library for Preparing Ligand Binding Studies

4.6.1 Introduction and Overview

Perl Molecule is a class library to convert pdb-files or pqr-files into QMPB input. It should provide all options needed to convert one or more pdb-files as they can be downloaded from the PDB database [127, 128] into calculation setups for QMPB. For crystal structures, hydrogen atoms need to be added. Some parts of the structure may be deleted, *i.e.,* water molecules or additives in the crystallization medium, but also maybe some domains of the protein. Some residues may need to be renamed, *e.g.,* the naming of cofactors often differs in crystal structures from different authors. Sometimes one may want to transfer some parts of one structure into another one, *e.g.,* there is a low resolution structure with a particular ligand bound and a high resolution structure without ligand or with another ligand bound. Therefore, the ligand of the low resolution structure might be copied into the high resolution structure for the calculations.

These pre-processing steps were done previously by shell scripts (using SED, AWK and GREP extensively) and specialized helper programs. A problem with this method is that these shell scripts are very error prone. Easily atoms are removed (or not removed), because the regular expression patterns were not exact enough (or did not match). These scripts are usually not transferable between different crystal structures, because small differences make some steps fail. Since the shell tools and most of the helper programs do not use any topological information on the molecules, but instead work on a *per* line (or *per* `ATOM` record) basis, certain operations are complicated and the programs can not sufficiently guide the user. The helper programs used in the shell scripts were often written with a particular application in mind. Therefore, they fail for pdb-files with “unusual” content. For example, I often had problems with crystal structures, where different occupancies for sidechains were given. Some programs ignored all occupancies except one, others did not work at all. Some programs even had problems with molecules consisting of more than one chain or simply the chain label column in the pdb-file. Additional pre- and post-processing was necessary to split the occupancies or chains into separate files, apply the programs to each of them and afterwards to combine the results. Since independent programs were responsible for the steps, often information had to be removed and later added again. Usually it was not practical, to automatically store and load this information, but it was removed once and the user had to take care to add it in a later step. CHARMM [96–99] allows many modeling tasks, but it requires a lot of pre-processing to convert a (non-trivial) pdb-file into a format, which is readable by the program. Unfortunately, many programs do not read or write pdb-file format properly, so that adjustments on the files were often necessary passing the information from one to the next program.

In most cases all these steps are not necessary anymore, if a rather simple program is written using my Perl Molecule library. It stores the molecular data in a detailed data structure representing the chemical view on the molecule. Each biomolecular complex can be in different conformers (globally different structures), consists of one or more chains, which in turn consist of residues. The residues are build up from atoms, which have coordinates and charges.

Due to this hierarchical structure (Fig. 4.19), operations like removing or renaming a residue or a chain need to act on one object only. If a chain is deleted, for example, automatically all residues in the chain and all atoms in each residue are deleted. Perl Molecule is aware of the topology of the molecule, *i.e.,* which atom is bonded to which other atoms. Since instances should always have an integer charge in the PBE model, the program keeps track of charge groups, *i.e.,* sets of atoms having in total an integer charge. Therefore, if an atom is part of an instance all other atoms belonging to the charge group are automatically included in the instance. If an instance will belong to a `QMsite` in QMPB, Perl Molecule can automatically select the atoms which are necessary for computing the correction energy (Fig. 3.7). For each atom Q1, which has bonded atoms not belonging to the `QMsite`, these bonded atoms M1 and the atoms bonded to these atoms M2 (*i.e.,* atoms in an angle relationship with the atom in the `QMsite`, which themselves do not belong to the `QMsite`) are selected. By the same method atoms Q2, which are bonded to the atom Q1 at the `QMsite` boundary and belong to the `QMsite` are searched. For these atoms, atoms M1 outside the `QMsite` in an angle relationship are searched and selected.

These examples show, that Perl Molecule exploits chemical and physical knowledge implemented in the program to assist the user setting up QMPB calculations. Perl Molecule aims to gain the knowledge necessary to make its decisions from existing data. For example, the topological information and the information about charge groups is required also for molecular mechanics packages like CHARMM. A considerable effort has been made to parameterize amino acid residues, DNA and RNA bases and common cofactors in so-called topology files. Therefore, Perl Molecule is able to read CHARMM topology files to exploit this information.

For specific tasks Perl Molecule can use external programs. For example, for adding hydrogens to the structure, Hwire (section 2.4.3) can be used. It is favored over similar programs, because it places hydrogens only based on geometrical criteria and not based on an energy function. Energy functions for hydrogen placement are generally different from the energy function of QMPB and would therefore lead to an inconsistent physical model. If different hydrogen rotamers are possible, for each a hydrogen atom is placed. Perl Molecule includes these atoms as rotamer instances. The rotamer energy (torsional and internal electrostatic energy) is taken from a potential energy file (appendix A.4.3), which was pre-computed by a molecular mechanics program like CHARMM. The current module interfacing Perl Molecule and Hwire is quite complex, because it has to split the structure and reconstruct it after running Hwire because the program can not treat occupancies (or rotamers in general) properly. Also some other shortcomings of Hwire were found, which could have been avoided in an own implementation (for example it is not possible to have different hydrogen bonding criteria *e.g.,* for hydrogen bonds involving oxygen and sulfur). Tabulation of potential energies as function of the torsion angle is only feasible for simple cases, *e.g.,* the single torsion angle of a hydroxyl group. For complex rotamers with multiple torsion angles, such tabulation is space and time consuming. It is easier to read the force field parameters and calculate the energy function inside a module of Perl Molecule, but this feature is not implemented yet.

For side chain rotamers, Perl Molecule can read the Dunbrack rotamer database (section 2.4.2 and appendix A.5.3) and convert the probability of the rotamer form into an estimate of its energy.

Charge forms can not only be specified by the st-file format of Multiflex (for backward compatibility), but also by the est-file format allowing multiple instances for a residue, the xst-file format, allowing sites consisting of multiple residues and in the fst-file format, allowing combinations of rotamer forms and charge forms for multiple residues. Charge set file formats are documented in appendix A.2. Except the st-file format, all file formats can be used for characterizing `QMsite` and `MMsite` objects in QMPB. The large number of different file formats is usefull to allow on one hand very general definitions of simple sites (like histidine), on the other hand very complex sites (like the Cu_B center in Cytochrome *c* oxidase, section 5.4).

Perl Molecule can write input files for QMPB exploiting all its features. Future versions of QMPB may use Perl Molecule objects directly making the intermediate step of writing input files and pqr-files optional. Perl Molecule might be extended to also become usefull as preprocessor for other programs like DMC or even CHARMM.

4.6.2 Example: Replacing Multiflex2qmpb by Perl Molecule

Since Perl Molecule is a class library and not a monolithic program, I decided to illustrate its use by two example applications.

Here, a simple program is shown, which reads a pqr-file, topology file and a sites file and writes a QMPB input. Therefore, it is very similar in function to Multiflex2qmpb, but it uses the Perl Molecule library. Features, like the topological information, instances described by `QMsite` objects in QMPB or the more advanced charge form file formats (xst-files and fst-files, additionally to st-files and est-files) can be used.

```
1  #!/usr/bin/perl -w
2  use strict;
3  use error_messages;
4  use molecule3;
5
6  error_messages::set_blab(2);
7  ################################
8  # DHC2 chain A
9  ################################
10 my $mol = molecule->add(undef, 'dhc2');
11 $mol->read_pqr('dhc2_mod.pqr');
12 ####################################
13 #
14 my %topology = (
15            method              => 'topology',
16            forcefield          => 'charmm',
17            file                => 'top_all27_prot_na_frank.inp',
18            patch               => [
19                                     'ACEX_28',
20                                     'CT3_97',
21                                   ]
22          );
23 #
24 $mol->setup_charge_groups(\%topology);
```

```
25 $mol->setup_graph(\%topology);
26 # assign additional charges from .st/.est files
27 #
28 $mol->setup_charge_sites ('dhc2_a.sites');
29 $mol->setup_sites;
30 # qmpb general options:
31 my %qmpb = (
32              meadpath => '$MEADPATH/bin',
33              T => $mol->get_T,
34              I => 0.1,
35              backfile => 'background.pqr',
36              OGMpoints => [61, 61, 131, 131],
37              OGMspace => [4.0, 1.0, 0.5, 0.2],
38              MGMpoints => 131,
39              MGMspace => 0.2,
40              epsin1 => 1,
41              epsin2 => 4,
42             );
43 # write qmpb input
44 $mol->write_qmpb (\%qmpb);
```

The first two lines of the Perl program turn on warnings and make the syntax checking more strict. These two lines are generally adviceable in Perl programs. Then in line 3, the Perl Molecule module `error_messages` is loaded, which is dealing with error messages. It is used in line 6 by calling the procedure `set_blab`. This procedure currently sets the verbosity of the output of Perl Molecule in a range of 0 to 5. The value of two is an intermediate level, at which usefull information and warnings are given as it is usefull for production runs. Higher levels are primarily for debugging Perl Molecule based programs. Line 4 loads the module containing the `molecule` class. The `molecule` class contains all chemical objects in a hierarchical structure, which will be discussed in detail in section 4.6.3. For the current program, all methods belong to this class. For more advanced programs, methods of the other classes need to be accessed directly, as it will be shown in section 4.6.4. In line 10 an empty object of the `molecule` class is constructed. The first parameter of the constructor `add` is a reference to the superior object. In this case, the `molecule` object is the root node of the hierarchy and the parameter is set to `undef`. The second parameter specifies a label for the object, in this case the molecule name. In this example, the molecule is Diheme cytochrome *c*2 (DHC2), which was studied by Frank Dickert [142]. However, analogous programs are used by other group members for much more complex proteins like Cytochrome *c* oxidase (section 5.4) or the bacterial reaction center. Line 11 reads the pqr-file into the Perl Molecule data structure. In lines 24 and 25 the charge groups and topological information (in form of a graph) are constructed from the information provided in a topology file (methods `setup_charge_groups` and `setup_graph`, respectively). Lines 14 to 22 serve to define a hash, which is given as parameter to the methods. The key `method` defines by the value `topology` that the information is given in form of a topology file. Then, the key `forcefield` specifies with the value `charmm`, that a CHARMM-type topology file should be read. If reader modules would be available, one could for example specify AMBER [143] or GROMACS [144] to use the topology files of other force fields. Next, the key `file` specifies the topology file to read. The key `patch` allows to specify patches, especially for the N-

and C-terminus of each chain, for additional atoms and bonding topology. The patch option is not yet as advanced as in CHARMM, where patches can include multiple residues. Currently, such complex patches are also not required, because in CHARMM they are mostly used to assign non-standard charges to groups of atoms. In Perl Molecule, non-standard charges can for example be assigned by xst-files (maybe with only one instance). Incompleteness in the bonding topology due to missing patches was so far not found to be problematic since only static structures were used. Line 28 calls the the method `setup_charge_sites`, which gets a sites file as parameter (section A.4.2). For each st-file or est-file specified therein, a charge set is created with as many charge forms as specified in the file. The more advanced charge set file formats (xst-files and fst-files) have to be specified in a form similar as it is done in the example in section 4.6.4. `setup_sites` in line 29 creates objects internally, which represent instances and sites. In this example it is a rather simple process, in which each charge set is described by a site and each charge form by an instance. In general, it is more complex, when including coordinate sets and rotamer forms. Then, all permutations between charge forms and rotamer forms must be generated. Also the size of the site must be increased, if the atoms of the coordinate set and the charge set only partially overlap, so that the site contains all atoms of both sets and possibly additional atoms to complete charge groups. Finally, in line 44 `write_qmpb` writes the QMPB input file. All information necessary for `QMsite` and `MMsite` blocks is given in the charge set files. `Modelsite` blocks are generated as required. The model compounds are not generated as in Multiflex or Multiflex2qmpb by pattern matching of atom names, but using the topological and charge group information known by Perl Molecule. Currently, the rule is that a model compound is the residue containing the site (or all residues belonging to the site) and the charge groups of atoms, which are bound to atoms of the residues.

For relative binding energy calculations, four cases were distinguished in section 3.4 and section 4.3.3, if there are rotamer forms. Perl Molecule can generate the input appropriately, because it knows, if there are multiple charge forms (*i.e.,* if it is a ligand binding or non-ligand binding site). The reference rotamer form is selected to be the rotamer form with lowest torsion energy. Therefore, Perl Molecule also can differentiate reference and non-reference rotamer forms. The choice of the reference rotamer form to be the one with lowest torsion energy is reasonable, since usually the rotamer form is not determined together with the binding energy. Instead the binding energy is measured for an ensemble of structures of the model compound. It can be assumed, that the rotamer with lowest energy is populated most and therefore contributes most to the measured binding energy.

The QMPB input also has a general block, where parameters for the helper programs are specified, *e.g.,* grid parameters and dielectric constants. The required parameters for this block are given as a Perl hash in lines 31 to 42 and passed to the method `write_qmpb` as parameter.

4.6.3 Class Hierarchy and Ontology of Perl Molecule

Currently, Perl Molecule consists of 34 classes and about 10000 lines of code. So it is significantly more complex than QMPB with 6 classes and about 2000 lines of code. This complexity naturally sets a limit to the detail at which the classes can be described here. I review the classes and discuss the most important methods a bit more in detail.

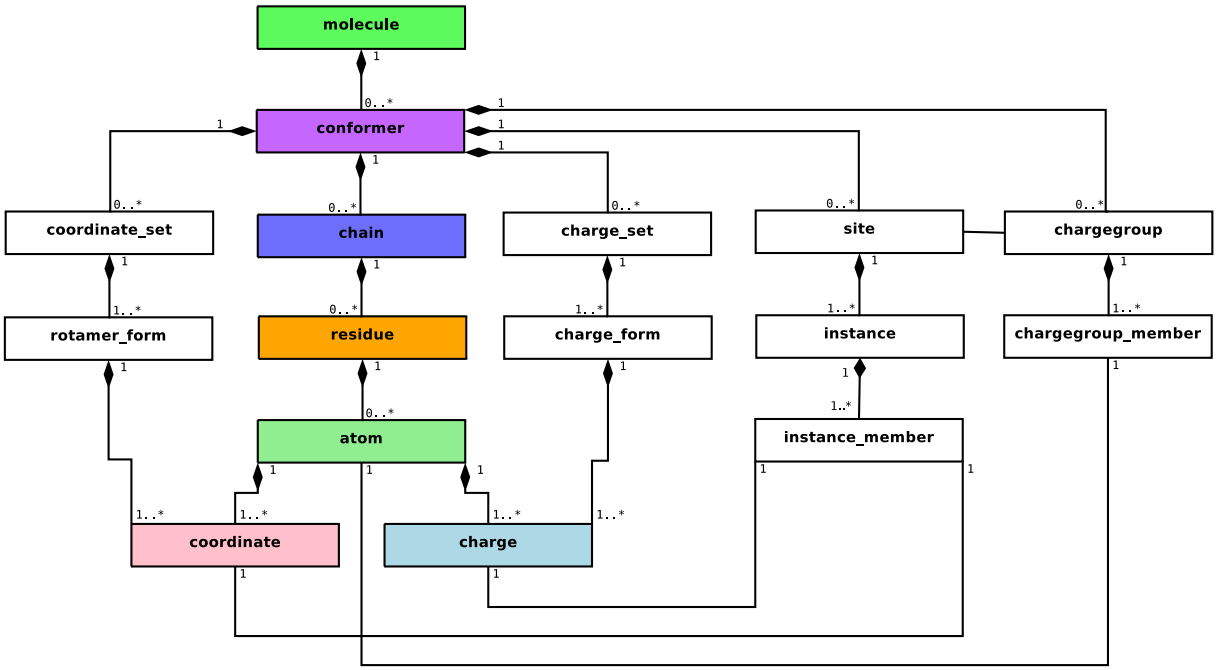


Figure 4.19. The composition hierarchy diagram shows the taxonomic dependence of classes in the ontology of Perl Molecule. The arrows with lozenge point towards the aggregation class and the other end to the components. The numbers give the order of association. The lines indicate important associations, which are no compositions. The classes are color coded in this section.

Perl Molecule implements an ontology[2], *i.e.*, a number of concepts are described by classes. Already in the introduction to this section it was pointed out, that biomolecular complexes (here just called molecules) can be seen in a hierarchical way. A molecule is described by a set of conformers, each conformer consists of chains, each chain consists of residues and each residue consists of atoms. Such a relationship of complex objects can be modeled by component objects, where each object is usually part of another object (composition). Many operations (methods) act with a cascaded semantic, *i.e.*, deleting a residue deletes also all atoms, which are part of the residue.

[2] An ontology is "A systematic arrangement of all of the important categories of objects or concepts which exist in some field of discourse, showing the relations between them. When complete, an ontology is a categorization of all of the concepts in some field of knowledge, including the objects and all of the properties, relations, and functions needed to define the objects and specify their actions. A simplified ontology may contain only a hierarchical classification (a taxonomy) showing the type subsumption relations between concepts in the field of discourse. An ontology may be visualized as an abstract graph with nodes and labeled arcs representing the objects and relations.
Note: The concepts included in an ontology and the hierarchical ordering will be to a certain extent arbitrary, depending upon the purpose for which the ontology is created. This arises from the fact that objects are of varying importance for different purposes, and different properties of objects may be chosen as the criteria by which objects are classified. In addition, different degrees of aggregation of concepts may be used, and distinctions of importance for one purpose may be of no concern for a different purpose." (From The Collaborative International Dictionary of English v.0.48)

The Container Class

To realize a composite pattern [135], a general abstract class `container` (Fig. 4.20) was written. It implements the methods to manage a set of `container` objects it stores. Each container object has one reference to the superior[3] `container` object it is part of. The methods `add` and `delete` call the constructor and destructor internally, but also call methods of the superior object to add or delete the current object. There are methods for moving the object from one container to another or to join the components of two containers into one. The members are generally referred to by a name (label), but unlike hashes in Perl, the `container` class keeps track of their order. By default, the order is the order in which the components were added, but there are also methods for sorting the components. There are various accessor methods for the components (*e.g.,* by label, by index in the current order or by index number specific for a particular object type). There is an `foreach` iterator method, which allows operations, *e.g.,* saying "foreach atom in chain A..." instead of having a method in the `chain` class, which iterates over all residues and all atoms in each residue.

The `container` class is general and can be used for many applications. For example it would make sense to derive the class `site_master` in QMPB from this class, because a major function of this class is to manage `site` objects. (This requires, that also the `site` class is derived from the `container` class, even it contains no further objects as it also is the case *e.g.,* for `coordinate` and `charge` classes in Perl Molecule).

The MMcontainer Class

From the `container` class the `MMcontainer` class is derived (Fig. 4.20), which adds some methods usefull to all molecular modeling child classes. For example, the method `printstatistics` is implemented here, which iterates over the object composition to gain information as how many atoms are in a residue and how many residues with how many atoms in total are in a chain *etc.* It also provides a general `write` method, which allows writing all atoms contained (indirectly) in an object to a file in various pdb or pqr formats.

The Molecule Class

The root of the ontological tree of Perl Molecule is the `molecule` class (Fig. 4.21). It describes a biomolecular complex by a number of discrete conformers. In Perl Molecule, it provides an interface to the objects it is build up from by cascading or delegating method calls. A number of methods were already described in the previous section. In the next section, some more advanced operations are shown, which require to call methods of other classes as well. From the 47 methods of this class only a few can be discussed briefly: The methods `read_pdb` and `read_pqr` aim to be very flexible reading routines for the two file formats. They should be able to handle many of the various versions of the file formats different programs produce. Internally, a hash of the data is generated from each `ATOM` or `HETATM` line and passed to the method `setup` (defined in the `MMcontainer` class), which generates appropriate objects at each level of the Perl Molecule ontological hierarchy. The methods `write_pdb` and `write_pqr`

[3] In the code, the instance variable `inferior` is used, looking from the leaves down to the roots of the tree. Consistently, the abstract method `superior_type` returns the classname of the inferior class for all derived classes. It seems that the biology oriented study left its traces...

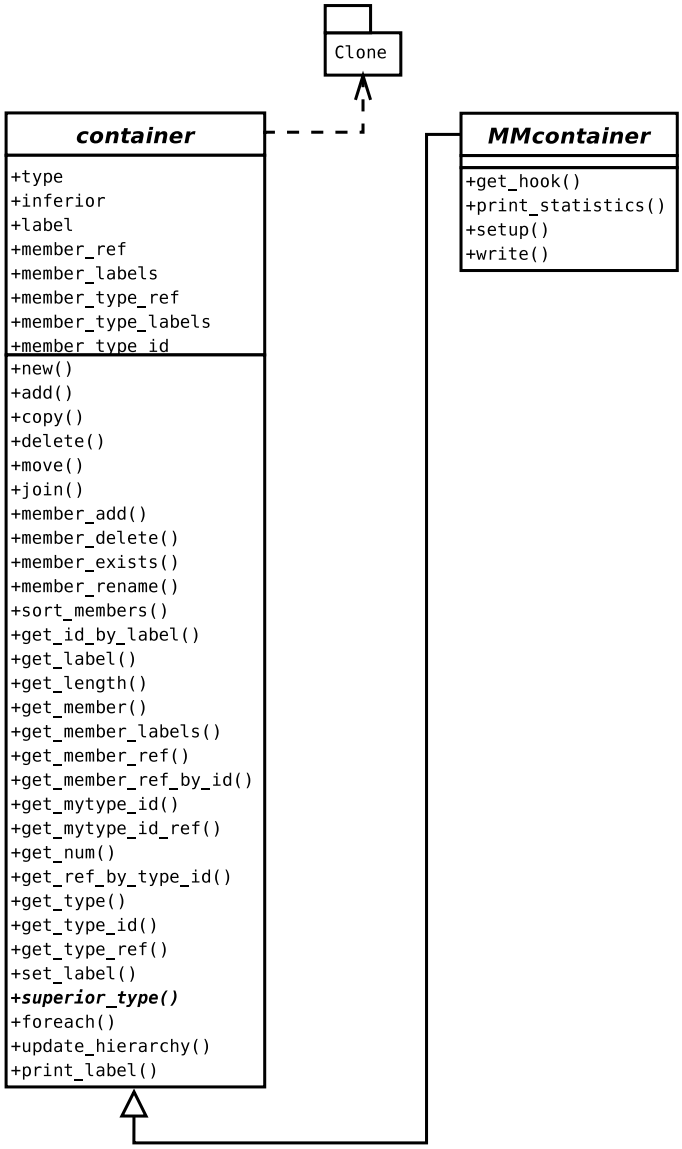


Figure 4.20. From the abstract `container` class most Perl Molecule classes are derived. The Perl module `Clone` is used by the copy method. The abstract `MMcontainer` class is derived from the `container` class.

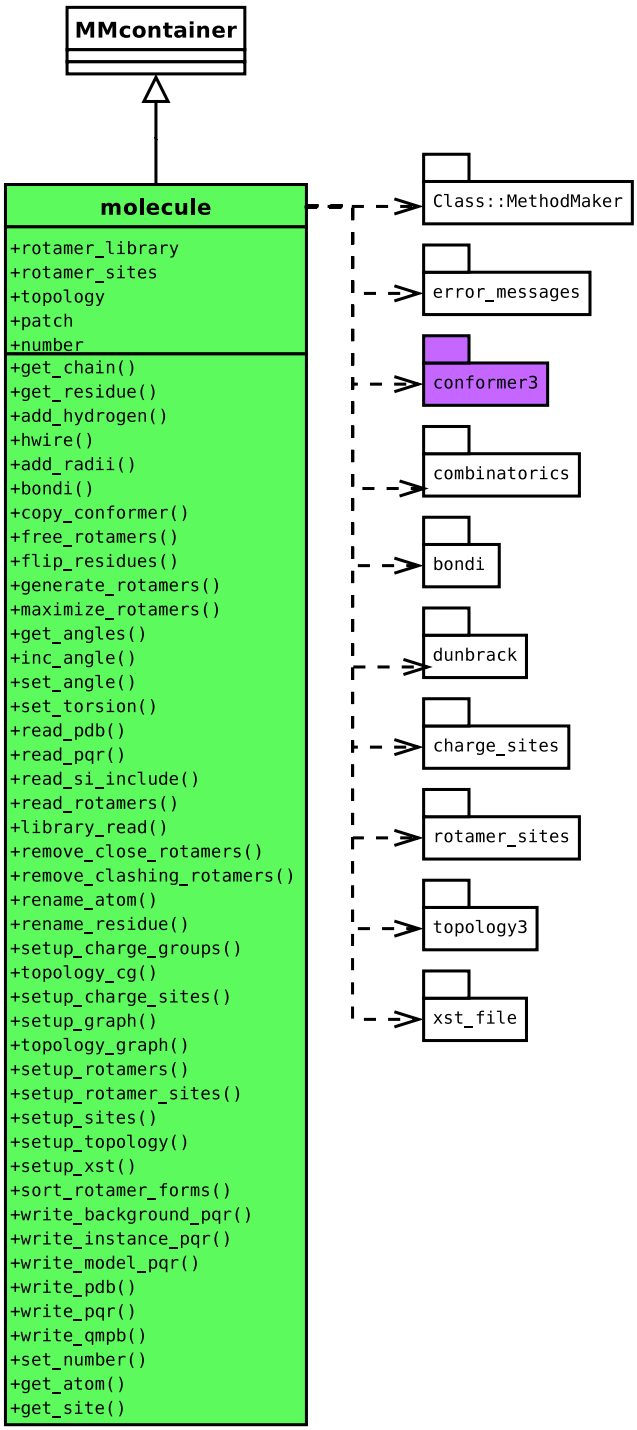


Figure 4.21. The `molecule` class is derived from the `MMcontainer` class. It contains objects of the `conformer` class. Classes which are imported are linked by an arrow originating at the class name.

call the method `write` for each atom with parameters appropriate for the two file formats. The method `add_hydrogen` includes the code for splitting the molecule into parts, which can be processed by Hwire and joining the parts after the run of the external program into a single molecular structure. The method `add_radii` constructs an object of the class `bondi` to read radii from file and then iterates over all atoms by calling their method `set_radius` with the method `get_element` of the `bondi` object as parameter. The method `read_rotamers` creates rotamer library objects of the `dunbrack` class. Using this class, the backbone and non-backbone dependent rotamer library of Dunbrack [101, 102] can be read and used to generate sidechain rotamers. The method `setup_topology` is called by the methods `setup_graph` and `setup_charge_groups` to construct an object from the `topology` class, which contains a method of the pattern `read_charmm` or `read_amber` for each supported force field topology format.

The Conformer Class

The `conformer` class describes a discrete global conformation of the molecule (Fig. 4.22). In the generalized ligand binding theory, all instances of all sites have to be recomputed for each conformer. In contrast, for rotamers only additional instances need to be calculated, which requires a significantly lower computational effort. However, due to the approximations made for rotamers, they are only valid for small local structural changes.

Each conformer can contain components of different classes. For the molecular description it contains one or more objects of the `chain` class. The ligand binding sites of the molecule are stored in objects of the `site` class in the `conformer` object. Coordinate and charge sets are stored in objects of the `coordinate_set` and `charge_set` class and charge groups are stored in objects of the `charge_group` class. Since a site, coordinate set, charge set or charge group can extend multiple chains, but is always associated with a particular conformer, the branching of the ontological graph at this point is reasonable. Most functionality of the `conformer` class is inherited from the `container` and `MMcontainer` classes. It provides some methods for cascading calls to components. The most important method maybe is `conformer::write_qmpb`[4], which is called by the method `molecule::write_qmpb`. It checks the elements of the `%qmpb` hash for completeness and writes them into the general section of a QMPB input file in a subdirectory with the conformer name. For each instance of each site `instance::write_qmpb` is called.

The Chain Class

The `chain` class represents a particular polymer chain in a certain conformation (Fig. 4.22). It only contains objects of the `residue` class. All its functionality is inherited from the `container` and `MMcontainer` classes.

The Residue Class

The `residue` class represents a particular residue in a polymer chain (Fig. 4.22). It only contains objects of the `atom` class. Most of its functionality is inherited from the `container`

[4] I will use the shorthand notation `class::method` to refer to a method of a particular class. If no class is specified, I refer to the class discussed in the current paragraph.

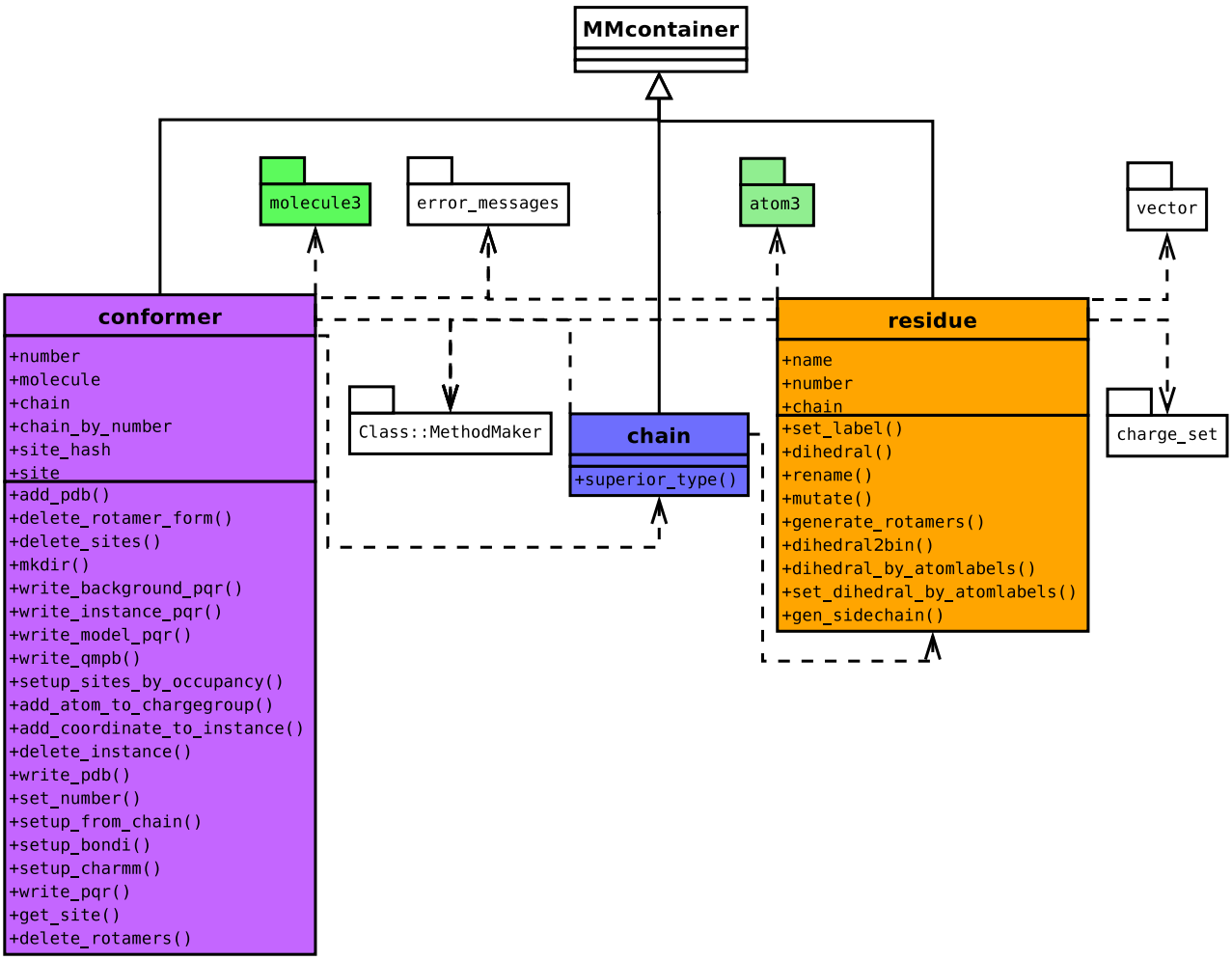


Figure 4.22. The `conformer`, `chain` and `residue` class are derived from the `MMcontainer` class. The three class contain objects of the `chain`, `residue` and `atom` class, respectively, and other classes. Classes which are imported are linked by an arrow originating at the class name.

and MMcontainer classes. The residue class has a number of methods dealing with dihedral angles and rotamers of the side chain: The method dihedral_by_atomlabels calculates the dihedral angle between four atoms of this residue. If the atom has multiple coordinates, only the first coordinate is taken into account. The label "-C" refers to the carbonyl carbon of the previous residue, which is resolved by the topology graph. The dihedral is computed by the method vector::get_torsion. The method gen_sidechain generates recursively a hash of sidechain atom names as keys and references to the objects as values. The method starts at a given atom and includes all atoms into the hash, which are bound to this atom unless they are either in a hash of excluded atoms or already in the hash of found sidechain atoms. For example, starting at the atom "CB" and excluding the atom "CA", the method would find the sidechain of each amino acid except glycine and proline based on topological information. The method set_dihedral_by_atomlabels changes the dihedral angle between two atoms given by their atom labels. The method takes a hash of sidechain atoms as generated by gen_sidechain and the target dihedral angle as further parameters. First, the rotation matrix is calculated by vector::get_rotation_matrix. Then for each atom in the hash of sidechain atoms the torsion matrix is applied using vector::my_change_tor_matrix. The method generate_rotamers is called by the method of same name of the molecule class. The method allows to generate all rotamers in a rotamer library using the methods discussed for this class.

The Atom Class

The atom class represents a particular atom in a residue (Fig. 4.23). It contains objects of the coordinate and charge class. The atom radius is assumed to be fixed for a given atom and not variable in different instances. Some of its functionality is inherited from the container and MMcontainer classes, but also a substantial amount of functionality is added by over thirty additional methods. The setup method is overwritten, because a number of instance variables are set, an object of the coordinate class generated for each coordinate of the atom and an object of the charge class generated (*via* the add_charge method) for each charge. The write method is called by the methods molecule::write_pdb and molecule::write_pqr as well as by many other methods which need to write parts of the structure in pdb or pqr format. The write method itself is mainly concerned about printing a line for each instance the atom is associated with. The actual formated printing is done by the method write_pdb. The method setup_bonded creates a hash of atoms bonded to this atom. This information is used by methods iterating through the topology of the molecule. The method setup_charge_groups, called by the method molecule::setup_charge_groups, constructs a new object, if it does not yet exist, of the chargegroup class as component of the conformer object to which the atom object belongs. It calls the method chargegroup::setup_members for the charge group defined in the topology object and adds the current atom by the method chargegroup::add_atom. The method atom::setup_sites is called by the method molecule::setup_sites to create objects of the site class from rotamer forms and charge forms. A new object of the site class is constructed, if the site of the atom is undefined, but a charge_set or coordinate_set is defined for the atom. Each atom contained in the charge_set or coordinate_set is added to the new site by the method site::add_atom. If an atom already belongs to another site, the two sites are joined to a larger site. Finally, the method site::setup is called. The setup_rotamers method adds the atom to a

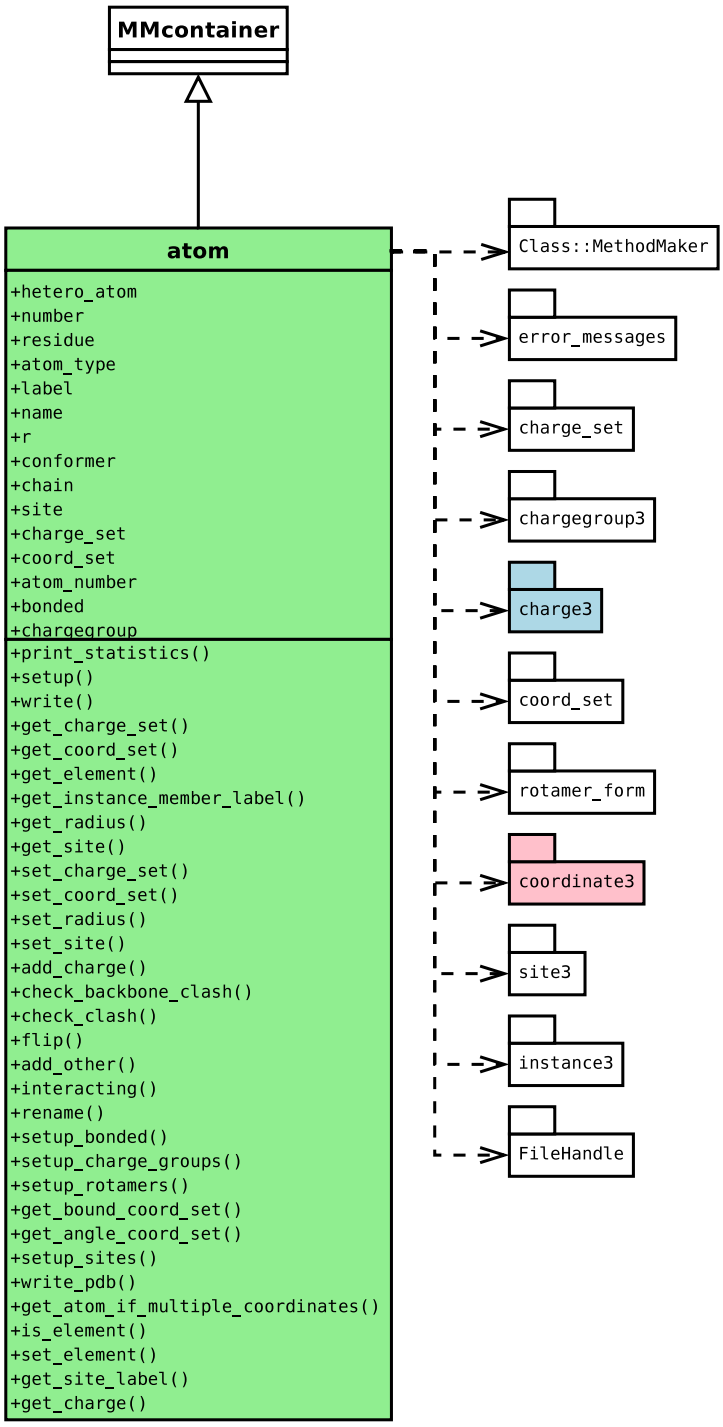


Figure 4.23. The `atom` class is derived from the `MMcontainer` class. It contains objects of the `coordinate` and `charge` class. Classes which are imported are linked by an arrow originating at the class name. An `atom` object uses the attributes `site`, `charge_set`, `coord_set` and `chargegroup` to store references to the objects of the respective classes to which it belongs.

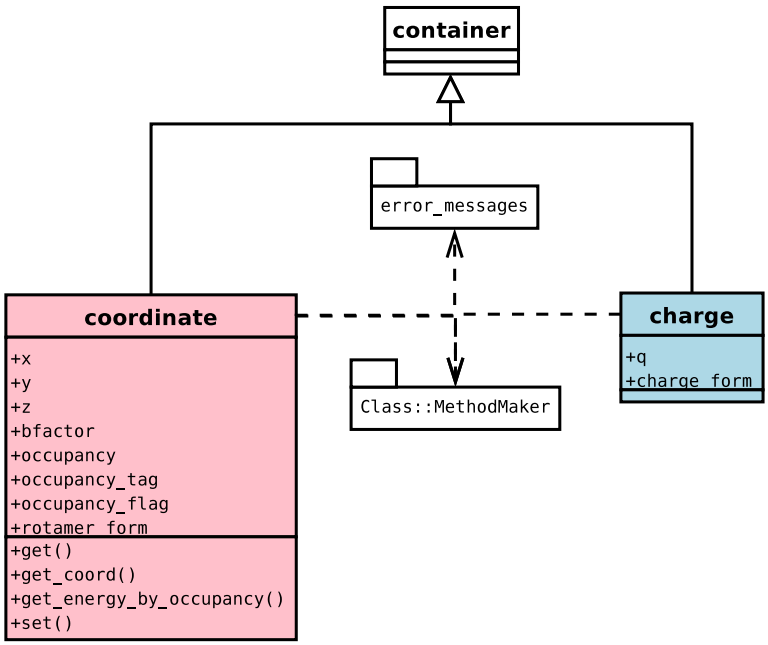


Figure 4.24. The `coordinate` and `charge` class are derived from the `container` class. Classes which are imported are linked by an arrow originating at the class name. The most important attributes are for the `coordinate` class the three cartesian coordinates `x`, `y` and `z` and for the `charge` class the charge `q`.

`coordinate_set` and its `coordinate` objects to a `rotamer_form` object. If a `atom` does not yet belong to a `coordinate_set` object, but has more than one object of the `coordinate` class as component, the method checks if any bound atom or atom bound to a bound atom (angle relationship) already belongs to a `coordinate_set`. If so, the `atom` is added by the `add_atom` method to the `coordinate_set`. If no `coordinate_set` is found, a new `coordinate_set` object is constructed as component of the `conformer` object to which the `atom` belongs. The `atom` is added *via* `add_atom` as before. For each `coordinate` object of the `atom` it is checked if a `rotamer_form` with same occupancy tag already exists for the `coordinate_set`. Otherwise it is created by the method `coordinate_set::add_form`. The `coordinate` is added by the method `rotamer_form::add_coordinate`. A rotamer energy is either computed from the probability of the occupancy as given for crystal structures or looked up from pre-calculated force-field energies according to a calculated torsion angle and assigned to the `rotamer_form` by the method `set_energy`.

The Coordinate Class

The `coordinate` class represents a particular coordinate of an `atom` (Fig. 4.24). Functionality is inherited from the `container` class, even it is not thought to contain any components. However, it uses the methods of the parent class for adding itself to and deleting itself from the superior `atom` object. The most important attributes are the three cartesian coordinates `x`, `y` and `z`, but also the `occupancy` and `bfactor` are stored as they were given in the pdb-file. The method `get_energy_by_occupancy` calculates an energy by inverting the equation of the

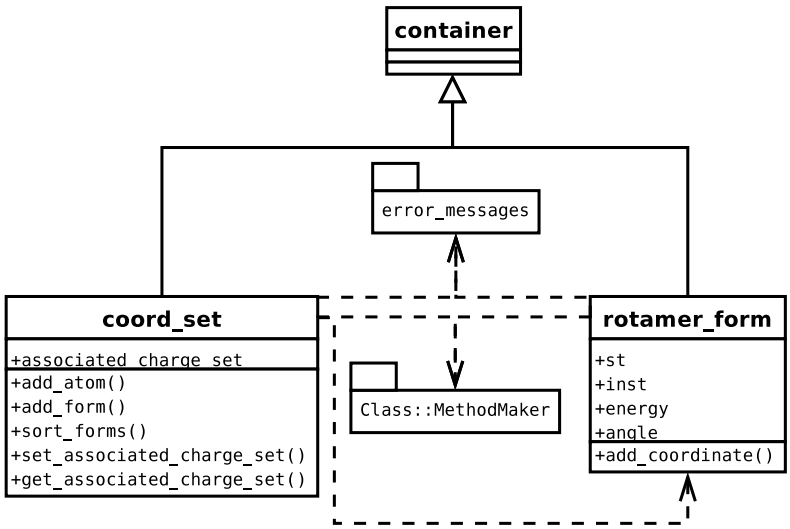


Figure 4.25. The `coordinate_set` and `rotamer_form` class are derived from the `container` class. Classes which are imported are linked by an arrow originating at the class name.

Boltzmann probability (see section 2.4.2). Most other methods are simple accessor methods, which can be generated automatically by the Perl module `Class::MethodMaker` as it is also done for attributes of other classes of Perl Molecule.

The Charge Class

The `charge` class represents a particular charge of an `atom` (Fig. 4.24). Functionality is inherited from the `container` class, even it is not thought to contain any components. However, it uses the methods of the parent class for adding itself to and deleting itself from the superior `atom` object. The most important attribute is the charge `q`, for which accessor methods are automatically generated by `Class::MethodMaker`.

The Coordinate Set Class

The `coordinate_set` class represents a particular coordinate set in a conformer (Fig. 4.25). It contains objects of the `rotamer_form` and `atom` class. Most of its functionality is inherited from the `container` class. The methods `add_atom` and `add_form` add components which are objects of the `atom` and `rotamer_form` class, respectively. The method `delete` removes `atom` components and also takes care, that the `coordinate` components of the `rotamer_form` objects are removed. Perl Molecule allows the definition of associations between charge sets and coordinate sets, *i.e.*, a certain rotamer form only occurs with a particular charge form. Usually, all permutations between charge forms and rotamer forms are generated. Such cases can be described by fst-files in Perl Molecule scripts. The `coordinate_set` class has the attribute `associated_charge_set` with appropriate accessor functions to refer to the associated charge set.

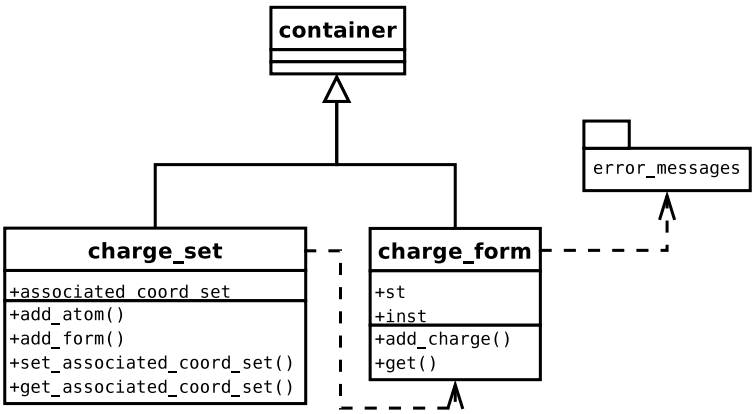


Figure 4.26. The `charge_set` and `charge_form` class are derived from the `container` class. Classes which are imported are linked by an arrow originating at the class name.

The Rotamer Form Class

The `rotamer_form` class represents a particular rotamer form in a coordinate set (Fig. 4.25). It contains objects of the `coordinate` class. Most of its functionality is inherited from the `container` class. The attribute `energy` contains the rotamer energy of the form. The method `add_coordinate` adds objects of the `coordinate` class as components.

The Charge Set Class

The `charge_set` class represents a particular charge set in a conformer (Fig. 4.26). It contains objects of the `charge_form` and `atom` class. Most of its functionality is inherited from the `container` class. Like the `coordinate_set` class, it has the `add_atom` and `add_form` methods to add its components. It has the attribute `associated_coord_set` and accessor functions to refer to the associated coordinate set.

The Charge Form Class

The `charge_form` class represents a particular charge form in a charge set (Fig. 4.26). It contains objects of the `charge` class. Most of its functionality is inherited from the `container` class. The method `setup` initializes the two attributes `st` and `inst`, which are a reference to a `st_file`, `est_file`, `xst_file` or `fst_file` object and an instance number, respectively. The method `get` is called with a accessor method name of the object stored in `st` and an instance number as parameters. It returns the result of the accessor method call for the instance `inst`. The method `add_charge` adds objects of the `charge` class as components.

The Chargegroup Class

The `chargegroup` class represents a particular charge group in a conformer (Fig. 4.27). It contains objects of the `chargegroup_member` class. Most of its functionality is inherited from the

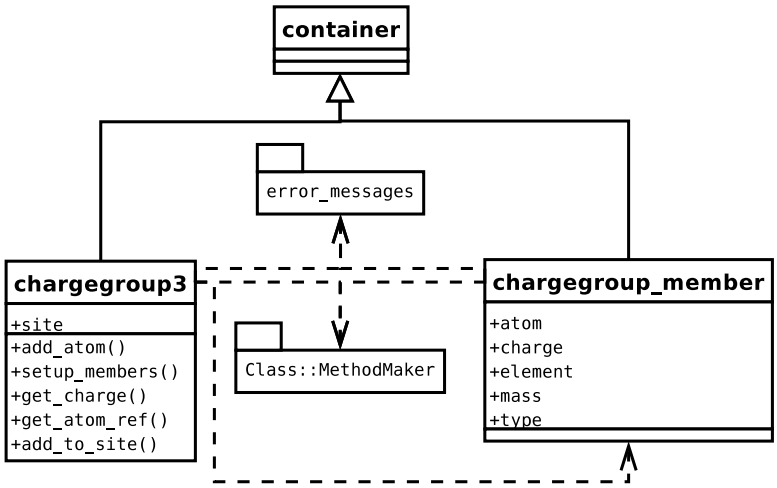


Figure 4.27. The `chargegroup` and `chargegroup_member` class are derived from the `container` class. Classes which are imported are linked by an arrow originating at the class name.

`container` class. The method `setup_members` is called by the method `atom::setup_chargegroups`. It constructs an object of the `chargegroup_member` class for each atom returned by the functions `topology::get_chargegroup` or `topology::get_patch_chargegroup`. The method also initializes the attributes of the new `chargegroup_member` object *via* the provided accessor methods. The method `add_atom` associates an `atom` object with this `chargegroup` object by setting the `chargegroup` variable of the `atom` object to point to this `chargegroup` object. The `chargegroup_member` object, which is the representation of an `atom` in a `chargegroup`, is linked with the `atom` object by calling the `set_atom` method. If the `chargegroup` belongs to a `site`, the `atom` is set to belong to the same `site`.

The Chargegroup Member Class

The `chargegroup_member` class represents a particular `atom` in a `chargegroup` (Fig. 4.27). Functionality is inherited from the `container` class, even it is not thought to contain any components. However, it uses the methods of the parent class for adding itself to and deleting itself from the superior `chargegroup` object. The most important attribute is `atom`, which is a reference to the `atom` object, which is associated with the `chargegroup_member` object. Additionally, each instance contains variables to store information contained about the atom in the topology file, *i.e.,* the `charge`, the `element`, the `mass` and `type`. The attribute `charge` is used to assign charges to atoms read from a pdb-file. The attribute `element` is used to look up atom radii based on the chemical element. The other attributes are currently not used. For attributes accessor methods are automatically generated by `Class::MethodMaker`.

The Site Class

The `site` class represents a particular site in a conformer. It contains objects of the `instance` and `atom` class (Fig. 4.28). Most of its functionality is inherited from the `container` and

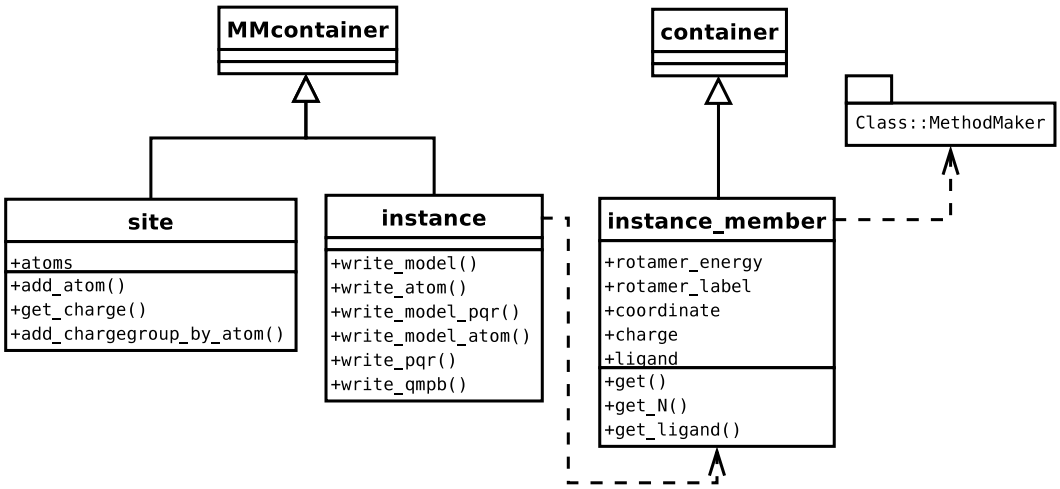


Figure 4.28. The site and instance class are derived from the MMcontainer class. The instance_member class is derived from the container class. Classes which are imported are linked by an arrow originating at the class name.

MMcontainer classes. The add_atom method adds the given atom to the site. The method recursively also adds all atoms, which are in the same charge group as the atom, making sure, that a site always has an integer charge. A site can be associated with objects of the coordinate_set and charge_set class to which the atom objects belongs. If all atom objects only belong to a single set, each instance object can be associated with a particular rotamer_form or charge_form object. In general, a site may be associated with a set of both types, because different atom objects belong to different charge_set and coordinate_set objects. Then, instance objects have to be generated as all permutations of rotamer_form and charge_form objects unless the coordinate_set and charge_set are associated (see above). Each permutation can be described by a unique vector in analogy to a state vector of a microstate. The length of the vector equals the sum of the number of coordinate_set and charge_set objects associated with the site. The maximum value at each position of the vector equals the number of rotamer_form or charge_form objects of the particular coordinate_set or charge_set object represented by the position. All permutations are generated in the method site::setup using an object of the combinatorics class as helper. For each vector an object of the instance class is created and for each atom component in the site object an instance_member object is created. The instance_member object stores a particular combination of coordinate and charge for a single atom. The instance object stores instance_member objects, one for each atom in the site.

The Instance Class

The instance class represents a particular instance in a site. It contains objects of the instance_member class (Fig. 4.28). By that, an instance represents a particular combination of coordinates and charges for a set of atoms. Most of its functionality is inherited from the container and MMcontainer classes. The methods write and write_model write a pqr-file for the particular instance of a site and for the associated model compound. A number of

helper methods are used, which finally call the method `atom::write_pdb` with appropriate parameters. The model compound includes all atoms of the site, all atoms of each residue, which has atoms belonging to the site, and the atoms of charge groups bonded to an atom either belonging to the site or to an included residue. The method `write_qmpb` writes the QMPB input file section for the instance.

The Instance Member Class

The `instance_member` class represents a particular combination of a `coordinate` and a `charge` for a single atom (Fig. 4.28). Functionality is inherited from the `container` class, even it is not thought to contain any components. However, it uses the methods of the parent class for adding itself to and deleting itself from the superior `instance` object. The most important attributes are a reference each to the associated `coordinate` and `charge` object, which are set in the `setup` method.

Other Classes

Additionally to the classes discussed above, there are other classes, which mostly are object oriented representation of files. Usually, a file of the appropriate type is parsed and the contense is provided to other objects by accessor methods. The classes `st_file`, `est_file`, `xst_file` and `fst_file` read the different charge set file formats (Fig. 4.29). The `topology` class reads topology files (currently only of CHARMM); the `potential` class reads files with stored rotamer energies based on torsion potentials; the `dunbrack` class reads the backbone dependent and backbone independent rotamer database of Dunbrack [101, 102]; the `bondi` class reads atom radii from a file (Fig. 4.30) and the `charge_sites` class reads a file containing the sites, for which different charge forms based on st-files (appendix A.2.1) and est-files (appendix A.2.2) should be applied.

Some classes encapsulate mathematical routines, *i.e.,* the `vector` class simplifies vector algebra, the `combinatorics` class contains functions to generate state vectors (especially the iterator discussed in section 4.2) and the `kabsch` class is a SWIG interface (section 4.4.4) to coordinate superimposition routines by Matthias Ullmann. The `kabsch` class is used in combination with fst-files, ensuring that the atoms bonded to atoms outside the atom set given in the fst-file are superimposing with minimal root-mean-square deviation. The associated coordinate and charge sets in fst-files are usually generated by QM geometry optimization and charge fitting, which may change the coordinates of the link atoms. It is common, that the files are transferred between different crystal structures, where the sites are at different coordinate positions.

A third group of classes contain information, which did not fit into any other class. The `PhysCond` class encapsulates variables and constants, which reflect the physical conditions of the calculation. Currently it only contains accessor functions for the absolute temperature and the gas constant. The module `error_messages` contains some procedures to adjust the verbosity of the output generated by Perl Molecule.

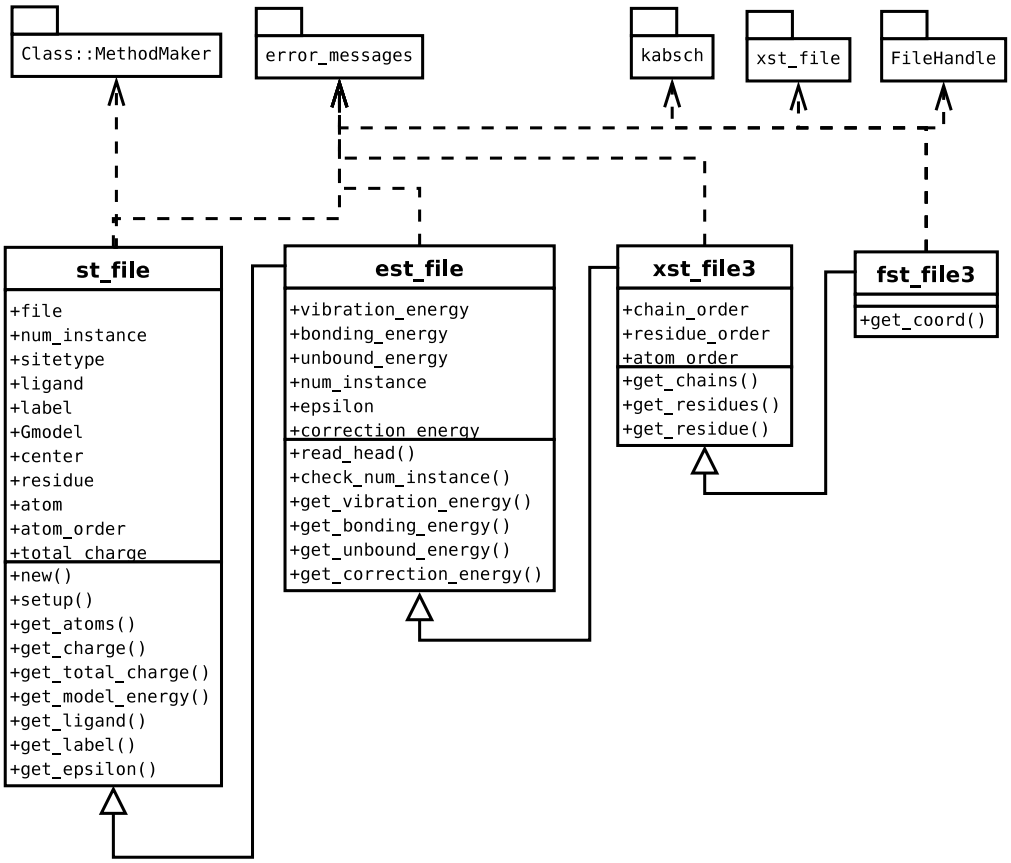


Figure 4.29. The classes reading different charge set file formats. The files are increasingly complex, so that each class can inherit functionality from the classes reading the less complex charge set files. Classes which are imported are linked by an arrow originating at the class name.

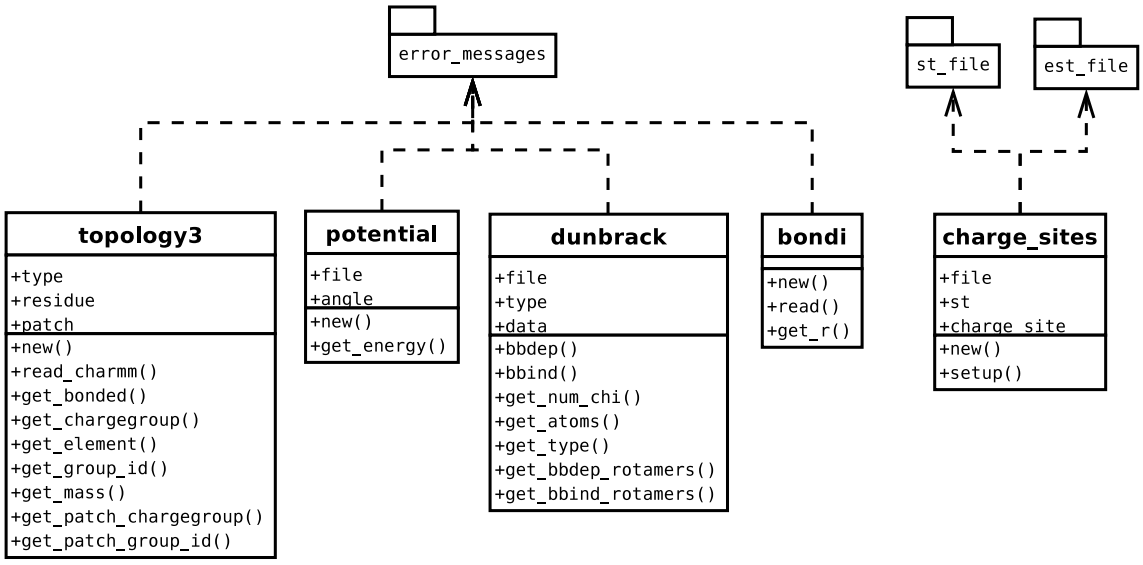


Figure 4.30. Classes reading different file formats and representing the contense. Classes which are imported are linked by an arrow originating at the class name.

4.6.4 Example: Modeling with Perl Molecule

In section 4.6.2 a simple example was given how to replace Multiflex2qmpb by Perl Molecule. In section 4.6.3 all the classes of Perl Molecule were discussed in a systematic way. Here, a more complex example is given, which starts with the crystal structure 1CZP [145] of *Anabaena* ferredoxin as it can be downloaded from the PDB database [127, 128]. The crystallographic unit cell contains two molecules (named chain A and chain B), which were refined independently. The [2Fe-2S] centers are given without chain label in residues FES 599 and 699. Since the crystal structure has a very high resolution of 1.17 Å, a number of atoms are refined with more than one occupancy. Most interestingly, a peptide bond next to the iron-sulfur center is seen in this (partially) reduced structure in two different orientations (called "CO-in" and "NH-in") forming an additional hydrogen bond in the NH-in orientation. Since this so-called "peptide flip" was not observed in the fully oxidized structure it was assumed, that NH-in structure is characteristic for the reduced form. Results of this study are reported in section 5.3.

In this script, the structural information contained in the pdb-file is divided into four conformers (chain A and B for the two molecules mol-1 and mol-2; occupancy A and B for the peptide orientation CO-in and NH-in, leading to the four structures "mol-1 CO-in", "mol-1 NH-in", "mol-2 CO-in" and "mol-2 NH-in"). The iron-sulfur center, the four ligating cysteins and the backbone atoms forming the first layer of hydrogen bonds with the iron-sulfur center were geometry optimized quantum chemically in the oxidized and two reduced forms (reduction of Fe1 and Fe2, respectively). Based on the QM calculations, charges were fitted to the nuclei of the atoms. In the calculation shown here, the charges are stored in xst-files. Alternatively, fst-files can be used analogously including the coordinate changes during the QM geometry optimization.

```
1  #!/usr/bin/perl -w
2  use strict;
3  use error_messages;
4  use molecule3;
5  use xst_file;
6
7  error_messages::set_blab(2);
8  ################################
9  # 1CZP
10 ################################
11 my $mol = molecule->add(undef, '1CZP');
12 $mol->read_pdb('1CZP.pdb');
```

The first lines are identical to the example in section 4.6.2. In line 5 the `xst_file` package is included since it will be needed later. Line 11 creates an empty object of the molecule class with the name 1CZP. In the next line it is filled by reading the pdb-file using the method `read_pdb`. At this point in the previous example the method `read_pqr` was used to read the pqr-file. Currently, the object structure of Perl Molecule contains one molecule with one conformer and three chains. The pdb-file contains the two molecules with chain labels A and B. The atoms of the iron-sulfur centers and all crystallographic water molecules have no chain label and are therefore stored in a third chain with label X. The next step is to move the iron-sulfur centers into their respective molecule and delete all remaining water molecules in chain X.

```
13 # move FES599 from chain X to chain A
14 my $res = $mol->get_residue('FES_599');
15 my $chain = $mol->get_chain('A');
16 $res->move($chain);
17 ## move FES699 from chain X to chain B
18 $res = $mol->get_residue('FES_699');
19 $chain = $mol->get_chain('B');
20 $res->move($chain);
21 # rename FES699 to FES599
22 $res->set_label('FES_599');
23 # delete chain X
24 $chain = $mol->get_chain('X');
25 $chain->delete;
```

Therefore, in line 14 and 18 the method `get_residue` is called for the residues `FES_599` and `FES_699`. It is one of the helper methods provided by the `molecule` class which does not require, that the user of Perl Molecule knows the conformer and chain labels in which the residues are stored or even the object tree structure underlying Perl Molecule. If the ontology is extended by adding or removing levels of the taxonomy, scripts using the method will still work. Since the `molecule` class does not contain any residues directly, it recurses through the object tree until it reaches objects of the `chain` class and checks if they contain a `residue` object with the given label. If none or more than one object with the given label is found the program stops with an error message. On success a reference to the residue is returned. The

method get_chain used in lines 15 and 19 is analogous to get_residue. It recurses until it finds a chain with the given label and returns a reference to it. In line 16 and 20 the method move of the residue objects are called with the reference of the destination chain as parameter. The method move was implemented in the container class and inherited to all derived classes. The residue object removes itself from its superior chain container class by calling member_delete and adds itself to the chain class given as parameter by the method member_add. If a member with same label exists in the destination container class (member_exists returns TRUE) the method returns FALSE, else it returns a reference to the object (in the shown script it is not checked for the return value). For consistency in line 22 the residue FES_699 of chain B is renamed into FES_599 as the iron-sulfur center is called in chain A. In line 24 a reference to chain X containing all the water molecules is obtained and in line 25 the chain is deleted. The method delete (also provided by the container class) removes the object by recursively deleting all objects it contains. In the next step, the molecule in chain with label B is moved into a separate conformer with label "mol-2 CO-in".

```
26 # move chain B into new conformer
27 my $conf = conformer->add($mol, 'mol-2 CO-in');
28 $chain = $mol->get_chain('B');
29 $chain->move($conf);
30 # rename chain B to chain A
31 $chain->set_label('A');
```

Therefore, in line 27 a new empty conformer object is created with the label "mol-2 CO-in" and the molecule as container class. In line 28 again the get_chain method is used to obtain a reference to the chain object with label B (mol-2) and move it into the new conformer (line 29). Here the same move method of the container class is used as it was used before to move residues. The only chain in conformer "mol-2 CO-in" should carry the label A, which is changed using the set_label method (line 31).

In the following a CHARMM topology file should be read and used to define charge groups and the molecular topology graph. The topology file uses the CTER patch, which calls the C-terminal carboxyl oxygens OT1 and OT2.

```
32 # set O and OXT to OT1 and OT2
33 $mol->rename_atom('A', 'TYR_98','O', 'OT1');
34 $mol->rename_atom('A', 'TYR_98','OXT', 'OT2');
35 # set all HIS to HSP
36 $mol->rename_residue('A', 'HIS', 'HSP');
```

Therefore, the atoms O and OXT are renamed in both conformers using the method rename_atom. The method takes the chain label, residue label and the old and new atom name as parameters. Internally it recurses through the object structure and calls rename for each object of type atom. The rename method of the atom class returns immediately if the chain, residue or old atom label do not match, else it uses the set_label method to set the new atom label. The CHARMM topology file contains a residue HSP, which is a doubly protonated histidine. Since the hydrogen placement procedure in the following has to place both the ϵ- and δ-proton, all histidine residues have to be named HSP. The method rename_residue in line 36 is analogous to rename_atom in recursing through the object tree and renaming all residues, which match the given parameters.

The hydrogen adding procedure will add rotateable hydrogen (*e.g.,* in hydroxyl groups) as many times as there are hydrogen bond acceptors in the vicinity (according to geometric criteria). To assign energies for the different rotamers, potential energies were calculated and stored as function of the torsion angle. The method `setup_rotamer_sites` (line 37) reads a file `charmm.potentials` (see appendix A.4.3), which contains the residue names and four atoms defining the dihedral angle of the torsion together with a file name, in which the pre-computed torsion angles and energies are stored. Those files are read and stored into `potential` objects.

```
37 $mol->setup_rotamer_sites('charmm.potentials');
38 my %topology = (
39           method              => 'topology',
40           forcefield          => 'charmm',
41           file                => [
42                                   'top_all27_prot_na.rtf',
43                                   'fes.rtf'
44                                  ],
45           patch               => [
46                                   'NTER_1',
47                                   'CTER_98',
48                                  ]
49         );
50 $mol->setup_charge_groups(\%topology);
51 $mol->setup_graph(\%topology);
52 $mol->setup_rotamers;
53 $mol->setup_sites;
```

The lines 38 to 51 are analogous to the simpler example. An object of the `topology` class is generated and used to define charge groups (*via* `chargegroup` and `chargegroup_member` objects). The `topology` object is also used to construct the graph of bonded atoms to define the topology of the molecule. In line 52 the method `setup_rotamers` is called, which constructs `coordinate_set` and `rotamer_form` objects by calling `setup_rotamers` for each atom (described in detail in the section about the `atom` class). At this point only `rotamer_form` objects are created for the sidechains with different occupancies and for the backbone section including the peptide flip. Energies of the rotamers are calculated from the occupancy values interpreted as probabilities. The method `molecule::setup_sites` calls the method `atom::setup_sites` for each `atom` object. The method `atom::setup_sites` constructs a `site` object and generates `instance` and `instance_member` objects *via* calling the `site::setup` method. (Also these methods were described in more detail before.)

In the next step, the two rotamer forms for the peptide flip (CO-in and NH-in) should be separated into conformers. Since two conformers are already present (from mol-1 and mol-2), this leads to four conformers ("mol-1 CO-in", "mol-1 NH-in", "mol-2 CO-in" and "mol-2 NH-in").

```
54 # coordinate set N_ALA_45_A is peptide flip, I separate the two rotamer forms
55 # into two conformers
56 # conf 0 (id: 0)          -> mol-1 CO-in (coordinate set N_ALA_45_A only
57 #                                         rotamer form A_1CZP.pdb)
58 #                         -> mol-1 NH-in (coordinate set N_ALA_45_A only
```

```
59 #                                              rotamer form B_1CZP.pdb)
60 # conf mol-2 CO-in (id: 1) -> mol-2 CO-in (coordinate set N_ALA_45_A only
61 #                                              rotamer form A_1CZP.pdb)
62 #                             -> mol-2 NH-in (coordinate set N_ALA_45_A only
63 #                                              rotamer form B_1CZP.pdb)
64 $conf = $mol->copy_conformer(0, 'mol-1 NH-in');
65 $conf->delete_rotamer_form('N_ALA_45_A', 'A_1CZP.pdb');
66 $conf = $mol->get_ref_by_type_id(0, 'conformer');
67 $conf-> delete_rotamer_form('N_ALA_45_A', 'B_1CZP.pdb');
68 $conf = $mol->copy_conformer(1, 'mol-2 NH-in');
69 $conf->delete_rotamer_form('N_ALA_45_A', 'A_1CZP.pdb');
70 $conf = $mol->get_ref_by_type_id(1, 'conformer');
71 $conf->delete_rotamer_form('N_ALA_45_A', 'B_1CZP.pdb');
72 $mol->get_member('0')->set_label('mol-1 CO-in');
```

The method `copy_conformer` copies in line 64 the first `conformer` (mol-1, index 0) into a new `conformer` (with label "mol-1 NH-in") within the same `molecule`. From `conformer` "mol-1 NH-in" the `rotamer_form` CO-in with label A_1CZP.pdb of the `coordinate_set` with label N_ALA_45_A is deleted (`delete_rotamer_form` in line 65). In line 66 a reference to the conformer with index 0 is obtained and used in line 67 to delete the `rotamer_form` NH-in with label B_1CZP.pdb. The `conformer` with index 1 (and label "mol-2 CO-in", line 27) is processed analogously. Finally, in line 72 the label for `conformer` 0 (default, since no label is given in the pdb-file) is changed to "mol-1 CO-in". As result, the `molecule` object has four conformers, where "mol-1 CO-in" and "mol-1 NH-in" originate from chain A (mol-1) and "mol-2 CO-in" and "mol-2 NH-in" originate from chain B (mol-2). "mol-1 CO-in" and "mol-2 CO-in" contain the peptide flip in the orientation CO-in, which had the occupancy label A. "mol-1 NH-in" and "mol-2 NH-in" contain the peptide flip in the orientation NH-in, which had the occupancy label B.

The next step is to add hydrogens using the external program Hwire.

```
73 # multiple hydrogens as occupancies/sites
74 my %param = (
75             method              => 'instance_x',
76             program             => 'hwire',
77             binary              => '/home/essigke/bin/hwire -f',
78             hwire_parameter     => {
79                   par           => [
80                                     'charmm22_hpath.dat',
81                                     'fes.dat'
82                                    ],
83                   d1max         => '3.8',
84                   d2max         => '4.5',
85                   alpha         => '110',
86                   epsilon       => '60',
87                   zeta          => '60',
88                   hydrogens     => 'on',
89                   rotamers      => 'on',
90                   lonepairs     => 'off',
```

```
91                  patch           => [
92                                      'NTER 1 A',
93                                      'CYS_FE1 41 A',
94                                      'CYS_FE1 46 A',
95                                      'CYS_FE2 49 A',
96                                      'CYS_FE2 79 A'
97                                     ]
98                                     },
99          );
100 $mol->add_hydrogen(\%param);
101 $mol->setup_charge_groups(\%topology);
102 $mol->setup_graph(\%topology);
103 $mol->setup_rotamers;
104 $mol->setup_sites;
```

In lines 74 to 99 a hash is filled, which can be passed to the method `add_hydrogen` in line 100, which performs the addition of hydrogens. Since there are so many programs available for this purpose, all with their strength and weaknesses, I assumed that it would be usefull to have this part of Perl Molecule variable by a plug-in infrastructure (similar to the potential usage of different topology file formats). Currently, only Hwire is available, which was discussed with some of its parameters in section 2.4.3. A problem of Hwire is, that it can not deal properly with rotamers as they originate here from sidechains with multiple occupancies. The `add_hydrogen` method therefore generates separate conformers, lets Hwire add hydrogens to each of them and finally joins the conformers to retrieve the rotamers with hydrogens added. The generation of conformers from rotamers requires to deal with the different combinations of rotamers in a structure. Internally, `add_hydrogen` uses the `combinatorics` class, which provides different methods of generating state vectors. Usually one wants to use the method `combinatorics::full_x`, which generates a complete set of state vectors. For example, a structure with rotamers in five `coordinate_set` objects and three `rotamer_form` objects each, would lead to $3^5 = 243$ conformations to which hydrogens have to be added (or 972 for the four conformations of 1CZP). For such a large number of structures, hydrogen adding would be very time consuming. The method `combinatorics::instance_x` generates a structure for each instance (or `rotamer_form` in this case), leading to three conformers in the previous example (with state vectors (00000), (11111) and (22222)). The method `combinatorics::site_x` generates a structure for each site (`coordinate_set`) in each instance `rotamer_form`, while all other sites are in their reference instance. In the example, it would lead to eleven structures ($(3-1)\cdot 5+1 = 11$, with state vectors of *e.g.*, (00000), (10000), (20000), (01000), $(02000)\ldots$). For the 1CZP structure, it was found that the `instance_x` method is sufficient, since the sidechains with different occupancies are well separated on the surface of the protein. However, for other structures this method may not be sufficient. Probably the best way to determine which method to use, if it can not be decided by quick visual inspection, is to try the different methods in a test calculations. If the methods including more conformers do not increase the number of hydrogen positions for rotateable hydrogen, the cheaper method is sufficient. Usually, Hwire will place rotateable hydrogens in slightly different positions, which complicates this analysis. To remove these artefacts, the method `molecule::remove_close_rotamers` is provided, which takes a threshold dihedral angle as parameter. All rotamer forms and associ-

ated hydrogen coordinates are deleted, which are closer than the threshold dihedral angle to an accepted rotamer form.

After adding hydrogen, the new atoms have to be assigned to the present charge groups and the molecular topology has to be updated to include the bonds to the hydrogen (line 101 and 102). New coordinate sets are created due to hydrogen rotamers or existing coordinate sets get more rotamer forms due to a combination of, *e.g.,* sidechain occupancies and hydrogen rotamers. Therefore `setup_rotamers` is called again (line 103). Since the `coordinate_set` and `rotamer_form` objects changed, the `site`, `instance` and `instance_member` objects have to be adjusted by `setup_sites` (line 104).

The pdb-file does not contain any information on the atom radii, unlike the pqr-file in the previous example. Therefore, the radii are assigned from a file. Also different charge forms have to be assigned to protonateable groups of the protein and the reduction forms of the iron-sulfur center.

```
105 # setup bondi (assign radii to atoms)
106 %param = (
107           method              => 'bondi',
108           file                => 'bondi.rad'
109           );
110 $mol->add_radii(\%param);
111 #
112 # assign additional charges from .st/.est files, include as instances
113 # assign energies for instances based on model pK or ADF energy
114 #
115 $mol->setup_charge_sites ('vFdx.sites');
116 #
117 # setup_xst - est files containing multiple residues from multiple chains
118 foreach my $conf_label (qw( 'mol-1_CO-in' 'mol-1_NH-in' 'mol-2_CO-in' 'mol-2_NH-in')) {
119   my $xst_name = "1CZP_".$conf_label.".xst";
120   my $xst = xst_file3->new($xst_name);
121   my $conf = $mol->get_member($conf_label);
122   $xst->setup($conf);
123 }
124 $mol->setup_sites;
```

The method `add_radii` in line 110 takes also a hash as parameter. The hash (line 106 to 109) contains analogously to the topology hash a key `method` and a key `file`. The `method` allows extensions for other methods or file formats to determine radii (*e.g.,* from some force field, PARSE radii [146], van-der-Waals radii *etc.*). Currently only a single file format used for Bondi [147] radii is available (appendix A.5.4). In line 115 the method `setup_charge_sites` is used to define `charge_set` and `charge_form` objects as in the previous example. By this method only charge sets can be defined, which are not larger than one residue, because the sites file is residue based. Future versions may change the sites file to allow also to specify xst-files and fst-files. A different xst-file should be used for each of the four conformers. It is done in this example (line 118 to 123) by iterating over the conformer labels, constructing a xst-file name, reading the file by constructing a `xst` object and passing a `conformer` object (selected by the label) to the `xst::setup` method. In line 124 the method `setup_sites` is called to

regenerate the `site`, `instance` and `instance_member` objects including the charge sets and charge forms. Now the Perl Molecule data structure contains all information necessary for writing the QMPB input.

```
125 #
126 # write qmpb input for each conformer
127 #
128 # qmpb general options:
129 my %qmpb = (
130              meadpath => '$MEADPATH/bin',
131              T => $mol->get_T,
132              I => 0.1,
133              backfile => 'background.pqr',
134              OGMpoints => [61, 61, 131, 131],
135              OGMspace => [4.0, 1.0, 0.5, 0.2],
136              MGMpoints => 131,
137              MGMspace => 0.2,
138              epsin1 => 1,
139              epsin2 => 4,
140             );
141 $mol->write_qmpb (\%qmpb);
142 #
143 $mol->print_statistics;
144 $mol->write_pdb ('1CZP_h.pdb');
145 $mol->write_pqr ('1CZP_h.pqr');
```

The lines 129 to 141 are again analogous to the simple example. At the end the methods `print_statistics`, `write_pdb` and `wite_pqr` are called for debugging purpose.

4.6.5 More Features of Perl Molecule

Despite the length of this section, only some of the many features of Perl Molecule could be shown. For example, one method not discussed so far is invoked by `molecule::flip_residues`. The orientation of the sidechains of histidine, glutamine and asparagine can usually not be determined doubtless by x-ray crystallography, because the electron densities of nitrogen, carbon and oxygen are too similar. In fact two rotamers would fit the electron density equally well, differing in the rotation of a torsion angle by 180° next to the functional group. The correct orientation is usually assigned based on the hydrogen bond network by visual inspection of the crystallographer or by using protein structure validation programs like Whatif. However, in some cases a reasonable hydrogen bond network is possible in both orientations or the hydrogen bond network is inverted by an unusual protonation form of a surrounding residue. Therefore, it can be valueable to include histidine, glutamine and asparagine by two rotamer forms, avoiding any bias and leaving the decission on the most probable instance to the Monte Carlo procedure.

Another aspect, which could not be shown in appropriate detail, is the generation of instances of complex sites by `molecule::setup_sites`. Each atom, which has more than one coordinate, is associated with a coordinate set and each coordinate is member of a rotamer form.

Bonded atoms with more than one coordinate are assigned to the same coordinate set, *e.g.,* using the occupancy information given in a pdb-file. Charge sets are usually defined by one of the charge set file types (st-file, est-file, xst-file or fst-file). Upon generation of sites, coordinate sets and charge sets are combined. It is possible that a charge set contains atoms of different coordinate sets and a coordinate set contains atoms of different charge sets. Since sites have to have an integer charge, not only the atoms belonging to coordinate sets and charge sets are included, but all atoms belonging to the same charge group as any atom of one of the sets. It is possible, that the atoms, which were included due to charge groups also belong to additional coordinate sets or charge sets, which need to be added to the site. The method acts according to these rules until a consistent set of atoms for the site can be found. Then, all coordinate sets and charge sets are combined by generating instances for all permutations of rotamer forms and charge forms. Therefore, sites constructed from many coordinate sets and charge sets can have very many instances. Without assistance by the program, it would be very hard for a human to make the right choice of atoms to ensure that all rules are fulfilled and that all permutations of coordinates and charges are included in the subsequent QMPB calculations.

The feature to generate sidechain rotamers based on a rotamer database, is also not shown in one of the examples, but it will be discussed in section 5.3.3. More capabilities should be relatively easy to add to Perl Molecule because of the detailed class hierarchy and the chemical and physical knowledge already implemented. Needs of future projects will drive the development on Perl Molecule.

4.7 Summary

To make the theory developed in the previous chapter applicable for larger biomolecules, a set of programs was developed. QMPB is the program to compute intrinsic energies and interaction energies. The intrinsic energy contains contributions, which can be obtained experimentally, by molecular mechanical or quantum chemical calculations and have to be given as input to QMPB. Furthermore, pqr-files are required for each instance specifying coordinates and charges for each atom of the site. QMPB interprets its input file to set up continuum electrostatic calculations to compute transfer energies. A LPBE solver is not implemented in QMPB, but the program uses a set of external programs to perform the calculations. This approach has the advantage, that on one hand, the PB solver can be easily exchanged by replacing the programs, on the other hand parallelization of the time consuming step solving the LPBE, does not need to be done within QMPB. Instead, QMPB writes a shell script, which can be executed directly and performs the calls to the PB solver sequentially. Alternatively, a additional program is used splitting the shell script into program calls to be executed on different computers. The program can also use an existing queuing system to distribute the workload on a compute cluster. Therefore it is easily possible to optimize the program for the local computing environment without modifying QMPB directly. In a second run, QMPB collects the output of the helper programs, calculates the intrinsic energies and interaction energies and writes them into suitable files for further processing.

During development of QMPB, it was an aim to keep the program as transparent as possible. Splitting the calculation in several steps with extensive input and output files and making

the electrostatic calculations repeatable without recomputing the rest of the systems allows the user to follow every step. This transparency is valueable teaching students and simplifies extensions to the program as the progress of research requires them.

For small systems the output of QMPB can be used to solve the grand canonical partition function, *e.g.,* by the program SMT. For larger systems, the probability of instances can be approximated, *e.g.,* by the Monte Carlo program GMCT. In practice, probabilities of instances at a fixed set of chemical potentials are not of as much interest as the titration behavior of a site, when changeing the chemical potentials. Therefore, each chemical potential is scanned in small steps over a wide range, which in most cases is computationally more demanding than solving the LPBE. In fact, QMPB scales linearly with the number of instances, while the Monte Carlo algorithm has an exponential scaling behavior with the number of ligand types, *i.e.,* dimensions in which the chemical potential needs to be varied. Since for many biological questions studies with many ligand types are of great interest and the generalized titration theory and QMPB do not impose limits in this respect anymore, future applications will certainly require such high-dimensional titration calculations. The adaptive mesh refinement algorithm (AMR) is an approach to reduce the computational cost. The basic idea is to exclude instances, which do not titrate in a region of chemical potential space on a coarse grid for calculations on finer grids. This reduces the number of points, at which probabilities need to be calculated and reduces the state vector length for analytical or Monte Carlo calculations for the remaining sites.

QMPB relies on a large number of input files, which can be complex for certain sites. Manual generation of these input files is tedious and error prone. Shell scripts written for this purpose suffer from readability, reliability and reueseability. One source of these problems is the lack of complex data structures in shell script, another the inherent complexity to process a biomolecular structure to generate input for electrostatic calculations. A significant amount of chemical and physical knowledge is required to set up the calculations in a meaningful way, *e.g.,* knowledge of the topology of the molecule is required to choose the right atoms for sites. Since all instances of a site should have an integer charge, charge groups - known from force fields - are a helpful concept to group atoms. Sites may be complex containing several sets of atoms changing their coordinates in a concerted way (rotamer forms of a coordinate set) and sets of atoms changeing their charge (charge forms of a charge set). All permutations of rotamer forms and charge forms need to be generated as instances. To ease these and other tasks, the class library Perl Molecule was written. It implements an ontology following a common hierarchical view on biomolecules. By reading topological descriptions of molecules and other parameter files, sufficient information is gathered to guide the user through the process of input preparation. However, it was felt too inflexible to write a single program for this purpose, because biomolecules are too complex to cover all cases. Future developments or just different opinions of different users are likely to change the protocol. Therefore Perl Molecule is a class library, which allows the user to write powerful Perl scripts with little effort, but sufficient flexibility for a large variety of use cases.

CHAPTER 5

EXAMPLES FOR LIGAND BINDING STUDIES

5.1 Lysozyme as Test Case for QMPB

The use of hen egg white lysozyme (HEWL) as test case for protein electrostatics has a long history [27, 32, 33, 35, 46, 48–52]. HEWL is a small enzyme of 129 residue length and 21 titrateable residues (1 histidine, 2 glutamate, 7 aspartate, 3 tyrosine and 7 lysine, C-terminal leucine). Most titrateable residues are solvent exposed and their $\mathrm{p}K_a$ values were experimentally determined. Therefore, the protein was often used as benchmark system, assuming that a low RMSD between the calculated and experimental $\mathrm{p}K_a$ values is a good measure for the quality of a method in predicting $\mathrm{p}K_a$ values. However, the studies have shown, that rather simple methods using a high dielectric constant for the protein (20 and higher [48, 50]) give lower RMSD values than physically more sophisticated methods using lower dielectric constants [32]. In contrast, the buried active site residues can be predicted better with low dielectric constants and a physically better model. The reason is that for a small protein most titrateable residues are on the surface and are very flexible. This flexibility is hard to account for by continuum electrostatics, especially if crystal structures are used showing only a single rotamer. Implicitly a higher dielectric constant accounts for this flexibility and reduces the RMSD. Nevertheless, little can be learned from such a model, because usually the most important residues are buried at least partially and have to be in a specific orientation. Georgescu *et al.* [33] have shown, that as good RMSD $\mathrm{p}K_a$ values can be achieved with a dielectric constant of 4, if sidechain rotamers generated by a rotamer database were taken into account. In their calculation the RMSD gets worse for higher dielectric constants as one would expect for a model taking flexibility into account.

Donald Bashford, the author of the MEAD package, used HEWL as test enzyme for his program Multiflex. He provides the input data for his early work [35] and scripts to run his programs and to reproduce his results. I provide the same example as test case for Multiflex2qmpb and QMPB to compare to Multiflex.

Fig. 5.1 shows that the titration curves obtained with Multiflex and Karlsberg [34] are identical with those obtained with Multiflex2qmpb, QMPB and GMCT within sampling errors of the Monte Carlo procedure. Therefore, the new programs can replace the old programs, but add additional functionality, which is demonstrated in the following example applications.

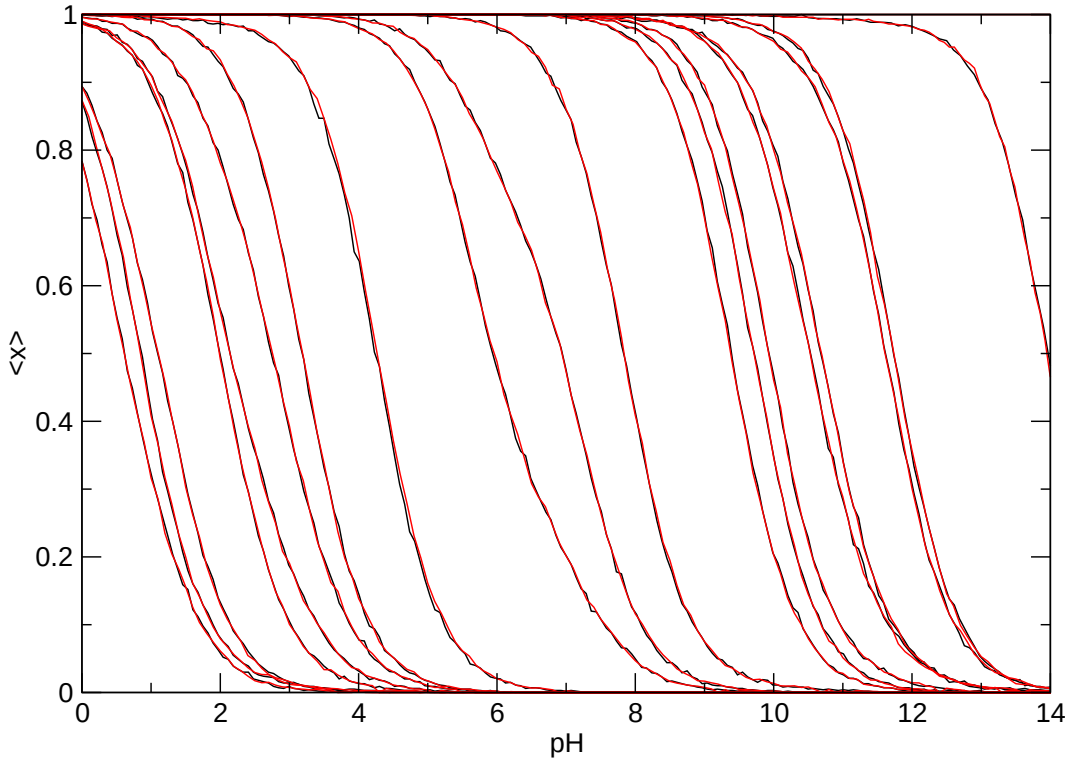


Figure 5.1. Titration curves of titrateable residues in lysozyme calculated with the old set of programs Multiflex and Karlsberg (black) as well as calculated with the new set of programs Multiflex2qmpb, QMPB and GMCT (red). The curves are identical within the sampling error of the Monte Carlo procedures.

5.2 Effects of Serine Phosphorylation and Histidine Protonation and Phosphorylation on HPr

This project was done in collaboration with Nadine Homeyer, Heike Meiselbach and Heinrich Sticht (University of Erlangen). The results are published [148, 149].

In both publications, the molecular dynamics simulations and most of the structural analysis were performed by Nadine Homeyer in the group of Heinrich Sticht, while the protonation probability calculations (using QMPB) and the calculations using the microstate model were done by me. Matthias Ullmann contributed by helpful suggestions, discussion and formulation of the manuscripts.

5.2.1 The Biological Role of HPr

The histidine-containing phosphocarrier protein (HPr) is involved in the regulation of a number of processes associated with carbohydrate uptake and consumption in various bacterial species. It plays a central role in the phosphoenolpyruvate:sugar phosphotransferase system

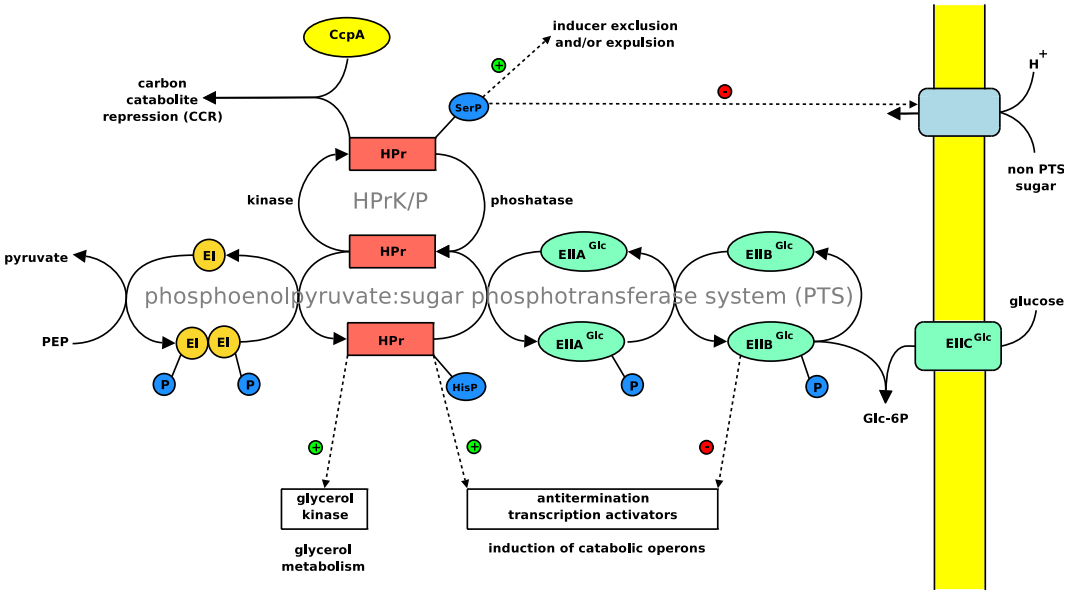


Figure 5.2. Role of HPr in Gram-positive bacteria. HPr is involved in the phosphoenolpyruvate:sugar phosphotransferase system (PTS) in transferring phosphate from enzyme I (EI) to enzyme IIA (EIIA). The phosphate transfer involves transiently phosphorylating His 15 of HPr. Alternatively, HPr can also be phosphorylated on Ser46, regulating catabolite repression *via* the catabolite control protein A (CcpA) on the gene expression level. Additionally, HPr can play a regulatory role at a number of points of carbohydrate metabolism. Adapted from [152].

(PTS), where it was discovered in 1964 [150], but it modulates also the activity of carbohydrate specific regulators, catabolic enzymes, permeases, and transcription factors [151].

The PTS simultaneously transports sugars into the cell and phosphorylates them, because cell membranes are impermeable for sugar phosphates. Unlike the sugar transporters of other cells, which hydrolyze two ATPs to actively transport and phosphorylate the sugar, the PTS only hydrolyzes a single phosphoenolpyruvate (PEP) for both transport and phosphorylation. The energy-efficiency is particularly remarkable recalling that glycolysis gains only two molecules of ATP *per* glucose molecule. In the first step one molecule of ATP is used for phosphorylating glucose, in the last step only one molecule of ATP is gained *per* PEP molecule. If an additional ATP would be consumed for the transport, the energetic gain of glycolysis would be halved.

In the PTS phosphate is transferred from PEP to a soluble enzyme I (EI), which phosphorylates HPr on His15. The phosphate of HPr-His15P is in turn used to phosphorylate enzyme II (EII). EII is sugar specific (*i.e.*, there are different EII for glucose, fructose, mannose *etc.*), so that as many as fifteen different EII were found in *Bacillus subtilis* [153]. It contains a transmembrane protein, which transports the sugar into the cell and phosphorylates it. The EII gene may encode for a single protein with one or more intracellular domains involved in phosphate transfer and regulation additionally to the transmembrane domain or separate proteins. The number of cellular EII gene products (sometimes called EIII *etc.*) depends on the organism and sugar.

In Gram-negative bacteria the PTS regulation is particularly well studied for the glucose transport of *Escherichia coli* [154]. Here, phosphorylated EIIA^{Glc} activates adenylate cyclase, which produces cAMP as a ligand for the catabolite gene activator protein. It enhances transcription of catabolic genes in the absence of glucose. Dephosphorylated EIIA$^{\mathrm{Glc}}$, occurring in the presence of glucose, inhibits the influx of lactose into the bacterium and thus prevents the induction of the lac operon.

In Gram-positive bacteria, carbon catabolite repression (CCR) operates by phosphorylation of HPr at Ser46 by the ATP dependent kinase/phosphatase HPrK/P [155–158]. At high levels of ATP as in glycolysing bacteria, the kinase activity of HPrK/P is dominant, producing HPr-Ser46P, which is only slowly phosphorylated by EI. Therefore, the PTS is down-regulated due to a depletion of unphosphorylated HPr. At low levels of ATP, the phosphatase activity of HPrK/P becomes dominant, dephosphorylating HPr and therefore up-regulating the PTS.

Additionally to this direct regulation of the PTS, HPr-Ser46P binds to the transcriptional regulator catabolite control protein A (CcpA) to convert it into its DNA-binding-competent conformation. CcpA binds to operator-like catabolite responsive element (*cre*) sequences, which are common motifs in front of many operons involved in carbohydrate degradation. Experimental data indicate, that the expression of 8% of all *B. subtilis* genes is regulated by CcpA [159, 160]. Thus, HPr-Ser46P is directly involved in regulation of gene expression.

Also HPr-His15P is involved in regulation. The antiterminator proteins of the *bgl-sac* family are regulated by phosphorylation at conserved histidine residues in the PTS regulatory domains (PRDs). Transcription of several operons involved in carbohydrate utilization appears to be controlled by this mechanism. HPr was also shown to allosterically stimulate glycogen phosphatase in *E. coli*. In Gram-positive bacteria, glycerol kinase is activated by phosphorylation at a histidine by HPr-His15P coordinating glycerol utilization with utilization of other carbon sources [161].

HPr is an α-β protein. The core is formed by a four-stranded, antiparallel β-sheet, which is connected by two long helices ($\alpha1$, $\alpha3$) and one short helix ($\alpha2$). The two phosporylation sites of His15 and Ser46 are shown in Fig. 5.3.

The aim of the project was to study the structure, dynamics and physicochemical properties of Ser46 phosphorylation and His15 protonation and phosphorylation to draw conclusions how HPr performs its regulatory functions.

5.2.2 Phosphorylation of Ser46

The first part of the study analyzed HPr-Ser46P in solution and its complex formation with CcpA [148]. The major part of the work was performed by Nadine Homeyer in the group of Heinrich Sticht, who did structural studies, MD simulations, and *in silico* alanine scanning. Their results are summarized here only in brief. An important question within this project, however, was the protonation state of the phosphate bound to HPr, which I calculated by continuum electrostatics using my program QMPB.

Until now, there is no experimentally determined $\mathrm{p}K_a$ value for HPr to compare the results directly. However, studies on model peptides containing phosphoserine gave a $\mathrm{p}K_a$ value of 5.96 [163]. Since due to the protein environment, the $\mathrm{p}K_a$ value might shift significantly, it

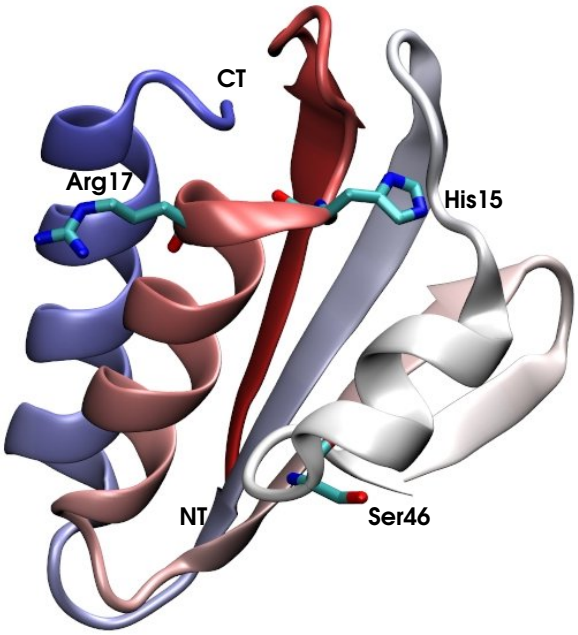


Figure 5.3. Structure of HPr. The backbone is colored by a gradient from N-terminus (red) to the C-terminus (blue). Important residues are shown. The figure was prepared with VMD based on the crystal structure 1PTF [162].

could not be ruled out *a priori*, that the phosphate group could also be singly protonated at neutral pH.

MD simulations were done staring from models 1 and 16 of the NMR structure 1JEM [164]. Therefore, the phosphorylation on His15 was removed and modeled onto Ser46. The phosphate was assumed to be doubly negative charged, unprotonated or singly negative charged, singly protonated. During the MD simulation it was observed, that the sidechain of Asn43 and the phosphate oxygens are frequently within hydrogen bonding distance (48% and 66% of the time for 1JEM - 1 and 1JEM - 16, respectively, using a donor-acceptor distance of smaller than 2.2 Å and an $O \cdots H - N$ angle larger than $150°$ as hydrogen bond criteria). In the alternative, non-hydrogen-bonded conformation Ser46P was observed in an orientation, where the phosphate group is pointing towards the solvent and does not form any intramolecular interactions. We assumed, that the hydrogen bond may have an influence on the apparent $\mathrm{p}K_a$ value of the phosphate group. Therefore, I performed calculations of the apparent $\mathrm{p}K_a$ value with and without hydrogen bond present from snapshots of the MD trajectories started assuming full deprotonation or single protonation of the phosphate. The results are shown in Tab. 5.1.

It is evident, that my results of the protonation probability calculations depend to a significant degree on the input structure. Calculations based on structures taken from the trajectory where protonation was assumed, led to results with increased protonation probability, *i.e.*, a higher apparent $\mathrm{p}K_a$ value. If the hydrogen bond with Asn43 is present, the deprotonated form is stabilized, leading to lower apparent $\mathrm{p}K_a$ values. Such a biasing of $\mathrm{p}K_a$ calculations

Structure -	hydrogen-bonded		non-hydrogen-bonded	
Model	MD time (ps)	$\mathrm{p}K_a$	MD time (ps)	$\mathrm{p}K_a$
doubly deprotonated Ser46P in MD trajectory				
1JEM - 1	1000	5.4	1500	6.2
	4000	5.7	5500	6.2
1JEM - 16	2000	5.9	2800	6.1
	3500	6.0	5000	5.9
singly protonated Ser46P MD trajectory				
1JEM - 1	2300	6.0	500	6.8
	2800	6.2	1000	6.8
1JEM - 16	1600	6.6	500	6.6
	2300	6.0	1000	6.3

Table 5.1. Calculated apparent $\mathrm{p}K_a$ values of HPr-Ser46P. The structures are snapshots of MD simulations started assuming a doubly deprotonated and singly protonated phosphate, respectively. The snaphosts were chosen to have a hydrogen bond between Asn43 (HD22) and one of the phosphate oxygens present or not.

by MD simulation was observed by others before [165]. Nevertheless, this bias is such small, that the calculated $\mathrm{p}K_a$ value varies in less than one unit from the value for the model peptide. Assuming, that the snapshots are representative for the ensemble, one can conclude that the doubly deprotonated form is predominant at neutral pH.

Analysis of the sequence of HPr proteins revealed, that the asparagine forming the hydrogen bond during the MD simulation is strictly conserved within the Gram-positive bacteria that possess HPrK/P and CcpA, but no conserved residue is present in Gram-negative bacteria. Therefore, it was assumed, that Asn43 plays an important role in CCR *via* Ser46 phosphorylation. Structural analysis and MD simulations of the interaction of HPr-Ser46P with CcpA indicated, that Asn43 might be important for fixation of the phosphate group in a conformation consistent with binding. Thus, the form with hydrogen bond is the binding-competent form. In the bound form, however the hydrogen bond between phosphate group and Asn43 is not present, but the intramolecular hydrogen bond is replaced by intermolecular interactions. The role of several residues in the binding interface between the proteins were studied by alanine scanning giving insight into the discrimination of CcpA between different phosphorylation states at His15 and Ser46 of HPr.

5.2.3 Protonation of His15

The second part of the study [149] investigated the protonation and conformational change of His15 in HPr. Unphoshorylated His15 was found in two conformations, a CLOSED confor-

mation, where His15($N\delta1$) is in short distance to Arg17(N) and an OPEN conformation, where the histidine sidechain points into the solvent. The structures obtained with the OPEN conformation were determined at low pH (1FU0 at pH 4.2 [152] and 1PTF at pH 5.0 [162]), while the CLOSED conformations were found in structures around pH 7 [164, 166–171]. However, it was unclear, if the conformational change of His15 is correlated with a protonation of the imidazole ring.

I performed protonation probability calculations on OPEN and CLOSED structures separately based on different crystal and NMR structures. The CLOSED structures are primarily protonated on $N\epsilon2$ in the singly-protonated form and have an apparent pK_a value of about 4. The OPEN structures are primarily protonated on $N\delta1$ in the singly-protonated form and have an apparent pK_a value of below 7 (Tab. 5.2). We aimed to decompose the apparent pK_a value of His15 into Born, background and interaction energy (as discussed in section 3.2). However, the interaction energy is a tensor containing the interaction of a particular instance of a site (here His15), with all other instances of all other sites. Instead, for the energy decomposition, a single energy is required, which represents the mean-field according to the probability $\langle x_i(l_m, \{\mu_\lambda\})\rangle$ of the instances m of other sites l in conformer i at a particular set of thermodynamic variables $\{\mu_\lambda\}$. I formulated a mean-field interaction energy $\langle G_{\mathrm{inter,i}}(j_k, \{\mu_\lambda\})\rangle$ for instance k of site j (here His15) as:

$$\langle G_{\mathrm{inter,i}}(j_k, \{\mu_\lambda\})\rangle = \sum_{l\neq j}^{N_{\mathrm{site},i}} \sum_{m}^{N_{\mathrm{instance},i,l}} G_{\mathrm{inter},i}(j_k, l_m)\langle x_i(l_m, \{\mu_\lambda\})\rangle \tag{5.1}$$

This formulation is similar to the Tanford-Roxby approximation [110], but here the probability $\langle x_i(l_m, \{\mu_\lambda\})\rangle$ is either taken from the statistical average (eq. 3.11) or approximated by a Monte Carlo procedure (section 3.1.4). Tanford and Roxby approximated the probability in an iterative scheme until self-consistency was reached. Due to the dependence on the pH (or thermodynamic variables $\{\mu_\lambda\}$ in general), the mean-field approximation is only exact for the pH equal to the pK_a value.

From the energy decomposition shown in Tab. 5.2 it is obvious, that for the CLOSED conformation a significant contribution to the apparent pK_a value is due to the mean-field interaction energy. For the OPEN conformation this contribution is close to zero. The Born energy is positive and depends more on the structure than on the His15 conformation. The background energy is positive for the CLOSED conformation and negative for the OPEN conformation. An analysis of the change in mean-field interaction energy of His15 *per* residue revealed, that it is clearly dominated by the interaction with Arg17, so that the contribution can be attributed to the hydrogen bond between His15($N\delta1$) and Arg17(N) in the CLOSED conformation.

In parallel, MD simulations were started from OPEN and CLOSED structures in different protonation states. Nadine Homeyer found, that the CLOSED structures in the singly-protonated form and OPEN structures in the doubly-protonated form were stable in the simulations. Doubly-protonated CLOSED structures instead changed the histidine conformation into OPEN, while singly-protonated OPEN structures changed the histidine conformation into CLOSED (Fig. 5.4).

To include the conformational change into the protonation probability calculations, I derived a four microstate model shown in Fig. 5.5. The reference state is the singly-protonated CLOSED form, I assigned a microstate energy of zero ($\Delta G_{\mathrm{micro},1}$ = 0). The doubly-protonated CLOSED

Structure	$\Delta\Delta G_{\mathrm{Born}}$	$\Delta\Delta G_{\mathrm{back}}$	$\langle\Delta G_{\mathrm{inter}}\rangle$	$\Delta G_{\mathrm{model}}$	Sum	$\mathrm{p}K_{\mathrm{app}}$	Conformation
2HPR	1.6	0.1	1.2	-9.1	-6.2	4.5	CLOSED
1KA5	2.2	1.8	0.7	-9.1	-4.3	3.2	CLOSED
1KKM	1.6	0.3	1.3	-9.1	-5.9	4.2	CLOSED
1PTF	1.6	-1.4	0.1	-9.6	-9.3	6.8	OPEN
1FU0 A	2.4	-2.2	-0.1	-9.6	-9.5	6.8	OPEN
1FU0 B	1.2	-0.7	-0.1	-9.6	-9.3	6.4	OPEN

Table 5.2. Energy decomposition of apparent $\mathrm{p}K_a$ values into Born $\Delta\Delta G_{\mathrm{Born}}$, background $\Delta\Delta G_{\mathrm{back}}$, model $\Delta G_{\mathrm{model}}$, and mean-field interaction $\langle\Delta G_{\mathrm{inter}}\rangle$ energy difference of His15 tautomers in $\frac{\mathrm{kcal}}{\mathrm{mol}}$. The mean-field interaction energy is calculated using probabilities at the pH equal to the apparent $\mathrm{p}K_a$ value. For 1PTF and 1FU0 the protonation reaction at $N\epsilon 2$ (doubly-protonated minus δ-protonated), for the other structures the protonation reaction at $N\delta 1$ (doubly-protonated minus ϵ-protonated) is decomposed. The model energy $\Delta G_{\mathrm{model}}$ is calculated from the model $\mathrm{p}K_a$ value as $\Delta G_{\mathrm{model}}$= $-RT\ln 10\ \mathrm{p}K_{a,\mathrm{model}}$ with $\mathrm{p}K_{a,\mathrm{model}}$ of 7.0 and 6.6 for $N\epsilon 2$ and $N\delta 1$, respectively.

form has a microstate energy of $\Delta G_{\mathrm{micro},2}$ = $RT\ln 10$ (pH- $\mathrm{p}K_{\mathrm{app1}}$), where $\mathrm{p}K_{\mathrm{app1}}$ is the apparent $\mathrm{p}K_a$ value of the CLOSED conformation calculated before. The singly-protonated OPEN form has a microstate energy $\Delta G_{\mathrm{micro},3}$ = ΔG_{conf}, with a conformational energy ΔG_{conf} containing the torsional energy of the $\chi 1$ and $\chi 2$ angles of His15, but also all other strains in the structure building up or relaxing upon the conformational change. Finally, the doubly-protonated OPEN form has a microstate energy $\Delta G_{\mathrm{micro},4}$ = ΔG_{conf}+ $RT\ln 10$(pH- $\mathrm{p}K_{\mathrm{app2}}$), where $\mathrm{p}K_{\mathrm{app2}}$ is the apparent $\mathrm{p}K_a$ value of the OPEN conformation calculated before.

The population of each microstate n is calculated by

$$\langle x_n(\Delta G_{\mathrm{conf}},\mathrm{pH})\rangle = \frac{\exp\left(-\frac{\Delta G_{\mathrm{micro},n}}{RT}\right)}{\sum_{m=1}^{4}\exp\left(-\frac{\Delta G_{\mathrm{micro},m}}{RT}\right)} \tag{5.2}$$

and the probability of the CLOSED conformation equals

$$\langle p_{\mathrm{CLOSED}}(\Delta G_{\mathrm{conf}},\mathrm{pH})\rangle = \langle x_1(\Delta G_{\mathrm{conf}},\mathrm{pH})\rangle + \langle x_2(\Delta G_{\mathrm{conf}},pH)\rangle. \tag{5.3}$$

The resulting probability of the CLOSED conformation and population of the four microstates are plotted in Fig. 5.6 A and C-F, respectively. The values $\mathrm{p}K_{\mathrm{app1}}$ = 4.5 (2HPR - CLOSED) and $\mathrm{p}K_{\mathrm{app2}}$ = 6.8 (1PTF - OPEN) used for the plots were obtained from calculations at atomic detail for the highest resolution crystal structures available. The conformational energy ΔG_{conf} was estimated from the experimental $\mathrm{p}K_a$ values of *Bacillus subtilis* (5.4) and *Enterococcus faecalis* (6.1) of His15 HPr by setting $\langle p_{\mathrm{CLOSED}}(\Delta G_{\mathrm{conf}},5.4)\rangle = 0.5$ and $\langle p_{\mathrm{CLOSED}}(\Delta G_{\mathrm{conf}},6.1)\rangle = 0.5$, respectively. Solving the equation leads to conformational energies ΔG_{conf} of 1.8 $\frac{\mathrm{kcal}}{\mathrm{mol}}$ and 0.7 $\frac{\mathrm{kcal}}{\mathrm{mol}}$, respectively (marked by dashed lines in Fig. 5.6). At pH 7 and positive conformational energy, the singly-protonated and CLOSED microstate (Fig. 5.6 C) is favored, while at lower pH

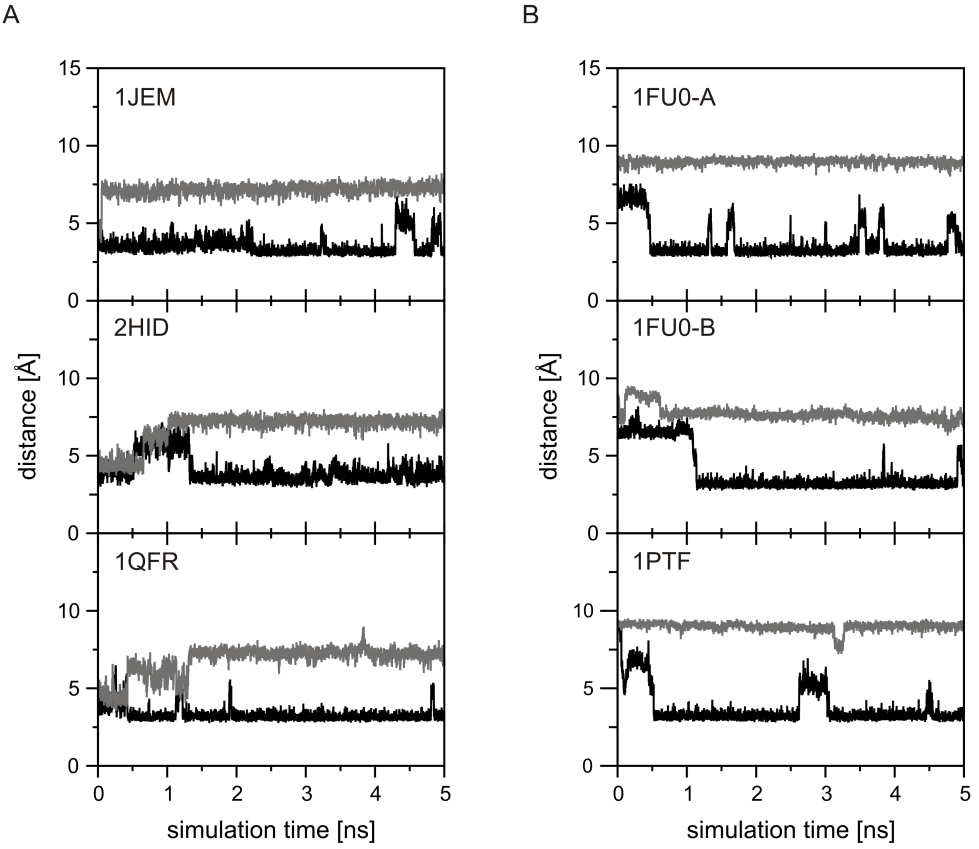


Figure 5.4. Orientation of His15 in dependence of the protonation form during MD simulations done by Nadine Homeyer. The His15($N\delta1$)-Arg17(N) distance is plotted as function of the simulation time. A: Simulations of CLOSED structures, which show an opening motion in the doubly-protonated form (gray), while they are stable in the singly-protonated form (black). B: Simulations of OPEN structures, which show a closing motion in the singly-protonated form (black), while they are stable in the doubly-protonated form (gray).

the doubly-protonated and OPEN microstate (Fig. 5.6 F) is favored. In order to populate the two remaining microstates (CLOSED, doubly-protonated and OPEN, singly-protonated), very large positive conformational energies stabilizing the CLOSED conformation in the doubly-protonated form or negative conformational energies ΔG_{conf} stabilizing the OPEN conformation are required.

Thus, we obtain a simple picture of our system from this model. His15 of HPr has two conformations: A CLOSED conformation with a low apparent $\mathrm{p}K_a$ value, which is populated in the singly-protonated form and an OPEN conformation with a high apparent $\mathrm{p}K_a$ value, which is populated in the doubly-protonated form. The experimental $\mathrm{p}K_a$ value describes a mixture of both conformations, which causes that the $\mathrm{p}K_a$ value is in between the values for the OPEN and the CLOSED conformation. The CLOSED to OPEN transition in the singly-protonated form requires His15 to adopt a less favorable conformation with higher energy, but this energy is compensated by protonation of this residue.

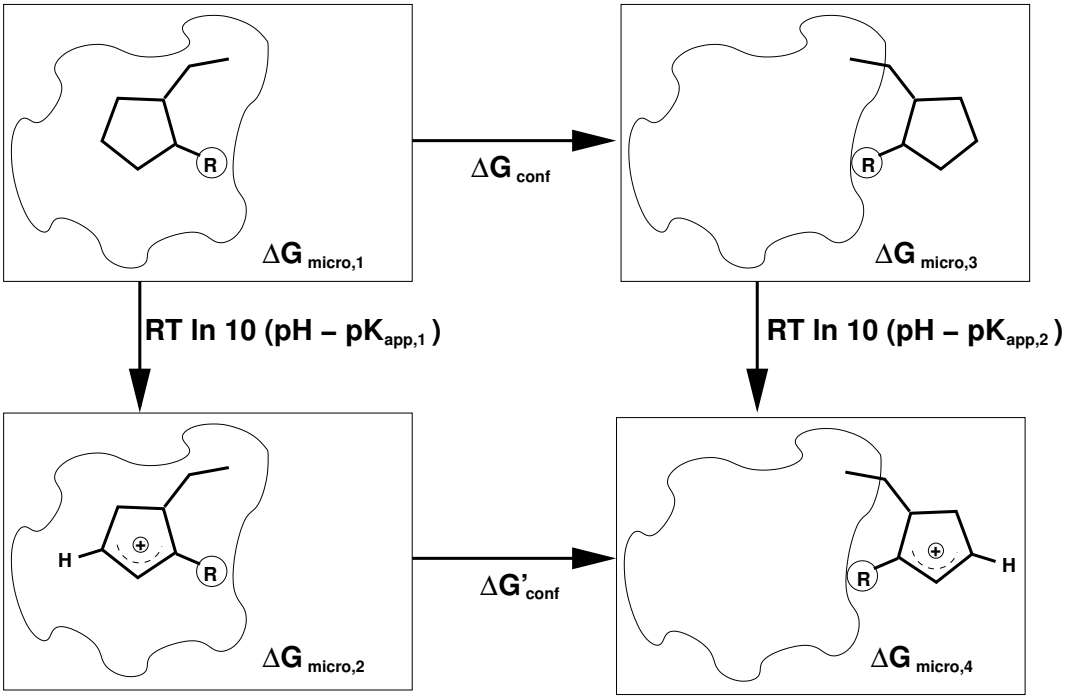


Figure 5.5. Thermodynamic cycle with schematic pictures of the four microstates in a simple model. The moiety R is a hydrogen in His15 HPr or a phosphate group in His15P HPr. The relative population of histidine tautomers is not included in the figure.

The model explains very well the observations from experiment and MD simulations showing a pH or protonation state dependence for observing the OPEN and CLOSED conformation.

5.2.4 Phosphorylation of His15

Structures for phosphorylated His15 are only available at pH 7.4 (1JEM [164]) and 7.5 (1PFH [167]) showing a CLOSED conformation. It was unclear, if the phosphorylated form would also show a conformational transition at lower pH. Taking into account the experimental $\mathrm{p}K_a$ values of 7.7-8.3 [172] for His15P, the imidazole ring has to be assumed to be predominantly protonated (positively charged) in the experimental structures, which lead to CLOSED to OPEN transitions in the unphosphorylated form. We applied the same methods as before to study HPr-His15P.

The apparent $\mathrm{p}K_a$ values calculated for seven models taken from the 25 NMR structures in the 1JEM ensemble of phosphorylated HPr are in the range from 5.8 to 7.3 (Tab. 5.3), which is considerably lower than the experimental results. However, the snapshots from MD simulations (by Nadine Homeyer) on phosphorylated HPr showed significantly higher $\mathrm{p}K_a$ values in excellent agreement with experiment. Therefore, we concluded, that the lower $\mathrm{p}K_a$ values result from local inaccuracies of the analyzed structures due to lack of direct NOE distance restraints for the phosphate group. Additionally, neither electrostatic interactions nor water molecules were included in the NMR refinement, which might cause problems for the highly charged phosphate group. The MD simulation led to a refinement of the structure.

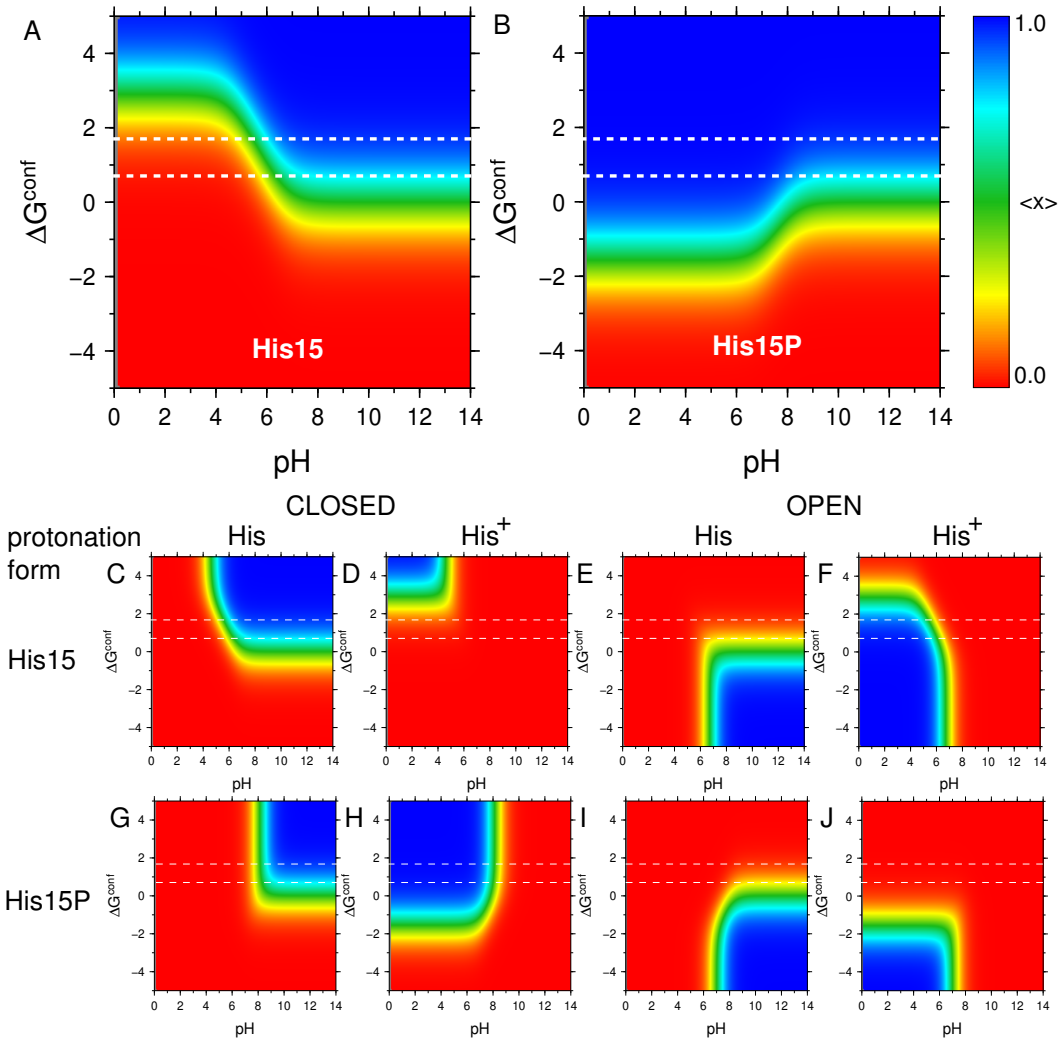


Figure 5.6. Probability plots of the four microstate model of His15 (A, C-F) and His15P (B, G-J). Apparent $\mathrm{p}K_a$ values of 4.6 and 6.8 were used for His15 in CLOSED and OPEN conformation, respectively. For His15P the respective values were 8.0 and 7.0. The conformational energy ΔG_{conf} is given in $\frac{\mathrm{kcal}}{\mathrm{mol}}$. A: Probability of the CLOSED conformation (both protonation forms) of His15. B: Probability of the CLOSED conformation (both protonation forms) of His15P. C-F: Protonation of the CLOSED (C,D) and OPEN (E,F) conformation of HPr-His15 in dependence on the two different protonation forms (denoted as His and His^+, respectively). G-J: Probability of the CLOSED (G,H) and OPEN (I,J) conformation of HPr-His15P in dependence on the two different protonation forms (denoted as His and His^+, respectively).

Model	$\Delta\Delta G_{\mathrm{Born}}$	$\Delta\Delta G_{\mathrm{back}}$	$\langle\Delta G_{\mathrm{inter}}\rangle$	$\Delta G_{\mathrm{model}}$	Sum	$\mathrm{p}K_{\mathrm{app}}$	d1
1	-2.6	1.9	0.9	-9.6	-9.4	6.9	3.57
2	-2.1	1.9	1.8	-9.6	-8.0	5.8	3.60
3	-2.2	1.8	0.9	-9.6	-9.1	6.6	2.21
13	-2.2	1.3	0.4	-9.6	-10.1	7.3	2.21
14	-2.1	1.4	1.2	-9.6	-9.1	6.6	2.20
16	-2.2	1.4	1.0	-9.6	-9.5	6.9	2.92
22	-2.0	1.8	0.9	-9.6	-8.9	6.5	2.20
MD 5200 ps	-3.3	1.8	-0.5	-9.6	-11.7	8.5	1.91
MD 5934 ps	-2.6	1.7	-0.5	-9.6	-11.1	8.1	1.88
MD 8300 ps	-3.3	2.4	-0.5	-9.6	-11.1	7.9	4.50
MD 8600 ps	-2.4	1.5	-0.5	-9.6	-11.0	7.9	4.60

Table 5.3. Protonation of the imidazole of His15P in seven models of the 1JEM ensemble and four snapshots from the MD simulation. The energy (in $\frac{\mathrm{kcal}}{\mathrm{mol}}$) is decomposed as before. The shortest distance (d1 in Å) between a phosphate group oxygen and Arg17 amide hydrogen is given.

Interestingly, a hydrogen bond of the phosphate group oxygen with the backbone of Arg17 or Ala16 is present in about 70% of the structures collected during the MD simulation. Therefore, we were analyzing if the presence of the hydrogen bond has an effect on the calculated apparent $\mathrm{p}K_a$ value, but this was not the case.

The four microstate model was also applied to HPr-His15P to test whether a correct description can be obtained. The apparent $\mathrm{p}K_a$ value for the CLOSED conformation was set to $\mathrm{p}K_{\mathrm{app1}}$ = 8.0, which was calculated before for the structures after MD simulation and is in excellent agreement with experiment. In a hypothetical OPEN conformation, the imidazole ring will show no major interactions with the rest of the protein, and therefore $\mathrm{p}K_{\mathrm{app2}}$ = 7.0 is approximated by the model $\mathrm{p}K_a$ value for phosphohistidine in aqueous solution. It can be assumed that the conformational energy ΔG_{conf} of HPr-His15P is in the same range as for HPr-His15 since this energy term is primarily the torsion energy around $\chi 1$ and $\chi 2$ of the histidine side chain, which should not be strongly affected by the phosphorylation. The same analysis as for His15 showed, that His15P is always in the CLOSED conformation in the range of conformational energy estimated before (region between dashed lines in Fig. 5.6 B). Also the probability calculations on the four microstates (Fig. 5.6 G-J) showed that the OPEN conformations require a negative conformational energy to be populated. It is remarkable, that the same four microstate model can explain both the pH dependent conformational change of HPr-His15 and the conformational rigidity of HPr-His15P.

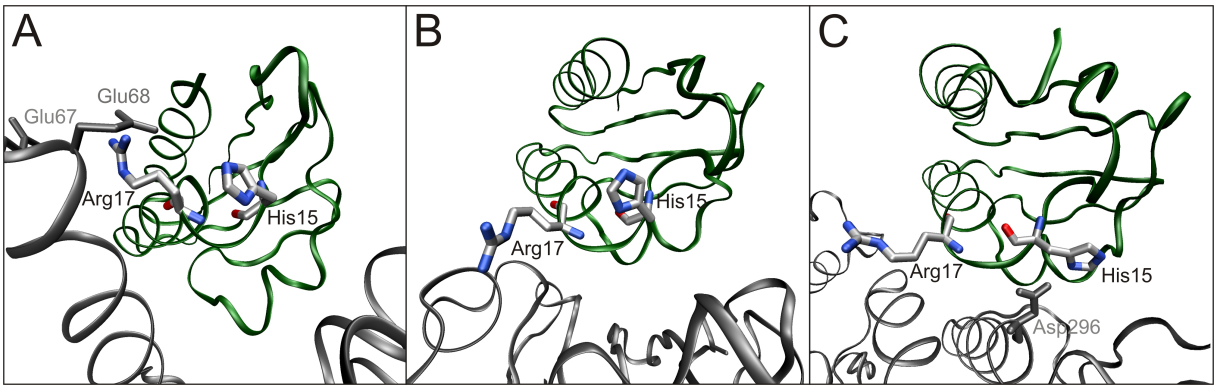


Figure 5.7. Conformation of His15 and Arg17 in three HPr-containing complexes. A: HPr-EI, B: HPr-EIIA$^{\mathrm{Glc}}$ and C: HPr-CcpA. Taken from [149].

5.2.5 Conclusions and Biological Implications of the Project

We have studied the phosphorylation of HPr at Ser46 and His15 as well as the protonation state dependent conformational equilibrium of unphosphorylated His15.

HPr-Ser46P can have two conformations, characterized by the presence of a hydrogen bond of one of the phosphate oxygens with the sidechain of Asn43. The phosphate group is predominantly doubly-deprotonated at neutral pH, independent of the presence of the hydrogen bond and other structural variations. The hydrogen bond was found to be important in the binding process of HPr to CcpA to fulfil the regulatory function.

HPr-His15 has two experimentally observed conformations, we termed CLOSED and OPEN and characterized by the distance of His15($N\delta1$) and Arg17(N). The conformations are coupled to a specific protonation form of His15 as it could be seen by MD simulations. We calculated significantly different apparent $\mathrm{p}K_a$ values for the CLOSED and OPEN conformation and found that the measured values are a mixture of both. We derived a four microstate model which allows to estimate the conformational energy based on our calculated and the measured $\mathrm{p}K_a$ values. The estimated positive conformational energies explain a direct transition from the singly-protonated CLOSED to the doubly-protonated OPEN conformation of His15.

For HPr-His15P a hydrogen bond of one of the phosphate oxygens with Arg17(N) was observed in the MD simulation to be present only from time to time. The hydrogen bond has no significant influence onto the protonation probability of the imidazole ring. The 1JEM structure, however, contains artefacts from the NMR structure refinement method, which had to be removed by MD simulations, before reliable apparent $\mathrm{p}K_a$ values could be calculated. The observed structures of His15P show a conformation very similar to the CLOSED conformation of HPr-His15. Both the MD simulations and the electrostatic calculations based on the four microstate model showed that a OPEN conformation is not adopted. Additionally, the four microstate model states, that such a conformational transition could only be observed for negative conformational energies, but not for the range of positive conformational energies estimated from calculations on HPr-His15.

The biological role of conformational rigidity of His15P on one hand and conformational flexibility of His15 on the other hand can be understood by studying the available complexes of HPr with EI, EIIA and CcpA shown in Fig. 5.7. HPr-His15 is found to be primarily in the singly-$N\epsilon 2$-protonated CLOSED conformation at pH7 appropriate for complex formation with EI and primed to accept the phosphate at $N\delta 1$. HPr-His15P is rigid in the CLOSED conformation, as it is also retained in the complexes of HPr with various structurally different EIIA proteins [173–175]. Phosphate transfer from EI to HPr and further onto EIIA can be accomplished without larger structural rearrangements of His15P. While the OPEN conformation does not seem to play a role in phosphate transfer within the PTS, it is important for the regulatory function of HPr accomplished by binding to CcpA. In this complex His15 adopts an OPEN conformation, which is stabilized by a hydrogen bond to the side chain of Asp296 of CcpA. In contrast, HPr phosphorylated on His15 was found not to bind to CcpA, probably due to the electrostatic repulsion between the phosphate group and Asp296. The conformational flexibility of HPr-His15 might be essential to fulfill its dual role as phosphotransfer-protein and as regulator.

5.3 The Reduction Potential of Ferredoxin

I started to calculate absolute reduction potentials of small iron-sulfur proteins, *i.e.,* on the ferredoxin of the cyanobacterium *Anabaena*, in the group of Rebecca Wade. Despite of excellent crystallographic data, the project turned out to be very complicated. Many problems arose, which in part could be solved by developing Perl Molecule and QMPB. Here, I take ferredoxin also as an example to show key features of my programs.

5.3.1 Previous Structural and Theoretical Work

The vegetative ferredoxin (Fdx) from the cyanobacterium *Anabaena* is a small α/β protein with a single chain of 98 amino acids (Fig. 5.8). It contains a [2Fe-2S] iron-sulfur center coordinated by four cystein. Its biological function is to transfer electrons from photosystem I to ferredoxin-NADP-oxidoreductase (FNR). The reduction of NADP requires two electrons, which are transported in two subsequent steps by two Fdx molecules. Fdx is biochemically and biophysically very well characterized. Nevertheless, potential redox state dependent conformational changes and electron transfer mechanisms are subjects of debate.

The structure of Fdx and analogous proteins is well known for many years [176–178]. In 1999 Morales *et al.* [145] published the crystal structure of oxidized (at 1.3 Å resolution, PDB code 1QT9) and reduced (at 1.17 Å resolution, PDB code 1CZP) *Anabaena* Fdx. Interestingly, they observed a single conformation in the oxidized structure, but two conformations in the reduced structure. The conformational change can be characterized by a rotation of the peptide bond linking Cys46 and Ser47, but requires also some changes of the backbone between Ala45 and Thr48 as well as the sidechains of Ala45, Cys46 and Ser47. In the reduced structure the additional conformation has a hydrogen bond formed between the amide hydrogen, Ser47 (HN), and S*2 of the bridging sulfurs (NH-in). A conformation, where the carbonyl oxygen, Cys46 (O), is pointing inside the [2Fe-2S] cluster binding loop is observed in both the oxidized (1QT9) and reduced structure (1CZP, CO-in). The occupancy of the CO-in conforma-

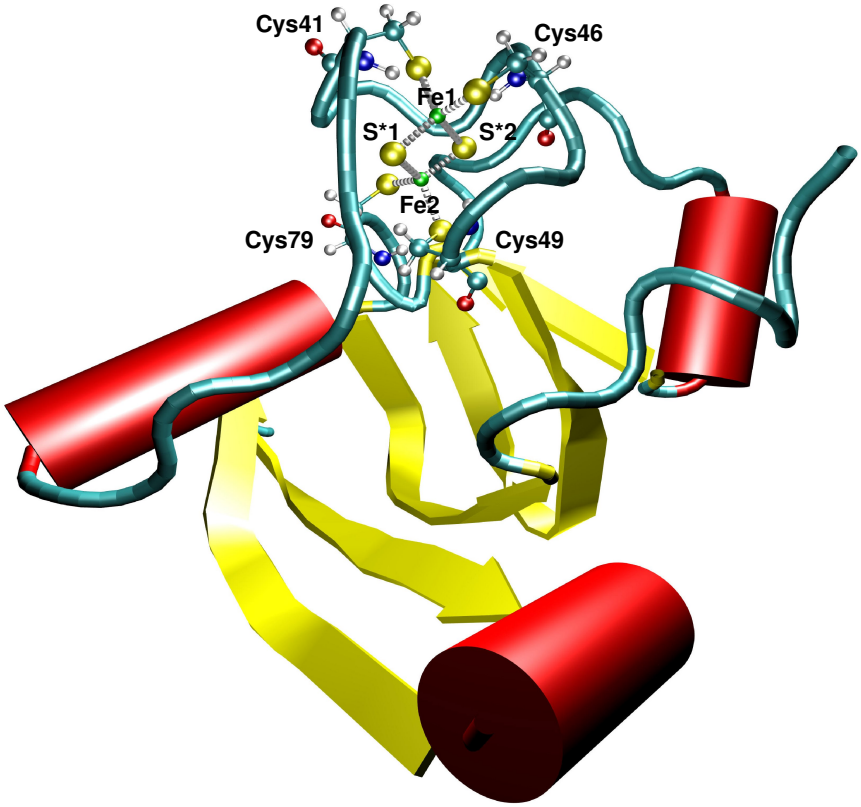


Figure 5.8. Cartoon representation of the structure of ferredoxin. The [2Fe-2S] cluster and the ligating cysteines are shown in ball-and-stick representation.

tion in the two crystallographically independent molecules in the unit cell, mol-1 and mol-2 of 1CZP, was 0.60 and 0.45, respectively. It was assumed by the crystallographers, that the NH-in conformation is characteristic for the reduced state and therefore only 40 to 55 % of the molecules were reduced by soaking with dithionite. The small discrepancy between the two independent molecules in the unit cell was attributed to different crystal packing conditions and accessibility for dithionite. An alternative interpretation could be, that in the oxidized form only the CO-in conformation is energetically accessible, but in the reduced form CO-in and NH-in are about equi-energetic. Thus the structure 1CZP could be fully reduced.

^{57}Fe Mössbauer spectroscopy has shown that the [2Fe-2S] cluster contains two ferric Fe^{3+} in the oxidized form and a localized mixed valence Fe^{2+}/Fe^{3+} in the reduced form [179]. Temperature-dependent proton NMR of the homologous spinach Fdx has identified as Fe^{2+} the iron being the closest one to the molecular surface [180, 181], which is called Fe1 in case of the structure 1CZP.

The reduction potential of Fdx was measured by several groups using different methods leading to discrepancies of about 60 meV. An overview is given in Tab. 5.4. Attempts to calculate reduction potentials were already made by other groups [38, 187], but only based on oxidized crystal structures at lower resolution. Stephens *et al.* [187] calculated the reduction potential

	$\Delta G_{\text{redox}}[\frac{\text{kcal}}{\text{mol}}]$	$\mathcal{E}$ [mV]
measured reduction potentials		
Böhme & Schrautemeier [182]	-92.17 $\pm$ 0.23	-433 $\pm$ 10
Salamon & Tollin [183]	-92.24 $\pm$ 0.46	-430 $\pm$ 20
Hurley *et al.* [184]	-92.01 $\pm$ 0.35	-440 $\pm$ 15
Vidakovic *et al.* [185]	-92.79 $\pm$ 0.23	-406 $\pm$ 10
Vidakovic *et al.* [185]	-92.03 $\pm$ 0.23	-439 $\pm$ 10
Hurley *et al.* [186]	-93.30 $\pm$ 0.07	-384 $\pm$ 3
previous calculations		
Stephens *et al.* [187]	-96.00	-267
Stephens *et al.* [187]	-87.10	-653
Li *et al.* [38]	-80.13	-955

Table 5.4. Previously measured and calculated reduction potentials of ferredoxin

of vegetative Fdx relative to the experimental value of heterocyst Fdx using the PDLD method and the MD-PDLD method. The approach of Li *et al.* [38] is very similar to the one used here, but based on the structure 1FXA [176] at 2.5 Å and using a ionic strength of I = 50 mM.

Pizzitutti *et al.* [188] has performed MD simulations based on the structure 1CZP and using the charges derived by Li *et al.* to calculate free energy profiles of the conformational change. They found, that the activation barrier of the CO-in to NH-in transition is modulated by the distance between the residues Ser47 and Glu94. For short distances the CO-in conformer is favored in the reduced form, whereas for large distances the NH-in conformer is favored. The hydrogen bond between Ser47 (Oγ) and Cys46 (O) was found to be essential to lock in the NH-in conformer. The results were interpreted in terms of their implications on electron transfer. Morales *et al.* [189] calculated the electron delocalization between the iron-sulfur cluster, Ser47, Phe65 and Glu94, proposing that Phe65 acts as intermediate carrier receiving the reducing electron prior to its transfer to FNR.

5.3.2 Different Calculation Approaches

Conciderable effort has been made to synthesize [2Fe-2S] clusters [190]. However, the iron-sulfur complex is very instable in aqueous solution outside the protein environment that large aromatic groups and organic solvents as DMF or acetonitrile are required for stabilization. Due to the aromatic groups most of the synthetic model compounds differ in their electronic structure from the protein and thus the measured reduction potentials between -1.13 and -1.73 V are significantly different to the values in the protein [191, 192]. Since the model

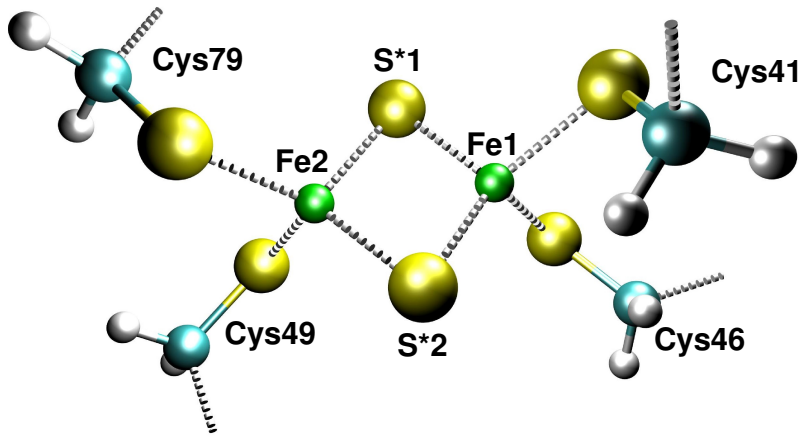


Figure 5.9. The [2Fe-2S] cluster and the ligating cystein are shown in ball-and-stick representation as it was used as small model.

compounds are quite large for treatment by QM methods, they were not considered a good choice for relative reduction potential calculations. Hence, so called absolute calculations based on QM calculations were performed.

The Small Model of the Iron-Sulfur Cluster

The small model is very similar to the one used by Li *et al.* [38]. It includes the [2Fe-2S] center, and the four coordinating histidines up to Cβ (Fig. 5.9). The Cβ atoms were fixed in space. The starting geometry was taken from the PDB file 1CZP of both molecules, mol-1 and mol-2. The two molecules were taken as an internal error estimate for the computational methods applied. Since both structures were determined independently from the same crystal structure, the chemical conditions were identical and only the physical environment due to crystal packing is slightly different. Therefore, very similar energies were expected to be obtained from the different starting geometries. The geometry optimization was done by ADF for the oxidized form and two reduced forms (reduced on Fe1 and Fe2). Since Fe1, S*2 and the sidechain of Cys46 were refined in two alternative positions in the two conformers, in total twelve geometry optimizations were required for each basis set. Here the results of the basis sets TZP and TZ2P(+) are reported. The TZP basis set was also used by Li *et al.* [38], but TZ2P(+) was found to work better as also reported by Szilagyi [90]. Charges were fit to the atomic coordinates by the CHELPG implementations of Mouesca *et al.* [37] and Behera *et al.* [94] as well as the MDC method as implemented by Swart *et al.* [95] and Thomas Ullmann (unpublished). I limit the discussion to the results obtained with the program of Mouesca, since for the large model also no clear improvement was found by the other charge fitting methods.

In Tab. 5.5 the relative QM energies of the small model after geometry optimization are given. Despite the high symmetry of the model, the energies of the cluster reduced on Fe1 tend to be lower with the TZP basis set. For the CO-in conformer of molecule mol-2, however, the energy

1CZP	mol-1			mol-2		
	ox	red Fe1	red Fe2	ox	red Fe1	red Fe2
CO-in TZP	0.0	114.0	114.4	0.0	114.2	113.9
NH-in TZP	0.0	113.6	114.7	0.0	114.1	115.3
CO-in TZ2P(+)	0.0	113.7	113.7	0.0	114.9	114.9
NH-in TZ2P(+)	0.0	114.2	114.2	0.0	114.6	115.9

Table 5.5. Quantum chemical energies (in $\frac{\text{kcal}}{\text{mol}}$) relative to the oxidized form of the small model of the iron-sulfur cluster relative to the oxidized form.

for reducing on Fe2 is lower. Using the higher basis set TZ2P(+), usually the same energy is observed for reducing both irons as one would expect for a nearly symmetric molecule. The exception of the 1.3 $\frac{\text{kcal}}{\text{mol}}$ higher energy of the NH-in conformer of molecule mol-2 might be, because the system is convergence to a higher energy minimum, not the global minimum. Generally, the results using the higher basis set seem to be more stable and therefore more reliable.

The reduction potentials in the protein were calculated with Perl Molecule and QMPB. All glutamate, aspartate, histidine, arginine, lysine and tyrosine were allowed to titrate with standard $\text{p}K_{a,\text{model}}$ values [121]. The dielectric constants were set to one for the QM region, 4 for the rest of the protein and 80 for water. The ionic strength was $\text{I} = 100\text{mM}$. The reduction potentials calculated for the small center (Tab. 5.6) are generally much lower than the experimental results, but slightly closer to experiment than the results of Li *et al.* (Tab. 5.4). The reduction potentials calculated with the TZ2P(+) basis set are lower than their counterparts calculated with the TZP basis set. The NH-in conformation is always favorable over the CO-in conformation.

In summary, the small model is not suited for calculating accurate reduction potentials. It is assumed, that the different electronic polarization of the two irons due to the different protein environment, which is not included in this model, is the reason for the discrepancy to the experiment. The charge model is too symmetric to be a good description for a system with localized $\text{Fe}^{2+}/\text{Fe}^{3+}$ as observed by Mössbauer spectroscopy. Therefore, a larger model was made, including the first layer of hydrogen bonds to the iron-sulfur cluster.

The Large Model of the Iron-Sulfur Cluster

The distribution of hydrogen bonds is highly asymmetric for the sulfurs coordinating the two iron. The sulfurs of Cys41 and Cys46 as well as the bridging sulfur S*1 are stabilized by two strong hydrogen bonds. The bridging sulfur S*2 has one permanent strong hydrogen bond and an additional strong hydrogen bond in the NH-in structure. The sulfurs of Cys49 and Cys79 coordinating Fe2 are only stabilized by one strong hydrogen bond each. The potential hydrogen bonds of the sidechains of Thr48 and Thr78 are considered as weak, because other rotamers of the hydroxyl group are possible and tend to be more likely. Therefore, already by

1CZP	$\mathcal{E}$	ΔH_{vac}	$-\Delta G_{corr}$	$\Delta\Delta G_{Born}$	$\Delta\Delta G_{back}$	$\langle\Delta G_{inter}\rangle$	ΔG_{redox}
TZP							
mol-1 CO-in	-850	114.0	1.8	-180.5	-18.3	0.4	-82.6
mol-1 NH-in	-662	113.6	1.9	-181.4	-21.2	0.2	-86.9
mol-2 CO-in	-818	114.2	1.6	-180.8	-18.5	0.2	-83.3
mol-2 NH-in	-657	114.1	1.8	-181.7	-21.8	0.6	-87.0
TZ2P(+)							
mol-1 CO-in	-870	114.7	1.7	-180.9	-18.3	0.7	-82.1
mol-1 NH-in	-683	114.2	1.8	-181.2	-21.0	-0.2	-86.4
mol-2 CO-in	-887	114.9	1.5	-180.6	-18.2	0.7	-81.7
mol-2 NH-in	-701	114.6	1.9	-181.5	-21.5	0.5	-86.0

Table 5.6. Reduction potential $\mathcal{E}$ at pH 7.0 (in mV) and energy contributions (in $\frac{\text{kcal}}{\text{mol}}$) for the small model of the iron-sulfur cluster inside the protein. The energy contributions contain only the oxidized and on Fe1 reduced structures. The energy terms are explained in section 3.3. ΔG_{redox} is the sum of the energy contributions and $\mathcal{E}$ is calculated from this value.

counting the number of hydrogen bonds, it can be predicted that the reduction of Fe1 should be favored over the reduction of Fe2, because the negative charge of the additional electron is better stabilized on the sulfurs coordinating Fe1.

Smaller models do not fully describe this sophisticated hydrogen bond network and can therefore not describe the asymmetric environment in the charge model (*e.g.,* [38] or our small model). Morales *et al.* [189] used an asymmetric model, but it is imbalanced by only including some hydrogen bond partners on the Fe1 side. Especially at the beginning of the project, the disadvantage of the 93-atom model shown in Fig. 5.10 was clearly the enormous computational time required for geometry optimization. In 2002 and 2003, the calculations were run with the TZP basis set on the HELICS I cluster of the IWR of the University of Heidelberg (June 2002 the 35th fastest computer in the world[1]). The calculations took months and sometimes used over 300 of the 512 AMD Athlon 1.4 GHz CPUs [2]. Due to the increase in processor speed, better convergence with the higher basis set and improvements in the ADF code, the calculations using the higher TZ2P(+) basis set were feasible within weeks on the 96 AMD Opteron 2.4 GHz processor compute cluster of our group in Bayreuth.

In Tab. 5.7 the relative QM energies of the large model after geometry optimization are given. The energies for the reduced molecule relative to the oxidized molecule are lower compared to the small model, because the additional electron is stabilized by the protein environment included in the larger model. Due to the different environment of the two iron atoms, the reduction of Fe1 is always favored over the reduction on Fe2. However, for the lower basis set

[1] http://www.top500.org/list/2002/06/100

[2] The processor time granted by the IWR and support of the HELICS team is kindly acknowledged.

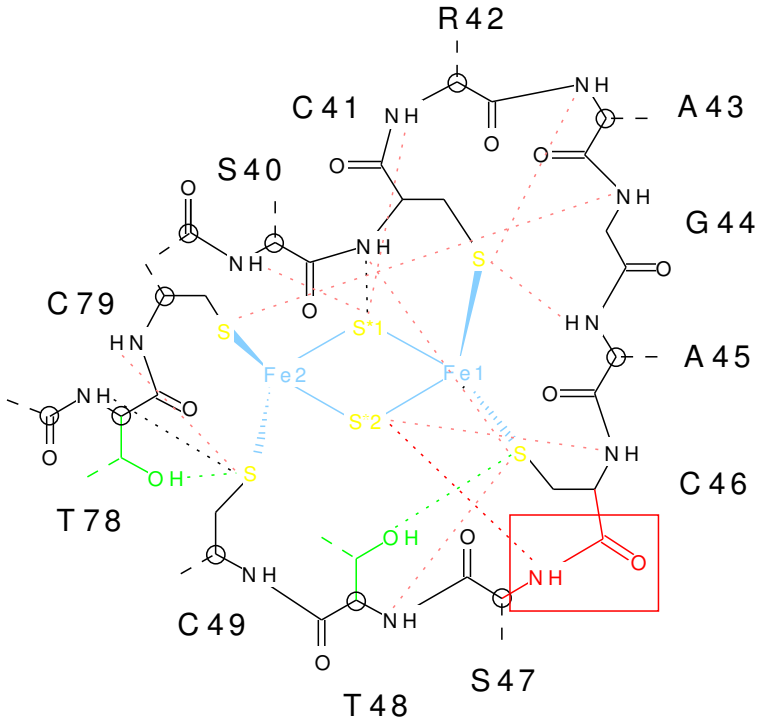


Figure 5.10. Schematic representation of the large model containing the [2Fe-2S] center (in blue and yellow) and all backbone atoms of the two loops forming hydrogen bonds with the iron-sulfur cluster (black). Weak hydrogen bonds are dashed in black and strong hydrogen bonds are dashed in pink. The peptide bond between Cys46 and Ser47, which was observed in two conformations, is shown in red (including the additional hydrogen bond in the NH-in conformation). The sidechains of Thr48 and Thr78 (in green) can form hydrogen bonds with the iron-sulfur cluster, but they were omitted from the QM calculation to allow for rotameric freedom in the continuum electrostatic computations. Rotameric forms which form hydrogen bonds with residues outside the large model are possible and maybe more likely. The atoms marked by a circle were fixed in space during geometry optimization to be able to fit the optimized structure back into the protein.

TZP, the results fluctuate more than for the higher basis set TZ2P(+), so that the maximum difference between equivalent forms in molecule mol-1 and mol-2 is 1.6 $\frac{\text{kcal}}{\text{mol}}$ for TZP, but only 0.5 $\frac{\text{kcal}}{\text{mol}}$ for TZ2P(+). According to the QM energies, reduction of the NH-in conformer requires about 3 $\frac{\text{kcal}}{\text{mol}}$ less energy than reducing the CO-in conformer.

The calculated reduction potentials based on the large model (Tab. 5.8) are much closer to the experimental results than for the small model, however, the reduction potentials are slightly too high. Again, the results for the higher basis set TZ2P(+) are better than the results for TZP compared to experiment and also in terms of consistency between mol-1 and mol-2. Still the energy for the NH-in conformer is lower than the CO-in conformer, but the calculated reduction potentials for CO-in are much closer to the experimentally determined reduction potentials.

1CZP	mol-1			mol-2		
	ox	red Fe1	red Fe2	ox	red Fe1	red Fe2
CO-in TZP	0.0	32.6	35.1	0.0	32.3	34.1
NH-in TZP	0.0	30.2	31.1	0.0	30.2	32.7
CO-in TZ2P(+)	0.0	33.5	36.9	0.0	33.7	37.4
NH-in TZ2P(+)	0.0	30.4	33.3	0.0	30.6	33.5

Table 5.7. Quantum chemical energies (in $\frac{\text{kcal}}{\text{mol}}$) relative to the oxidized form of the large model of the iron-sulfur cluster relative to the oxidized form.

In my calculations I observed, that the reduction potential is strongly influenced by the protonation probability of Glu94. This residue, shows an unusual protonation behavior and the protonation probability curve is largely shifted compared to solution. At pH 7.0 Glu94 has a significant probability of being protonated ($\text{p}K_{\text{app}}$ of 7.4, 9.1, 6.3 and 6.9 in mol-1 CO-in, mol-1 NH-in, mol-2 CO-in and mol-2 NH-in respectively). Because the crystal structure 1CZP (as all *Anabaena* Fdx structures) was determined at a low pH of 5.5, where glutamates have a significant probability of being protonated, it was taken into consideration, that the protonation of Glu94 could be an artifact of the structure. In fact, several acidic residues show increased calculated $\text{p}K_{\text{app}}$ values.

The simplest approach was to fix Glu94 in the deprotonated form during the electrostatic calculations. The results in Tab. 5.8 show clearly lower reduction potentials in better agreement with experimental results at pH 7.0. For the CO-in conformer, nearly perfect agreement with the results of Hurley *et al.* [186] could be obtained. The small remaining differences could be attributed either to other acidic groups, which are unrealistically stabilized in the protonated form by the protein structure, but interact less with the iron-sulfur center, or to other small inaccuracies in the calculation and reduction potential measurement.

A significant effort has been made to find an alternative set of sidechain rotamers in Fdx, which stabilizes Glu94 in its deprotonated form. Therefore, the Dunbrack rotamer library as implemented in Perl Molecule was used. Rotating Glu94 by about -120° to increase the distance between the negatively charges [2Fe-2S] cluster and the carboxy group did not change the protonation probability sufficiently. Only extensive modeling rotating the negatively charged Glu95 out of the vicinity of the carboxy group of Glu94 and rotating His92 into hydrogen bond distance with Glu94 lead to a sufficient stabilization of the deprotonated form of Glu94. By that, some positive electron density in the geometric shape of a carboxyl group could be used, which was not fitted when determining the structure 1CZP. Even the electron density of the new rotamer is much less than for the rotamer modeled by the crystallographers, it might be an indication for an additional rotamer, which is only little occupied under the low-pH conditions for which the structure was determined. However, such a rotamer might be dominating at higher pH.

1CZP	$\mathcal{E}$	ΔH_{vac}	$-\Delta G_{corr}$	$\Delta\Delta G_{Born}$	$\Delta\Delta G_{back}$	$\langle\Delta G_{inter}\rangle$	ΔG_{redox}
TZP							
mol-1 CO-in	-323	32.6	-2.0	-121.1	-5.2	1.0	-94.7
mol-1 NH-in	-198	30.2	-3.4	-119.5	-5.2	0.3	-97.6
mol-2 CO-in	-245	32.3	-2.0	-121.5	-6.2	0.9	-96.5
mol-2 NH-in	-241	30.2	-2.3	-120.3	-5.3	1.1	-96.6
TZ2P(+)							
mol-1 CO-in	-354	33.5	-2.1	-121.1	-5.4	1.1	-94.0
mol-1 NH-in	-241	30.4	-2.0	-120.5	-5.2	0.7	-96.6
mol-2 CO-in	-362	33.7	-2.2	-120.8	-5.8	1.3	-93.8
mol-2 NH-in	-284	30.6	-1.9	-120.2	-5.4	1.3	-95.6
Glu94 fixed to be deprotonated, TZ2P(+)							
mol-1 CO-in	-380	33.5	-2.1	-121.1	-5.4	1.7	-93.4
mol-1 NH-in	-289	30.4	-2.0	-120.5	-5.2	1.8	-95.5
mol-2 CO-in	-371	33.7	-2.2	-120.8	-5.8	1.5	-93.6
mol-2 NH-in	-289	30.6	-1.9	-120.2	-5.4	1.4	-95.5

Table 5.8. Reduction potential $\mathcal{E}$ at pH 7.0 (in mV) and energy contributions (in $\frac{\text{kcal}}{\text{mol}}$) for the large model of the iron-sulfur cluster inside the protein. The energy contributions contain only the oxidized and on Fe1 reduced structures. The energy terms are explained in section 3.3. ΔG_{redox} is the sum of the energy contributions and $\mathcal{E}$ is calculated from this value.

It has to be noted, that the energy of the NH-in conformer is much lower than the energy of the CO-in conformer. Not only that the difference in QM energy of reduced and oxidized Fdx is about 3 $\frac{\text{kcal}}{\text{mol}}$ lower for NH-in, but also the absolute QM energy of the oxidized state is 8.2 $\frac{\text{kcal}}{\text{mol}}$ and 7.0 $\frac{\text{kcal}}{\text{mol}}$ lower for NH-in in molecule mol-1 and mol-2, respectively. The QM calculations seem to greatly over-estimate the energy of this additional hydrogen bond by about 10 $\frac{\text{kcal}}{\text{mol}}$.

Therefore, the qualitative conclusion can be drawn, that it is possible to construct a set of rotamers, which stabilize a deprotonated Glu94 at neutral pH and this structure agrees well with experimentally measured reduction potentials. The calculated absolute reduction potentials are in much better agreement with experimental results than any other previous absolute or relative calculation attempts. The asymmetric reduction behavior of the two iron atoms was calculated in agreement with Mössbauer and NMR spectroscopic data.

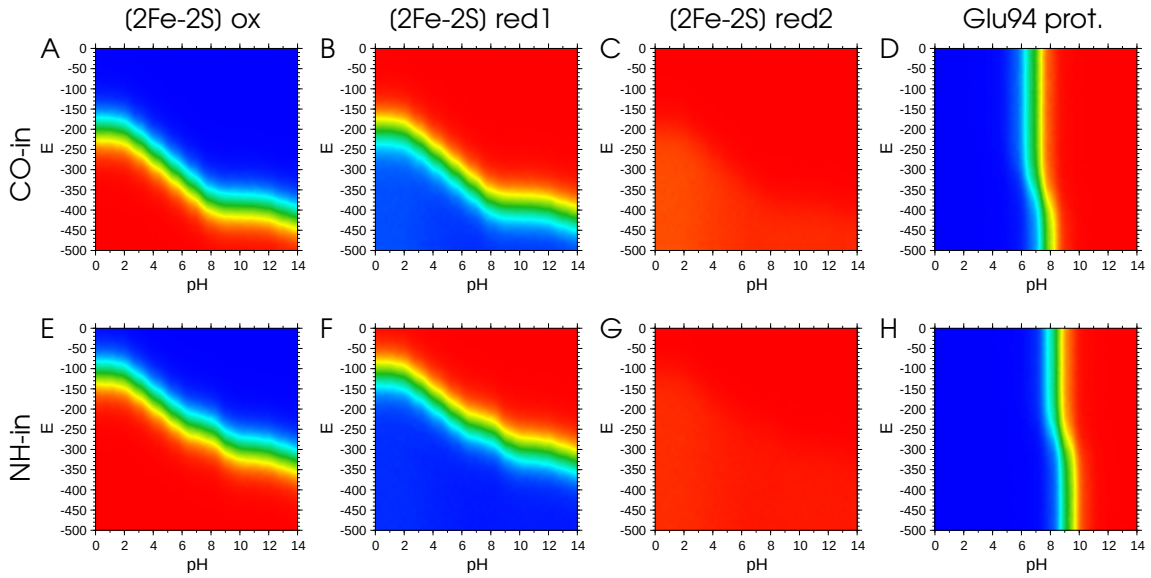


Figure 5.11. Reduction and protonation probability plots of the iron-sulfur center (large model, TZ2P(+)) and Glu94 in conformer CO-in (A - D) and conformer NH-in (E - H) of molecule mol-1 of structure 1CZP. The probability of the iron-sulfur center being oxidized (A and E), being reduced on Fe1 (B and F) and being reduced on Fe2 (C and G) is shown. Plots D and H give the probability of Glu94 being protonated. The color coding of the probability $\langle x \rangle$ is identical to Fig. 5.6.

5.3.3 Advantages of Using Perl Molecule and QMPB

The examples in the first two sections of this chapter could have been done with Multiflex as well, but already in the application calculating the $\mathrm{p}K_a$ value of His15 in HPr the capability of QMPB to deal with more than two instances was of advantage. The interpretation of the results was much more straight forward, than the procedures required for Multiflex (described in section 3.5.1). For Fdx absolute reduction potential calculations were required, for which no single program existed before (section 3.5.2). Even Multiflex3D, which allows the three dielectric regions required for this model, could not cope with the three instances required to describe the [2Fe-2S] center in one oxidized and two reduced forms. A consistent treatment of reduction and protonation events would have only been possible with major manual intervention using Multiflex3D. For systems like Fdx, the study of the dependence of the reduction or protonation probability dependent on more than one chemical potential is important, as it can be seen in Fig. 5.11. The reduction potential of the [2Fe-2S] center depends significantly on the protonation probability of other residues like Glu94 and *vice versa*.

The structure preparation of 1CZP with Perl Molecule was discussed in detail in section 4.6.4. With any other method the complex modeling task splitting the structure in four conformers and generating all the rotamer forms and charge forms would have been much more complicated and error prone. As discussed before, the method `add_hydrogen` adds several hydrogen rotamers for residues like serine or threonine, one for each possible hydrogen bonding partner. Rotamer energies were assigned by the method `setup_rotamer_sites` according to a file `charmm.potentials`, which in turn contains torsion potential files for each relevant torsion angle (appendix A.4.3). The torsion potential files were generated by incrementally changing the torsion potential and calculating the energy with CHARMM.

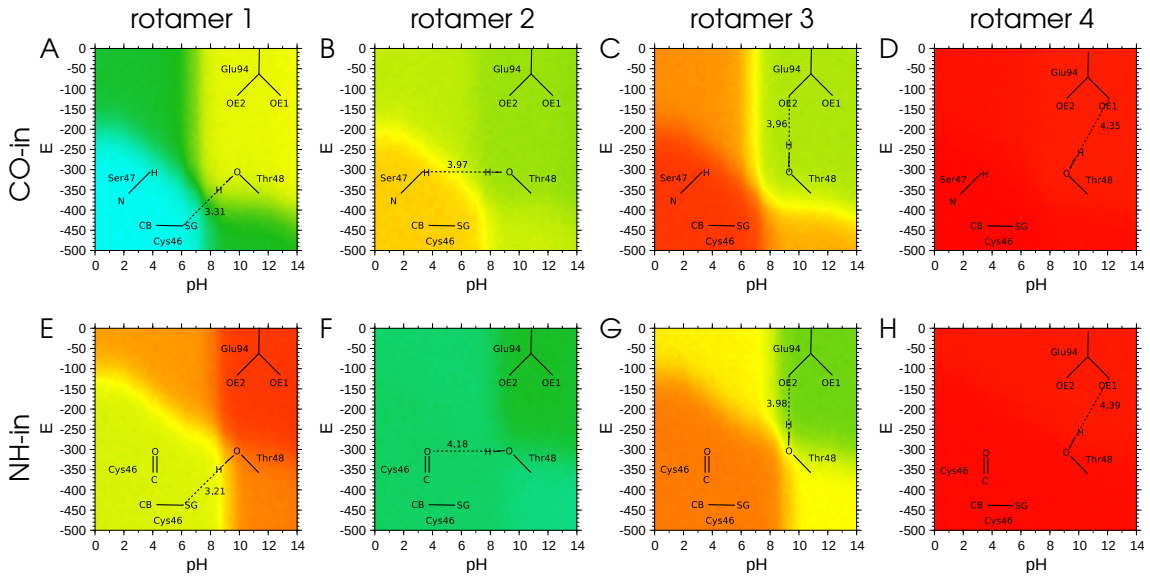


Figure 5.12. Probability plots of the four hydrogen rotamers of Thr48 in the CO-in conformer (A-D) and NH-in conformer (E-H) of molecule mol-1 of structure 1CZP. The color coding of the probability $\langle x \rangle$ is identical to Fig. 5.6. A schematic drawing of the hydrogen bond formed by each rotamer is laid on top of the probability plot. The distance in Å between Thr48 (Oγ) and the acceptor atom is given.

The importance of including several rotamers is shown by probability plots of instances of Thr48 in Fig. 5.12. The residue is described by four rotamer forms, but only three are populated. In the fourth rotamer, the Thr48 (Hγ1) would point in the direction of Glu94 (Oϵ1), which is significantly further away than the other potential (weak) hydrogen bond acceptors (Fig. 5.12 D and H). In the CO-in conformer, if the iron-sulfur center is reduced (Fig. 5.11 B) and Glu94 is protonated (Fig. 5.11 D), the rotamer of Thr48 forming a hydrogen bond with Cys46 (Sγ) is most likely (cyan area in Fig. 5.12 A, about 75%). If the iron-sulfur center is oxidized and Glu94 is deprotonated, the probability of this instance is only 25-30 % (yellow area). Instead the rotamer forming a (weak) hydrogen bond with Glu94 (Oϵ2) gets more likely (30-35 %, Fig. 5.12 C). Surprisingly, a rotamer between the two hydrogen bonds has over wide areas of the pH and reduction potential space a probability of 30 - 40 % (Fig. 5.12 B). The rotamer was included, because it describes the interaction with Cys46 (O) in the NH-in conformer, but in case of the CO-in conformer Thr48 (Hγ1) is pointing in the direction of Ser47 (HN). The overall doubly negative charge of the quantum center seems to over-compensate the repulsion of the positive partial charges on the two hydrogen atoms.

In case of the NH-in conformer, the second rotamer, in which Thr48 (Hγ1) is interacting with Cys46 (O), is overall the most populated instance (Fig. 5.12 F, $\geq$ 50%). If the iron-sulfur center is reduced (Fig. 5.11 F) and Glu94 is protonated (Fig. 5.11 H), the rotamer forming a hydrogen bond with Cys46 (Sγ) is also significantly populated (green-yellow area in Fig. 5.12 E, about 40%) and instead, if the iron-sulfur center is reduced and Glu94 is protonated, the rotamer forming the (weak) hydrogen bond with Glu94 (Oϵ2) gets more likely (about 50 %, Fig. 5.12 H). Inclusion of only a single rotamer form would bias the iron-sulfur center and Glu94 towards a particular instance and therefore shift the reduction and protonation probability, respectively.

Rotamers of sidechains can be generated by Perl Molecule using a rotamer database (section 2.4.2). The file format of the Dunbrack backbone dependent and backbone independent rotamer database (appendix A.5.3) can be read by the class `dunbrack` (section 4.6.3) using the method `molecule::read_rotamers`. The method `molecule::generate_rotamers` generates all rotamers in the database for a particular residue of a chain given as input parameter. The method `molecule::remove_clashing_rotamers` removes rotamers, which are too close to atoms of the background set or non-rotamer sites. A parameter can be given to scale the radii, which was thought to be useful to reduce the number of rejected rotamers. Some rotamers might be valid, if *e.g.,* smaller hydrogen radii as in force fields would be used or if some relaxation of the neighborhood of the new rotamer would be included in the protocol. Alternatively, only particular rotamers can be constructed by giving a hash of rotamer classification strings as third parameter for `molecule::generate_rotamers`. As energy for the rotamers, the rotamer probability stored in the database can be converted by Perl Molecule, or all energies are calculated separately and assigned to the rotamers in the QMPB input file. The rotamer energies derived from probabilities turned out not to be useful. Currently, the support for rotamers in Perl Molecule is limited, because the CHARMM energy function is not implemented to obtain the rotamer energies easily. Also a program calculating Lennard-Jones terms, *i.e.,* an interaction energy matrix of different rotamers is lacking. However, including this terms into the energy function would be very simple. For my more qualitative study on possible alternative rotamers and their effect on the reduction potential this improved energy function was not required.

5.3.4 Conclusions

By treating a larger part of the protein by QM methods, a higher basis set and a crystal structure of higher resolution, absolute reduction potentials for *Anabaena* Fdx could be calculated, which are in much better agreement with measured values than previous results. Also for the first time the asymmetric environment of the two iron atoms was included sufficiently into calculations leading to two energetically well separated reduced forms. The results are in agreement with the Mössbauer data on a localized mixed valence Fe^{2+}/Fe^{3+} state. Also the reduced iron Fe1 is in agreement with NMR data. However, obtaining accurate energies by QM calculations still remains a challenge. It seems that todays DFT functionals are not accurate enough for the purpose of this work. An additional problem is to derive accurate point charges form the electron density calculated by the QM method. This problem has no unique solution and different approaches find considerably different results. Certainly, the electrostatic calculations critically depend on the set of point charges used.

Another problem is that reduction potentials are usually measured at pH 7, but no crystal structure exists at this pH. The used crystal structure was determined at a pH of 5.5, at which acidic groups are likely to be protonated (especially in such an acidic protein as Fdx and with a negatively charge iron-sulfur center). Under such conditions, groups of acidic residues arrange into orientations stabilizing a bound proton and by that leading to an increased $\mathrm{p}K_a$ value compared to the groups separated in solution. Similarly, Glu94 adopts an orientation close to the negatively charged iron-sulfur cluster stabilizing its protonated form. Probably the residue would adopt a different orientation at higher pH. It seems that such pH dependent conformational changes are important in Fdx since a structure could be modeled,

which stabilizes a deprotonated Glu94 at neutral pH leading to very good agreement with experimental reduction potentials.

Due to the largely over-estimated favorable energy for the NH-in conformer by the QM method, it can only be speculated about the potential role of the conformational change in the peptide bond. The interpretation of Morales *et al.*, that the CO-in conformer is only populated in the oxidized form and the NH-in conformer is only populated in the reduced form seems unlikely at present. The calculations on the CO-in conformer show that a reduction potential close to experiment can be obtained for this conformation only. The energy difference between the CO-in and the NH-in conformer is currently by far too large to obtain a reduction potential in the order of magnitude of the measured values. Shifting the energy of the NH-in conformer by about 10 $\frac{\mathrm{kcal}}{\mathrm{mol}}$, would lead to results only showing the CO-in conformer in the oxidized form in agreement with experiment. The oxidized form of the NH-in conformer would be 2-3 $\frac{\mathrm{kcal}}{\mathrm{mol}}$ higher in energy and therefore not be significantly populated. For a lower energy shift, the CO-in and NH-in conformer would become equi-energetic and should have both been observed in the oxidized form or for energy shifts lower than, *e.g.,* 5 $\frac{\mathrm{kcal}}{\mathrm{mol}}$ only the NH-in conformer should have been observed in the oxidized form. For an assumed shift of 10 $\frac{\mathrm{kcal}}{\mathrm{mol}}$, the energy of the reduced NH-in conformer is about 1 $\frac{\mathrm{kcal}}{\mathrm{mol}}$ lower than the CO-in conformer. By that, both conformers would be populated as the crystallographic data on the reduced crystal could be interpreted. For higher energy shift values, again only the CO-in conformer would be populated also in the reduced form. By this thermodynamic interpretation, in which the two conformers are about equi-energetic in the reduced form and only the CO-in conformer is populated in the oxidized form, a direct role of the conformational change for the reduction potential could be ruled out. However, the conformational change may still influence the kinetics of electron transfer by lowering the barrier. The NH-in conformation seems to provide a more direct route for transferring electrons to FNR.

5.4 Protonation Probability Calculations of $\mathrm{Cu_B}$ Ligands in the Reaction Mechanism of Cytochrome *c* Oxidase

The reaction mechanism of Cytochrome *c* oxidase (CcO) is studied by Punnagai Munusami in our group. The complex protonation equilibria in the active site are studied using Perl Molecule and QMPB. Our work is described in [60].

Cytochrome *c* oxidase is the terminal enzyme in the respiratory chain. It reduces oxygen to water by consuming four protons. The oxygen reduction drives pumping of four additional protons across the membrane [193–195]. However, details of the reaction mechanism are not well understood and under debate [58, 59, 196–198].

The enzyme contains several metal prosthetic sites and 13 subunits in mammals. The $\mathrm{Cu_A}$ center accepts electrons from Cytochrome *c* and transfers them to heme *a* and finally to the heme a_3 - $\mathrm{Cu_B}$ binuclear center, where oxygen gets reduced. The $\mathrm{Cu_B}$ center consists of a copper ion coordinated by three histidines. At the fourth position the oxygen, which gets reduced to water, is coordinated (Fig. 5.13). One of the coordinating histidines, His240, is cross-linked with Tyr244.

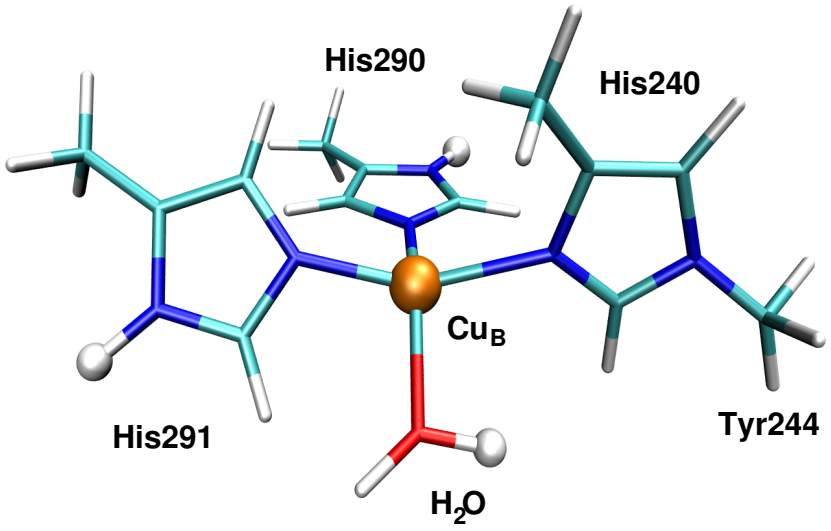


Figure 5.13. The model of the Cu_B center calculated by DFT. The $C\beta$ atoms of the three histidines were fixed in space at there crystallographic position (PDB code 2OCC [199]). The atoms marked by white spheres were considered for deprotonation. Tyr244 is modeled as a methyl group.

Popović *et al.* suggested that His291 deprotonates depending on the redox state of the Cu_B center playing a central role in proton pumping [58, 59]. They found by DFT calculations the pK_a value of His291 to be 8.6 and 13.2 in oxidized and reduced models of the Cu_B center in aqueous solution, respectively. Inside the protein pK_a values of 4.0 and 19.7 were calculated. In contrast, Fadda *et al.* [196] obtained a pK_a value of above 13.5 for His291 in the oxidized state, making a redox-coupled proton pumping involving this residue unlikely. The discrepancies led to some debate [197, 198] and weaknesses could be found in both studies: The structures were not fully relaxed in both studies leading to unrealistic bond lengths. Fadda *et al.* assumed standard protonation state for all amino acids in the protein. Both works also studied only the deprotonation of His291, but did not consider the other microstates of the Cu_B center to check for internal consistency.

We calculated all microscopic pK_a values of the Cu_B center (Fig. 5.13) using DFT methods (Gaussian 03, B3LYP/6-31G*) and a conductor like polarizeable continuum model (C-PCM) [200–202] fixing only the $C\beta$ carbons to their crystallographic positions. All titrateable amino acids in the protein were allowed to adjust to the chemical potential and their environment. The eight microstates of the Cu_B center are shown together with the twelve microscopic pK_a values in Fig. 5.14. We obtained pK_a values of 15.9 and 15.2 for the first deprotonation reactions of His290 and His291, respectively, in the oxidized state of the Cu_B center in aqueous solution. Obviously, the pK_a values are too high to allow deprotonation at physiological pH and the values are even higher in the reduced state. According to the QMPB calculations, also in the protein environment only the state is populated at pH 7, where His290, His291 and the water molecule are protonated. Interestingly, however, the pK_a value for deprotonating

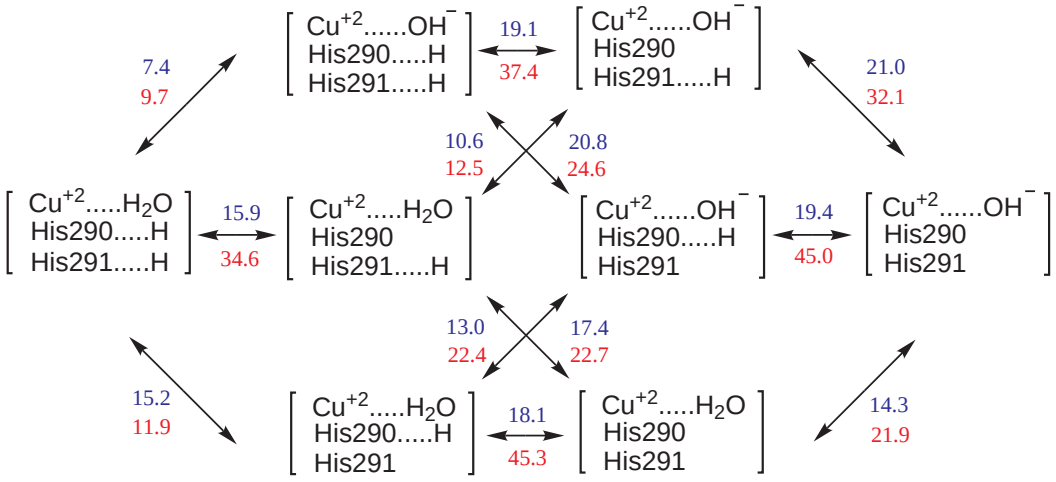


Figure 5.14. Microstates of the $\mathrm{Cu_B}$ center and microscopic $\mathrm{p}K_a$ values in aqueous solution (blue) and in the protein with aquo-ferrous heme a_3 (red).

His291 in the fully protonated state is the only $\mathrm{p}K_a$ value, which drops by about 3 units, while all other $\mathrm{p}K_a$ values increase upon transfer from aqueous solution into the protein. Albeit the value is still too high to be deprotonated, it shows a tendency of the protein to lower the $\mathrm{p}K_a$ value specifically. A group with even higher $\mathrm{p}K_a$ value could abstract the proton form His291. Tryptophane was suggested by us to fulfill this function, because it was recently discovered to be involved in CcO reactions [203–206]. A possible mechanism could be, that a deprotonated tryptophane radical is formed upon O-O bond cleavage. After rereduction, the tryptophane with a solution $\mathrm{p}K_a$ value of about 17 could protonate by abstracting a proton from His291. This model would allow a proton pumping mechanism as suggested by Popović *et al.* [58], but only in presence of a very strong base.

5.5 Reduction and Protonation Reactions of Quinones

Coenzyme Q plays a central role in a lot of biochemical processes, *e.g.,* as electron and proton carrier in photosynthesis, the mitochondrial respiratory chain, lysosomes, regulation of physicochemical membrane properties, as antioxidant and to regenerate other antioxidants, in aging to stimulate cell growth and inhibit cell death [207–209]. Due to the large number of different reduction and protonation states quinones are an ideal example of sites with many instances binding two different ligand types.

Ubiquinone is studied as a part of the bacterial reaction center by Eva-Maria Krammer in our group. However, ubiquinone has two flexible methoxy groups and a long (3-10 monomer) isoprenoid chain as substituents (Fig. 5.15 B). Therefore we chose the related organic molecule *p*-benzoquinone (Fig. 5.15 A) for quantum mechanical studies to determine an optimal method to compute protonation and reduction equilibrium constants.

Due to the yellow color of *p*-benzoquinone and the colorless benzoquinol the reduction state can be easily monitored by UV/VIS absorption spectroscopy. However, its poor solubility in

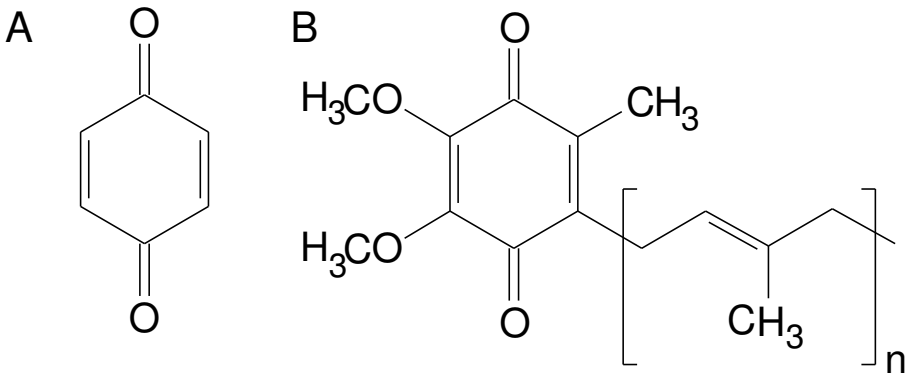


Figure 5.15. Different quinone types. A: Benzoquinone, B: Ubiquinone ($n = 3 \ldots 10$)

water makes measurements difficult in the solvent most important for biological reactions. The solubility in ethanol is significantly larger, thus most experiments were done in some alcohol containing aqueous solutions. All experiments were done in solutions of neutral or basic pH, so we concentrated on the neutral or negatively charged instances of benzoquinone shown in Fig. 5.16. In the beginning the literature base of reviews and book articles seemed large, but most articles reference only a very few independent measurements. The reduction potential of the overall reaction from benzoquinone to benzoquinole (reaction r3 in Fig. 5.16) could be measured by Conant and Fieser to be 0.699 V [210–212]. Baxendale and Hardy [213] determined the $\mathrm{p}K_a$ values of reaction p2 and p3 to be 12.1 and 14.6, respectively. Bishop and Tong [214] discovered a quinone-hydroxide adduct formation at high pH and corrected the $\mathrm{p}K_a$ values to be 9.8 and 11.4, respectively. Further more, the authors determined the semiquinone formation constant K_4 for the reaction

$$BQ + BQ^{2-} \rightleftharpoons 2BQ^{-} \tag{5.4}$$

to be 4.2. Willson determined the $\mathrm{p}K_a$ value of reaction p1 to be 4.1 by pulse radiolysis [215]. For the other reactions no reliable measurements could be found, so that we relied on the calculations of Ilan *et al.* [216] based on the cited measurements.

We tried a spectrum of basis sets and charge fit methods: The best performing method of the Fdx study was used, which is geometry optimization with ADF [87–89], the TZ2P(+) basis set and VWN [81] and PW [84, 85] exchange correlation functions. Charges were fit by the CHELPG implementation of Mouesca *et al.* [37]. Additionally, Gaussian [134] was used with three different basis sets (6-31+G*, 6-31++G**, 6-311+G*, section 2.3.1) and the B3LYP [10, 217] functional. Charges were fit by a number of different methods: Mulliken charges are based on a Mulliken population analysis [218–221]; the CHELPG method [93] is briefly described in section 2.3.2; the natural population analysis (NPA, [222–226]) is based on natural bond orbitals; Merz-Kollman charges [227, 228]; the conductor like PCM (C-PCM, [200, 202]) method, the PCM [20, 229–231] method and an integral equation formalism PCM (IEF-PCM, [20, 230, 232–235]) method. The results of the last three methods were found to be nearly identical, such that only the C-PCM results are reported. Using the different charge fits, sol-

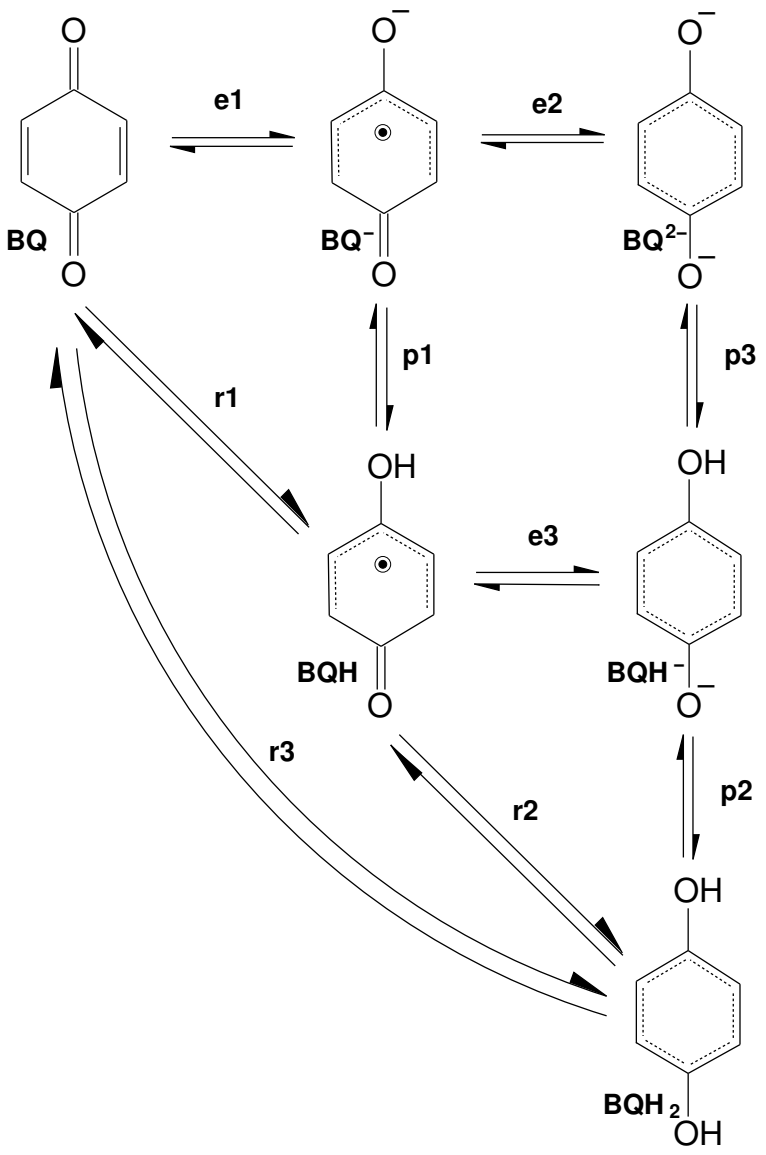


Figure 5.16. Reduction (horizontal) and protonation (vertical) reactions of benzoquinone (BQ) to benzoquinole (BQH_2). The intermediates are deprotonated benzosemiquinone (BQ^-), protonated benzosemiquinone (BQH), singly deprotonated benzoquinole (BQH^-) and doubly deprotonated benzoquinole (BQ^{2-}). The reduction reactions are labeled e1 to e3 and the protonation reactions p1 to p3. Reactions r1 and r2 describe the uptake of a hydrogen radical, $H^{\cdot}$, and r3 the formal uptake of H_2. The positively charged reaction intermediates were excluded from the study, because no experimental or theoretical data could be found for comparison.

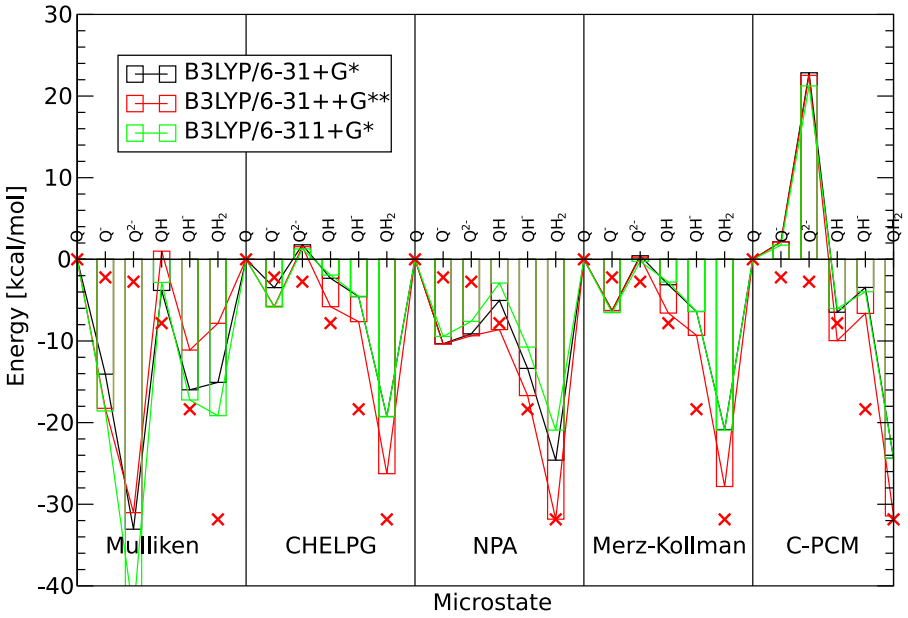


Figure 5.17. The energies (in $\frac{\text{kcal}}{\text{mol}}$) of the six microstates of BQ (shown in Fig. 5.16) are plotted for three basis sets and five charge fitting methods relative to the BQ microstate. The experimental data (or data calculated based on experimental data) is marked by red crosses. The bars indicating the energy of the microstates are connected by lines, because the height of the bar would be hard to see otherwise, especially for the basis sets B3LYP/6-31+G* and B3LYP/6-311+G*.

vation energies were calculated using Solvate from the MEAD suite of programs, except for the PCM methods were solvation energies were taken directly form the Gaussian output.

The results for the microstate energies strongly depend on the charge fit method used. The worst results were obtained with Mulliken and C-PCM charges, which lead to a difference in the result of over 60 $\frac{\text{kcal}}{\text{mol}}$ for BQ^{2-} (Fig. 5.17). The difference due to different basis sets is not that large, but discrepancies of over 5 $\frac{\text{kcal}}{\text{mol}}$ are frequent. The triple split valence basis 6-311+G* allows for more flexibility in the variation of PGTOs, than 6-31+G* and therefore should perform better. However, in all instances it performes equal or worse than the double split valence basis set. A big improvement is gained from including polarization of the hydrogen atoms and p-type diffusion functions. It seams reasonable, that for a molecule formed by s and sp2 orbitals, an accurate description of these orbitals by many functions is important. Clearly the charge fitting method with best agreement to experiment is NPA. For the uncharged forms BQH and BQH_2 agreement within 1 $\frac{\text{kcal}}{\text{mol}}$ with experiment could be obtained (-0.79 $\frac{\text{kcal}}{\text{mol}}$ and 0.02 $\frac{\text{kcal}}{\text{mol}}$, respectively; Fig. 5.18). For the charged instances BQH^-, BQ^- and BQ^{2-} the error is considerably larger (1.66 $\frac{\text{kcal}}{\text{mol}}$, -8.18 $\frac{\text{kcal}}{\text{mol}}$ and -6.63 $\frac{\text{kcal}}{\text{mol}}$, respectively). As consequence, the reaction free energies between the uncharged instances agree well (r1 -0.79 $\frac{\text{kcal}}{\text{mol}}$, r2 0.81 $\frac{\text{kcal}}{\text{mol}}$ and r3 0.44 $\frac{\text{kcal}}{\text{mol}}$). The calculation of r3 also compares favorably with the calculation of Wass *et al.* [14], who calculated the reaction energy by B3LYP/6-311+G** and IEF-PCM. Only after correcting the gas phase reaction enthalpy by 8.3 $\frac{\text{kcal}}{\text{mol}}$ according to experimental values, the error in the calculation reduced from -2.21 $\frac{\text{kcal}}{\text{mol}}$ to -0.23 $\frac{\text{kcal}}{\text{mol}}$.

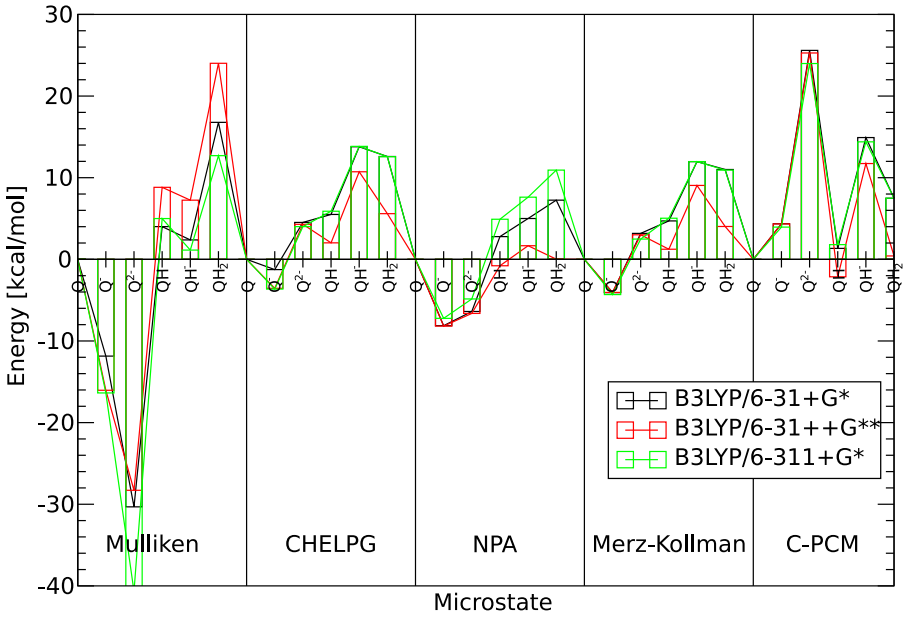


Figure 5.18. The microstate energies (in $\frac{\text{kcal}}{\text{mol}}$) are plotted as difference to the experimental value and relative to the BQ microstate. The data is the same as in Fig. 5.17. The plot clearly shows, that the NPA chargefit method together with B3LYP/6-31++G** is in particularly good agreement with the experiment.

The first $\mathrm{p}K_a$ value of BQH_2 is reasonably predicted (error of 1.20 $\mathrm{p}K_a$ units for p2, Fig. 5.19 and Tab. 5.9), but the second $\mathrm{p}K_a$ value and the $\mathrm{p}K_a$ value of the semiquinone form have a considerable error (-6.05 $\mathrm{p}K_a$ units for p3 and -5.39 $\mathrm{p}K_a$ units for p1). An excellent agreement for the p2 reaction was obtained by Liptak *et al.* [236] using a CBS-QB3 method with C-PCM. Our reduction potential calculations are considerably better for reducing the semiquinone (error of -67 meV for e2 and of -106 meV for e3) than for reducing BQ (error of 355 meV for e1).

It is remarkable, that the experimental values, which are probably the easiest to measure and therefore the most accurate, *i.e.*, r3 and p2 are among the best predicted reaction free energies by Gaussian NPA. The $\mathrm{p}K_a$ value of p3 is hard to measure in aqueous solution due to its high value and p1 due to the complex radical chemistry involved to generate the semiquinone form. However, the calculated $\mathrm{p}K_a$ value for p3 is unrealistic, because by chemical intuition one would assume that the value should be higher than the value of p2, which only could be found using the PCM methods or ADF. Also a negative $\mathrm{p}K_a$ value for p1 seems not to be realistic, but clearly positive values could only be obtained using the PCM methods or ADF as well. Therefore, the problem seems rather to be a systematic error in the calculation of charged instances by the QM method, than experimental inaccuracies. It can be hoped, that adding two or three diffusion functions to better describe the loosely bound electrons of charged species will resolve the remaining problems. Another option would be to choose a different experimental value from literature [237, 238] for the potential of the standard hydrogen electrode and the solvation energy of a proton, which are closely related quantities. We use the value of -260.5 $\frac{\text{kcal}}{\text{mol}}$ [120] for the solvation energy of the proton in consistency with

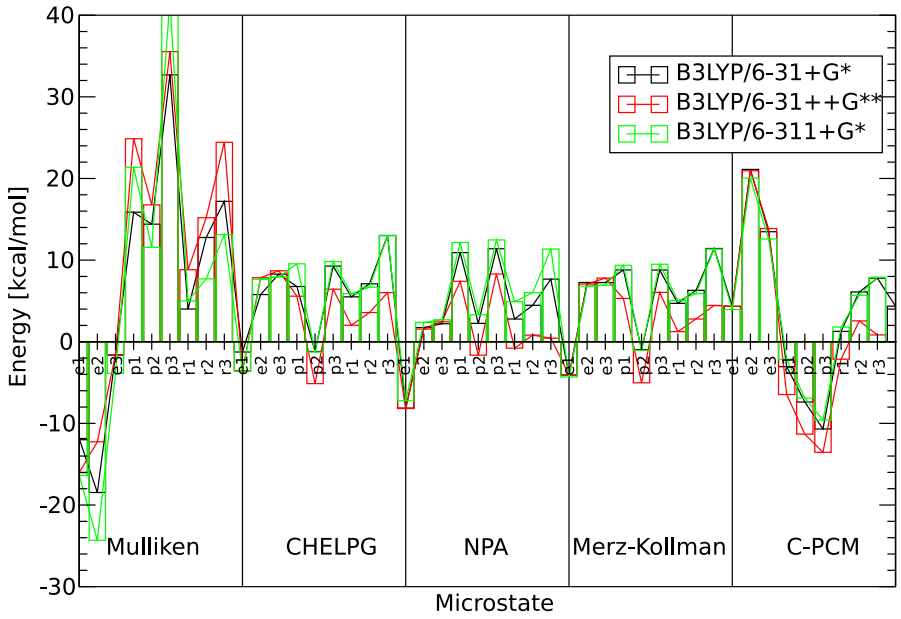


Figure 5.19. The reaction free energy difference (in $\frac{\text{kcal}}{\text{mol}}$) of the nine microscopic reactions (shown in Fig. 5.16) are plotted. Particularly good agreement is also here obtained by the 6-31++g** basis set and the NPA chargefit method.

reaction	Gaussian NPA			ADF CHELPG			experiment			Reference
in units of	$\frac{\text{kcal}}{\text{mol}}$	pK_a	eV	$\frac{\text{kcal}}{\text{mol}}$	pK_a	eV	$\frac{\text{kcal}}{\text{mol}}$	pK_a	eV	
e1	-10.39	7.6	0.451	12.20	-8.9	-0.529	-2.21	1.6	0.096	
e2	1.02	-0.7	-0.044	0.78	-0.6	-0.034	-0.53	0.4	0.023	
e3	-8.09	5.9	0.351	-11.61	8.5	0.503	-10.53	7.7	0.457	
p1	1.78	-1.3	-0.077	-15.41	11.2	0.668	-5.62	4.1	0.244	[215]
p2	-15.14	11.0	0.656	-24.97	18.2	1.083	-13.49	9.8	0.585	[213, 214]
p3	-7.33	5.3	0.318	-27.80	20.2	1.205	-15.62	11.4	0.677	[213, 214]
r1	-8.61	6.3	0.373	-3.21	2.3	0.139	-7.82	5.7	0.339	[216]
r2	-23.22	16.9	1.007	-36.58	26.6	1.586	-24.03	17.5	1.042	[216]
r3	-31.83	23.2	1.380	-39.80	29.0	1.726	-32.26	23.4	1.398	[210–212]

Table 5.9. Calculated and experimental reaction energies in different energy units. The experimental value of 0.699 V *per* electron is doubled for the two electron, two proton reaction r3. The RMSD for the Gaussian B3LYP/6-31++G** calculations is 1.58 $\frac{\text{kcal}}{\text{mol}}$, for the ADF TZ2P(+) calculations 3.19 $\frac{\text{kcal}}{\text{mol}}$ relative to the experimental values. Experimental values were measured for p1, p2, p3 and r3. Ilan [216] calculated r1 and r2 based on these experimental values and we added e1, e2 and e3 following the thermodynamic cycle in Fig. 5.16.

older works [13, 38, 126] using ADF. Instead, in the work on CcO, Punnagai Munusami used a value of -264.61 $\frac{\text{kcal}}{\text{mol}}$ [236, 239], which seems to work better with Gaussian C-PCM.

The problems arising for the rather simple organic molecule *p*-benzoquinone highlight the challenge to compute reaction free energies in complex metall clusters, like the [2Fe-2S] cluster of Fdx and the Cu_B center of CcO. In both projects QM methods were used, which perform significantly worse for *p*-benzoquinone. However, at the moment it seems, that different methods are good for different molecules, *i.e.,* higher basis sets and using NPA gave worse results for the Cu_B center. It is questionable, if the currently used GTO basis sets, polarization and diffusion functions are high enough to describe metal centers as in Fdx as accurate as the TZ2P(+) STO basis set of ADF. Therefore, future work has to investigate further, which method is suitable for a broad spectrum of molecules to obtain accurate reaction free energies.

5.6 Proton Transfer Through the Gramicidin A Channel

A Dynamical Monte Carlo algorithm (DMC) was developed in the group by Mirco Till, Torsten Becker and Matthias Ullmann. This algorithm allows to study transfer processes inside proteins, *i.e.,* long range proton transfer reactions. On the basis of electrostatic energies calculated by QMPB, probabilities of transitions between microstates within a time interval are calculated. My contribution to the work was on one hand assistance in setting up the system for QMPB calculations. since the results of the QMPB calculation were a necessary input for the DMC calculations. On the other hand I contributed some ideas on an efficient caching algorithm (section 4.2.2) speeding up the simulation time by a factor of 13 and thus making simulations on the model system gramicidin A (gA) possible with acceptable computational effort. A manuscript describing the DMC algorithm and the application on gA is in preparation [240].

The study of long range proton transfer processes is important in many biological processes, *e.g.,* the generation of a proton gradient across a membrane in photosynthesis and respiration. The theoretical study of such processes was hindered on one hand by the inability of standard MD simulations to incorporate proton binding and release reactions, on the other hand by the time scale of microseconds to milliseconds required for such proton transfer processes. While the time scale imposes a challenge to MD simulations, it is not feasible with QM/MM methods allowing for bond-breaking and -forming. The DMC approach combines electrostatic calculations in the grand canonical ensemble, which seamlessly allows ligand binding reactions (*e.g.,* protonation probability calculations), with a master equation to describe the time evolution of the system. A Monte Carlo approach is applied to approximate the solution of the master equation.

The microstate energy is described in section 3.1.2 extended by an additional term to the intrinsic energy describing the contribution due to a membrane potential [39]. The equilibrium probability of a particular state is given by eq. 3.9, including the membrane potential into the

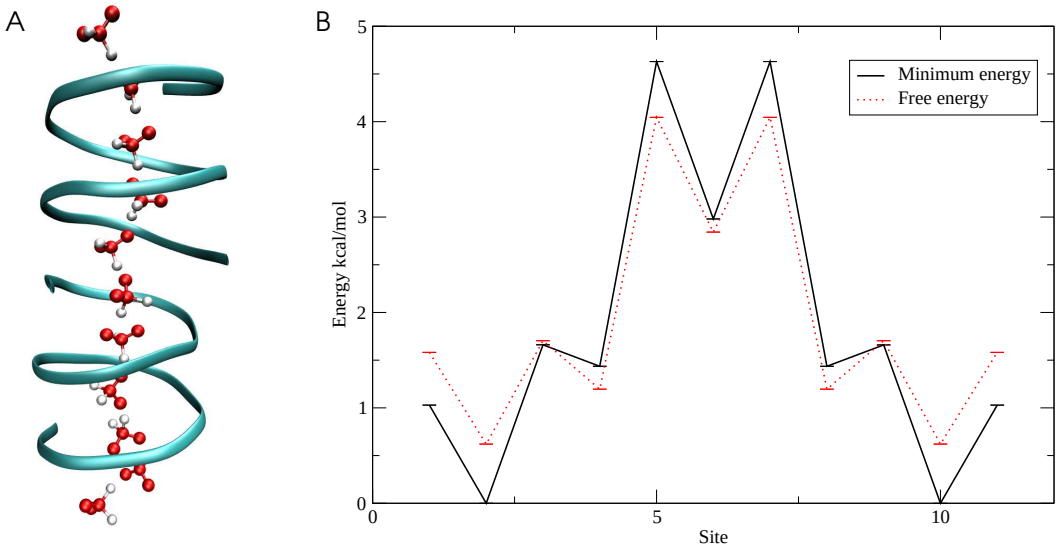


Figure 5.20. Proton transfer through gramicidin A (gA). A: The system consists of eleven water molecules burried inside the gA membrane channel. The dummy atoms describing the membrane are not shown. B: Energy profiles for the gA channel with exactly one proton inside the channel. The solid line depicts the minimum energy for the proton on each site, shifted up by 10 $\frac{\text{kcal}}{\text{mol}}$. The dottet line is the free energy profile.

set of thermodynamic variables $\{\mu_\lambda\}$. Abbreviating the probability of the microstate $n = 1$ at point t in time by $P_1(t) = \langle \vec{x}_{\text{i},1}(\{\mu_\lambda\})\rangle(t)$ the master equation can be written as

$$\frac{d}{dt}P_1(t) = \sum_n^{N_{\text{micro},i}} k_{1n}P_n(t) - \sum_n^{N_{\text{micro},i}} k_{n1}P_1(t). \tag{5.5}$$

In words, the probability of a particular microstate (here P_1) at a point in time is given by the probability of all other microstates n and the rate constant k_{1n} of the reaction from microstate n to microstate 1 (formation of microstate 1) minus the decay of microstate 1 into any of the other microstates. This set of equations as function of time is solved by a Monte Carlo method starting from a single initial microstate and allowing the algorithm to populate other microstates in the course of the simulation. Calculated properties are averaged over multiple simulations to obtain statistically meaningfull results.

To test the algorithm, gramicidin A (gA) was chosen, which is an antibiotic forming a cation permeable water filled channel in the plasma membrane (Fig. 5.20 A). Two molecules form a channel containing a single file of 9 water molecules. Two additional water molecules were added as proton uptake and release site to and from the bulk solvent on both sides of the membrane. For the water molecules a simplified, tetrahedral model was derived, which allows to describe the protonated water molecule by four instances and the neutral water molecule by six instances. This model allows to describe the protonation and rotation of each water by a single site with ten instances in QMPB. The charges on the atoms and lone pairs were fitted to model the monopoles and dipoles of protonated and neutral water. Instead of calculating new coordinates for the hydrogen positions at each step and calculating their electrostatic

interaction with the environment, this discrete model allows to pre-compute all energies for all permutations of protons and lone pairs in the tetrahedral coordinate system of each water molecule. This approximation leads to a dramatic time saving making the simulations possible at all.

Nevertheless, 10^{11} microstates (11 water molecules with 10 instances each) remain, for which sets of reaction rates and probabilities need to be calculated. Here the caching algorithm discussed in section 4.2.2 together with several other optimizations turned out to be valueable.

An energy profile for the transfer process can be obtained (Fig. 5.20 B), showing that the energy barrier for the long range proton transfer through the gA channel is about 3.4 $\frac{\text{kcal}}{\text{mol}}$ for the free energy profile and 4.6 $\frac{\text{kcal}}{\text{mol}}$ for the minimum energy profile. The DMC approach allows to simulate the proton transfer through gA on a microsecond to millisecond timescale. The calculated proton flux reproduces experimentally derived values over a wide range of pH values and membrane potentials (see [240]). The calculations indicate, that the rate of the proton transfer is limited by the gaps between the proton transfer events, not by transfer times of single protons. For physiological membrane potentials, the electrostatic energy barrier is probably the rate limiting factor for a single proton transfer and not the orientation of the hydrogen bonded network. However, both factors influence the rate constants at the same order of magnitude and are therefore important for the long range proton transfer process.

CHAPTER 6

CONCLUSIONS AND OUTLOOK

In my thesis the commonly used theory of ligand binding energetics was generalized to allow for any number of ligand types, which can bind to any number of sites having any number of instances. A consistent microstate description is given in the grand canonical ensemble to allow treatment of sites parameterized based on experimental data or on quantum mechanical computations. The choice of a different reference state led to a significant simplification of the equations compared to previous formulations of the less general energy function. Focussing on ligand binding reactions as function of the chemical potential of the ligand type in the bulk solvent renders the equations more concise, while making them more general at the same time. Previous formulations were increasing in complexity with the number of ligand types, *e.g.,* adding terms for the $\mathrm{p}K_a$ value and the reduction potential calculation. They led to some confusion of the sign of values due to the definition of the $\mathrm{p}K_a$ value based on a deprotonation (unbinding) reaction, while the standard reduction potential refers to the binding reaction of electrons. The discussion based on energies in common units (*e.g.,* $\frac{\mathrm{kcal}}{\mathrm{mol}}$) makes it easier to compare contributions and avoids odd results, *e.g.,* reduction potentials in $\mathrm{p}K_a$ units.

An important contribution to the energy function are electrostatic terms calculated by solving the Linearized Poisson-Boltzmann (LPBE) equation. Classifying the electrostatic energies as homogeneous and heterogeneous transfer energies dependent on the origin of the transfer made the equations more compact. Sites can be parameterized by calculating the energy of formation and vibration quantum mechanically (QM). The electrostatic contributions are a homogeneous transfer energy (from vacuum into the protein) and a correction energy to exclude electrostatic interactions, which are also usually excluded in force-field calculations. For each ligand an energy of the free ligand in bulk solution is included. This term includes some experimental value (*e.g.,* the solvation energy of a proton or the potential of the standard hydrogen electrode) and potentially the energy of formation of the ligand as well as entropy estimates of the binding reaction or energy terms related to volume work. So called relative ligand binding energy calculations rely on the experimentally determined ligand binding energy of model compounds in solution for parameterizing sites. The LPBE calculation determines the shift of the binding energy upon transferring the site from the model compound into the protein environment. Since the rotamers are usually not determined together with the ligand binding energy, energy contributions for changing the rotamer form must be estimated *e.g.,* by a molecular mechanics (MM) force field. Details of the calculation vary if the instance is a reference rotamer or not, or if no rotamers are included for this site at all.

Ligand binding calculations are usually done in three steps: First, the three dimensional structural information of the molecule of interest, known by x-ray crystallography or NMR spectroscopy, is processed to prepare a format suitable for further computations. This step can be rather simple for simple molecules, *e.g.,* transferring the cartesian coordinates for each atom into a file and adding charges from a force field, but it can also be the most complicated step of the calculation, if substantial modeling is necessary. Previously, this step was mainly left to the user to do manually or to write own scripts. This approach led to a substantial amount of repetitive work and was a major source of errors.

For the purpose of structure preparation and input generation an object library, Perl Molecule, was written, which allows to write short Perl programs doing all necessary steps. Since powerful object library methods are used, mistakes by the user are less likely. The structure preparation is easier to control and the protocol is transferable to similar proteins. However, the design as a library or tool box allows to implement variations on the protocol easily and is open for future extensions.

The second step is the calculation of electrostatic energies. For this purpose QMPB was written, which is an interface to programs solving the LPBE. By separating the interface from the Poisson-Boltzmann solver, the PB solver library can be exchanged with little effort. QMPB is minimalistic by design, relying on one side on proper input preparation by a preprocessor program and on the other side on an external PB solver. By this design it should be easy to adopt QMPB to changing energy functions, *e.g.,* additional energy terms found to be necessary to include. At the same time it should be as flexible as the equations for ligand binding. All complexity, which may arise from complicated sites, is left to the preprocessor. QMPB concentrates on implementing the energy functions as general as they were derived. All calculation steps should be transparent to the user to follow each step and detect possible mistakes. Each calculation step can be reproduced by the user for education and error checking. However, for non-trivial systems the number of analogous steps is too large to be done manually requiring a program like QMPB.

QMPB scales linear with the total number of instances of the system. Since the LPBE calculations of instances are independent of each other, parallel execution on any number of processors is possible without any inter-processor communication. Therefore, linear scaling with the number of processors up to the total number of instances, which ranges from some hundred to several thousand for biological systems, is expected. The parallelization or adoption to particular computing environments does not require any changes in the QMPB program, but can be done by external scripts or programs dependent on the local requirements. This separation of the parallelization from the main program is another advantage of the design decision to perform the LPBE calculations by external programs.

The third step in the calculation is to calculate ligand binding probabilities dependent on thermodynamic variables, *i.e.,* sets of chemical potentials of ligands in the bulk solvent. For very small systems an analytical evaluation of the partition function is possible and can be performed, *e.g.,* by SMT[1]. For larger systems ligand binding probabilities have to be approximated, which can be done by a Monte Carlo procedure as implemented in GMCT[1]. The programs rely on the intrinsic energies and interaction energies calculated and tabulated by QMPB. For larger systems, especially with more than one or two ligand types, the Monte Carlo procedure

[1] SMT and GMCT were written by Matthias Ullmann.

is the time limiting step of the whole calculation. Parallelization can help to reduce the wall-clock time a user needs to wait, but the computational effort increases exponentially with the number of ligand types. With the Adaptive Mesh Refinement (AMR) algorithm suggested in this work a substantial saving of computational time on the calculation of probabilities can be expected. The probability of most instances is constant over wide areas in chemical potential space, *e.g.,* only the fraction of the chemical potential range, where the instance titrates, is of interest to be sampled by a fine grid. Also the correlation with other sites binding other ligands is usually not strong, so that instances tend to be rather constant in dimensions of chemical potential space representing ligands not binding to this site. Therefore, an algorithm exploiting these properties can reduce the computational cost substantially.

Currently, the savings by AMR are still modest, which is in part due to the test system chosen, which represents a kind of worst case scenario, and in part due to limitations in the implementation, not fully exploiting the power of the algorithm. At present, titration calculations are only done up to two dimensions, but many current and future projects of the group will demand studies of more ligand types. The generalized theory and QMPB permit using any number of ligand types, but the exponential scaling behavior of the Monte Carlo procedure will become a hindrance. Therefore, future work on AMR and similar approaches will be valueable and strongly required.

My implementation of the ligand binding theory does not include membrane potentials, because my primary focus was on soluble proteins. The theory of including membrane potentials in electrostatic potentials was worked out by Elisa Bombarda. Thomas Ullmann extendend the LPBE solver libraries of MEAD by a sophisticated membrane model and functions to include membrane potentials. Since the membrane potential can be described by an additive term to the intrinsic energy in my equations, it was easy to replace the external LPBE solver programs by those being able to perform membrane potential calculations. Here the modular design already proved to be usefull. Future versions of Perl Molecule and QMPB should be able to prepare also the input for membrane proteins, which is currently done by external helper programs. In the probability calculation of instances, the internal and external chemical potential of protons will be treated as different ligand types and therefore contribute to the scaleability problem of the Monte Carlo calculation step.

The current implementation of rotamers in Perl Molecule is probably good for hydrogen atoms. However, the group of Marylin Gunner and others have shown, that including sidechain rotamers can help to improve on the agreement between calculated and experimentally measured $\mathrm{p}K_a$ values. Probably sidechain flexibility can also help in binding studies of other ligands. Currently, the support for sidechain rotamers by Perl Molecule is quite limited, because no force field energy function is implemented. The energies derived from probabilities of rotamers in databases was not found to be meaningfull and tabulating pre-computed rotamer energies by an external program is not feasible for longer sidechains. Also the interaction with the protein environment as well as with other rotateable sidechains needs to be included. The formalism of QMPB splitting the energy function in an intrinsic energy independent of all other sites and an interaction term accounting for the pairwise interactions of two instances of two sites can remain. Just an additional program is required calculating these energy contributions. This program could also solve the problem of the omitted Lennard-Jones terms in the current implementation, which is probably negligible for hydrogen rotamers, but gets important for sidechain rotamers. Also in this direction work is done by Thomas Ullmann.

Perl Molecule and QMPB have proven to be usefull in a number of projects outlined in the application section. In fact, most members of the group are currently using QMPB for their research and it has already become a part of the research practical for students to learn using it. The programs should be made available for any interested user via the internet under the GPL, free of charge allowing to modify and extend the programs for future applications.

However, the applications of QMPB to sites parameterized by quantum chemical calculations has not yet provided fully reliable results. In fact, it was expected to be one of the main advantages of QMPB to be able to treat sites, which can not sufficiently be characterized by experiment. In this field it is to my knowledge also the only program so far, which can satisfyingly solve the multiple site titration problem. Actually, the problems which arose, are not directly related to Perl Molecule or QMPB, but are rooted in the limitations of current charge fitting methods and inaccuracies of currently available basis sets and exchange correlation functionals. Since QMPB takes only the charges and energies no matter which way they were determined, it is open to any new development in the field of quantum chemistry.

By no means, the formalism and implementation of my programs is limited to thermodynamics of ligand binding only. As the application on gramicidin A shows, QMPB suites well to provide the thermodynamic basis for further kinetic studies on ligand transfer reactions. Currently work is done by Eva-Maria Krammer and Mirco Till applying DMC to proton transfer reactions in the bacterial reaction center and future projects may include electron transfer reactions as well. Other applications of QMPB have been stability predictions of proteins after site directed mutagenesis. Different amino acids at a position can be treated as different instances. Also for such projects, inclusion of better energy terms for sidechain rotamers is of interest.

Since already in such a short time Perl Molecule and QMPB have proven to be usefull for different projects in the domain of applications, for which they were written, and outside of this domain, it can be expected that the programs will be of great use in the theoretical biophysics community as soon as they will be made available to a larger group of users.

NOMENCLATURE

Here the meaning of certain terms is explained as they are used in this thesis and in the Perl Molecule ontology. However, no general definitions are attempted.

atom
Atom in a particular residue of a chain and conformer of the biomolecule. An atom is described by a radius (r in Å), a set of coordinates and a set of charges.

background (charge) set
Set of atoms of a conformer, which does not belong to any site.

background energy
Energy of a charge due to interaction with fixed charges in the system (background charges).

binding free energy
For a molecule the binding energy is the difference in microstate energy of the bound and unbound microstate. For a site the binding energy is the difference in energy of the bound and unbound instance of the site.

Born energy
Energy of a (solvated) charge due to a set of dielectric boundaries.

chain
Usually identical with a polymer chain of the biomolecule in a particular conformer. A chain is a set of residues.

charge form
Set of atoms with their respective partial charges keeping an integer total charge. Defined by a column in a charge set file (appendix A.2).

charge group member
Additional information associated with an atom defined in a topology file (appendix A.5.1). Most important is the partial charge.

charge group
Set of charge group members with their respective partial charges keeping an integer total charge. Charge groups are usually defined by topology files (appendix A.5.1).

charge
Partial charge q of an atom in units of a fraction of an elementary charge.

charge set
Set of charge forms. Defined by a charge set file, see appendix A.2.

conformation
Synonymous with conformer, *i.e.*, globally different structure.

conformational energy
The conformational energy is the energy of a particular conformation relative to a reference conformation.

conformer
Discrete conformation, *i.e.*, globally different structure. Independent sets of electrostatic calculations (with different sets of dielectric boundaries) need to be performed for each conformer. A conformer is a set of chains.

coordinate
Cartesian coordinate of an atom (x, y, z in Å).

coordinate set
Set of rotamer forms.

instance member
Particular, single atom with coordinates and charges taken from a particular combination of rotamer form and charge form.

instance
Set of atoms with coordinates and charges taken from a particular combination of rotamer form and charge form.

interaction energy
The interaction energy is due to the interaction of the potential of an instance of a site with the charges of an instance of another site.

intrinsic energy
The intrinsic energy is the energy of an instance of a site in the dielectric boundaries of the molecule and interacting with the background (charge) set. The charges of all atoms belonging to another site are set to zero.

microstate energy
Energy of a molecule in a particular microscopic state at a given set of thermodynamic variables.

molecule
Biomolecular system, *e.g.*, a ribosome. A molecule is described as set of conformers. Because all studies concentrated on proteins, “protein” is often used synonymously with biomolecule, not excluding that the described methods can be applied to other biomolecules as well..

reaction field
Effective electrostatic field acting back on the charge, which created the field.

reaction free energy
The reaction free energy is the energy difference of the system before and after a reaction.

residue
Usually identical with a residue of the biopolymer in a particular chain and conformer. A residue is a set of atoms.

rotamer form
Set of atoms with their respective coordinates.

site
Set of atoms associated with a coordinate set and/or charge set. A site is a set of instances.

solvation energy
Homogeneous transfer energy from vacuum into a given solvent.

state vector
Vector characterizing a microstate. The length of the state vector equals the number of sites and the value at each position refers to a particular instance of the site.

Stern layer
Ion exclusion layer around the protein to preserve a solvation layer around the ions.

Abbreviations and Acronyms

cre	Carbon **c**atabolite **r**esponse **e**lement.
ADF	**A**msterdam **d**ensity **f**unctional.
AMBER	**A**ssisted **m**odel **b**uilding and **e**nergy **r**efinement.
AMD	**A**dvanced **M**icro**d**evices.
AMR	**A**daptive **m**esh **r**efinement.
APBS	**A**daptive **P**ossion-**B**oltzmann **s**olver.
API	**A**pplication **p**rogramming **i**nterface.
ASCII	**A**merican **s**tandard **c**ode for **i**nformation **i**nterchange.
ATP	**A**denosine **t**ri**p**hosphate.
AWK	Interpreter language named by the authors Alfred **A**ho, Peter **W**einberger and Brian **K**ernighan.
B. subtilis	*Bacillus subtilis*.
C-PCM	**C**onductor like PCM.
cAMP	**C**yclic **a**denosine **m**ono**p**hosphate.
CcO	**C**ytochrome **c** **o**xidase.
CcpA	Carbon **c**atabolite **c**ontrol **p**rotein **A**.
CCR	**C**arbon **c**atabolite **r**epression.
CHARMM	**C**hemistry at **Har**vard **m**acromolecular **m**echanics.
CHELP	**Ch**arges from **el**ectrostatic **p**otentials.

CHELPG	**Ch**arges from **el**ectrostatic **p**otentials using a **g**rid based method.
CMCT	**C**onformational **M**onte **C**arlo **t**itration.
COSMO	**Co**nductor like **s**creening **mo**del.
CPAN	**C**omprehensive **P**erl **a**rchive **n**ework.
CPU	**C**entral **p**rocessing **u**nit.
DelPhi	Finite difference Poisson-Boltzmann solver by the Honig laboratory.
DFT	**D**ensity **f**unctional **t**heory.
DHC2	**Di**-**h**eme **c**ytochrome **c2**.
DMC	**D**ynamical **M**onte **C**arlo.
DMF	**Di**-**m**ethyl**f**ormamide.
DNA	**D**eoxyribo**n**ucleic **a**cid.
E. coli	*Escherichia coli*.
e.s.u.-c.g.s	**E**lectro**s**tatic **u**nits based on **c**entimeter, **g**ram, **s**econd and other base units.
EI	**E**nzyme **I**.
EII	**E**nzyme **II**.
EIII	**E**nzyme **III**.
EVB	**E**mpirical **v**alence **b**ond method.
Fdx	**F**erre**d**o**x**in.
FIFO	**F**irst **i**n **f**irst **o**ut buffer.
FNR	**F**erredoxin-**N**ADP-oxido**r**eductase.
FORTRAN	**For**mula **tran**slation.
gA	**G**ramicidin **A**.
GAMESS	**G**eneral **a**tomic and **m**olecular **e**lectronic **s**tructure **s**ystem.

GMCT	**G**eneralized **M**onte **C**arlo **t**itration.
GNU	**G**NU is **n**ot **U**nix.
GPL	**G**NU **p**ublic **l**icense, `http://www.gnu.org/copyleft/gpl.html`.
GREP	**G**lobal **r**egular **e**xpression **p**rint.
GROMACS	**Gro**ningen **ma**chine for **c**hemical **s**imulations.
GTO	**G**aussian **t**ype **o**rbital.
HELICS	**He**idelberg **Li**nux **c**luster **s**ystem.
HEWL	**H**en **e**gg **w**hite **l**ysozyme.
HF	**H**artree-**F**ock.
HPr	**H**istidine-containing phosphocarrier **pr**otein.
HPrK/P	**HPr K**inase/**P**hosphatase.
IEF-PCM	**I**ntegral **e**quation **f**ormalism PCM.
IO	**I**nput/**o**utput.
IPC	Interprocess communication.
IR	**I**nfra**r**ed spectroscopy.
IWR	German: **I**nstitut für **w**issenschaftliches **R**echnen, English: Institute for scientific computing.
KS	**K**ohn-**S**ham.
LCAO	**L**inear **c**ombination of **a**tomic **o**rbitals.
LPBE	**L**inearized **P**oisson-**B**oltzmann **e**quation.
LSDA	**L**ocal **s**pin **d**ensity **a**pproximation.
MC	**M**onte **C**arlo.
MCCE	**M**ulti**c**onformational **c**ontinuum **e**lectrostatics.
MCTI	**M**onte **C**arlo **ti**tration.

MD	**M**olecular **d**ynamics.
MD-PDLD	**M**olecular **d**ynamics averaging of **PDLD** calculations.
MDC	**M**ultipole **d**erived **c**harges.
MEAD	**M**olecular **e**lectrostatics at **a**tomic **d**etail.
MEP	**M**olecular **e**lectrostatic **p**otential.
MM	**M**olecular **m**echanics.
MMC	**M**etropolis **M**onte **C**arlo.
MPI	**M**essage **p**assing **i**nterface.
NADP	**N**icotinamide **a**denine **d**inucleotide.
NMR	**N**uclear **m**agnetic **r**esonance spectroscopy.
NOE	**N**uclear **O**verhauser **e**ffect.
NPA	**N**atural **p**opulation **a**nalysis.
NUMA	**N**on-**u**niform **m**emory **a**ccess or **a**rchitecture.
OMG	**O**bject **M**anagement **G**roup.
OOP	**O**bject-**o**riented **p**rogramming.
PARSE	**Par**ameters for **s**olvation **e**nergies.
PB	**P**oisson-**B**oltzmann.
PBE	**P**oisson-**B**oltzmann **e**quation.
PCM	**P**olarizable **c**ontinuum **m**odel.
PDB	**P**rotein **d**ata **b**ank.
PDLD	**P**rotein **d**ipoles **L**angevin **d**ipoles.
PEP	**P**hospho**e**nol**p**yruvat.
Perl	**P**ractical **e**xtraction and **r**eport **l**anguage.
PGTO	**P**rimitive **G**aussian **t**ype **o**rbital.
PTS	Phosphoenolpyruvat:sugar **p**hospho**t**ransferase **s**ystem.

PVM **P**arallel **v**irtual **m**achine.

PW Gradient-corrected XC energy functional of **P**erdew and **W**ang.

QM **Q**uantum **m**echanics (here synonymously used with quantum chemistry).

QMPB **Q**uantum **m**echanics based **P**oisson-**B**oltzmann.

RAID **R**edundant **a**rray of **i**nexpensive **d**isks.

RAM **R**andom **a**ccess **m**emory.

RMSD **R**oot **m**ean **s**quare **d**eviation. The RMSD is used as a measure for similarity. It is calculated by

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n}(\vec{r}_{\mathrm{ai}} - \vec{r}_{\mathrm{bi}})^2}{n}}, \tag{6.1}$$

where $\vec{r}_{\mathrm{ai}}$ and $\vec{r}_{\mathrm{bi}}$ are the positions of atom i of structure a and b, respectively. The number of atoms if given by n. In this work, RMSD is also used for scalars synonymously with a standard deviation.

RNA **R**ibo**n**ucleic **a**cid.

SCF **S**elf-**c**onsistent **f**ield.

SCSI **S**mall **c**omputer **s**ystems **i**nterface.

SED **S**tream **ed**itor.

SHE **S**tandard **h**ydrogen **e**lectrode.

SI **I**nternational **s**ystem of units based on kilogram, meter, second, ampere, kelvin, mole, candela.

SMP **S**ymmetric **m**ulti**p**rocessor architecture.

SMT **S**tatistical **m**echanics **t**reatment.

SOR **S**uccessive **o**ver-**r**elaxation.

SPE **S**ynergistic **p**rocessing **e**lements.

STO **S**later **t**ype **o**rbital.

SVD **S**ingular **v**alue **d**ecomposition.

SWIG	**S**implified **w**rapper and **i**nterface **g**enerator.
TZ2P	Core double-ζ, valence **t**riple-ζ, doubly (**2**) **p**olarized basis set of ADF.
TZ2P(+)	TZ2P+ for transition metals Sc-Zn and TZ2P otherwise.
TZ2P+	As TZ2P, but with 4 d-functions instead of 3 (only for transition metals Sc-Zn).
TZP	Core double-ζ, valence **t**riple-ζ, **p**olarized basis set of ADF.
UHBD	**U**niversity of **H**ouston **B**rownian **D**ynamics.
UML	**U**nified **m**odeling **l**anguage.
UV	**U**ltra **v**iolet light absorbtion spectroscopy.
VIS	**Vis**ual light absorbtion spectroscopy.
VMD	**V**isual **m**olecular **d**ynamics.
VWN	LSDA parameterization of **V**osko, **W**ilk and **N**usair.
XC	E**x**change and **c**orrelation.
XML	E**x**tensible **m**arkup **l**anguage.

Equation Symbols

$E[\rho(\vec{r})]$	Energy functional.
$E_{XC}[\rho(\vec{r})]$	Exchange and Correlation Energy.
E	Total energy.
$G_{\mathrm{Born,homo,i}}(j_k)$	Born energy of instance j_k in the homogeneous environment.
$G_{\mathrm{Born,model,i}}(j_k)$	Born energy of instance j_k in the model compound.
$G_{\mathrm{Born,protein,i}}(j_k)$	Born energy of instance j_k in the protein.
G_{Born}	Born energy.
$G_{\mathrm{back,homotrans,i}}(j_k)$	Background energy of instance j_k in the protein.
$G_{\mathrm{back,model,i}}(j_k)$	Background energy of instance j_k in the model compound.
$G_{\mathrm{back,protein,i}}(j_k)$	Background energy of instance j_k in the protein.
G_{back}	Background energy.
$G_{\mathrm{trans}}(H^+)$	Translational energy of the proton.
$G_{\mathrm{conf},i}$	Conformational energy of conformer i.
$G_{\mathrm{corr},i}(j_k)$	Correction of the background energy of instance k of site i in conformer i.
$G_{\mathrm{free},i}(j_k)$	Energy of the unbound ligand molecules for instance k of site i in conformer i.
$G_{\mathrm{inter},i}(j_k, l_m)$	Interaction energy of instance k of site i with instance m of site l in conformer i.
$G_{\mathrm{intr},i}(j_k)$	Intrinsic energy of instance k of site i in conformer i.
G_{micro}	Energy of a particular microstate.
$G_{\mathrm{vib},i}(j_k)$	Vibrational energy obtained from QM calculations on instance k of site i in conformer i.

$H_{\mathrm{vac},i}(j_k)$	Total bonding energy (or energy of formation) obtained from QM calculations on instance k of site i in conformer i.
H	Hamiltonian.
I	Ionic strength.
N_A	Avogadro constant.
N_ϑ	Number of electrons with spin $\vartheta = \alpha, \beta$.
N_e	Number of electrons.
N_λ	Number of ligands λ bound to conformer i of the molecule.
N_{atom}	Number of atoms in the molecule.
$N_{\mathrm{back,i}}$	Number of atoms in the background set of conformer i.
$N_{\mathrm{back,model,i,j}}$	Number of atoms in the background set of the model compound for site j of conformer i.
N_{conf}	Number of conformers.
N_{iontype}	Number of ion types.
$N_{\mathrm{instance},i,j}$	Number of instances in site j in conformer i.
$N_{\mathrm{instance},i,l}$	Number of instances in site l in conformer i.
N_{ligand}	Total number of reactive ligand types λ.
$N_{\mathrm{micro},i}$	Number of microstates in conformer i.
N_{moves}	Total number of MC moves.
N_{scans}	Total number of MC scans.
$N_{\mathrm{site},i}$	Number of sites in conformer i.
$N_{\mathrm{steps},\lambda}$	Total number of steps, in which the chemical potential range is sampled for ligand type λ.
P	Pressure.
Q_{a}	Background charge of atom a.
R	Gas constant.
S	Entropy.
$T[\rho(\vec{r})]$	Kinetic energy.
T	Absolute temperature.
$U[\rho(\vec{r})]$	Classical electrostatic energy.

V	Volume.
W_{elec}	Electrostatic Energy.
Z_{a}	Nuclear charge of atom a.
$\Delta G_{\mathrm{Born,heterotrans,i}}(j_k)$	Difference in Born energy for the transfer of instance j_k from a heterogeneous environment into the protein.
$\Delta G_{\mathrm{Born,homotrans,i}}(j_k)$	Difference in Born energy for the transfer of instance j_k from a homogeneous environment into the protein.
$\Delta G_{\mathrm{back,heterotrans,i}}(j_k)$	Difference in background energy for the transfer of instance j_k form a heterogeneous environment into the protein.
ΔG_{conf}	Difference in conformational energy.
$\Delta G_{\mathrm{heterotrans,i}}(j_k)$	Difference in energy for the transfer of instance j_k from a heterogeneous environment into the protein.
$\Delta G_{\mathrm{homotrans,i}}(j_k)$	Difference in energy for the transfer of instance j_k from a homogeneous environment into the protein.
$\Delta G_{\mathrm{inter}}$	Difference in interaction energy.
$\Delta G_{\mathrm{micro}}$	Difference in microstate energy.
$\Delta G_{\mathrm{model}}$	Energy difference of reaction of model compound.
$\Delta G_{\mathrm{rotamer}}(j_k)$	Energy difference of a particular rotamer k of site j relative to a reference rotamer. The charges have to be unchanged in the reference rotamer and the rotamer j_k.
$\Delta G_{\mathrm{solv,i}}(j_k)$	Solvation energy of instance j_k.
$\Delta G_{\mathrm{Born},i}(j_k)$	Difference in Born energy of instance k of site j in conformer i.
$\Delta G_{\mathrm{back},i}(j_k)$	Difference in background energy of instance k of site j in conformer i.
ΔG_{bind}	Binding free energy.
ΔG_{reac}	Reaction free energy.
$\Delta G_{\mathrm{selfback},i}(j_{r,\mathrm{protein}})$	Homogeneous transfer energy of the background charge set due to changing boundaries caused by the rotamer j_r.
$\Delta\Delta G_{\mathrm{Born}}$	Difference in Born energy.
$\Delta\Delta G_{\mathrm{back}}$	Difference in background energy.
$\Delta\Delta G_{\mathrm{inter,i}}(j,l)$	Interaction energy between site j and site l in conformer i in the titration theory using a binary state vector.

Λ	Chemical species number, including molecules M and λ ligand types.
Ψ	Wavefunction.
Ξ	Grand canonical partition function.
$\bar{\kappa}^2(\vec{r})$	Modified and squared inverse Debye length $\bar{\kappa}^2(\vec{r}) = \varepsilon(\vec{r})\kappa^2(\vec{r})$.
$\bar{\kappa}_0^2$	Modified and squared inverse Debye length at point 0 of the grid used for solving the LPBE numerically.
β	$\frac{1}{RT}$.
χ_e	Electrical susceptibility of the material.
$\epsilon_{XC}[\rho_\alpha, \rho_\beta, \nabla\rho_\alpha, \nabla\rho_\beta]$	Gradient-corrected XC energy density at point $\vec{r}$ in space.
$\epsilon_{XC}[\rho_\alpha, \rho_\beta]$	XC energy density at point $\vec{r}$ in space.
$\kappa(\vec{r})$	Inverse Debye length.
λ	Ligand type number, $1 \dots N_{\text{ligand}}$.
$\langle G_{\text{inter,i}}(j_k, \{\mu_\lambda\})\rangle$	Mean-field interaction energy of instance k of site j in conformer i for a given set of thermodynamic variables $\{\mu_\lambda\}$.
$\langle c_\lambda(\vec{r})\rangle$	Mean concentration of ion type λ.
$\langle x_i(l_m, \{\mu_\lambda\})\rangle$	Statistical average probability of instance m, of site l in conformer i for a given set of thermodynamic variables $\{\mu_\lambda\}$.
$\mathcal{E}^\circ$	Standard Reduction potential.
$\mathcal{E}$	Reduction potential.
$\mathcal{F}$	Faraday constant.
$\mathcal{K}_0$	Value at each point of the grid used for calculating the LPBE marking if $\bar{\kappa}^2(\vec{r})$ needs to be calculated at this point, *i.e.*, outside of the molecule and the Stern layer, or not (inside the Stern layer). At the Stern layer the value is averaged from the six surrounding values.
$\mathcal{S}$	Surface.
$\mathcal{W}_\lambda(\vec{r})$	Potential of mean force experienced by ion type λ at position $\vec{r}$.
μ_Λ°	Standard chemical potential of chemical species Λ.
μ_λ°	Standard chemical potential of ligand type λ.
μ_Λ	Chemical potential of chemical species Λ.

μ_λ	Chemical potential of ligand type λ.
$\mu_{\mathrm{H}^+} = -RT \ln 10 \mathrm{pH}$	Chemical potential of protons.
$\mu_{\mathrm{e}^-} = -\mathcal{F}\mathcal{E}$	Chemical potential of electrons.
∇	Gradient.
ν_Λ	Stoichiometric factor of chemical species Λ.
$\nu_{\lambda,i}(j)$	Stoichiometric factor *e.g.,* the maximal number of ligands λ site j can bind.
ϕ	Electrical flux (net flow) across a surface $\mathcal{S}$.
ρ	Density of free electric charge.
e_0	Elementary charge.
z_λ	Charge number (valency) of ion type λ.
σ	Symmetry number or degeneracy factor.
ε_0	Electrical permittivity of vacuum in farad per meter ($\frac{F}{m}$).
ε_r	Relative electrical permittivity or dielectric constant, unit-less ($\varepsilon_\mathrm{r} = \frac{\varepsilon}{\varepsilon_0}$).
ε	Electrical permittivity of the material in farad per meter ($\frac{F}{m}$).
$\varphi(\vec{r})$	Electrostatic potential at point $\vec{r}$.
φ_0	Electrostatic potential at point 0 of the grid used for solving the LPBE.
$\vec{A}$	Surface area element with normal vector orthogonal to the surface plane.
$\vec{D}$	Electric displacement field or electrical flux density.
$\vec{E}$	Electric field.
$\vec{P}$	Polarization density.
$\vec{\nabla}$	Divergence, in cartesian coordinates of an Euclidean space $\nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z}$.
$\vec{n}$	Surface normal vector.
$\vec{r}$	Point in space.
$\vec{x}$	State vector.
ξ	Progress variable or extent of reaction.

$\{\mu_\lambda\}$	Set of thermodynamic variables. In particular the chemical potential μ_λ of each ligand type λ is of importance for the calculations. Other thermodynamic variables, *i.e.*, T and V are usually not varied.
$c_{\mathrm{bulk},\lambda}$	Concentration of ion type λ in the bulk solvent.
h	The spacing of the grid used for solving the LPBE numerically.
k_B	Boltzmann constant.
n_Λ	Number of molecules of chemical species Λ.
$n_{\lambda,i}(j_k)$	Number of bound ligands of type λ in instance k of site i in conformer i.
q_{a}	Pointcharge of atom a.
$v_{XC,\vartheta}(\vec{r})$	XC potential.
a	Index for an atom.
pH	Negative decadic logarithm of the proton concentration.
$\mathrm{p}K_a$	Negative decadic logarithm of the equilibrium constant K_a of a deprotonation reaction.
$\mathrm{p}K_{a,\mathrm{intr},i}$	Intrinsic $\mathrm{p}K_a$ value.
$\mathrm{p}K_{a,\mathrm{model}}$	$\mathrm{p}K_a$ value of model compound.
$\mathrm{p}K_{\mathrm{app}}$	Apparent $\mathrm{p}K_a$ value.

BIBLIOGRAPHY

[1] M. Shirts and V. S. Pande. Computing: Screen savers of the world unite! *Science*, 290(5498):1903–1904, Dec 2000.

[2] M. R. Shirts and V. S. Pande. Mathematical analysis of coupled parallel simulations. *Phys. Rev. Lett.*, 86(22):4983–4987, May 2001.

[3] C. D. Snow, Y. M. Rhee, and V. S. Pande. Kinetic definition of protein folding transition state ensembles and reaction coordinates. *Biophys. J.*, 91(1):14–24, Jul 2006.

[4] G. Jayachandran, V. Vishal, and V. S. Pande. Using massively parallel simulation and Markovian models to study protein folding: examining the dynamics of the villin headpiece. *J. Chem. Phys.*, 124(16):164902, Apr 2006.

[5] G. Jayachandran, V. Vishal, A. E. García, and V. S. Pande. Local structure formation in simulations of two small proteins. *J. Struct. Biol.*, 157(3):491–499, Mar 2007.

[6] D. L. Ensign, P. M. Kasson, and V. S. Pande. Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *J. Mol. Biol.*, 374(3):806–816, Nov 2007.

[7] J. Åqvist and A. Warshel. Simulation of enzyme reactions using valence bond force fields and other hybrid quantum/classical approaches. *Chem. Rev.*, 93:2523–2544, 1993.

[8] A. Warshel. Computer simulations of enzyme catalysis: Methods, progress, and insights. *Annu. Rev. Biophys. Biomol. Struct.*, 32:425–443, 2003.

[9] U. Ryde. Combined quantum and molecular mechanics calculations on metalloproteins. *Curr. Opin. Chem. Biol.*, 7(1):136–142, 2003.

[10] A. D. Becke. Density-functional thermochemistry .3. The role of exact exchange. *J. Chem. Phys.*, 98(7):5648–5652, Apr 1 1993.

[11] C. J. Cramer and D. G. Truhlar. Implicit solvation models: Equilibria, structure, spectra, and dynamics. *Chem. Rev.*, 99(8):2161–2200, 1999.

[12] M. Orozco and F. J. Luque. Theoretical methods for the description of the solvent effect in biomolecular systems. *Chem. Rev.*, 100(11):4187–4225, 2000.

[13] J. Li, C. L. Fisher, J. L. Chen, D. Bashford, and L. Noodleman. Calculation of redox potentials and pK_a values of hydrated transition metal cations by a combined density functional and continuum dielectric theory. *Inorg. Chem.*, 35:4694–4702, 1996.

[14] J. R. T. J. Wass, E. Ahlberg, I. Panas, and D. J. Schiffrin. Quantum chemical modeling of the reduction of quinones. *J. Phys. Chem. A*, 110(5):2005–2020, Feb 9 2006.

[15] A. Klamt and G. Schüürmann. COSMO: A new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *J. Chem. Soc., Perkin Trans. 2*, 5:799–805, 1993.

[16] A. Klamt, V. Jonas, T. Burger, and J. C. W. Lohrenz. Refinement and parametrization of COSMO-RS. *J. Phys. Chem. A*, 5639(98):5074–5085, 1998.

[17] A. Klamt, F. Eckert, and M. Diedenhofen. First principles calculations of aqueous pK_a values for organic and inorganic acids using COSMO-RS reveal an inconsistency in the slope of the pK_a scale. *J. Phys. Chem. A*, 107(44):9380–9386, 2003.

[18] J. Tomasi and M. Persico. Molecular interactions in solution: An overview of methods based on continuous distributions of the solvent. *Chem. Rev.*, 94:2027–2094, 1994.

[19] A. Fortunelli and J. Tomasi. The implementation of density functional theory within the polarizable continuum model for solvation. *Chem. Phys. Lett.*, 231(1):34–39, 1994.

[20] E. Cances, B. Mennucci, and J. Tomasi. A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to isotropic and anisotropic dielectrics. *J. Chem. Phys.*, 107(8):3032–3041, 1997.

[21] K. Y. Sanbonmatsu, S. Joseph, and C. S. Tung. Simulating movement of tRNA into the ribosome during decoding. *Proc. Natl. Acad. Sci. USA*, 102(44):15854–15859, Nov 1 2005.

[22] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Natl. Acad. Sci. USA*, 98(18):10037–10041, Aug 2001.

[23] M. E. Davis, J. D. Madura, B. A. Luty, and J. A. McCammon. Electrostatics and diffusion of molecules in solution - Simulations with the University-of-Houston-Brownian Dynamics program. *Comp. Phys. Comm.*, 62(2-3):187–197, MAR 1991.

[24] J. D. Madura, J. M. Briggs, R. C. Wade, M. E. Davis, B. A. Luty, A. Ilin, J. Antosiewicz, M. K. Gilson, B. Bagheri, L. R. Scott, and J. A. McCammon. Electrostatics and diffusion of molecules in solution - Simulations with the University-of-Houston Brownian Dynamics program. *Comp. Phys. Comm.*, 91(1-3):57–95, SEP 1995.

[25] W. Rocchia, E. Alexov, and B. Honig. Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions. *J. Phys. Chem. B*, 105(28):6507–6514, JUL 2001.

[26] B. Honig and A. Nicholls. Classical electrostatics in biology and chemistry. *Science*, 268(5214):1144–1149, May 1995.

[27] A. S. Yang, M. R. Gunner, R. Sampogna, K. Sharp, and B. Honig. On the calculation of pK_as in proteins. *Proteins*, 15(3):252–265, Mar 1993.

[28] A. Nicholls and B. Honig. A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. *J. Comput. Chem.*, 12(4):435–445, 1991.

[29] I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig. Focusing of electric fields in the active site of Cu-Zn superoxide dismutase: Effects of ionic strength and amino-acid modification. *Proteins*, 1(1):47–59, Sep 1986.

[30] D. Bashford and K. Gerwert. Electrostatic calculations of the pK_a values of ionizable groups in bacteriorhodopsin. *J. Mol. Biol.*, 224(2):473–486, Mar 1992.

[31] D. Bashford. An object-oriented programming suite for electrostatic effects in biological molecules. In Yutaka Ishikawa, Rodney R. Oldehoeft, John V. W. Reynders, and Marydell Tholburn, editors, *Scientific Computing in Object-Oriented Parallel Environments*, volume 1343 of *Lecture Notes in Computer Science*, pages 233–240. ISCOPE97, Springer, 1997.

[32] E. G. Alexov and M. R. Gunner. Incorporating protein conformational flexibility into the calculation of pH-dependent protein properties. *Biophys. J.*, 72(5):2075–2093, May 1997.

[33] R. E. Georgescu, E. G. Alexov, and M. R. Gunner. Combining conformational flexibility and continuum electrostatics for calculating pK_as in proteins. *Biophys. J.*, 83(4):1731–1748, Oct 2002.

[34] B. Rabenstein and E. W. Knapp. Calculated pH-dependent population and protonation of carbon-monoxy-myoglobin conformers. *Biophys. J.*, 80(3):1141–1150, Mar 2001.

[35] D. Bashford and M. Karplus. pK_as of ionizable groups in proteins: Atomic detail from a continuum electrostatic model. *Biochemistry*, 29(44):10219–10225, Nov 1990.

[36] D. Bashford and M. Karplus. Multiple-site titration curves of proteins: An analysis of exact and approximate methods for their calculations. *J. Phys. Chem.*, 95:9557–9561, 1991.

[37] J.-M. Mouesca, J. L. Chen, L. Noodleman, D. Bashford, and D. A. Case. Density functional/Poisson-Boltzmann calculations of redox potential s for iron-sulfur clusters. *J. Am. Chem. Soc.*, 116(26):11898–11914, 1994.

[38] J. Li, M. R. Nelson, C. Y. Peng, D. Bashford, and L. Noodleman. Incorporating protein environments in density functional theory: A self-consistent reaction field calculation of redox potentials of [2Fe2S] clusters in ferredoxin and phthalate dioxygenase reductase. *J. Phys. Chem. A*, 5639(98):6311–6324, 1998.

[39] E. Bombarda, T. Becker, and G. M. Ullmann. Influence of the membrane potential on the protonation of bacteriorhodopsin: Insights from electrostatic calculations into the regulation of proton pumping. *J. Am. Chem. Soc.*, 128(37):12129–12139, Sep 2006.

[40] N. Calimet and G. M. Ullmann. The influence of a transmembrane pH gradient on protonation probabilities of bacteriorhodopsin: The structural basis of the back-pressure effect. *J. Mol. Biol.*, 339(3):571–589, Jun 2004.

[41] E. J. Leggate, E. Bill, T. Essigke, G. M. Ullmann, and J. Hirst. Formation and characterization of an all-ferrous Rieske cluster and stabilization of the [2Fe-2S]^0 core by protonation. *Proc. Natl. Acad. Sci. USA*, 101(30):10913–10918, 2004.

[42] D. Sengupta, R. N. Behera, J. C. Smith, and G. M. Ullmann. The alpha helix dipole: Screened out? *Structure*, 13(6):849–855, 2005.

[43] G. M. Ullmann. The coupling of protonation and reduction in proteins with multiple redox centers: Theory, computational method, and application to cytochrome c3. *J. Phys. Chem. B*, 104(26):6293–6301, 2000.

[44] G. M. Ullmann, L. Noodleman, and D. A. Case. Density functional calculation of pK_a values and redox potentials in the bovine Rieske iron-sulfur protein. *J. Biol. Inorg. Chem.*, 7(6):632–639, 2002.

[45] T. Essigke. Reaction mechanism of Drosophila alcohol dehydrogenase. Master's thesis, Freie Universiät Berlin, Oct 2000.

[46] J. E. Nielsen, K. V. Andersen, B. Honig, R. W. Hooft, G. Klebe, G. Vriend, and R. C. Wade. Improving macromolecular electrostatics calculations. *Protein Eng.*, 12(8):657–662, Aug 1999.

[47] J. E. Nielsen and G. Vriend. Optimizing the hydrogen-bond network in Poisson-Boltzmann equation-based pK_a calculations. *Proteins*, 43(4):403–412, Jun 2001.

[48] E. Demchuk and R. C. Wade. Improving the continuum dielectric approach to calculating pK_as of ionizable groups in proteins. *J. Phys. Chem.*, 100(43):17373–17387, Oct 1996.

[49] R. Roxby and C. Tanford. Hydrogen ion titration curve of lysozyme in 6 M guanidine hydrochloride. *Biochemistry*, 10(18):3348–3352, Aug 1971.

[50] J. Antosiewicz, J. A. McCammon, and M. K. Gilson. Prediction of pH-dependent properties of proteins. *J. Mol. Biol.*, 238(3):415–436, May 1994.

[51] T. J. You and D. Bashford. Conformation and hydrogen ion titration of proteins: A continuum electrostatic model with conformational flexibility. *Biophys. J.*, 69(5):1721–1733, 1995.

[52] J. Antosiewicz, J. A. McCammon, and M. K. Gilson. The determinants of pK_as in proteins. *Biochemistry*, 35(24):7819–7833, Jun 1996.

[53] H. Ishikita and E.-W. Knapp. Electrostatic role of the non-heme iron complex in bacterial photosynthetic reaction center. *FEBS Lett.*, 580(18):4567–4570, Aug 2006.

[54] H. Ishikita, W. Saenger, J. Biesiadka, B. Loll, and E.-W. Knapp. How photosynthetic reaction centers control oxidation power in chlorophyll pairs P680, P700, and P870. *Biochemistry*, 103(26):9855–9860, 2006.

[55] H. Ishikita and E.-W. Knapp. Function of redox-active tyrosine in photosystem II. *Biophys. J.*, 90(11):3886–3896, Jun 2006.

[56] H. Ishikita, A. Galstyan, and E.-W. Knapp. Redox potential of the non-heme iron complex in bacterial photosynthetic reaction center. *Biochim. Biophys. Acta*, 1767(11):1300–1309, Nov 2007.

[57] J. Koepke, E.-M. Krammer, A. R. Klingen, P. Sebban, G. M. Ullmann, and G. Fritzsch. pH modulates the quinone position in the photosynthetic reaction center from Rhodobacter sphaeroides in the neutral and charge separated states. *J. Mol. Biol.*, 371(2):396–409, Aug 2007.

[58] D. M. Popović and A. A. Stuchebrukhov. Electrostatic study of the proton pumping mechanism in bovine heart cytochrome c oxidase. *J. Am. Chem. Soc.*, 126(6):1858–1871, Feb 2004.

[59] D. M. Popović, J. Quenneville, and A. A. Stuchebrukhov. DFT/electrostatic calculations of pK_a values in cytochrome c oxidase. *J. Phys. Chem. B*, 109(8):3616–3626, Mar 2005.

[60] M. Punnagai, T. Essigke, and G. M. Ullmann. Does deprotonation of Cu_B ligands play a role in the reaction mechanism of cytochrome c oxidase? *J. Am. Chem. Soc.*, 2007. Submitted.

[61] G. M. Ullmann. Relations between protonation constants and titration curves in polyprotic acids: A critical view. *J. Phys. Chem. B*, 107:1263–1271, 2003.

[62] J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 3rd edition, 1999. Includes bibliographical references (p. 785-790) and index.

[63] M. J. Holst. Adaptive numerical treatment of elliptic systems on manifolds. *Adv. Comp. Math.*, 15(1-4):139–191, 2001.

[64] J. G. Kirkwood. Theory of solutions of molecules containing widely separated charges with special application to zwitterions. *J. Chem. Phys.*, 2(7):351–361, 1934.

[65] M. Daune. *Molekulare Biophysik*. Vieweg, 1st edition, 1997.

[66] H. Sklenar, F. Eisenhaber, M. Poncin, and R. Lavery. *Including Solvent and Counterion Effects in the Force Fields of Macromolecular Mechanics: The Field Integrated Electrostatic Approach (FIESTA)*, pages 317–335. In: Theoretical Biochemistry & Molecular Biophysics. Eds. David L. Beveridge & Richard Lavery, 1990.

[67] M. J. Holst and F. Saied. Multigrid solution of the Poisson-Boltzmann equation. *J. Comput. Chem.*, 14(1):105–113, Jan 1993.

[68] M. J. Holst and F. Saied. Numerical-solution of the nonlinear Poisson-Boltzmann equation - Developing more robust and efficient methods. *J. Comput. Chem.*, 16(3):337–364, Mar 1995.

[69] R. E. Bank and M. J. Holst. A new paradigm for parallel adaptive meshing algorithms. *SIAM Review*, 45(2):291–323, 2003.

[70] T. J. You and D. Bashford. An analytical algorithm for the rapid determination of the solvent accessibility of points in a three-dimensional lattice around a solute molecule. *J. Comput. Chem.*, 16(6):743–757, 1995.

[71] B. Lee and F. M. Richards. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.*, 55(3):379–380, 1971.

[72] W. H. Press and B. P. Flannery. *Numerical Recipes in C The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1993.

[73] T. Weinmaier. Creating a Perl interface to the MEAD library with applications to pH titration. Master's thesis, Universität Bayreuth, Mar 6 2006.

[74] L. J. Bartolotti and K. Flurchick. *An Introduction to Density Functional Theory*, volume 7 of *Rev. Comp. Chem.*, chapter 4, pages 187–216. Wiley, 1995.

[75] A. St-Amant. *Density Functional Methods in Biomolecular Modeling*, volume 7 of *Rev. Comp. Chem.*, chapter 5, pages 217–259. Wiley, 1995.

[76] E. Schrödinger. Quantisierung als Eigenwertproblem. (Erste Mitteilung.). *Ann. Phys.*, 79(4):361–376, 1926.

[77] E. Schrödinger. Quantisierung als Eigenwertproblem. (Zweite Mitteilung.). *Ann. Phys.*, 6:479–527, 1926.

[78] E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev.*, 28(6):1049–1070, 1926.

[79] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3B):B864–B871, Nov 1964.

[80] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, Nov 1965.

[81] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: A critial analysis. *Can. J. Phys.*, 58:1200, 1980.

[82] D. M. Ceperley. Ground state of the fermion one-component plasma: A Monte Carlo study in two and three dimensions. *Phys. Rev. B*, 18(7):3126–3138, Oct 1978.

[83] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45(7):566–569, Aug 1980.

[84] J. P. Perdew and Y. Wang. Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation. *Phys. Rev. B*, 33(12):8800–8802, Jun 1986.

[85] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B*, 46(11):6671–6687, Sep 1992.

[86] A. D. Becke. Basis-set-free density-functional quantum-chemistry. *Int. J. Quant. Chem.*, pages 599–609, 1989.

[87] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. F. Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler. Chemistry with ADF. *J. Comput. Chem.*, 22(9):931–967, 2001.

[88] C. Fonseca Guerra, J. G. Snijders, G. te Velde, and E. J. Baerends. Towards an order-N DFT method. *Theor. Chem. Acc.*, 99(6):391–403, Oct 1998.

[89] E. J. Baerends, J. Autschbach, A. Bérces, F. M. Bickelhaupt, C. Bo, P. M. Boerrigter, L. Cavallo, D. P. Chong, L. Deng, R. M. Dickson, D. E. Ellis, M. van Faassen, L. Fan, T. H. Fischer, C. Fonseca Guerra, S. J. A. van Gisbergen, J. A. Groeneveld, Gritsenko O. V., M. Grüning, F. E. Harris, P. van den Hoek, C. R. Jacob, H. Jacobsen, L. Jensen, G. van Kessel, F. Kootstra, E. van Lenthe, D. A. McCormack, A. Michalak, J. Neugebauer, V. P. Nicu, V. P. Osinga, S. Patchkovskii, P. H. T. Philipsen, D. Post, C. C. Pye, W. Ravenek, P. Ros, P. R. T. Schipper, G. Schreckenbach, J. G. Snijders, M. Solà, M. Swart, D. Swerhone, G. te Velde, P. Vernooijs, L. Versluis, L. Visscher, O. Visser, F. Wang, T. A. Wesolowski, E. M. van Wezenbeek, G. Wiesenekker, S. K. Wolff, T. K. Woo, A. L. Yakovlev, and T. Ziegler. *ADF2007.01*. SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, 2007. http://www.scm.com.

[90] R. K. Szilagyi and M. A. Winslow. On the accuracy of density functional theory for iron - sulfur clusters. *J. Comput. Chem.*, 27(12):1385–1397, 2006.

[91] W. J. Hehre, R. Ditchfield, and J. A. Pople. Self-consistent molecular-orbital methods .12. Further extensions of Gaussian-type basis sets for use in molecular-orbital studies of organic-molecules. *J. Chem. Phys.*, 56(5):2257–2261, 1972.

[92] R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople. Self-consistent molecular-orbital methods .20. Basis set for correlated wave-functions. *J. Chem. Phys.*, 72(1):650–654, 1980.

[93] C. M. Breneman and K. B. Wiberg. Determining atom-centered monopoles from molecular electrostatic potentials. The need for high sampling density in formamide conformational analysis. *J. Comput. Chem.*, 11(3):361–373, 1990.

[94] R. N. Behera and G. M. Ullmann. Conformational dependence of charge fits. in preparation.

[95] M. Swart, P. T. van Duijnen, and J. G. Snijders. A charge analysis derived from an atomic multipole expansion. *J. Comput. Chem.*, 22(1):79–88, 2001.

[96] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–217, 1983.

[97] M. Karplus and G. A. Petsko. Molecular dynamics simulations in biology. *Nature*, 347:631–639, 1991.

[98] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnik, T. Ngo, D. T. Nguyen, B. Prodhom, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, J. Wiorkiewicz-Kuczera, and M. Karplus. Self-consistent parameterization

of biomolecules for molecular mod eling and condensed phase simulation. *FASEB J.*, 6(A):143, 1992.

[99] C. L. Brooks. *CHARMM Documentation*. The Scripps Research Institute, http://www.scripps.edu/brooks/.

[100] A. C. Vaiana, Z. Cournia, I. B. Costescu, and J. C. Smith. AFMM: A molecular mechanics force field vibrational parametrization program. *Comput. Phys. Commun.*, 167(1):34–42, 2005.

[101] R. L. Dunbrack. Rotamer libraries in the 21st century. *Curr. Opin. Struct. Biol.*, 12(4):431–440, 2002.

[102] R. L. Dunbrack and F. E. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci.*, 6:1661–1681, 1997.

[103] I. K. McDonald and J. M. Thornton. Satisfying hydrogen bonding potential in proteins. *J. Mol. Biol.*, 238:777–793, 1994.

[104] I. L. McDonald. *HBPlus.man - A HBPlus v3.06 Manual*. available at http://www.biochem.ucl.ac.uk/ mcdonald/hbplus/.

[105] J. M. Word, S. C. Lovell, T. H. LaBean, M. C. Taylor, M. E. Zalis, B. K. Presley, J. S. Richardson, and D. C. Richardson. Visualizing and quantifying molecular goodness-of-fit: Small-probe contact dots with explicit hydrogen atoms. *J. Mol. Biol.*, 285:1711–1733, 1999.

[106] J. M. Word, S. C. Lovell, J. S. Richardson, and D. C. Richardson. Asparagine and glutamine: Using hydrogen atom contacts in the choice of side-chain amide orientation. *J. Mol. Biol.*, 285:1735–1747, 1999.

[107] G. Vriend. WHAT IF: A molecular modeling and drug design program. *J. Mol. Graphics*, 8(1):52–56, 29, Mar 1990.

[108] R. W. Hooft, C. Sander, and G. Vriend. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. *Proteins*, 26:363–376, 1996.

[109] X. Lopez, M. Schaefer, A. Dejaegere, and M. Karplus. Theoretical evaluation of pK_a in phosphoranes: Implications for phosphate ester hydrolysis. *J. Am. Chem. Soc.*, 124(18):5010–5018, 2002.

[110] C. Tanford and R. Roxby. Interpretation of protein titration curves. Application to lysozyme. *Biochemistry*, 11(11):2192–2198, May 1972.

[111] M. K. Gilson. Multiple-site titration and molecular modelling: Two rapid methods for computing energies and forces for ionizable groups in proteins. *Proteins*, 15:266–282, 1993.

[112] A. Karshikoff. A simple algorithm for the calculation of multiple site titration curves. *Protein Eng.*, 8:243–248, 1995.

[113] P. Beroza, D. R. Fredkin, M. Y. Okamura, and G. Feher. Protonation of interacting residues in a protein by a Monte Carlo method: Application to lysozyme and the photosynthetic reaction center. *Proc. Natl. Acad. Sci. USA*, 88:5804–5808, 1991.

[114] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[115] M. K. Gilson. Introduction to continuum electrostatics, with molecular applications. http://gilsonlab.umbi.umd.edu, Mar 2 2002.

[116] M. Born. Volumen und Hydrationswärme der Ionen. *Z. Phys.*, 1:45–48, 1920.

[117] S. J. Wodak and J. Janin. Analytical approximation to the accessible surface area of proteins. *Proc. Natl. Acad. Sci. USA*, 77(4):1736–1740, Apr 1980.

[118] W. Hasel, T. F. Hendrickson, and W. C. Still. A rapid approximation to the solvent accessible surface areas of atoms. *Tetrahedron Computer Methodology*, 1:103–116, 1988.

[119] R. M. Noyes. Thermodynamics of ion hydration as a measure of effective dielectric properties of water. *J. Am. Chem. Soc.*, 1(1956):513, 1962.

[120] H. Reiss and A. Heller. The absolute potential of the standard hydrogen electrode: A new estimate. *J. Phys. Chem.*, 89(1966):4207–4213, 1985.

[121] G. M. Ullmann and E.-W. Knapp. Electrostatic models for computing protonation and redox equilibria in proteins. *Eur. Biophys. J.*, 28(7):533–551, 1999.

[122] G. M. Ullmann. Relations between protonation constants and titration curves in polyprotic acids: A critical view. *J. Phys. Chem. B*, 107(5):1263–1271, 2003.

[123] D. Bashford, D. A. Case, C. Dalvit, L. Tennant, and P. E. Wright. Electrostatic calculations of side-chain pK_a values in myoglobin and comparison with NMR data for histidines. *Biochemistry*, 32(31):8045–8056, Aug 1993.

[124] Z. Zhu and M. R. Gunner. Energetics of quinone-dependent electron and proton transfers in Rhodobacter sphaeroides photosynthetic reaction centers. *Biochemistry*, 44(1):82–96, Jan 2005.

[125] Gernot Kieseritzky. *The TAPBS documentation*. AG Knapp, FU Berlin, http://agknapp.chemie.fu-berlin.de/karlsberg/, 0.1 edition, 2007.

[126] G. M. Ullmann, L. Noodleman, and D. A. Case. Density functional calculation of pK_a values and redox potentials in the bovine Rieske iron-sulfur protein. *J. Biol. Inorg. Chem.*, 7(6):632–639, Jun 2002.

[127] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucl. Acids Res.*, 28(1):235–242, Jan 2000.

[128] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112(3):535–542, May 1977.

[129] Object Management Group (OMG), http://www.omg.org. *Unified Modeling Language (UML)*, 2.1.1 edition, Feb 2007.

[130] M. Hitz, G. Kappel, E. Kapsammer, and W. Retschitzegger. *UML@Work*. dpunkt.verlag, 2005. ISBN: 3898642615.

[131] *Grid computing: Making the global infrastructure a reality*, Wiley series in communications networking & distributed systems, New York, 2003. J. Wiley. Includes bibliographical references and index.

[132] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery. General atomic and molecular electronic-structure system. *J. Comput. Chem.*, 14(11):1347–1363, Nov 1993.

[133] M. S. Gordon and M. W. Schmidt. Advances in electronic structure theory: Gamess a decade later. In C.E. Dykstra, G. Frenking, K.S. Kim, and G.E. Scuseria, editors, *Theory and Applications of Computational Chemistry: The first forty years*, pages 1167–1189, Amsterdam, 2005. Elsevier.

[134] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople. Gaussian 03, Revision C.02, 2004. Gaussian, Inc., Wallingford, CT, 2004.

[135] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns Elements of Reusable Object-Oriented Software*. Addison Wesley, Oct 2005.

[136] M. J. Dominus. *Higher-Order Perl*. Elsevier, Amsterdam, 2005.

[137] W. Wulf and M. Shaw. Global variables considered harmful. In *ACM SIGPLAN Notices*, volume 8 (2), pages 28–34, Feb 1973.

[138] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, 53:484–512, 1984.

[139] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comp. Phys.*, 82:64–84, 1989.

[140] D. M. Beazley. SWIG: An easy to use tool for integrating scripting languages with C and C++. In *4th Annual Tcl/Tk Workshop*, Monterey, CA, Jul 1996.

[141] W. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *J. Mol. Graphics*, 14:33–38, 1996.

[142] F. Dickert. The coupling of reduction and protonation in diheme cytochromes. Master's thesis, Universität Bayreuth, Dec 2005.

[143] D. A. Case, T. A. Darden, T. E. Cheatham III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, K. M. Merz, D. A. Pearlman, M. Crowley, R. C. Walker, W. Zhang, B. Wang, S. Hayik, A. Roitberg, G. Seabra, K. F. Wong, F. Paesani, X. Wu, S. Brozell, V. Tsui, H. Gohlke, L. Yang, C. Tan, J. Mongan, V. Hornak, G. Cui, P. Beroza, D. H. Mathews, C. Schafmeister, W. S. Ross, and P. A. Kollman. *AMBER 9*. University of California, San Francisco, 2006.

[144] H. Bekker, H. J. Berendsen, E. J. Dijkstra, S. Achterop, R. van Drunen, D. van der Spoel, A. Sijbers, H. Keegstra, B. Reitsma, and M. K. R. Renardus. *Physics Computing*, volume 92, chapter GROMACS: A parallel computer for molecular dynamics simulations. World Scientific, Singapore, 1993.

[145] R. Morales, M. H. Charon, G. Hudry-Clergeon, Y. Pétillot, S. Norager, M. Medina, and M. Frey. Refined x-ray structures of the oxidized, at 1.3 Å, and reduced, at 1.17 Å, [2Fe-2S] ferredoxin from the cyanobacterium Anabaena PCC7119 show redox-linked conformational changes. *Biochemistry*, 38(48):15764–73, Nov 1999.

[146] D. Sitkoff, K. A. Sharp, and B. Honig. Accurate calculation of hydration free energies using macroscopic solvent models. *J. Phys. Chem.*, 98:1978–1988, 1994.

[147] A. Bondi. Van der Waals volumes and radii. *J. Phys. Chem.*, 68(3):441–451, 1964.

[148] N. Homeyer, T. Essigke, H. Meiselbach, G. M. Ullmann, and H. Sticht. Effect of HPr phosphorylation on structure, dynamics, and interactions in the course of transcriptional control. *J. Mol. Model.*, 13(3):431–444, Mar 2007.

[149] N. Homeyer, T. Essigke, G. M. Ullmann, and H. Sticht. Effects of histidine protonation and phosphorylation on histidine-containing phosphocarrier protein structure, dynamics, and physicochemical properties. *Biochemistry*, 46(43):12314–12326, Oct 2007.

[150] W. Kundig, S. Ghosh, and S. Roseman. Phosphate bound to histidine in a protein as an intermediate in a novel phospho-transferase system. *Proc. Natl. Acad. Sci. USA*, 52:1067–1074, Oct 1964.

[151] F. Titgemeyer and W. Hillen. Global control of sugar metabolism: A gram-positive solution. *Antonie Van Leeuwenhoek*, 82(1-4):59–71, Aug 2002.

[152] G. F. Audette, R. Engelmann, W. Hengstenberg, J. Deutscher, K. Hayakawa, J. W. Quail, and L. T. Delbaere. The 1.9 Å resolution structure of phospho-serine 46 HPr from Enterococcus faecalis. *J. Mol. Biol.*, 303(4):545–553, Nov 2000.

[153] J. Reizer, S. Bachem, A. Reizer, M. Arnaud, M. H. Saier, and J. Stülke. Novel phosphotransferase system genes revealed by genome analysis - The complete complement of PTS proteins encoded within the genome of Bacillus subtilis. *Microbiology*, 145 (Pt 12):3419–3429, Dec 1999.

[154] J. W. Lengeler, K. Jahreis, and U. F. Wehmeier. Enzymes II of the phosphoenol pyruvate-dependent phosphotransferase systems: Their structure and function in carbohydrate transport. *Biochim. Biophys. Acta*, 1188(1-2):1–28, Nov 1994.

[155] A. Galinier, M. Kravanja, R. Engelmann, W. Hengstenberg, M. C. Kilhoffer, J. Deutscher, and J. Haiech. New protein kinase and protein phosphatase families mediate signal transduction in bacterial catabolite repression. *Proc. Natl. Acad. Sci. USA*, 95(4):1823–1828, Feb 1998.

[156] J. Reizer, C. Hoischen, F. Titgemeyer, C. Rivolta, R. Rabus, J. Stülke, D. Karamata, M. H. Saier, and W. Hillen. A novel protein kinase that controls carbon catabolite repression in bacteria. *Mol. Microbiol.*, 27(6):1157–1169, Mar 1998.

[157] D. Brochu and C. Vadeboncoeur. The HPr(Ser) kinase of Streptococcus salivarius: Purification, properties, and cloning of the HPrK gene. *J. Bacteriol.*, 181(3):709–717, Feb 1999.

[158] M. Kravanja, R. Engelmann, V. Dossonnet, M. Blüggel, H. E. Meyer, R. Frank, A. Galinier, J. Deutscher, N. Schnell, and W. Hengstenberg. The HPrK gene of Enterococcus faecalis encodes a novel bifunctional enzyme: the HPr kinase/phosphatase. *Mol. Microbiol.*, 31(1):59–66, Jan 1999.

[159] M. S. Moreno, B. L. Schneider, R. R. Maile, W. Weyler, and M. H. Saier. Catabolite repression mediated by the CcpA protein in Bacillus subtilis: Novel modes of regulation revealed by whole-genome analyses. *Mol. Microbiol.*, 39(5):1366–1381, Mar 2001.

[160] G. L. Lorca, Y. J. Chung, R. D. Barabote, W. Weyler, C. H. Schilling, and M. H. Saier. Catabolite repression and activation in Bacillus subtilis: dependency on CcpA, HPr, and HPrK. *J. Bacteriol.*, 187(22):7826–7839, Nov 2005.

[161] V. Charrier, E. Buckley, D. Parsonage, A. Galinier, E. Darbon, M. Jaquinod, E. Forest, J. Deutscher, and A. Claiborne. Cloning and sequencing of two enterococcal glpK genes and regulation of the encoded glycerol kinases by phosphoenolpyruvate-dependent, phosphotransferase system-catalyzed phosphorylation of a single histidyl residue. *J. Biol. Chem.*, 272(22):14166–14174, May 1997.

[162] Z. Jia, M. Vandonselaar, W. Hengstenberg, J. W. Quail, and L. T. Delbaere. The 1.6 Å structure of histidine-containing phosphotransfer protein HPr from Streptococcus faecalis. *J. Mol. Biol.*, 236(5):1341–1355, Mar 1994.

[163] E. A. Bienkiewicz and K. J. Lumb. Random-coil chemical shifts of phosphorylated amino acids. *J. Biomol. NMR*, 15(3):203–206, Nov 1999.

[164] B. E. Jones, P. Rajagopal, and R. E. Klevit. Phosphorylation on histidine is accompanied by localized structural changes in the phosphocarrier protein, HPr from Bacillus subtilis. *Protein Sci.*, 6(10):2107–2119, Oct 1997.

[165] S. T. Wlodek, J. Antosiewicz, and J. A. McCammon. Prediction of titration properties of structures of a protein derived from molecular dynamics trajectories. *Protein Sci.*, 6(2):373–382, Feb 1997.

[166] N. A. van Nuland, I. W. Hangyi, R. C. van Schaik, H. J. Berendsen, W. F. van Gunsteren, R. M. Scheek, and G. T. Robillard. The high-resolution structure of the histidine-containing phosphocarrier protein HPr from Escherichia coli determined by restrained molecular dynamics from nuclear magnetic resonance nuclear Overhauser effect data. *J. Mol. Biol.*, 237(5):544–559, Apr 1994.

[167] N. A. van Nuland, R. Boelens, R. M. Scheek, and G. T. Robillard. High-resolution structure of the phosphorylated form of the histidine-containing phosphocarrier protein HPr from Escherichia coli determined by restrained molecular dynamics from NMR-NOE data. *J. Mol. Biol.*, 246(1):180–193, Feb 1995.

[168] T. Maurer, R. Döker, A. Görler, W. Hengstenberg, and H. R. Kalbitzer. Three-dimensional structure of the histidine-containing phosphocarrier protein (HPr) from Enterococcus faecalis in solution. *Eur. J. Biochem.*, 268(3):635–644, Feb 2001.

[169] O. Herzberg, P. Reddy, S. Sutrina, M. H. Saier, J. Reizer, and G. Kapadia. Structure of the histidine-containing phosphocarrier protein HPr from Bacillus subtilis at 2.0- Å resolution. *Proc. Natl. Acad. Sci. USA*, 89(6):2499–2503, Mar 1992.

[170] H. R. Kalbitzer, A. Görler, H. Li, P. V. Dubovskii, W. Hengstenberg, C. Kowolik, H. Yamada, and K. Akasaka. 15N and 1H NMR study of histidine containing protein (HPr) from Staphylococcus carnosus at high pressure. *Protein Sci.*, 9(4):693–703, Apr 2000.

[171] T. Maurer, S. Meier, N. Kachel, C. E. Munte, S. Hasenbein, B. Koch, W. Hengstenberg, and H. R. Kalbitzer. High-resolution structure of the histidine-containing phosphocarrier protein (HPr) from Staphylococcus aureus and characterization of its interaction with the bifunctional HPr kinase/phosphorylase. *J. Bacteriol.*, 186(17):5906–5918, Sep 2004.

[172] E. B. Waygood. The structure and function of HPr. *Biochem. Cell Biol.*, 76(2-3):359–367, 1998.

[173] G. Cornilescu, B. R. Lee, C. C. Cornilescu, G. Wang, A. Peterkofsky, and G. M. Clore. Solution structure of the phosphoryl transfer complex between the cytoplasmic A domain of the mannitol transporter II. Mannitol and HPr of the Escherichia coli phosphotransferase system. *J. Biol. Chem.*, 277(44):42289–42298, Nov 2002.

[174] G. Wang, J. M. Louis, M. Sondej, Y. J. Seok, A. Peterkofsky, and G. M. Clore. Solution structure of the phosphoryl transfer complex between the signal transducing proteins HPr and IIA (glucose) of the Escherichia coli phosphoenolpyruvate:sugar phosphotransferase system. *EMBO J.*, 19(21):5635–5649, Nov 2000.

[175] D. C. Williams, M. Cai, J. Y. Suh, A. Peterkofsky, and G. M. Clore. Solution NMR structure of the 48-kDa IIA Mannose-HPr complex of the Escherichia coli mannose phosphotransferase system. *J. Biol. Chem.*, 280(21):20775–20784, May 2005.

[176] W. R. Rypniewski, D. R. Breiter, M. M. Benning, G. Wesenberg, B. H. Oh, J. L. Markley, I. Rayment, and H. M. Holden. Crystallization and structure determination to 2.5- Å resolution of the oxidized [2Fe-2S] ferredoxin isolated from Anabaena 7120. *Biochemistry*, 30(17):4126–4131, Apr 1991.

[177] J. K. Hurley, A. M. Weber-Main, A. E. Hodges, M. T. Stankovich, M. M. Benning, H. M. Holden, H. Cheng, B. Xia, J. L. Markley, C. Genzor, C. Gomez-Moreno, R. Hafezi, and G. Tollin. Iron-sulfur cluster cysteine-to-serine mutants of Anabaena [2Fe-2S] ferredoxin exhibit unexpected redox properties and are competent in electron transfer to ferredoxin:NADP+ reductase. *Biochemistry*, 36(49):15109–15117, Dec 1997.

[178] M. Martinez-Julvez, M. Medina, J. K. Hurley, R. Hafezi, T. B. Brodie, G. Tollin, and C. Gomez-Moreno. Lys75 of Anabaena ferredoxin-NADP(+) reductase is a critical residue for binding ferredoxin and flavodoxin during electron transfer. *Biochemistry*, 37(39):13604–13613, Sep 1998.

[179] H. Beinert, R. H. Holm, and E. Münck. Iron-sulfur clusters: Nature's modular, multipurpose structures. *Science*, 277(5326):653–659, Aug 1997.

[180] L. B. Dugad, G. N. La Mar, L. Banci, and I. Bertini. Identification of localized redox states in plant-type two-iron ferredoxins using the nuclear Overhauser effect. *Biochemistry*, 29(9):2263–2271, Mar 1990.

[181] L. Banci, I. Bertini, and C. Luchinat. The H-1-NMR parameters of magnetically coupled dimers - The Fe2S2 proteins as an example. *Struct. Bond.*, 72:113–136, 1990.

[182] H. Boehme and B. Schrautemeier. Comparative characterization of ferredoxins from heterocysts and vegetative cells of Anabaena variabilis. *Biochim. Biophys. Acta*, 891(1):1–7, Mar 1987.

[183] Z. Salamon and G. Tollin. Cyclic voltammetric behavior of [2Fe-2S] ferredoxins at a lipid bilayer modified electrode. *Bioelectrochem. and Bioenerg.*, 27(3):381–391, Jun 1992.

[184] J. K. Hurley, Z. Salamon, T. E. Meyer, J. C. Fitch, M. A. Cusanovich, J. L. Markley, H. Cheng, B. Xia, Y. K. Chae, and M. Medina. Amino acid residues in Anabaena ferredoxin crucial to interaction with ferredoxin-NADP+ reductase: Site-directed mutagenesis and laser flash photolysis. *Biochemistry*, 32(36):9346–9354, Sep 1993.

[185] M. Vidakovic, G. Fraczkiewicz, B. C. Dave, R. S. Czernuszewicz, and J. P. Germanas. The environment of [2Fe-2S] clusters in ferredoxins: The role of residue 45 probed by site-directed mutagenesis. *Biochemistry*, 34(42):13906–13913, Oct 1995.

[186] J. K. Hurley, A. M. Weber-Main, M. T. Stankovich, M. M. Benning, J. B. Thoden, J. L. Vanhooke, H. M. Holden, Y. K. Chae, B. Xia, H. Cheng, J. L. Markley, M. Martinez-Júlvez, C. Gómez-Moreno, J. L. Schmeits, and G. Tollin. Structure-function relationships in Anabaena ferredoxin: Correlations between x-ray crystal structures, reduction potentials, and rate constants of electron transfer to ferredoxin:NADP+ reductase for site-specific ferredoxin mutants. *Biochemistry*, 36(37):11100–11117, Sep 1997.

[187] P. Stephens, D. Jollie, and A. Warshel. Protein control of redox potentials of iron-sulfur proteins. *Chem. Rev.*, 96(7):2491–2514, Nov 1996.

[188] F. Pizzitutti, P. Setif, and M. Marchi. Theoretical investigation of the "CO in"- "CO out" isomerization in a [2Fe-2S] ferredoxin: Free energy profiles and redox states. *J. Am. Chem. Soc.*, 125(49):15224–15232, 2003.

[189] R. Morales, M. Frey, and J.-M. Mouesca. An approach based on quantum chemistry calculations and structural analysis of a [2Fe-2S] ferredoxin that reveal a redox-linked switch in the electron-transfer process to the Fd-NADP+ reductase. *J. Am. Chem. Soc.*, 124(23):6714–6722, Jun 2002.

[190] P. V. Rao and R. H. Holm. Synthetic analogues of the active sites of iron-sulfur proteins. *Chem. Rev.*, 104(2):527–559, Feb 2004.

[191] J. J. Mayerle, S. E. Denmark, B. V. Depamphilis, J. A. Ibers, and R. H. Holm. Synthetic analogs of active-sites of iron-sulfur proteins .11. Synthesis and properties of complexes containing Fe2S2 core and structures of Bis[Ortho-Xylyl-α,α'-Dithiolato-μ-Sulfido-Ferrate(III)] and Bis[Para-Tolylthiolato-μ-Sulfido-Ferrate(III)] dianions. *J. Am. Chem. Soc.*, 97(5):1032–1045, 1975.

[192] P. L. Fabre, D. Demontauzon, and R. Poilblanc. Potential-PSR diagram of thiophenolate iron-sulfur derivatives. *Bull. Soc. Chim. Fr.*, 1(2):123–129, Mar-Apr 1991.

[193] H. Michel, J. Behr, A. Harrenga, and A. Kannt. Cytochrome c oxidase: Structure and spectroscopy. *Annu. Rev. Biophys. Biomol. Struct.*, 27:329–356, 1998.

[194] S. Ferguson-Miller and G. T. Babcock. Heme/copper terminal oxidases. *Chem. Rev.*, 96(7):2889–2907, Nov 1996.

[195] O. M. H. Richter and B. Ludwig. Cytochrome c oxidase - Structure, function, and physiology of a redox-driven molecular machine. *Rev. Physiol. Biochem. Pharmacol.*, 147:47–74, 2003.

[196] Elisa Fadda, Nilmadhab Chakrabarti, and Régis Pomès. Acidity of a Cu-bound histidine in the binuclear center of cytochrome c oxidase. *J. Phys. Chem. B*, 109(47):22629–22640, Dec 2005.

[197] E. Fadda, N. Chakrabarti, and R. Pomes. Reply to "Comment on acidity of a Cu-bound histidine in the binuclear center of cytochrome c oxidase". *J. Phys. Chem. B*, 110(34):17288–17289, 2006.

[198] A. A. Stuchebrukhov and D. M. Popović. Comment on "Acidity of a Cu-bound histidine in the binuclear center of cytochrome c oxidase". *J. Phys. Chem. B*, 110(34):17286–17287; discussion 17288–17289, Aug 2006.

[199] S. Yoshikawa, K. Shinzawa-Itoh, R. Nakashima, R. Yaono, E. Yamashita, N. Inoue, M. Yao, M. J. Fei, C. P. Libeu, T. Mizushima, H. Yamaguchi, T. Tomizaki, and T. Tsukihara. Redox-coupled crystal structural changes in bovine heart cytochrome c oxidase. *Science*, 280(5370):1723–1729, Jun 1998.

[200] V. Barone and M. Cossi. Quantum calculation of molecular energies and energy gradients in solution by a conductor solvent model. *J. Phys. Chem. A*, 102(11):1995–2001, Mar 1998.

[201] A. M. Toth, M. D. Liptak, D. L. Phillips, and G. C. Shields. Accurate relative pK_a calculations for carboxylic acids using complete basis set and Gaussian-n models combined with continuum solvation methods. *J. Chem. Phys.*, 114(10):4595–4606, Mar 8 2001.

[202] M. Cossi, N. Rega, G. Scalmani, and V. Barone. Energies, structures, and electronic properties of molecules in solution with the C-PCM solvation model. *J. Comp. Chem.*, 24(6):669–681, Apr 2003.

[203] C. Ribacka, M. I. Verkhovsky, I. Belevich, D. A. Bloch, A. Puustinen, and M. Wikstrom. An elementary reaction step of the proton pump is revealed by mutation of tryptophan-164 to phenylalanine in cytochrome c oxidase from Paracoccus denitrificans. *Biochemistry*, 44(50):16502–16512, Dec 2005.

[204] F. MacMillan, K. Budiman, H. Angerer, and H. Michel. The role of tryptophan 272 in the Paracoccus denitrificans cytochrome c oxidase. *FEBS Lett.*, 580(5):1345–1349, Feb 2006.

[205] P. Lemma-Gray, S. T. Weintraub, C. A. Carroll, A. Musatov, and N. C. Robinson. Tryptophan 334 oxidation in bovine cytochrome c oxidase subunit I involves free radical migration. *FEBS Lett.*, 581(3):437–442, Feb 6 2007.

[206] F. G. M. Wiertz, O. M. H. Richter, A. V. Cherepanov, F. MacMillan, B. Ludwig, and S. de Vries. An oxo-ferryl tryptophan radical catalytic intermediate in cytochrome c and quinol oxidases trapped by microsecond freeze-hyperquenching (MHQ). *FEBS Lett.*, 575(1-3):127–130, Sep 2004.

[207] H. Nohl, A. V. Kozlov, K. Staniek, and L. Gille. The multiple functions of coenzyme Q. *Bioorg. Chem.*, 29(1):1–13, Feb 2001.

[208] F. L. Crane. Biochemical functions of coenzyme q_{10}. *J. Am. Coll. Nutr.*, 20(6):591–598, Dec 2001.

[209] M. Turunen, J. Olsson, and G. Dallner. Metabolism and function of coenzyme Q. *Biochim. Biophys. Acta*, 1660(1-2):171–199, Jan 2004.

[210] J. B. Conant and L. F. Fieser. Free and total energy changes in the reduction of quinones. *J. Am. Chem. Soc.*, 44(11):2480–2493, 1922.

[211] J. B. Conant and L. F. Fieser. Reduction potentials of quinones. I. The effect of the solvent on the potentials of certain benzoquinones. *J. Am. Chem. Soc.*, 45(9):2194–2218, 1923.

[212] J. B. Conant and L. F. Fieser. Reduction potentials of quinones. II. The potentials of certain derivatives of benzoquinone, naphthoquinone and anthraquinone. *J. Am. Chem. Soc.*, 46(8):1858–1881, 1924.

[213] J. H. Baxendale and H. R. Hardy. The ionization constants of some hydroquinones. *Trans. Faraday Soc.*, 49(10):1140–1144, 1953.

[214] C. A. Bishop and L. K. J. Tong. Equilibria of substituted semiquinones at high pH. *J. Am. Chem. Soc.*, 87(3):501–505, 1965.

[215] R. L. Willson. Pulse radiolysis studies of electron transfer in aqueous quinone solutions. *Trans. Faraday Soc.*, 67(586):3020–3029, 1971.

[216] Y. A. Ilan, G. Czapski, and D. Meisel. One-electron transfer redox potentials of free-radicals .1. Oxygen-superoxide system. *Biochim. Biophys. Acta*, 430(2):209–224, 1976.

[217] A. D. Becke. A new mixing of Hartree-Fock and local density-functional theories. *J. Chem. Phys.*, 98(2):1372–1377, Jan 1993.

[218] R. S. Mulliken. Electronic population analysis on LCAO-MO molecular wave functions .1. *J. Chem. Phys.*, 23(10):1833–1840, 1955.

[219] R. S. Mulliken. Electronic population analysis on LCAO-MO molecular wave functions .2. Overlap populations, bond orders, and covalent bond energies. *J. Chem. Phys.*, 23(10):1841–1846, 1955.

[220] R. S. Mulliken. Electronic population analysis on lcao-mo molecular wave functions .3. effects of hybridization on overlap and gross ao populations. *J. Chem. Phys.*, 23(12):2338–2342, 1955.

[221] R. S. Mulliken. Electronic population analysis on LCAO-MO molecular wave functions .4. Bonding and antibonding in LCAO and valence-bond theories. *J. Chem. Phys.*, 23(12):2343–2346, 1955.

[222] J. E. Carpenter and F. Weinhold. Analysis of the geometry of the hydroxymethyl radical by the different hybrids for different spins natural bond orbital procedure. *J. Mol. Struct. (Theochem)*, 46:41–62, Aug 1988.

[223] J. P. Foster and F. Weinhold. Natural hybrid orbitals. *J. Am. Chem. Soc.*, 102(24):7211–7218, 1980.

[224] A. E. Reed and F. Weinhold. Natural bond orbital analysis of near-Hartree-Fock water dimer. *J. Chem. Phys.*, 78(6):4066–4073, 1983.

[225] A. E. Reed, R. B. Weinstock, and F. Weinhold. Natural-population analysis. *J. Chem. Phys.*, 83(2):735–746, 1985.

[226] A. E. Reed, L. A. Curtiss, and F. Weinhold. Intermolecular interactions from a natural bond orbital, donor-acceptor viewpoint. *Chem. Rev.*, 88(6):899–926, Sep-Oct 1988.

[227] B. H. Besler, K. M. Merz, and P. A. Kollman. Atomic charges derived from semiempirical methods. *J. Comp. Chem.*, 11(4):431–439, May 1990.

[228] U. C. Singh and P. A. Kollman. An approach to computing electrostatic charges for molecules. *J. Comp. Chem.*, 5(2):129–145, 1984.

[229] M. Cossi, V. Barone, B. Mennucci, and J. Tomasi. Ab initio study of ionic solutions by a polarizable continuum dielectric model. *Chem. Phys. Lett.*, 286(3-4):253–260, Apr 1998.

[230] B. Mennucci and J. Tomasi. Continuum solvation models: A new approach to the problem of solute's charge distribution and cavity boundaries. *J. Chem. Phys.*, 106(12):5151–5158, Mar 1997.

[231] M. Cossi, G. Scalmani, N. Rega, and V. Barone. New developments in the polarizable continuum model for quantum mechanical and classical calculations on molecules in solution. *J. Chem. Phys.*, 117(1):43–54, Jul 1 2002.

[232] B. Mennucci, E. Cances, and J. Tomasi. Evaluation of solvent effects in isotropic and anisotropic dielectrics and in ionic solutions with a unified integral equation method: Theoretical bases, computational implementation, and numerical applications. *J. Phys. Chem. B*, 101(49):10506–10517, Dec 4 1997.

[233] J. Tomasi, B. Mennucci, and E. Cances. The IEF version of the PCM solvation method: An overview of a new method addressed to study molecular solutes at the QM ab initio level. *J. Mol. Struct. (Theochem)*, 464(1-3):211–226, May 1999.

[234] D. M. Chipman. Reaction field treatment of charge penetration. *J. Chem. Phys.*, 112(13):5558–5565, Apr 1 2000.

[235] E. Cances and B. Mennucci. Comment on "Reaction field treatment of charge penetration". *J. Chem. Phys.*, 114(10):4744–4745, Mar 8 2001.

[236] M. D. Liptak, K. C. Gross, P. G. Seybold, S. Feldgus, and G. C. Shields. Absolute pK_a determinations for substituted phenols. *J. Am. Chem. Soc.*, 124(22):6421–6427, Jun 5 2002.

[237] J. O'M. Bockris and S. U. M. Khan. *Surface electrochemistry: A molecular level approach*. Plenum, New York, 1993. Includes bibliographical references and index.

[238] M. D. Tissandier, K. A. Cowen, W. Y. Feng, E. Gundlach, M. H. Cohen, A. D. Earhart, J. V. Coe, and T. R. Tuttle. The proton's absolute aqueous enthalpy and Gibbs free energy of solvation from cluster-ion solvation data. *J. Phys. Chem. A*, 5639(98):7787–7794, 1998.

[239] M. D. Liptak and G. C. Shields. Accurate pK_a calculations for carboxylic acids using complete basis set and Gaussian-n models combined with CPCM continuum solvation methods. *J. Am. Chem. Soc.*, 123(30):7314–7319, Aug 1 2001.

[240] M. S. Till, T. Becker, T. Essigke, and G. M. Ullmann. Simulating the proton transfer in gramicidin A by a sequential dynamical Monte Carlo method. *J. Phys. Chem.*, 2007. In Preparation.

[241] J. Westbrook, N. Ito, H. Nakamura, K. Henrick, and H. M. Berman. PDBML: The representation of archival macromolecular structure data in XML. *Bioinformatics*, 21(7):988–992, Apr 2005.

Appendices

APPENDIX A

FILE TYPES AND FORMATS

A.1 Coordinate Containing Files

A.1.1 The PDB File

The pdb-file format is documented at[1].

```
1         1         2         3         4         5         6         7         8
2 12345678901234567890123456789012345678901234567890123456789012345678901234567890
3 ATOM    145  N   VAL A  25      32.433  16.336  57.540  1.00 11.92           N
4 ATOM    146  CA  VAL A  25      31.132  16.439  58.160  1.00 11.85           C
5 ATOM    147  C   VAL A  25      30.447  15.105  58.363  1.00 12.34           C
6 ATOM    148  O   VAL A  25      29.520  15.059  59.174  1.00 15.65           O
7 ATOM    149  CB AVAL A  25      30.385  17.437  57.230  0.28 13.88           C
8 ATOM    150  CB BVAL A  25      30.166  17.399  57.373  0.72 15.41           C
9 ATOM    151  CG1AVAL A  25      28.870  17.401  57.336  0.28 12.64           C
10 ATOM   152  CG1BVAL A  25      30.805  18.788  57.449  0.72 15.11           C
11 ATOM   153  CG2AVAL A  25      30.835  18.826  57.661  0.28 13.58           C
12 ATOM   154  CG2BVAL A  25      29.909  16.996  55.922  0.72 13.25           C
```

The `ATOM` and `HETATM` records are character position based format without field separators. The file format was dictated by the IBM 80 column punch card format introduced 1928 but common until the mid-1970s. However, most programs do not follow this definition strictly, but provide reading and writing routines, which deviate from the original format in one way or another. Since alternative location indicators (position 17) are not very common and splitting whitespace separated columns is very simple to program in most languages, it is done in many programs. Another problem is that the documentation does not mention the alignment of fields precisely. For example, the atom name field is character position 13 to 16, but in the example the atom name starts at position 14. If hydrogen atoms are added, *e.g.,* the same atom bound to `CG1` of `VAL` might be called `1HG1` or `HG11`. In the first case, the free character position 13 is used for the additional number. In the second case the first character of the atom name is at position 13 unlike for other atoms or the atom name uses position 14 to 17, excluding the use of alternative location indicators. Since many fields are optional (*e.g.,* the alternative location indicator (position 17) or the chain identifier (position 22)) programs using

[1] `http://www.wwpdb.org/documentation/format23/sect9.html`

column based parsing may fail if no further checks are performed, *e.g.,* that column 5 is a number, not a letter.

Recently, PDBML a XML based representation of the data was published [241]. Many high-level languages like, Java, C++, Perl or Python provide parsers for XML, so that supporting this format should be easier than parsing the original pdb-file format. However, currently PDBML is supported only by few programs, human readability is lower and files get larger.

A.1.2 The PQR File

The pqr-file is a derivative of a pdb-file, containing the coordinates, charges and radii of each atom. It is commonly used for electrostatic calculations (*e.g.,* by MEAD and APBS). Unlike the pdb-file, it is not defined by character positions, but by columns separated by one or more whitespaces. To be compatible with the reading routines of molecular viewers, it is common to format the pqr-file similar to a pdb-file, *i.e.,* to have the same character position for the coordinates.

```
1  ATOM    1 N    VAL  1   −17.730  −6.192  6.148  −0.263  1.550
2  ATOM    2 HN1  VAL  1   −17.000  −6.429  5.486   0.312  1.200
3  ATOM    3 HN2  VAL  1   −18.260  −7.025  6.378   0.312  1.200
4  ATOM    4 HN3  VAL  1   −18.348  −5.501  5.737   0.312  1.200
5  ATOM    5 CA   VAL  1   −17.118  −5.637  7.383   0.151  1.700
6  ATOM    6 HA   VAL  1   −16.561  −4.848  7.124   0.048  1.200
7  ATOM    7 CB   VAL  1   −18.227  −5.144  8.350  −0.012  1.700
8  ATOM    8 HB   VAL  1   −18.341  −5.887  9.111   0.024  1.200
9  ATOM    9 CG1  VAL  1   −17.848  −3.818  9.051  −0.091  1.700
10 ATOM   10 HG11 VAL  1   −16.926  −3.962  9.635   0.031  1.200
11 ATOM   11 HG12 VAL  1   −17.684  −3.036  8.293   0.031  1.200
12 ATOM   12 HG13 VAL  1   −18.664  −3.512  9.723   0.031  1.200
13 ATOM   13 CG2  VAL  1   −19.597  −4.897  7.750  −0.091  1.700
14 ATOM   14 HG21 VAL  1   −19.521  −4.127  6.967   0.031  1.200
15 ATOM   15 HG22 VAL  1   −19.979  −5.831  7.310   0.031  1.200
16 ATOM   16 HG23 VAL  1   −20.286  −4.553  8.537   0.031  1.200
17 ATOM   17 C    VAL  1   −16.197  −6.686  7.983   0.616  1.700
18 ATOM   18 O    VAL  1   −16.601  −7.858  8.050  −0.504  1.500
```

The format has columns for the atom number, the atom name, the residue name, the residue number, the x-, y-, and z-coordinate, the charge and the radius.

A.1.3 The Extended PQR File

The pqr-file format has the deficiency, that chains identifiers are not available[2]. Additionally information on the conformer number, site identifier and instance identifiers were needed. Therefore, the extended pqr-file format, was introduced:

[2] MEAD version 2.2.5 introduced an optional chain identifier as additional column at the end. Before chains were differentiated by increasing the residue number by 1000. The current MEAD pqr-file format and the extended pqr-file format are currently incompatible.

```
1  ATOM  668  N    VAL  113   3.823  −5.789  −7.826  −0.470  1.550  1  1  0  0
2  ATOM  669  H    VAL  113   2.967  −6.294  −7.745   0.310  1.000  1  1  0  0
3  ATOM  670  CA   VAL  113   4.091  −4.866  −6.714   0.070  1.700  1  1  0  0
4  ATOM  671  HA   VAL  113   5.016  −4.343  −6.921   0.090  1.000  1  1  0  0
5  ATOM  672  CB   VAL  113   2.956  −3.825  −6.534  −0.090  1.700  1  1  0  0
6  ATOM  673  HB   VAL  113   2.069  −4.362  −6.117   0.090  1.000  1  1  0  0
7  ATOM  674  CG1  VAL  113   3.379  −2.713  −5.551  −0.270  1.700  1  1  0  0
8  ATOM  675  1HG1 VAL  113   2.574  −1.952  −5.460   0.090  1.000  1  1  0  0
9  ATOM  676  2HG1 VAL  113   3.570  −3.115  −4.535   0.090  1.000  1  1  0  0
10 ATOM  677  3HG1 VAL  113   4.297  −2.200  −5.910   0.090  1.000  1  1  0  0
11 ATOM  678  CG2  VAL  113   2.556  −3.220  −7.883  −0.270  1.700  1  1  0  0
12 ATOM  679  1HG2 VAL  113   1.771  −2.446  −7.749   0.090  1.000  1  1  0  0
13 ATOM  680  2HG2 VAL  113   3.434  −2.747  −8.374   0.090  1.000  1  1  0  0
14 ATOM  681  3HG2 VAL  113   2.150  −3.995  −8.566   0.090  1.000  1  1  0  0
15 ATOM  682  C    VAL  113   4.298  −5.676  −5.425   0.510  1.700  1  1  0  0
16 ATOM  683  O    VAL  113   3.661  −5.434  −4.388  −0.510  1.500  1  1  0  0
```

The additional columns contain the conformer number, chain number, site number and instance number.

A.1.4 The FPT File

The fpt-file can be read by Potential, Solvate and Solinprot to calculate reaction and protein field. It specifies the "field points" by white space separated x, y, z coordinates. Optionally, the three numbers can be surrounded by parenthesis and separated by commas. Line breaks are considered the same as any other white space, *e.g.*,

```
1  16.429  18.299  −2.502
2  17.431  17.898  −2.555
3  ...
4  (16.429  18.299  −2.502),(16.429  18.299  −2.502),(19.247
5  11.424  5.439)
```

The user has to multiply the charges by appropriate charges and convert the result into the desired energy units.

A.1.5 The Extended FPT File

The extended fpt-file can be read by My_3Diel_Solver (appendix 4.4.2) to calculate interaction energies (section 3.2.7). The file format is:

```
1  0 0  16.429  18.299  −2.502  0.210
2  0 0  17.431  17.898  −2.555  0.100
3  ...
4  0 1  16.429  18.299  −2.502  0.210
5  ...
6  0 2  16.429  18.299  −2.502  0.210
```

```
7  ...
8  1 0 19.247 11.424 5.439 0.210
9  ...
10 1 1 19.247 11.424 5.439 0.210
11 ...
```

The first column specifies the site-id, the second column the instance-id. The following three columns (column three to five) specify a coordinate of an atom and the sixth the charge of the atom. The file is ordered by increasing site-id and instance-id starting from zero. For calculating interaction energies with My_3Diel_Solver, the extended fpt-file is expected to contain all atom coordinates and charges of all instances of all sites. The electrostatic potential at each coordinate is multiplied by the associated charge and the energy is converted in $\frac{\text{kcal}}{\text{mol}}$ (unless the `-econv` parameter is given to My_3Diel_Solver). My_3Diel_Solver returns the energy *per* instance of each site, as it is required for the interaction energy file.

A.2 Charge Set Files

A.2.1 The ST File

The st-file specifies two charge forms (*i.e.,* two protonation forms) of a site for Multiflex, Multiflex3D (section 4.4.5), Multiflex2qmpb (section 4.5) or Perl Molecule (section 4.6), *e.g.,*

```
1 4.4
2 GLU  CG   −0.21  −0.28
3 GLU  HG1   0.09   0.09
4 GLU  HG2   0.09   0.09
5 GLU  CD    0.75   0.62
6 GLU  OE1  −0.36  −0.76
7 GLU  OE2  −0.36  −0.76
```

The first line specifies the model $\mathrm{p}K_a$ $\mathrm{p}K_{a,\mathrm{model}}$, of the site (eq. 3.44) for Multiflex or it is converted into a model energy, $\Delta G_{\mathrm{model}}$, (eq. 3.37) to be used for QMPB. The following lines specify the residue name, the atom name and the charge in the protonated and deprotonated form. Multiflex uses the first atom specified as the center of the coordinate system if `ON_CENT_OF_INTR` is specified in the ogm-file or mgm-file. Multiflex2qmpb and Perl Molecule allow the optional keyword `center` which specifies an atom name, which should be taken as center. The ligand type label "proton" is used for the two charge forms. The ligand type label is important if the st-file is used in combination with other charge set file formats.

A.2.2 The EST File

The st-file does not fit very well to the extended theory permitting more than two charge forms. Therefore, the est-file is used as an extended st-file, which can be read by Multiflex2qmpb or Perl Molecule (section 4.5 or section 4.6, respectively), *e.g.,* for relative $\mathrm{p}K_a$ calculations on histidine (following section 3.4) the est-file looks like

```
1  label      HSP      HSD        HSE
2  Gmodel     0        9.602486   9.05377
3  proton     2        1          1
4  HSP CB    −0.05    −0.09      −0.08
5  HSP CG     0.19    −0.05       0.22
6  HSP ND1   −0.51    −0.36      −0.70
7  HSP HD1    0.44     0.32       0.00
8  HSP CD2    0.19     0.22      −0.05
9  HSP HD2    0.13     0.10       0.09
10 HSP CE1    0.32     0.25       0.25
11 HSP HE1    0.18     0.13       0.13
12 HSP NE2   −0.51    −0.70      −0.36
13 HSP HE2    0.44     0.00       0.32
14 HSP HB1    0.09     0.09       0.09
15 HSP HB2    0.09     0.09       0.09
```

The file consists of a header with key-value pairs (the value is usually an array with an element *per* column) and a footer with lines for each atom of the site. Empty lines and comments (starting with `#` and ending with the end of the line) are permitted. Each line of the footer contains the residue name and atom name followed by one column for each charge form. The number of charge forms is arbitrary (one or more). Following keywords are available for the header:

label: Mandatory label for each charge form. The label is used for user friendly labeling of charge forms in the input and output files of QMPB.

center: Optional center of the grid used for solving the LPBE, when the keyword `ON_CENT_OF_INTR` is given in the ogm-file or mgm-file. The value specifies an atom by atom name. By default the geometric center of all atoms in all instances of the site is taken as center (by QMPB).

sitetype: Switches between absolute and relative intrinsic energy calculations (only Perl Molecule). For absolute intrinsic energy calculations (section 3.3), the value has to be `QMsite`. For relative intrinsic energy calculations (section 3.4), the value has to be `MMsite`. Multiflex2qmpb only supports calculations `MMsite` calculations, *i.e.,* ΔG_{model} has to be given.

epsilon: Optional dielectric constant of the site. By default, the dielectric constant is 1 for `QMsites` and 4 for `MMsites`.

Gmodel: Mandatory and to be used with `sitetype MMsite` or Multiflex2qmpb calculations. The energy of the model compound, ΔG_{model}, is specified for each charge form (in $\frac{\text{kcal}}{\text{mol}}$). Usually, one of the charge forms is chosen as reference ($\Delta G_{\text{model}} = 0$) and the other model energies are given relative to the reference.

bonding_energy: Mandatory and to be used with `sitetype QMsite`. The values are the total bonding energy, $H_{\text{vac},i}(j_k)$, for each charge form obtained by QM calculations.

vibration_energy: Mandatory and to be used with `sitetype QMsite`. The values are the vibrational energy, $G_{\text{vib},i}(j_k)$, for each charge form obtained by QM normal model analysis.

unbound_energy: Mandatory and to be used with `sitetype QMsite`. The keyword must be given for each ligand type of this site. The first value is the ligand type label, the second value is the energy of a single free ligand of that type, $G_{\mathrm{free},i}(j_k)$ (section 3.3.2).

Arbitrary non-keywords are interpreted as ligand type labels (*e.g.,* "proton") and a value for each charge form specifies the number of bound ligands of this type. The ligand type labels within charge set files for a calculation have to be consistent.

A.2.3 The XST File

The est-file is limited to charge sets consisting of only a single residue, which is sufficient for amino acids with different tautomers (like histidine) or different reduction and protonation forms (*e.g.,* tyrosine can be protonated, deprotonated, in a radical form and in a protonated radical form). The xst-file is an extended est-file, which allows including multiple residues of potentially different chains into one charge set. It is only supported by Perl Molecule. Charge sets composed of different residues could be metal centers, where the charge of the coordinating residues changes with the reduction state of the metal. The following lines are taken from the iron-sulfur center of ferredoxin:

```
label                ox          red1        red2
sitetype QMsite
epsilon 1 # This is default for a QMsite
electron             0           1           1
bonding_energy       -14915.0    -14882.4    -14879.9
# change in vibration energy due to electron binding is ignored
vibration_energy     0           0           0
# energy of a free electron: F* Delta SHE = 23.06 * (-4.43) kcal/mol
unbound_energy electron -102.1558
A  PHE_39    C       0.47232     0.46481     0.47450
A  PHE_39    O       -0.55295    -0.58717    -0.59104
A  SER_40    N       -0.49477    -0.47273    -0.49589
A  SER_40    CA      0.15185     0.15623     0.17354
A  SER_40    C       0.31169     0.33632     0.34860
A  SER_40    O       -0.50474    -0.54141    -0.54427
A  SER_40    HN      0.33784     0.32364     0.33576
A  SER_40    HA      0.08592     0.06664     0.06183
...
```

The file consists of a header and footer as the est-file. All key-value pairs of the header are explained in appendix A.2.2. Here, an absolute reduction calculation is shown. Therefore, `sitetype QMsite` is selected and the ligand type label is "electron". Each line of the footer contains the chain label, the residue label (consisting of residue name and residue number concatenated by an underscore), the atom name and the charge in each charge form. If the keyword `center` is given in the header, the atom has to be specified by the chain label, residue label and atom name.

A.2.4 The FST File

The fst-file is an extended xst-file which allows to specify instances as association between a particular charge form and a particular rotamer form. It is also only supported by Perl Molecule. Usually, each charge form is geometry optimized by QM methods leading to slightly different geometries, *e.g.,* bond lengths change with the charge of the system. This effect is usually ignored and the original crystallographic or NMR coordinates are used in a xst-file. A fst-file allows to specify a pqr-file for each instance containing both, coordinates and charges. Here, the above example is given as fst-file:

```
1  label                ox          red1        red2
2  sitetype QMsite
3  epsilon 1 # This is default for a QMsite
4  electron             0           1           1
5  bonding_energy       −14915.0    −14882.4    −14879.9
6  # change in vibration energy due to electron binding is ignored
7  vibration_energy     0           0           0
8  # energy of a free electron: F* Delta SHE = 23.06 * (−4.43) kcal/mol
9  unbound_energy electron −102.1558
10 pqr 1CZP_AA_ox.pqr 1CZP_AA_red1.pqr 1CZP_AA_red2.pqr
```

The header is the same as in appendix A.2.3, but the footer is replaced by a single line containing the keyword `pqr` and a pqr-file name for each instance.

A.3 Grid Files

MEAD needs a file specifying the grid on which the LPBE should be solved. All programs expect an ogm-file, which specifies the grid mesh of the object of interest (*e.g.,* macromolecule or site). For Multiflex mgm-files have to exist, which specify the grid mesh of model compounds. Different grid files for sites and model compounds allow to start focussing with coarser and larger grids for the macromolecule than for the model compound. However, to cancel grid artefacts, the grid specifications for the finest grid have to be identical. My_3Diel_Solver uses ogm-files and My_2Diel_Solver uses mgm-files. An example for an ogm-file or mgm-file is given below:

```
1 ON_ORIGIN 81 1.0
2 ON_GEOM_CENT 81 0.5
3 ON_CENT_OF_INTR 81 0.25
```

The example shows three focussing steps with 81 grid points *per* dimension and a grid spacing of 1.0, 0.5 and 0.25 Å, respectively. There are three possibilities to center the grid, *i.e.,* on the origin of the coordinate system (`ON_ORIGIN`), on the geometric center of the molecule (`ON_GEOM_CENT`) or on the center of interest (`ON_CENT_OF_INTR`). The center of the molecule refers to the first pqr-file given to Solinprot or My_3Diel_Solver, not to the center of the protein given by both pqr-files. The center of interest is only available for Multiflex and Multiflex3D focussing the grid onto the geometric center of the site. QMPB uses this option to calculate the geometric center of all instances of a site and passes the coordinate to My_2Diel_Solver

and My_3Diel_Solver. In QMPB the PB solvers only get the coordinates of a single instance and therefore the geometric center would be dependent on the rotamer and ligands bound. As a result grid artefacts would not cancel, which is necessary to obtain meaningful results.

A.4 Sites Files

A.4.1 The Multiflex Sites File

The sites file of Multiflex and Multiflex3D has two columns. The first gives the residue number, the second the st-file name. For example:

```
1  NTLYS
15 HSC
7  GLU
35 GLU
18 ASP
...
```

The same file format can be used for Multiflex2qmpb to set up QMPB calculations. Here, additionally also est-file can be used. If a st-file and an est-file exist with the same prefix, the est-file is used.

A.4.2 The Perl Molecule Charge Sites File

Charge sets can be specified in Perl Molecule by st-file, est-file, xst-file and fst-file. Currently, only st-file and est-file can be defined by a charge sites file. The other two charge sets allow to include multiple residues of multiple chains and can therefore not to be used on a *per*-residue basis. Usually, only a few complex sites are described by xst-file or fst-file, but many simple sites exist in proteins, which can be described by st-file or est-file.

An example for a charge sites file is:

```
A GLU_72 glu.st
B GLU_94 glu.st
C GLU_95 glu.st
A ASP_69 asp.st
B ASP_86 asp.st
C ASP_96 asp.st
A HSP_16 HSP.est
B HSP_92 HSP.est
A ARG_42 arg.st
...
```

The first column specifies the chain and the second the residue name and number. Therefore, the residue can be identified uniquely in a conformer in Perl Molecule. The third column contains the st-file name or est-file name. Unlike for the sites file of Multiflex2qmpb, the full name including postfix is used avoiding confusion due to preferring one filetype over the other.

A.4.3 The Perl Molecule Rotamer Sites File

The hydrogen placement procedure of Perl Molecule may generate multiple hydrogen rotamers in case different hydrogen bond networks are possible. Such groups will be treated as rotamer sites (section 3.4), with each hydrogen position as an instance. The rotamer sites file is used to assign rotamer energies, $\Delta G_{\mathrm{rotamer}}(j_k)$, to the instances. The format of the file is as follows:

```
1 SER HG1 OG  CB CA  SER_CA_CB_OG_HG1.dat
2 THR HG1 OG1 CB CA  THR_CA_CB_OG1_HG1.dat
3 TYR HH  OH  CZ CE1 TYR_CE1_CZ_OH_HH.dat
4 LYS HZ1 NZ  CE CD  LYS_CD_CE_NZ_HZ1.dat
5 ALA HT1 N   CA HA  ALA_HA_CA_N_HT1.dat
```

The first column gives the residue name, the next four columns define a torsion angle starting with the rotateable hydrogen and last column gives a file name, in which the torsion potential for this dihedral is stored. The torsion potential file has two columns, the first containing the dihedral angle and the second the associated energy (in $\frac{\mathrm{kcal}}{\mathrm{mol}}$), *e.g.*,

```
1 −180     3.9950
2 −179     3.9954
3 −178     3.9931
4 ...
5 178      3.9862
6 179      3.9920
7 180      3.9950
```

A.5 Force Field

A.5.1 The CHARMM Topology File

The documentation for the CHARMM topology file can be found at[3]. The file is organized in blocks, *e.g.*, defining residues and patches. Here some part of the block for residue alanine is shown:

```
1  RESI ALA          0.00
2  GROUP
3  ATOM N    NH1    −0.47  !      |
4  ATOM HN   H       0.31  !  HN–N
5  ATOM CA   CT1     0.07  !      |      HB1
6  ATOM HA   HB      0.09  !      |     /
7  GROUP                   !  HA–CA—CB–HB2
8  ATOM CB   CT3    −0.27  !      |     \
9  ATOM HB1  HA      0.09  !      |      HB3
10 ATOM HB2  HA      0.09  !   O=C
11 ATOM HB3  HA      0.09  !      |
12 GROUP                   !
```

[3] http://brooks.scripps.edu/charmm_docs/c32docs/c32a1/html/rtop.html

```
13 ATOM C     C          0.51
14 ATOM O     O         -0.51
15 BOND CB CA   N   HN   N   CA
16 BOND C   CA   C   +N   CA HA   CB HB1   CB HB2   CB HB3
17 DOUBLE O   C
18 ...
```

The block starts with the keyword `RESI`, the residue name and total charge of the residue. Atoms are grouped into chargegroups, which are marked by the keyword `GROUP` having a total integer charge. The name, type and charge for each atom are given in the lines starting with the keyword `ATOM`. The topology is defined by the keywords `BOND`, `DOUBLE` and `TRIPLE` followed by pairs of atom names. The exclamation mark comments out the rest of the line, *i.e.,* the ASCII diagram is not read by the program, but is only a help for the user.

A.5.2 The Hwire Parameter File

The documentation of the Hwire parameter file is part of the Hwire package. The file is also organized in blocks defining residues and patches. Here, the block for the residue serine is shown:

```
1 # column
2 # 1   2    3    4   5 6  7    8     9        10    11       12    13   14   15
3 residue SER
4 dono N    sp2  2   1 0  -C   CA    0.997    4.0   0.000    0.0   HN
5 nonp CA   sp3  3   1 0  N    C     1.080    0.0   0.000    0.0   HA
6 nonp CB   sp3  2   2 0  CA   OG    1.111    0.0   0.000    0.0   HB1  HB2
7 both OG   sp3  1   1 2  CB   ?     0.960  106.0   $lp_d  106.0   HG1  LG1  LG2
8 acce O    sp2  1   0 2  C    CA    0.000    0.0   $lp_d  $lp_a   L1   L2
9 end
```

The block is defined by the keywords `residue`, the residue name and the keyword `end`. Inside the block, there is one line *per* non-hydrogen atom. Column 1 specifies a keyword for the capability of the atom to form hydrogen bonds: The keyword is `dono`, if the atom is a hydrogen bond donor; it is `acce`, if the atom is an hydrogen bond acceptor; it is `both`, if the atom is an hydrogen bond donor and acceptor and `nonp` if the atom is non-polar. Column 2 gives the atom name and column 3 the hybridization (sp2 or sp3). In the fourth, fifth and sixth column the number of bonded non-hydrogen atoms, number of bonded hydrogen atoms and the number of lone-pairs of the atom are given, respectively. The next two columns specify the names of the first and second bonded non-hydrogen atom or "?" if no non-hydrogen atom is bound. Column 9 is the bond length between the atom and the hydrogen; column 10 the bond angle between the first bonded atom, the atom and the hydrogen atom (sp3 hybridization) or the angle deviation from the median vector between the first bonded atom, the atom and the inplane atom (sp2 hybridization). Column 11 and 12 are bond length between the atom and the lone-pair and the angle between the first bonded atom, the atom and the lone pair. Previously defined variables are prefixed by $. The names of the hydrogen atoms and lone-pairs are given in the remaining columns.

A.5.3 The Dunbrack Rotamer Library File

The rotamer library of Dunbrack and coworkers [101, 102] is documented at[4]. A backbone dependent and a backbone independent library are available. The backbone dependent library gives rotamer probabilities dependent on the ϕ and ψ angle of the protein backbone, while the backbone independent library does not do this differentiation. The backbone dependent library has the following format:

```
1  Res   phi   psi n  Rotamer  p(r1234)    chi1   chi2   chi3   chi4    sig1  sig2  sig3  sig4
2                      1 2 3 4
3  ARG  −180  −180 6  1 1 1 1  0.002977    55.4   79.7   62.4   82.3    19.8  16.1  15.0  11.9
4  ARG  −180  −180 6  1 1 1 2  0.006091    59.2   85.4   68.2  −166.5   23.2  16.6  15.4  25.6
5  ARG  −180  −180 6  1 1 1 3  0.000316    54.5   79.4   64.6  −103.2   21.5  17.2  16.6  12.5
6  ARG  −180  −180 6  1 1 2 1  0.004102    54.5   86.6  178.2   85.5    19.0  12.9  13.1  12.6
7  ARG  −180  −180 6  1 1 2 2  0.010949    51.1   86.5 −177.3  177.7    18.9  13.3  17.0  20.5
8  ARG  −180  −180 6  1 1 2 3  0.005369    53.5   87.8 −175.9  −81.0    20.2  11.8  17.5  12.2
9  ARG  −180  −180 6  1 1 3 1  0.000084    54.5   83.5  −78.4  102.9    21.5  14.5  20.8  14.2
10 ...
```

and the backbone independent library has the following format:

```
1  Res Rotamer n(r1)   n   p(r1234) sig p(r234|r1) sig   chi1 sig1 chi2 sig2   chi3 sig3   chi4 sig4
2      1 2 3 4        (r1234)
3  ARG 1 1 1 1  568    2   0.04     0.02  0.41      0.22  55.4 19.8 79.7 16.1   62.4 15.0   82.3 11.9
4  ARG 1 1 1 2  568    5   0.07     0.03  0.83      0.31  59.2 23.2 85.4 16.6   68.2 15.4 −166.2 25.4
5  ARG 1 1 1 3  568    0   0.00     0.01  0.04      0.07  54.5 21.5 79.4 17.2   64.6 16.6 −103.2 12.5
6  ARG 1 1 2 1  568    3   0.05     0.02  0.56      0.26  54.5 19.0 86.6 13.0  178.2 13.1   85.5 12.6
7  ARG 1 1 2 2  568    9   0.13     0.04  1.49      0.42  53.9 18.9 87.8 13.9 −178.5 18.0  178.1 22.0
8  ARG 1 1 2 3  568    4   0.06     0.03  0.73      0.29  53.5 20.2 87.8 11.8 −176.0 17.5  −81.1 12.2
9  ARG 1 1 3 1  568    0   0.00     0.00  0.01      0.04  54.5 21.5 83.5 14.5  −78.4 20.8  102.9 14.2
10 ...
```

The column headers have the following meaning:

Res: Residue name

phi: Backbone ϕ angle (in the backbone dependent rotamer library)

psi: Backbone ψ angle (in the backbone dependent rotamer library)

n: Number of samples of this rotamer found in the PDB

Rotamer: Rotamer classification of sidechain dihedrals $\chi 1$, $\chi 2$, $\chi 3$ and $\chi 4$ according to Dunbrack and Cohen [102]

p(r1234): Probability of the rotamer (and standard deviation `sig` in the backbone independent rotamer library)

p(r234—r1): Conditional probability of the rotamer (and standard deviation `sig` in the backbone independent rotamer library), that a sidechain will be in an r2 or r2,r3 or r2,r3,r4 rotamer given that r1 is a particular rotamer

[4] `http://dunbrack.fccc.edu/bbdep/bbdepformat.php`

chi1 - chi4: Average sidechain dihedrals $\chi 1$, $\chi 2$, $\chi 3$ and $\chi 4$

sig1 - sig4: Standard deviation of sidechain dihedrals $\chi 1$, $\chi 2$, $\chi 3$ and $\chi 4$

A.5.4 The Bondi Radii File

The Bondi radii [147] are read from a file with the following format:

```
1 RAD  FE  1.3
2 RAD  MG  1.6
3 RAD  CU  1.3
4 RAD  N   1.55
5 RAD  C   1.7
6 RAD  O   1.5
7 RAD  S   1.8
8 RAD  H   1.2
9 RAD  P   2.0
```

Each line starts with the keyword `RAD` followed by the name of the chemical element and the radius. Comments are prefixed with an exclamation mark.

A.6 The QMPB Input File

The input file of QMPB is described in section 4.3.3 based on examples. Here, a list of available keywords for each block is given for reference.

A.6.1 The General Block

Example:

```
1  meadpath = /home/essigke/bin
2  T = 300
3  I = 0.1
4  backfile = background.pqr
5  MGMcenter = ON_CENT_OF_INTR
6  MGMpoints = 131
7  MGMspace = 0.2
8  OGMcenter = ON_GEOM_CENT ON_CENT_OF_INTR
9  OGMpoints = 131 131
10 OGMspace = 1 0.2
11 epsin1 = 1
12 epsin2 = 4
13 Ligand_Labels = proton electron
```

meadpath: Path to the MEAD programs, *i.e.*, Pqr2SolvAccVol, My_2Diel_Solver and My_3Diel_Solver (mandatory).

T: Temperature in Kelvin (mandatory).

I: Ionic strength in $\frac{mol}{l}$ (mandatory).

backfile: Filename of the background pqr-file. The background pqr-file contains all atoms which are not part of any site. It must not contain atoms, which are part of any site (mandatory).

MGMcenter: As many strings (`ON_ORIGIN`, `ON_GEOM_CENT` or `ON_CENT_OF_INTR`, see appendix A.3) as there are focussing steps. The values define the center of the model grid mesh used for solving the PBE for a model compound (optional).

MGMpoints: As many uneven integer numbers as there are focussing steps. The values give the number of grid points in the model grid mesh used for solving the PBE for a model compound (optional).

MGMspace: As many positive floating point numbers as there are focussing steps. The values give the spacing between grid points in the model grid mesh used for solving the PBE for a model compound (optional).

OGMcenter: As many strings (`ON_ORIGIN`, `ON_GEOM_CENT` or `ON_CENT_OF_INTR`, see appendix A.3) as there are focussing steps. The values define the center of the object grid mesh used for solving the PBE for a `MMsite` or `QMsite` (optional).

OGMpoints: As many uneven integer numbers as there are focussing steps. The values give the number of grid points in the object grid mesh used for solving the PBE for a `MMsite` or `QMsite` (optional).

OGMspace: As many positive floating point numbers as there are focussing steps. The values give the spacing between grid points in the object grid mesh used for solving the PBE for a `MMsite` or `QMsite` (optional).

Ligand_Labels: As many strings as there are ligand types. The strings are the ligand type labels in the order of the **N** option in the other blocks (mandatory).

workdir: Name of the directory created by QMPB to write the `job.sh` script and all files needed to run the helper programs called by the script (optional, default `qmpb`).

epsin1: Dielectric constant of the region of atoms with quantum mechanically derived charges (mandatory, even if there are no atoms given for this dielectric region).

epsin2: Dielectric constant of the region of atoms with force field charges (mandatory).

A.6.2 An Instance of a QMsite

The block starts with a line

```
QMsite site_label instance_label
```

and ends with a line starting a block for another instance or with a keyword belonging to the general block. The `site_label` is a user defined tag for the site. Instances of the same site must have the same `site_label`, while different sites must have a different `site_label`.

The site_label may contain the residue name and number as well as a chain identifier. The instance_label is a user defined tag for the instance of the site. Instances of the same site must have different instance_labels.

Example:

```
1  QMsite site_C_PHE_39_A instance_0_ox
2    file=site_C_PHE_39_Ainstance_0_ox.pqr
3    sid=21
4    iid=0
5    eps= 1
6    Hqm=-14998.1
7    Gvib=98.693
8    N= 0 0
9    Gfree= 0  -102.1558
10   QM_corr C_PHE_39_A.pqr int_C_PHE_39_A.pqr
11   QM_corr O_PHE_39_A.pqr int_O_PHE_39_A.pqr
12 ...
```

Hqm: Energy of formation or total bonding energy, $H_{\mathrm{vac},i}(j_k)$, of the instance in the dielectric eps (mandatory).

Gvib: Vibrational energy, $G_{\mathrm{vib},i}(j_k)$, of the instance calculate by normal mode analysis (mandatory).

Gfree: Array of energies of unbound ligands, $G_{\mathrm{free},i}(j_k)$, for each ligand type λ. The length and order has to be consistent with the Ligand_Labels in the global block (mandatory).

Gcorr: Electrostatic correction energy, $G_{\mathrm{corr},i}(j_k)$ (section 3.3.1), of the instance calculated in a previous run. It can not be used in combination with the QM_corr keyword (optional).

QM_corr: This option does not follow the general key=value pair syntax and unlike other options is allowed to occur more than once in the block of an instance. The keyword is followed by two pqr-file names, the first containing an atom of the QMsite and the second containing all atoms outside the side, with which the atom is in a bond or angle relationship. The PBE is solved for each line with the QM_corr keyword to obtain a contribution to the correction energy, $G_{\mathrm{corr},i}(j_k)$. For most QMsites calculating the correction energy is computationally very costly. Therefore, a once computed correction energy can be given as input using the Gcorr keyword (optional).

file: Name of the pqr-file defining the coordinates and charges of atoms belonging to the instance (mandatory).

eps: Dielectric constant of the region to which the instance belongs. Currently, it has to be identical with either epsin1 or epsin2 in the general block (mandatory).

N: Array of numbers of bound ligand, $n_{\lambda,i}(j_k)$, for each ligand type λ. The length and order has to be consistent with the Ligand_Labels in the global block (mandatory).

OGMpoints, OGMspace, OGMcenter: The keywords have the same meaning as in the general block, but they define the grid for the site only. Therefore, the keywords can only be

given for one instance of each site. By that, it is possible to calculate most sites with a small grid as final focussing step (*e.g.,* fitting to an amino acid sidechain), but to define a larger grid for a particular site (*e.g.,* a heme), which is significantly larger than an average site (mandatory, if not defined in the general block).

center: If the string `ON_CENT_OF_INTR` is used in the grid definition, the geometric center of all instances of the site is calculated. Instead, it is possible to define a center coordinate, *e.g.,* (0.0,0.0,0.0), once for a site (optional).

sid: Numerical site identifier required by some of the post-processing programs. The output is sorted by numerical identifiers (unique for each site, optional).

iid: Numerical instance identifier required by some of the post-processing programs. The output is sorted by numerical identifiers (unique for a instance in a particular site, optional).

A.6.3 An Instance of a MMsite

The block starts with a line

```
MMsite site_label instance_label
```

and ends with a line starting a block for another instance or with a keyword belonging to the general block. The `site_label` is a user defined tag for the site. Instances of the same site must have the same `site_label`, while different sites must have a different `site_label`. The `site_label` may contain the residue name and number as well as a chain identifier. The `instance_label` is a user defined tag for the instance of the site. Instances of the same site must have different `instance_labels`.

Example;

```
1 MMsite site_HT1_ALA_1_A instance_0_C
2   file=site_HT1_ALA_1_Ainstance_0_C.pqr
3   sid=0
4   iid=0
5   ref=self
6   eps= 4
7   N= 0 0
8  Gmm=0.3299
```

ref: Reference by `instance_label` either pointing to a `Modelsite` or another rotamer of the same site with identical `N` array. If the reference points to a `Modelsite`, the energy of the site is calculated relative to the model energy. If the reference points to a different rotameric form, the energy is calculated relative to the energy of the other rotameric form. If the string `self` is used, the instance is a reference instance (section 3.4, mandatory).

Gmm: Rotamer energy, $G_{\mathrm{rotamer}}(j_k)$, of the instance. The term is used when calculating the energy of an instance relative to another rotameric form of the same instance (mandatory).

nohomo: The keyword disables the calculation of the site in a homogeneous dielectric to save time on sites without rotamer forms (keyword without value, optional).

file: Name of the pqr-file defining the coordinates and charges of atoms belonging to the instance (mandatory).

eps: Dielectric constant of the region to which the instance belongs. Currently, it has to be identical with either `epsin1` or `epsin2` in the general block (mandatory).

N: Array of numbers of bound ligand, $n_{\lambda,i}(j_k)$, for each ligand type λ. The length and order has to be consistent with the `Ligand_Labels` in the global block (mandatory).

OGMpoints, OGMspace, OGMcenter: The keywords have the same meaning as in the general block, but they define the grid for the site only. Therefore, the keywords can only be given for one instance of each site. By that, it is possible to calculate most sites with a small grid as final focussing step (*e.g.,* fitting to an amino acid sidechain), but to define a larger grid for a particular site (*e.g.,* a heme), which is significantly larger than an average site (mandatory, if not defined in the general block).

center: If the string `ON_CENT_OF_INTR` is used in the grid definition, the geometric center of all instances of the site is calculated. Instead, it is possible to define a center coordinate, *e.g.,* (0.0,0.0,0.0), once *per* site (optional).

sid: Numerical site identifier required by some of the post-processing programs. The output is sorted by numerical identifiers (unique for each site, optional).

iid: Numerical instance identifier required by some of the post-processing programs. The output is sorted by numerical identifiers (unique for a instance in a particular site, optional).

A.6.4 An Instance of a Modelsite

The block starts with a line

```
Modelsite site_label instance_label
```

and ends with a line starting a block for another instance or with a keyword belonging to the general block. The `site_label` is a user defined tag for the site. Instances of the same site must have the same `site_label`, while different sites must have a different `site_label`. The `site_label` may contain the residue name and number as well as a chain identifier. The `instance_label` is a user defined tag for the instance of the site. Instances of the same site must have different `instance_labels`.

Example:

```
1 Modelsite site_CE_LYS_4_A model_instance_0_p_ROT_-5
2   file=model_site_CE_LYS_4_Ainstance_0_p_ROT_-5.pqr
3   sid=1
4   iid=0
5   ref=instance_0_p_ROT_-5
6   Gmodel=-14.2665506152
7   eps= 4
8   N= 1 0
```

ref: Reference by `instance_label` pointing to the `MMsite` to which the `Modelsite` belongs (mandatory).

Gmodel: Energy of the model compound relative to a different instance of the same model compound, *e.g.,* the user defines a instance as the reference form for the site and gives all experimental values relative to this instance.

file: Name of the pqr-file defining the coordinates and charges of atoms belonging to the instance (mandatory).

eps: Dielectric constant of the region to which the instance belongs. Currently, it has to be identical with either `epsin1` or `epsin2` in the general block (mandatory).

N: Array of numbers of bound ligand, $n_{\lambda,i}(j_k)$, for each ligand type λ. The length and order has to be consistent with the `Ligand_Labels` in the global block (mandatory).

MGMpoints, MGMspace, MGMcenter: The keywords have the same meaning as in the general block, but they define the grid for the site only. Therefore, the keywords can only be given for one instance of each site. By that, it is possible to calculate most sites with a small grid as final focussing step (*e.g.,* fitting to an amino acid sidechain), but to define a larger grid for a particular site (*e.g.,* a heme), which is significantly larger than an average site (mandatory, if not defined in the general block).

center: If the string `ON_CENT_OF_INTR` is used in the grid definition, the geometric center of all instances of the site is calculated. Instead, it is possible to define a center coordinate, *e.g.,* (0.0,0.0,0.0), once *per* site (optional).

sid: Numerical site identifier required by some of the post-processing programs. The output is sorted by numerical identifiers (unique for each site, optional).

iid: Numerical instance identifier required by some of the post-processing programs. The output is sorted by numerical identifiers (unique for a instance in a particular site, optional).

APPENDIX B

MANUAL PAGES

B.1 QMPB

Name

QMPB - Quantum Mechanics bases Poisson Boltzmann

Synopsis

```
qmpb.pl <qmpb.in> -b|-a [-gmct] [-inputorder] [-pf]
```

Description

QMPB calculates ligand binding energies of ligands of a macromolecule by a continuum electrostatic approach.

Reference reactions can be obtained by quantum mechanics in vacuum or measured for model compounds in solution. The transfer energy of the reactants and products from the reference reaction environment into the macromolecule is calculated by helper programs solving the linearized Poisson-Boltzmann equation (LPBE). QMPB generates the input for the LPBE solvers in a pre-processor mode (option `-b`) and condenses the output into a file of intrinsic energies and a file of interaction energies in a post-processor mode (option `-a`). The program run is largely input file driven (`<qmpb.in>`), which is described in appendix A.6. Most of the work of QMPB is performed by objects, which are defined in module files. If the module files are not in the default Perl path, the path to the module files has to be given by setting the environment variable `PERL5LIB` appropriately.

Options

-a: Run in post-processor mode (after)

-b: Run in pre-processor mode (before)

-gmct: Write output (in post-processor mode) in a format suited for reading by SMT and GMCT, *i.e.,* site and instance identifiers start at one (optional). By default all counting in QMPB starts with zero.

-inputorder: Read site identifiers `sid` and instance identifier `iid` from input file (optional). By default the output is given in the order of internally generated Perl hashes.

-pf: Calculate protein field energies by passing this option to My_3Diel_Solver (optional). If the flag is set each coordinate and charge in the background pqr-file is copied to a fpt-file (`protein.fpt`), which is processed by My_3Diel_Solver. The output is not analyzed by QMPB, but has to be handled by external programs.

Environment

PERL5LIB: Directory to search for Perl modules of QMPB (see man perlrun).

B.2 Pqr2SolvAccVol

Name

Pqr2SolvAccVol - Calculate the analytical description of the "Solvent Accessible Volume" from a pqr-file.

Synopsis

```
pqr2SolvAccVol [-solrad <value>] [-ascii] [-blab1|blab2|blab3] <molname>
```

Description

The program Pqr2SolvAccVol reads a extended pqr-file (`<molname>.pqr`) as input and writes the analytical surface description of the solvent accessible volume(s) of the molecule(s) to an output file (`<molname>.txt` for ASCII format and `<molname>.dat` for binary format). The solvent accessible surface is calculated by a rolling probe (section 2.2.6) and described analytically [70]. For complex molecules the calculation is time consuming and redundant for all My_3Diel_Solver calculations. Therefore, pre-calculating the surfaces reduces the overall computational time. It is fast to map the analytical surface representation to dielectric constants on a discrete grid for a particular focussing step.

Additionally, by reading always the same surfaces for dielectric regions it is ensured that dielectric boundaries stay constant, even if the coordinates of some atoms change. Therefore My_3Diel_Solver reads different files for calculating boundaries and electrostatic energies.

To ensure that all atoms of rotameric forms are inside their dielectric region, the pqr-file generated by QMPB contains all instances of all sites. Therefore, for sites with different rotameric forms the dielectric region is larger as it would be if only the rotameric form of interest would be included. The error is small if the rotamers are similar, *i.e.,* if only hydrogen rotamers are included.

Internally, Pqr2SolvAccVol reads `<molname>.pqr` into an `AtomSet` object and a `SolvAccVol` object is created from the `AtomSet`. All classes and invoked methods are part of the unmodified MEAD library.

Options

-solrad: Set the probe radius (in Å) for the probe, which will be rolled over the molecular surface (default 1.4 Å). A smaller radius will give deeper cusps between the atoms while a larger radius will give a smoother surface. The default value is often used for water.

-ascii: Write the output in ASCII format. The default, binary format gives smaller files and is faster to read by the MEAD programs, but is not human readable.

-blab*: The "blab"-level determines the amount of debug output written to STDOUT. A higher value increases the verbosity.

B.3 My_3Diel_Solver

Name

My_3Diel_Solver - Solve the LPBE for an instance of a site in a three dielectric environment.

Synopsis

```
my_3diel_solver -epsin1 <value> -epsin2 <value> [-epsext <value>]
                [-epsvac <value>] [-solrad <value>] [-sterln <value>]
                [-ionicstr <value>] [-T <value>] [-kBolt <value>]
                [-conconv <value>] [-econv <value>]
                [-bohr_radius <value>] [-proton_charge <value>]
                [-converge_oldway <value>] [-blab1|blab2|blab3]
                [-eps1set <eps1set>] [-eps2set <eps2set>]
                [-fpt <extended fpt-file>] [-pf <extended fpt-file>]
                [-epshomo <value>] <instance> <background>
```

Description

The program My_3Diel_Solver solves the linearized Poisson-Boltzmann equation (LPBE) in a (up to) three dielectric environment. Therefore it is compareable with Solinprot, however it may not calculate a homogeneous transfer energy, but only a Born and a background energy. Grid artefacts may need to be canceled by an accompanied calculation of My_2Diel_Solver. It is used for computing electrostatic energies for instances of the `QMsite` and `MMsite` class in QMPB.

Options

Most options are defined in the documentation provided with the MEAD package. Here, only new options are explained.

-eps1set: Specifies a file `<eps1set>.pqr` containing all atoms in the region with dielectric constant `-epsin1`. If a file `<eps1set>.txt` or `<eps1set>.dat` exists, it is read additionally to the pqr-file, assuming that the file was generated by Pqr2SolvAccVol from the pqr-file. The file `<eps1set>.dat` is preferred for smaller file size and higher numerical accuracy. Else the analytical surface representation is calculated using the radius `-solrad`. Finally, the pqr-file is used to calculate the `ElectrolyteEnvironment`.

-eps2set: Specifies a file `<eps2set>.pqr` containing all atoms in the region with dielectric constant `-epsin2`. Analogously to the `eps1set`, analytical surface representations can be read from file.

-fpt: Specifies an extended fpt-file to be read (appendix A.1.5). The electrostatic potential is calculated at the given coordinates and multiplied by the given charges. The interaction energy is summed up for each instance of each site and given in the output.

-pf: Specifies an extended fpt-file to be read (appendix A.1.5). The electrostatic potential is calculated at the given coordinates and multiplied by the given charges. The electrostatic energy is given in the output file marked by the string "PF-TAG". This option, thought for debugging or detailed analysis of background or interaction energy terms, is ignored by QMPB.

-epshomo: Additionally to the calculation in the three dielectric environment a calculation in a homogeneous environment can be done. The homogeneous dielectric is set to the given value.

B.4 My_2Diel_Solver

Name

My_2Diel_Solver - Solve the LPBE for a model compound of an instance of a site in a two dielectric environment.

Synopsis

```
my_2diel_solver -epsin <value> [-epsext <value>] [-epsvac <value>]
                [-solrad <value>] [-sterln <value>] [-ionicstr <value>]
                [-T <value>] [-kBolt <value>] [-conconv <value>]
                [-econv <value>] [-bohr_radius <value>]
                [-proton_charge <value>] [-converge_oldway <value>]
                [-blab1|blab2|blab3]
                <instance> <model-background>
```

Description

The program My_2Diel_Solver solves the linearized Poisson-Boltzmann equation (LPBE) in a two dielectric environment. Therefore it is compareable with Solvate, however it does not calculate a homogeneous transfer energy, but only a Born and a background energy. Grid artefacts have to be canceled by an accompanied calculation with My_3Diel_Solver. It is used for computing electrostatic energies for model compounds of instances of sites of the `MMsite` class in QMPB.

While Solvate reads one pqr-file, which is used for both, calculating the dielectric boundaries and the electrostatic energies, My_2Diel_Solvertakes two extended pqr-files. The file `<instance>.pqr` contains the coordinates and charges of the instance. It is identical with the file used in the associated My_3Diel_Solver calculation. The file `<model-background>.pqr` contains all atoms of the model compound defining the dielectric boundary between the two regions with dielectric constant `-epsin` and `-epsext`. Charges belonging to atoms of the instance of the site must be set to zero, while charges of other atoms define the background charge set of the model compound. The file `<model-background>.pqr` is also used to calculate the `ElectrolyteEnvironment`.

Pre-calculating surfaces as in My_3Diel_Solver was not found to be useful, because each amino acid of the same type is probably in a different reference rotamer. Also some effort would be necessary to rotate and translate the surface of one site onto the coordinates of another.

Options

All options are defined in the documentation provided with the MEAD package.

B.5 Multiflex2qmpb

Name

Multiflex2qmpb - Convert Multiflex input into an input file for QMPB.

Synopsis

```
multiflex2qmpb.pl <molname> > qmpb.in
```

Description

The program Multiflex2qmpb converts input files written for Multiflex into an input file for QMPB. Only a subset of QMPBs options are available: Only relative ligand binding energies of sites without rotamers[1] can be calculated. Unlike Multiflex, Multiflex2qmpb can handle est-files of sitetype `MMsite` additionally to st-files. Therefore, it is possible to perform calculations with more than two charge forms and different ligand types.

The program expects the files `<molname>.pqr` (appendix A.1.2 or appendix A.1.3), `<molname>.sites` (appendix 3.3), `<molname>.ogm` and `<molname>.mgm` (appendix A.3). For each residue name given in the file `<molname>.sites`, a st-file or est-file is expected. st-files (appendix A.2.1) are assumed to contain charge sets for proton binding and energies given in $\mathrm{p}K_a$-units for compatibility with Multiflex. The est-file is documented in appendix A.2.2. If files of both types are present, the est-file is preferred. Residues with more than one tautomer (histidine, N- and C-terminus of residues with titrateable sidechain) have to be given in an est-file instead using two (or more) st-files as in Multiflex.

The pqr-file of the model compound is generated by taking the whole residue (not just the atoms given in the st-file or est-file) and additional atoms of the previous and next residue specified in the arrays `prev_res` and `next_res` inside the Multiflex2qmpb sourcecode. These atoms are expected to change rarely and are also hardcoded in Multiflex. Also the default parameters are adjustable in the Perl source code.

Absolute ligand binding energy calculations are not supported by Multiflex2qmpb and have to be added by modifying the QMPB input file `qmpb.in`.

To be compatible with Multiflex, multiple st-files can be specified per residue. Each charge column in the st-file has to have integer charge. One column has to have a charge sum of zero, because the uncharged form is chosen as reference form. All atoms which are mentioned in any st-file given for the residue are set to one of the charge forms from the st-file. The charges in the pqr-file are ignored.

Environment

PERL5LIB: Directory to search for Perl modules of Multiflex2qmpb (see man perlrun).

MEADPATH: `$MEADPATH/bin` is the value of the QMPB option `meadpath` (see appendix A.6).

[1] Simple rotamers can be included as charge forms.

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Ferner erkläre ich, dass ich anderweitig mit oder ohne Erfolg nicht versucht habe, diese Dissertation einzureichen. Ich habe keine gleichartige Doktorprüfung an einer anderen Hochschule endgültig nicht bestanden.

Timm Essigke

Bayreuth, den 20. Dezember 2007