Monte Carlo Methods for Simulation of Protein Folding and Titration

Inauguraldissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften
bei dem Fachbereichs Biologie – Chemie – Pharmazie
der Freien Universität Berlin

vorgelegt von
Björn Rabenstein
Diplom-Biochemiker aus Berlin

Berlin, August 31, 2000

1. Gutachter: Prof. Dr. E. W. Knapp, Freie Universität Berlin

2. Gutachter: Dr. H. Sklenar, Max-Delbrück-Centrum, Berlin

Tag der mündlichen Prüfung: 25. Oktober 2000

Preface

Folding and titration of proteins are the two major topics of this work. Seemingly unconnected, they await their unification at the very end of the last chapter. Another hidden connection is the concept of continuum electrostatics, which is used for both. Therefore, I summarize some aspects about the methods of continuum electrostatics in the first chapter including a brief derivation of the most important equations, hoping that this will give the reader a first aid for understanding the underlying principles. The first chapter is merely a toolbox used in later chapters, where my actual research work is presented.

The second chapter deals with the long-term dynamics of proteins, which means mainly to tackle the folding problem. This is one of the two great challenges of doing computer simulations of biomolecules. (The other is to understand the function of enzymes in detail by combining quantum mechanics and molecular mechanics.) A Monte Carlo method for the long-term dynamics of proteins has been developed by the Knapp group for many years. My own contribution reported in chapter 2 is the further improvement of the folding of peptides *in vacuo* and finally the addition of a solvent model.

The third chapter comprises the major part of my work. Titrating proteins in the computer means to calculate the protonation state of all titratable sites within a protein for different pH values. Redox active sites can be treated by very similar methods and are therefore included in the calculations. I applied the electrostatic methods for the calculation of titration behavior successfully to bacterial photosynthetic reaction centers and to myoglobin. The results did not only show the reliability of the methods by reproducing experimental measurements, but provided also new insights into the studied processes and enabled new interpretations of experimental findings that were previously not well understood. These results are an example that nowadays theory in bioscience can do much more than merely reproducing results everybody has known before.

Science without collaboration is impossible. And so I am especially happy to present the following long list of acknowledgments for all those helpful people and institutions who supported me during my work. At first, there is Ernst-Walter Knapp, who was an excellent *Doktorvater* for me, and his group, where I found perfect working conditions and plenty of inspiring discussion throughout all fields of science, from the foggy swamps of biology to the windy crests of mathematics. Specially, I am grateful to G. Matthias Ullmann (now working at the Scripps Institute) for his collaboration in the calculation of electron-transfer and protonation reactions of the bacterial photosynthetic reaction center (see chapter 3) and to Björn Kleier for the collaboration in preparing energy functions for CAMLAB++ (see chapter 2). The KARLSBERG program (see appendix E) would not exist, if Benedikt Dietrich had not introduced me to C++. It was a pleasure to take part in other projects of the group, especially Peter Vagedes' work on the deacylation step of acetylcholinesterase, Dragan Popović's work on the computer modeling of artificial cytochrome b, and Timm Essigke's study on using CAMLAB++ for solving NMR structures. To take care for the technical basis of our scientific work, a running network of computers, was a rewarding, informative, and mostly enjoyable task I was allowed to do together with Bernd Melchers, Timm Essigke and Daniel Winkelmann. I had the honor to contribute some theoretical parts to the experimental work of Jörg Wischhusen and Ioakim

Spyridopoulos at the university of Tübingen. I got financial support from the Deutsche Forschungsgemeinschaft, Sfb 312 and Sfb 498. I was allowed to use numerous computer programs free of charge: Specially, I thank Donald Bashford for providing MEAD and Daniel Hoffmann for providing CAMLAB. The typesetting of this thesis was done by LaTeX. Molecular graphics are drawn with MOLSCRIPT. Plots were done using XMGR. Most of the figures were prepared or postprocessed with XFIG.

At last, let me appreciate a change of paradigm that has taken place during my work in the Knapp group and is sometimes called the “open source revolution”. Thanks to the GNU project and expecially the Unix-like operating system Linux, we are now able to turn cheap off-the-shelf personal computers free of charge into all-round professional workstations, well equipped and stable enough to run our programs for weeks and months. We are now going to assemble lots of them to a massive parallel supercomputer, not more expensive than one or two of the workstations we used when I joined the Knapp group several years ago. The open source paradigm is deeply scientific in its desire to share information, ideas and improvements. The success of open source software made the world a little bit better by making it a little bit more scientific. Science benefits from open source software, but by doing so it should also feel exhorted not to forget the virtue of its own open principles.

BJÖRN RABENSTEIN

Freie Universität Berlin
July 2000

Contents

List of Figures

List of Tables

Appendix A

Abstract

The present work is divided into two parts: The first part presents the folding of peptides using a Monte Carlo method for the long-term dynamics of proteins. The second part contains studies of the titration and redox behavior of the bacterial photosynthetic reaction center and of myoglobin. By combining both parts, I suggest a new method to calculate titration behavior taking into account full conformational flexibility.

The Knapp group has been working on a Monte Carlo method for the long-term dynamics of proteins for a long time. I improved this method for the first part of this work. After that, it was possible to fold not only a model peptide to a helix–turn–helix motif, as already done before, but also to simulate the folding of a β-hairpin. However, so far these calculations all lacked a solvent model. Therefore, the Analytical Continuum Solvent model of Michael Schaefer was integrated into our method. Using this model, the formation and melting of a polyalanine helix could be simulated in agreement with results from a molecular dynamics simulation with an explicit solvent model. Also, the folding of a fragment of ribonuclease A was successful, but the folding of the β-hairpin forming peptide BH8 failed. This unveiled so far unknown problems of our protein model with rigid peptide planes and special effective torsion potentials for the backbone torsion angles. The folding of BH8 was successful using a fully flexible protein model. However, using the flexible model the folding is much less efficient as expected for the rigid model.

The calculation of protonation and redox states of titratable and redox-active groups of proteins is done on the basis of continuum electrostatics, similar to the solvent model used in the folding simulation described above. In continuum electrostatics, the solvent is represented by a medium of a high dielectric constant. By solving the Poisson-Boltzmann equation numerically on a grid, the resulting electrostatic potentials (and thus also the interaction energies) can be calculated. In this way, the total electrostatic energy of a specific protonation and redox state of a molecular system can be determined. A protein contains usually a lot of titratable or redox-active groups. For n of such groups, the total number of possible states is extremely large (2^n) so that the calculation of thermodynamical averages, which are necessary to determine titration curves and redox potentials, by exact summation is computationally infeasible. Therefore, I apply in the present work a Monte Carlo method where an importance sampling of the huge number of possible states is performed according to the Metropolis criterion. The method yields good results after relatively short sampling times. The statistical error of these results can be probed by evaluating a correlation function.

By applying these methods, I could gain a lot of valuable insights. For the first time, the energy of the electron transfer from $Q_A^{\cdot -}$ to Q_B in the bacterial reaction center was calculated correctly. The calculations also yield important hints for the so far unsettled sequence of the following electron transfer and protonation events involving the quinones of the reaction center. In addition, the conformational gating hypothesis of the electron transfer from $Q_A^{\cdot -}$ to Q_B could be supported from the viewpoint of theory, and the insight into the related processes was deepened. I present in this work several

approaches to include conformational ensembles and conformational flexibility in the calculation of titration behavior. By explicit consideration of conformational relaxation, the dielectric constant of the protein interior could be decreased remarkably in accordance with fundamental principles. By including an ensemble of x-ray structures, the pH induced conformational changes of myoglobin could be reproduced. The pK_a values calculated in this study are in better agreement with experimental values than any other result obtained by non-trivial methods before.

In the concluding outlook, I discuss the future improvements of the Monte Carlo method for the long-term dynamics of proteins, and the assets and drawbacks of existing methods to calculate titration behavior considering conformational flexibility. I show how to avoid most of the drawbacks by a new method on the basis of a combination of the Monte Carlo dynamics with the titration calculation. This method realizes the unrestricted and unbiased sampling of the conformational space and the space of titration states at the same time.

Appendix B

Zusammenfassung *(German abstract)*

Die vorliegende Arbeit gliedert sich in zwei Teile. Der erste Teil behandelt die Faltung von Peptiden mit einer Monte-Carlo-Methode zur Langzeitdynamik von Proteinen. Der zweite Teil enthält Studien zum Titrations- und Redoxverhalten des bakteriellen photosynthetischen Reaktionszentrums und von Myoglobin. Durch die Kombination beider Teile skizziere ich am Schluß der Arbeit eine neue Methode zur Titrationsberechnung von Proteinen mit vollständiger Konformationsflexibilität.

Die Arbeitsgruppe Knapp arbeitet bereits seit langer Zeit an der Entwicklung einer Monte-Carlo-Methode zur Langzeitdynamik von Proteinen. Diese Methode wurde von mir für den ersten Teil meiner Arbeit weiterentwickelt. Es gelang daraufhin nicht nur, wie bereits zuvor, ein Modellpeptid zu einem Helix–Turn–Helix-Motif zu falten, sondern auch die Faltung eines β-Hairpins zu simulieren. Diese Berechnungen beinhalteten jedoch noch kein Modell für das Lösungsmittel. In einem weiteren Schritt wurde daher das *Analytical Continuum Solvent*-Modell von Michael Schaefer in unsere Methode integriert. Mit diesem Modell konnte der Aufbau und das Schmelzen einer α-Helix aus Polyalanin in Übereinstimmung mit Ergebnissen einer Molekulardynamik mit explizitem Lösungsmittelmodell simuliert werden. Auch die Faltung eines Fragmentes der Ribonuclease A war erfolgreich, wohingegen die Faltung des β-Hairpin-bildenden Peptids BH8 mißlang. Hier zeigten sich bislang nicht erkannte Schwächen unseres Proteinmodells mit starrer Geometrie der Peptidebene und speziellen effektiven Potentialen für die Torsionswinkel. Die Faltung von BH8 konnte erfolgreich mit einem voll flexiblen Proteinmodell simuliert werden, allerdings weitaus weniger effizient als man es bei Verwendung des starren Proteinmodells erwarten würde.

Die Berechnung der Protonierungs- und Redoxzustände von titrierbaren bzw. redoxaktiven Gruppen in Proteinen erfolgte, wie auch schon beim für die Simulation der Proteinfaltung verwendeten Lösungsmittelmodell, auf Basis der Kontinuumselektrostatik, bei der das Lösungsmittel durch einen Bereich erhöhter Dielektrizitätskonstante simuliert wird. Durch numerisches Lösen der Poisson-Boltzmann-Gleichung auf einem Gitter lassen sich die resultierenden elektrostatischen Potentiale und damit auch die elektrostatischen Interaktionsenergien berechnen. Damit kann man die elektrostatische Energie eines bestimmten Protonierungs- und Redoxzustandes des Gesamtproteins bestimmen. Ein Protein verfügt in der Regel über eine große Zahl n von titrierbaren bzw. redoxaktiven Gruppen. Die Gesamtzahl an möglichen Zuständen (2^n) ist extrem groß, so daß sich thermodynamische Mittelwerte, die zur Bestimmung von Titrationskurven und Redoxpotentialen nötig sind, praktisch nicht exakt berechnen lassen. Daher verwende ich in der vorliegenden Arbeit eine Monte-Carlo-Methode, mit der eine Teilmenge der Vielzahl möglicher Zustände gemäß des Metropoliskriteriums durchmustert wird. Durch diese Methode erlangt man nach kurzer Zeit Ergebnisse hoher statistischer Genauigkeit, was durch Auswerten einer Korrelationsfunktion gezeigt werden kann.

Mit der genanten Vorgehensweise war es mir möglich, eine Vielzahl wertvoller Erkenntnisse zu sammeln. So konnte erstmalig die Energie des Elektronentransfers von $Q_A^{\cdot -}$ zu Q_B im bakteriellen Reaktionszentrum korrekt berechnet werden. Die Berechnungen ergaben auch wichtige Hinweise

für die zuvor ungeklärte weitere Reihenfolge der Elektronentransfer- und Protonierungsschritte der Chinone im Reaktionszentrum. Desweiteren konnte auch von theoretischer Seite die *conformational gating*-Hypothese betreffend den Elektronentransfers von $Q_A^{\cdot -}$ zu Q_B untermauert und das Verständnis der zugehörigen Vorgänge im Reaktionszentrum vertieft werden. Ich entwickele in der vorliegenden Arbeit mehrere Ansätze zur Einführung multipler Konformationen und von Konformationsflexibilität in die Titrationsberechnungen. In einem Fall konnte durch die explizit berücksichtigte strukturelle Relaxiation die Dielektrizitätskonstante für das Innere des Proteins entsprechend allgemeiner Prinzipien stark verringert werden. In einem anderen Fall wurde die Energetik pH-induzierter struktureller Änderungen von Myoglobin durch Einbeziehen eines Ensembles von Kristallstrukturen verstanden. Die dabei berechneten pK_a-Werte zeigen eine Übereinstimmung mit dem Experiment, die in dieser Genauigkeit niemals zuvor mit nicht-trivialen Methoden zur pK_a-Wertberechnung erzielt worden sind.

In einem Ausblick erörtere ich die in Zukunft möglichen Verbesserungen der Monte-Carlo-Methode zur Langzeitdynamik von Proteinen und die Vor- und Nachteile der existierenden Methoden zur Titrationsberechnung mit Konformationsflexibilität. Ich zeige auf, wie sich durch eine zukünftige Kombination der Monte-Carlo-Dynamik mit der Titrationsberechnung die meisten Nachteile der existierenden Methoden umgehen lassen. Diese Methode verwirklicht das uneingeschränkte und verzerrungsfreie gleichzeitige Durchmustern des Konformationsraumes und des Raumes der Titrationszustände.

Appendix C

Glossary

The following abbreviations and symbols are used in this work:

∇	Nabla operator, gradient
$\int_A$	surface integral
$\int_V$	volume integral
α	occupation probability
β	$(k_\mathrm{B}T)^{-1}$
Δ	Laplace operator
ε	dielectric constant
ε_p	dielectric constant of the protein interior
ε_s	dielectric constant of the solvent
ρ	density (usually charge density)
ϕ	electrostatic potential
A	area, surface
ACS	analytical continuum solvent
bRC	bacterial (photosynthetic) reaction center
c	concentration
CM	Cartesian move
CPU	central processing unit
$\vec{D}$	electrostatic displacement
DSSP	dictionary of secondary structures of proteins (method for secondary structure assignment)
$\vec{E}$	electrostatic field
FTIR	Fourier transform infrared (spectroscopy)
G	free energy, Gibb's energy
I	ionic strength
k_B	Boltzmann constant
MbCO	carbonmonoxymyoglobin
MD	molecular dynamics
MQ	menaquinone
P	special pair (of the bRC)
PBE	Poisson-Boltzmann equation
PDB	Brookhaven Protein Data Bank

q	charge
Q_B	primary quinone acceptor in the bRC
Q_B	secondary quinone acceptor in the bRC
R	molar gas constant
$\vec{r}$	coordinates of a point in three-dimensional space
Rb. sphaeroides	*Rhodobacter sphaeroides*
RP	Ramachandran plot based method of secondary structure assignment
Rps. viridis	*Rhodopseudomonas viridis*
SM	simple move
T	absolute temperature
u	(electrostatic) energy density
UQ	ubiquinone
V	volume
WM	window move

Appendix D

CAMLAB++ Manual

This is a manual of the additional commands and utilities of CAMLAB++ compared to the normal CAMLAB program. To use CAMLAB++ , you will also need the standard CAMLAB manual, which can be found in the WWW:
`http://cartan.gmd.de/compchem/hoffmann/CAMLab/manual/manual.html`
The documentation of the ACS parts of CAMLAB++, which were mainly implemented by Björn Kleier, are still in a preliminary state and not included here.

D.1 Commands

D.1.1 Subcommands of `read_energy_terms`

These commands can be used as *type*-specifier in the `read_energy_terms`-command.

D.1.1.1 `lj_pair`

Format of additional data:
`n_pairs` *numberOfLJPairs*
molname_1 atomnum_1 molname_2 atomnum_2 LJe LJr
...

Description:
`lj_pair` is used to introduce special Lennard-Jones interactions between two individual atoms. In each line two atoms are specified by molecule name (*molname*) and atom number (*atomnum*). The Lennard-Jones interaction is given by the two usual Lennard-Jones parameters energy (*LJe*) and radius (*LJr*).

D.1.1.2 `noe`

Format of additional data:
`temperature` *tempParam*
`scaling` *scaleFactor*
`n_distances` *numberOfNOEConstraints*
`a` *numberOfAtomsInGroupAOfConstraint1*
m1a a1a
m2a a2a
...
`b` *numberOfAtomsInGroupBOfConstraint1*

m1b a1b
m2b a2b
...
`para` *kmin rmin kmax rmax fmax*
`a` *numberOfAtomsInGroupAOfConstraint2*
...
...

Description:
`noe` is an energy function to use NOE constraints within CAMLab. It is very similar to `distance_1`, but all energy terms are multiplied by *tempParam* and *scaleFactor*. If you use the same parameter for *tempParam* and for the temperature parameter in the metropolis criterion, temperature will have no effect on the NOE energy terms in MC sampling.
In addition to `distance_1`, `noe` allows to give more than one atom on both side of the NOE constraint. All possible distances are then averaged by the following equation:

$$\bar{r} = \frac{1}{\sqrt[6]{\sum_{i,j} \frac{1}{r_{ij}^6}}}$$

D.1.1.3 `quadratic_repulsion`

Format of additional data:
`scaling` *scaleFactor*
`n_atoms` *numberOfAtoms*
moleculeName numberOfAtom1 radiusOfAtom1 forceConstant1
moleculeName numberOfAtom2 radiusOfAtom2 forceConstant2
...

`quadratic_repulsion` is an energy function which calculates the quadratic repulsion of `atom` i with each other `atom` j based on its radii r_i and r_j and their distance r_{ij}. The atomic force constants k_i and k_j are scaled by the scaling factor s.

$$E = s \sum_{i>j} \sqrt{k_i k_j} (r_{ij} - \frac{1}{2}(r_i + r_j))^2$$

D.1.1.4 `sartori`

Format of additional data:
`epsilonfactor` *number*
`epsilonfunction` *funtionString*
`epsilon14factor` *number*
`number_of_torsion_potentials` *numberOfTorsionPotentials*
`torsion_potential` *nameOfPotential_1*
`dimension` *n* ! number of values in one dimension
energyValue_phi_1_psi_1
energyValue_phi_1_psi_2
...
energyValue_phi_1_psi_n
energyValue_phi_2_psi_1

energyValue_phi_2_psi_2
...
energyValue_phi_n_psi_n
`torsion_potential` *nameOfPotantial_2*
`dimension` *numberOfValuesInOneDimension*
...
`number_of_molecules` *number*
`molecule` *nameOfMolecule_1*
`bonds` *numberOfBondTermsInMolec_1*
atomnumber_1 atomnumber_2 forceConstant equilibriumDistance
...
`bondangles` *numberOfBondangleTermsInMolec_1*
atomnum_1 atomnum_2 atomnum_3 forceConstant equilAngle
...
`torsions` *numberOfTorsionTermsInMolec_1*
atomnum_1 atomnum_2 atomnum_3 atomnum_4 forceConst multiplicity equilTorsion
...
`impropers` *numberOfImproperTermsInMolec_1*
atomnum_1 atomnum_2 atomnum_3 atomnum_4 forceConst equilImproper
...
`nonbondeds` *numberOfNonbondedsInMolec_1*
atomNum mass charge LJe LJr LJ14e LJ14r deltaR
...
`excluded_nonbonded_pairs` *numberOfExcludedNBPairs*
atomnum_1 atomnum_2
...
`torsions_with_potential` *numberOfTorsionsWithPotentialInMolec_1*
nameOfPotential phi_1 phi_2 phi_3 phi_4 psi_1 psi_2 psi_3 psi_4
...
`fixed_bonds` *numberOfFixedBondsInMolec_1*
atomnum_1 atomnum_2 bondLength
...
`fixed_angles` *numberOfFixedAnglesInMolec_1*
atomnum_1 atomnum_2 atomnum_3 angle
...
`fixed_torsions` *numberOfFixedTorsionsInMolec_1*
atomnum_1 atomnum_2 atomnum_3 atomnum_4 torsionAngle
...
`fixed_impropers` *numberOfFixedImpropersInMolec_1*
atomnum_1 atomnum_2 atomnum_3 atomnum_4 improperTorsionAngle
...
`molecule` *nameOfMolecule_1*
`bonds` *numberOfBondTermsInMolec_2*
...
...

Description:

`sartori` is very similar to `md_force_field`. It does everything what `md_force_field` does. In addition, it is possible to read in twodimensional torsion potentials for pairs of dihedral angles.

Also parameters for fixed bond lengths and bond angles can be given.
Here only the differences to the `md_force_file` subcommand are described.
To switch off the calculation of Coulomb interaction, set `epsilonfactor` to zero.(That is necessary if you want to substitute the Coulomb part by another forcefield, e.g. `acs`.)
The `epsilon14factor` is applied to Coulomb interaction of 1-4-linked atoms. It is 0.5 for the Charmm parameter-set of MSI, 0.4 for the Charmm PARAM19 set, and 1.0 for the Charmm PARAM22 set.
Torsion potentials are given as energy values on a twodimensional grid with n grid points in each dimension. The grid is torus-like. The first grid point in a grid row is identical to the last. Hence, with a 10° grid n is 37. *energyValue_phi_1_psi_1* is the energy value at $\phi = -180°$ and $\psi = -180°$. *energyValue_phi_n_psi_n* is the energy value at $\phi = +180°$ and $\psi = +180°$. Both gridpoints and values are identical. CAMLAB++ uses the torus-like form of the potential to check for consistency of the given data. The input data for the torsion potentials can be generated using the stand-alone program →`MakeSartoriPotentials`.
In the energy function of Fredo Sartori, there is a correction parameter for the Coulomb energy:

$$E_{Coulomb} = \frac{1}{4\pi\varepsilon\varepsilon_0}\frac{q_1 q_2}{r + \Delta R}$$

The parameter ΔR (in Å) can be given for each individual atom in the `nonbondeds` section (*deltaR*). Usually, it is only applied to H atoms.
The twodimensional torsion potentials usually include some non-bonded terms implicitly. These non-bonded interactions must be switched off individually using the `excluded_nonbonded_pairs` subcommand. The order of the pairs is critical. The first atom number must always be greater than the second. The first atom number in row $n+1$ must be less or equal than the first atom number in row n. If the first atom numbers in row $n+1$ and n are equal, the second atom number in row $n+1$ must be less than the second atom number in row n.
With the `torsions_with_potential` subcommand pairs of dihedral angles can be assigned to the twodimensional torsion potentials defined by the `torsion_potential` subcommand.
The remaining subcommands are used to define values for fixed degrees of freedom. These values are only used for things like chain rebuilding. They don't make sure that these degrees of freedom are really fixed. If you apply for example a cartesian move, the "fixed" degrees of freedom will probably change. Thus, you have to take care to keep the "fixed" degrees of freedom really fixed by an appropriate selection of moves.
Since the chain rebuilding procedure is not yet implemented in CAMLAB++, these commands are actually useless with the important exception of `fixed_bonds`. The parameters of this subcommand in connection with those of the `bonds` subcommand are used to create the bonded list. Each bond in the molecule must be representated either in the `bonds` section or in the `fixed_bonds` section.
The parameters for a molecule can be generated using the stand-alone program →`MakeSartoriEterms`.

D.1.2 Subcommands of `read_move`

D.1.2.1 `dihedral_window`

Format of additional data:
`old_energy` *EoldParaName*
`new_energy` *EnewParaName*
`selection_energy` *Eselection*
`number_of_tries` *NumberOfTries* ! (optional)
`conformational_change` *ConfParaName* ! (optional)

`acceptance` *AcceptParaName*
`backbone_dihedral_scaling` *BBDihedralScaleFact*
`sidechain_dihedral_scaling` *SCDihedralScaleFact*
`prerotation_dihedral_scaling` *PRDihedralScaleFact* ! (optional)
`cartesian_displacement` *Displacement* ! (optional)
`int_list` *WList*
`energy_subset_index` *EnergySubsetParaName*
`atom_subset_index` *Asubset*
`criterion` *CritType*
... (criterion data)
`end_criterion`
`number_of_molecules` *n_molecs*
`molecule` *molecName*
`number_of_chains` *ncs*
`start_chain` *chaintype aStart* ! (*chaintype* is `fixed` or `free` or `cyclic`)
`sc` *s1 s2 s3 s4 DeltaPhiS*
`l` *nls*
a_first_1 a_last_1
...
a_first_nl a_last_nl
`r` *nrs*
...
a_first_nr a_last_nr
...
`cartesian` *nCart* ! (optional)
a_first_1 a_last_1
...
a_first_nCart a_last_nCart
`bb_pivot` *apivot*
`bb` *b1 b2 b3 b4 DeltaPhiB*
`l` *nlb*
...
`r` *nrb*
...
`sc` *s1 s2 s3 s4 DeltaPhiS*
...
`cartesian` *nCart* ! (optional)
...
`bb_pivot` *apivot*
...
`end_chain` *chaintype aEnd*
`start_chain` *chaintype aStart*
...

Description:

The `dihedral_window` command of standard CAMLab has been replaced in CAMLAB++. The new `dihedral_window` command has some additional features but is nevertheless completely compatible to the old version. All new keywords are optional and can be omitted in the input file. However, if you want to use one of the new features, you have to insert the new keyword exactly at

the place given in the syntax description above.
All keywords not mentioned below have the same meaning as in the old version of `dihedral_window`.
`number_of_tries`: If after application of the prerotations no geometrical solution is possible, the program will try it again with new prerotations, until the number of tries reaches the value of this parameter.
`conformational_change`: If a new conformation (i. e. different from the original conformation) is selected, the value of this parameter is set to 1 (if not, it is set to 0). The value of this parameter will be 1 even if the new conformation is rejected by the respective `criterion`. With this parameter, you can distinguish between "acceptance due to unchanged conformation" (i. e. unchanged energy) and "real acceptance" (of a new conformation).
`prerotation_dihedral_scaling`: This parameter scales only the prerotations. If it is given, `dihedral_scaling` scales only unconstrained moves at a free end or start of a chain and dihedral changes that are not prerotations. If you set `prerotation_dihedral_scaling` to a small value and `dihedral_scaling` to a large value, you will get easily a lot of geometrical possible solutions of the window move. It is, however, questionable wether detailed balance is maintained or not.
`cartesian_displacement`: This parameter gives the maximal possible displacement for the cartesian move.
`cartesian`: In this section, atoms for the cartesian displacement, which is applied if a new conformation has been selescted, are selected. To select atoms in the sidechain, this block must appear between the two dihedral angles at the pivot atom.

D.2 Utilities

The items described here are not CAMLAB++ commands but stand-alone programs.

D.2.1 Generation of Input Data

D.2.1.1 `MakeSartoriPotentials`

Syntax:

`MakeSartoriPotentials` *inputfile potential-name* [*factor*]

This tool converts a two dimensional potential from a raw-ASCII format to the CAMLAB++ input format. All values are multiplied by *factor*.

D.2.1.2 `MakeSartoriEterms`

Syntax:

`MakeSartoriEterms` *masses.rtf parm.prm psf-file delta-R molecule-name output-file*

A CHARMM `psf`-file is converted to the generalized input format of CAMLAB++. For this process, also the CHARMM parameter file and `rtf`-file is needed. *delta-R* is the hydrogen-bond correction-value.

D.2.1.3 `MakePivot`

Syntax:

`MakePivot` *dihedral-intervall offset aa1* [*aa2* . . .]

This is a `perl` script that writes a CAMLab pivot file to `stdout`. The script is very specialized and will work only under special conditions. The peptide for which the pivot file is created must be built with CHARMM19 using the `NACT` and `CMAM` patch. Some customization can be done by setting a suitable *offset* for the atom numbers. However, you will have to edit the first residue. *dihedral-intervall* is the maximal change of a dihedral angle during a window move. *aa*n denotes the respective residue. Currently supported are `ALA`, `ARG`, `ASN`, `GLU`, `GLY`, `HSC` (protonated His), `ILE`, `LEU`, `LYS`, `PHE`, `THR`, `TYR`, `VAL`. Timm Essigke has written the corresponding program for CHARMM22.

D.2.1.4 `MakeDihedrals`

Syntax:

`MakeDihedrals` *dihedral-intervall offset aa1* [*aa2* ...]

This is analogous to `MakePivot`, but does not create a pivot file, but a dihedral file for the application of simple dihedral moves.

D.2.2 Processing of Output Data

D.2.2.1 `clt2dcd`

Syntax:

`clt2dcd` [`-h`] [`-f` *steps-per-record*] [`-n` *stepoffset*] [`-o` *dcd-out-file*] [`-s` *frame-skip*] [`-swap`] *clt-in-file(s)* ...

This program converts CAMLab trajectory files into CHARMM `dcd` trajectory files. All input files are merged to one output file in the order given at the command line. They must be all for the same molecule. The *stepoffset* is only applied once at the beginning, but *steps-per-record* must be valid for all input files. If no name for the *dcd-out-file* is given, a name is generated from the first input file name. With the `-swap` option given, the byte order is swapped so that the `dcd` file is usable on machines with different endian (only).

Appendix E

KARLSBERG Manual

This is the orginal version of the manual for KARLSBERG 0.4. I did not apply any changes to prevent confusion. So you will find a kind of self-reference to my PhD thesis within the manual. I hope this will not be the source of another kind of confusion. I assure you that the referred PhD thesis is really just the book you are reading now. The manual (perhaps even a newer version) can also be found at the KARLSBERG homepage:
`http://lie.chemie.fu-berlin.de/karlsberg/`

E.1 What is KARLSBERG?

KARLSBERG is a program for Monte-Carlo (MC) calculation of titration patterns of proteins. It needs intrinsic pK_a values of the titratable residues and an interaction matrix between them as input. It will calculate the protonation probabilities for these titratable sites at given pH values. This input usually comes from electrostatic programs, *e. g.* MEAD by Donald Bashford (Bashford & Karplus, 1990). So far, KARLSBERG is very similar to the MCTI program of Paul Beroza (Beroza *et al.*, 1991). All features of MCTI are also present in KARLSBERG: double moves, reduced site titration, error estimation by correlation functions. However, there are some additions: KARLSBERG is able to do parallel tempering (Hansmann, 1997), triple moves (Rabenstein *et al.*, 1998a), energy-biased MC (Beroza *et al.*, 1995) and, most important, sampling of multiple conformations using an MC version of the method described by Rabenstein *et al.* (1998a).
Papers where KARLSBERG was applied, are Rabenstein *et al.* (2000), Rabenstein and Knapp (2000a), Vagedes *et al.* (2000). Especially in Rabenstein and Knapp (2000a) you will find most of the scientific background described at one place. An even more complete source is my PhD thesis, which will be available as soon as it is published via my WWW publication list:
`http://lie.chemie.fu-berlin.de/~rabe/science/publications.html`
KARLSBERG itself is also available in the WWW:
`http://lie.chemie.fu-berlin.de/karlsberg/`

E.2 Installation

Untar the tarball.
Binaries for Linux and IRIX are part of the distribution (`karlsberg.Linux` and `karlsberg.IRIX`). Move the correct binary to your favored binary directory and rename it to `karlsberg`. The utilities (`make_curves` and `make_protonation_table`) are Perl programs, so you need Perl to run them.

KARLSBERG is written in C++, so you need an appropriate compiler, if you want to compile it yourself.
Go to the `src` subdirectory and type `make` (you must use GNU-make!). This will work on Linux and IRIX. The resulting binary is called `karlsberg.Linux` or `karlsberg.IRIX`, respectively. (They are created in the same directory as the sourcefiles.) For other platforms, you must unfortunately help yourself and modify the Makefile.

E.3 Running KARLSBERG

There are no command line parameters. All program parameters and special commands are read from stdin line by line. Usually, you will pipe a control file with all the parameters and commands into the program:

`karlsberg <` *input-file* `>` *log-file*

KARLSBERG tells you a lot about what it is doing to stdout. This output should be self-explaining. I/O from and to files is controlled by the commands given to stdin. As input you need at least a file with intrinsic pK_a values and a file with the interaction matrix. The real results are written to files, not to stdout.

E.3.1 Sample files

In the directory `sample` of the distribution, you find a set of sample files. Run KARLSBERG by typing
`karlsberg < myoglobin.in > myoglobin.out`
To understand the commands in `myoglobin.in`, read the following section.

E.3.2 Syntax of input to stdin

Each line starts either with a keyword or with the comment sign "#". Lines starting with "#" are ignored. You can use the comment sign also within a line. The part of the line after "#" is ignored. After a keyword, parameters may follow. If you give too many parameters, KARLSBERG will ignore the surplus. If you give too few, KARLSBERG will assign default values if there are any. Input is finished by a line starting with the keyword `end`. Everything after this line is ignored. If no such line is present, input will stop when `eof` is detected. Order of lines may be arbitrary with two exceptions: The one is the `end` line (of course), and the other are repeated keywords. Most keywords should only appear once in the input. If such a keyword is used more than once, only the first appearance is recognized, all others are ignored. In the following, all keywords with their respective parameters are listed (in alphabetical order) and explained. In the syntax description, square brackets ([. . .]) mark optional parameters. Default values for these parameters are given in the description.

`bias` *site_name energy* [*unit*]
This keyword can be used more than once. It defines a bias energy that is added to the protonation energy of the specified site. (For details see Beroza *et al.* (1995).) The default unit is e^2/Å. Other units may be specified as *unit*: `kJ/mol`, `kcal/mol`, `meV`, `eV`.

`conformation` *pkint_file interaction_matrix_file* [*reference_energy* [*unit*]]
This keyword can be used more than once. Each occurrence generates a new conformer. Data for intrinsic pK_a values and the interaction matrix are read from the given files. The default value for the reference energy is 0 (zero), the default unit is e^2/Å. Other units may be specified

as *unit*: `kJ/mol`, `kcal/mol`, `meV`, `eV`.
Format for *pkint_file*: One row per titratable site. Each row has three items separated by spaces. First item: intrinsic pK_a value. Second item: one letter, `A` (anionic) or `C` (cationic), anionic means uncharged reference state is protonated, cationic means uncharged reference state is unprotonated. The rest of the line is third item: name of site. It is recommended to use site names without spaces, since you can not give site names with spaces in the commands `bias` and `pH_independent`.
Format of *interaction_matrix_file*: One row per interaction (total number of rows is number of titratable sites squared). Each row has three items separated by spaces (items after the third are ignored). First item: number of first site (sites are numbered from 1 to *n* in the order they occur in *pkint_file*). Second item: number of second site. Third item: interaction energy in units of e^2/Å.

`conformation_moves_per_scan` [*real_number*]
Number of conformation-changing moves per MC scan. These moves are randomly mixed with the other moves. This parameter is a statistical expectation value. It may be smaller than one. Default value is 1.

`conformation_reference_state` *site_name state*
Sometimes the conformational reference energy can not be determined with all titratable sites in their uncharged reference state. The titration state of individual titratable sites during the calculation of the conformational reference energy can be given using this keyword, if it is different from the uncharged state used as reference state during the calculation of the intrinsic pK_a values. *state* is either `P` (protonated) or `U` (unprotonated). This keyword can be used more than once.

`correlation_limit` [*factor*]
This is the fraction of the variance that has to be reached by the correlation function to determine correlation time. Must be greater than zero and smaller than one. Default value is 0.1. (For details see Beroza *et al.* (1991).)

`end` *(no parameters)*
This keyword tells KARLSBERG to stop reading from stdin and to start the calculation.

`full_scans` [*number_of_scans*]
Number of MC scans with all titratable sites. Default value is 10,000. One scan comprises as many random attempts to change the titration states as different titration moves are eligible (single, double, and triple moves together) plus the given number of conformational and tempering moves (see `conformation_moves_per_scan` and `tempering_moves_per_scan`). In addition to the number of full scans given here, KARLSBERG will apply 10 equilibration scans before the production run is started.

`min_int_pairs` [*interaction-energy*]
Minimal interaction energy in pK_a units to consider a pair of titratable groups as strongly coupled. For such pairs, double moves are applied in addition to simple moves. Default value is 2.5 pK_a units. If 0 (zero) is given as parameter, no double moves are done at all. If you are using multiple conformations, for each conformation the interaction energies may be different. KARLSBERG will always look for the strongest available interaction energy of a specific site pair.

`min_int_triples` [*interaction-energy*]
Minimal interaction energy in pK_a units to consider a triple of titratable groups as strongly

coupled. For such pairs, triple moves are applied in addition to simple moves. Default value is 5 pK_a units. If 0 (zero) is given as parameter, no triple moves are done at all. Please note, that for three groups A, B, and C the triple criterion is fulfilled if A and B are coupled strongly enough and B and C, too. A and C do not need to be strongly coupled. If you are using multiple conformations, for each conformation the interaction energies may be different. KARLSBERG will always look for the strongest available interaction energy of a specific site pair.

`output` [*output_file_prefix*]
All output files are generated using this prefix. Default value is `karlsberg-output`.

`pH_end` [*pH_value*]
At this pH, titration stops. Must be greater than or equal `pH_end`. Default is equal to `pH_start`.

`pH_incr` [*increment_value*]
Increment for pH value. Must be positive. Default is 0.1.

`pH_independent` *site_name*
Not only titratable groups, but also redox groups can be treated by KARLSBERG. The redox potential of these groups is independent on pH. This keyword, which can be used more than once, marks a site as a pH-independent redox group. Which redox state is the pseudo-"protonated" and pseudo-"unprotonated" and which of them is the reference state, depends on your input.

`pH_start` [*pH_value*]
At this pH, titration begins. Must be smaller than or equal `pH_end`. Default is 7.0

`reduced_scans` [*number_of_scans*]
Number of MC scans with the reduced set of titratable sites (see also `reduced_set_tolerance`). Default value is 100,000. One scan comprises as many random attempts to change the titration states as different titration moves are eligible (single, double, and triple moves together).

`reduced_set_tolerance` [*tolerance_value*]
After the full MC calculation has finished, all titratable sites whose protonation probability has a difference to zero or unity smaller than this value are fixed in their respective state (fully protonated or unprotonated). The difference must be small enough in *all* conformations (that were sampled at least once) and at *all* temperatures (of parallel tempering). After that, reduced MC calculation is done, where the fixed titratable sites are not involved any longer. Default value is 0.000001.

`seed` [*positive_integer*]
Seed number for the random generator. Must be a positive integer. If 0 (zero) or nothing is given, the seed is taken from the present system time.

`temperature` [*temperature*]
Temperature for MC simulation in K. Default value is 300 K. This keyword can be given more than once. If this is the case, parallel tempering is done with one copy per given temperature. The lowest temperature given in this way is assumed to be the temperature, where you have calculated the intrinsic pK_a values (KARLSBERG uses this temperature for converting pK_a units to kJ/mol).

`tempering_moves_per_scan` [*real_number*]
Number of tempering moves per MC scan. The tempering moves are randomly mixed with the other moves. This parameter is a statistical expectation value. It may be smaller than one. The default value is 1.

E.3.3 Output

KARLSBERG generates one output file for each combination of pH, temperature, conformation and full / reduced sampling. File names look like

*prefix*_[full|red]_pH*pH_temp*K_conf*conf-nr*

Conformation number 0 (zero) means that all conformations are considered together. If only one conformation is considered, the last part of the filename is omitted. In each file, there is a single number in the first line. This number is the relative occupancy of the respective conformation during the MC simulation. After that, a table with five columns follows. The first column is the site number, the second the site name, the third the biased protonation probability, the fourth the statistical error (standard deviation), and the fifth the rebiased protonation probability. If you have not applied biased MC, the third and fifth column are equal. Statistical error for the rebiased protonation probability is not calculated. In the first row of the table (site number 0 (zero)), the sum of all sites is given.

E.4 Postprocessing of output

E.4.1 `make_curves`

Syntax:

`make_curves` *prefix*

<prefix> must include the `_full` or `_red` part of the filenames. The program will try to find all usable output files of KARLSBERG by itself.
This postprocessing program writes one file for each site, one for the sum of all sites, and one for the occupancy of the different conformations. It takes data from the last column of the KARLSBERG output files (rebiased protonation) and puts them for all pH values together. It will try to calculate the pK_a value and the slope of the Hill plot. The result is written to the comment section of the output files. After this comment section a table follows. In the first column of this table you will find the pH value and in the following columns the protonation probability for the different conformations. The file with the occupancies of the different conformations has a format similar to that of the protonation probability files. In the first column, you will find again the pH. In the following columns you will find the occupancies of the conformations. The output files can be visualized using a program like `xmgr` or `gnuplot`. The treatment of histidines is special (Bashford *et al.*, 1993).

E.4.2 `make_protonation_table`

Syntax:

`make_protonation_table` *file1* [*file2* [...]] [*start-conf end-conf*]

This postprocessing program writes a LaTeX file of a protonation table for the given files. It is a Perl script. See the script file itself for details. The treatment of histidines is special (Bashford *et al.*, 1993).

Appendix F

Curriculum vitae of the author

Björn Rabenstein

1971/10/20 born in Berlin, Germany

1978–1984 primary school: *Annedore-Leber-Grundschule*, Berlin-Tempelhof

1984–1991 high school: *Georg-Büchner-Oberschule (Gymnasium)*, Berlin-Tempelhof

1991 *Abitur* (≈high school diploma, grade 1.0, *sehr gut*)

1991–1992 personal *studium generale* at the *Freie Universität Berlin* in musicology and theology

1992–1997 study of biochemistry at the *Freie Universität Berlin* , including external research participations at the *Max-Volmer-Institut* of the *Technische Universität Berlin* and the *Weizmann Institute of Science*, Rehovot, Israel

1992–1997 scholar of the *German national scholarship foundation (Studienstiftung des deutschen Volkes)*

1994/07/04 *Vordiplom* in biochemistry (≈bachelor, grade: *sehr gut*)

1997/01/14 *Diplom* in biochemistry (≈master with thesis: *Protonierungs- und Redoxzustände der Chinone im bakteriellen photosynthetischen Reaktionszentrum von* Rps. viridis, grade: *sehr gut*)

1997–2000 PhD student in the macromolecular modelling group of Prof. E. W. Knapp

July 2000 PhD thesis: *Monte-Carlo Methods for Simulation of Protein Folding and Titration*

Parents:

Irene Rabenstein (administration secretary, programmer, catechist)

Ulf Rabenstein (construction engineer)

Teaching:

1995–1999 supervising several courses in basic lipid biochemistry and computational biochemistry

Present address:

office:

Institut für Chemie (Kristallographie)
Takustraße 6
D-14195 Berlin
Germany

e-mail: rabe@chemie.fu-berlin.de
phone: +49-30-838-53484
fax: +49-30-838-53464

private:

Mehringdamm 80
D-10965 Berlin
Germany

phone: +49-30-78954895

Publications

Journal articles:

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp (1998).** Energetics of Electron Transfer and Protonation Reactions in the Photosynthetic Reaction Center of *Rhodopseudomonas viridis*. *Biochemistry* **37**, 2488-2495

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp (1998).** Calculation of Protonation Patterns in Proteins with Structural Relaxation and Molecular Ensembles — Application to the Photosynthetic Reaction Center. *Eur. Biophys. J.* **27**, 626-637

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp (2000).** Electron Transfer between the Quinones in the Photosynthetic Reaction Center and its Coupling to Conformational Changes. *Biochemistry* **published on web**

- **P. Vagedes, B. Rabenstein, J. Åqvist, J. Marelius, E. W. Knapp (2000).** The Deacylation Step of Acetylcholinesterase: Computer Simulation Studies. *J. Am. Chem. Soc.* **in press**

- **B. Rabenstein, E. W. Knapp (2000).** Calculated pH-Dependent Population and Protonation of CO-Myoglobin Conformers. *Biophys. J.* **submitted**

- **I. Spyridopoulos, J. Wischhusen, B. Rabenstein, D. Axel, K.-U. Fröhlich, K. R. Karsch (2000).** Alcohol Enhances Oxysterol-Mediated Apoptosis by a Calcium-Dependent Mechanism. *Arterioscl. Throm. Vas.* **in press**

- **D. Popović, S. Zarić, B. Rabenstein, E. W. Knapp (2000).** Computer Modeling of Artificial Cytochrome b: Evaluation of Redox Potentials **in preparation**

Theses:

- **B. Rabenstein (1997).** Protonierungs- und Redoxzustände der Chinone im bakteriellen photosynthetischen Reaktionszentrum von *Rhodopseudomonas viridis*. Master thesis, Freie Universität Berlin.

- **B. Rabenstein (2000).** Monte Carlo Methods for Simulation of Protein Folding and Titration. PhD thesis, Fachbereich Biologie/Chemie/Pharmazie — Institut für Chemie, Freie Universität Berlin.

Talks at international conferences:

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp.** Protonation and Redox States of the Quinones in the Reaction Center of *Rps. viridis*. Talk at the 2nd European Biophysics Congress, EBSA97, Orléans July 13-17 1997. Abstract published in *Eur. Biophys. J.* **26**, 122.

- **B. Rabenstein, B. Kleier, D. Hoffmann, M. Schäfer, E. W. Knapp.** Folding of α-Helices and β-Hairpins Using a Monte-Carlo Method and an Implicit Solvation Model. Talk (and poster) at the International Scientific Congress of the *Belgian Biophysical Society* "Folding of Soluble and Membrane Proteins", Gembloux June 14-17 1999.

Book articles, conference proceedings etc.

- **B. Rabenstein, D. Hoffmann, E. W. Knapp (1999).** Simulation of oligopeptide folding or how do residues talk. In *Biological Physics - Third International Symposium,* AIP Conference Proceedings 487, American Institute of Physics, Melville, New York, pp. 54-68.

- **B. Rabenstein, E. W. Knapp (2000).** Problems Evaluating Energetics of Electron Transfer from Q_A^- to Q_B: the Light-Exposed and Dark-Adapted Bacterial Reaction Center. *submitted*

Poster presentations, talks of co-workers etc.

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp.** Energetics of Electron and Proton Transfer Reactions of the Quinones in the Photosynthetic Reaction Center of *Rps. viridis.* Poster at the International Meeting of the Deutsche Bunsen-Gesellschaft "Hydrogen Transfer: Experiment and Theory", Berlin September 10-13 1997.

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp.** Triple Moves in Monte Carlo Titration of Systems with Strongly Coupled Titratable Groups — Application to the Reaction Center of *Rps. viridis* with a Low Dielectric Constant. Poster at the HLRZ-Workshop "Monte Carlo Approach to Biopolymers and Protein Folding", Forschungszentrum Jülich December 3-5 1997.

- **B. Rabenstein, D. Hoffmann, F. Sartori, E. W. Knapp.** Folding of β-structure elements with a Monte-Carlo method. Poster at the Euroconference "Dynamics of Complex Molecular Liquids - Computer Simulations and Experiments", Vaalsbroek (NL) May 24-28 1998.

- **B. Rabenstein, G. M. Ullmann, E. W. Knapp.** Protonation Patterns of the Photosynthetic Reaction Center and Energetics of the Different Charge and Protonation States of the Quinones. Poster at the XIth International Congress on Photosynthesis, Budapest August 17-22 1998.

- **B. Rabenstein, D. Hoffmann, E. W. Knapp.** Simulation of oligopeptide folding or how do residues talk. Talk (speaker: E. W. Knapp) at the Third International Symposium on Biological Physics, Santa Fe September 20-24 1998.

- **B. Rabenstein, D. Hoffmann, F. Sartori, E. W. Knapp.** Folding of helix-turn-helix motifs and β-hairpins with a Monte-Carlo method. Poster at the annual meeting of the Deutsche Gesellschaft für Biophysik, Frankfurt/Main September 21-23 1998.

- **B. Rabenstein, B. Kleier, D. Hoffmann, M. Schäfer, E. W. Knapp.** Folding of α-Helices and β-Hairpins Using a Monte-Carlo Method and an Implicit Solvation Model. Poster (and talk) at the International Scientific Congress of the Belgian Biophysical Society "Folding of Soluble and Membrane Proteins", Gembloux June 14-17 1999.

- **E. W. Knapp, B. Rabenstein, G. M. Ullmann.** Energetics of the Electron Transfer from Q_A^- to Q_B for the Light and Dark Adapted Reaction Centers. Talk (speaker E. W. Knapp) at the 218^{th} American Chemical Society National Meeting, New Orleans August 22-26 1999.

- **E. W. Knapp, B. Rabenstein, G. M. Ullmann.** Electrostatic Interactions in Proteins. Talk (speaker E. W. Knapp) at the XIII International Biophysics Congress, New Delhi September 19-24 1999.

- **P. Vagedes, D. Popović, B. Rabenstein, E. W. Knapp.** Electrostatic Interactions in Proteins: Redox Reactions and Enzyme Catalysis. Talk (speaker E. W. Knapp) at the Third European Conference on Computational Chemistry, Budapest September 4-8 2000.

- **B. Rabenstein, E. W. Knapp.** Calculated pH-Dependent Population and Protonation of CO-Myoglobin Conformers. Poster at the 3rd European Biophysics Congress, München September 9-13 2000. .

- **P. Vagedes, D. Popović, B. Rabenstein, E. W. Knapp.** Electrostatic Interactions in Proteins: Redox Reactions and Enzyme Catalysis. Talk (speaker E. W. Knapp) at the International Meeting “Understanding Protein Electrostatics”, Huddinge (Sweden) September 15-18 2000.