QVADIS: A package to compute proton transfer pathways in proteins

Kerstin Kunz, Volkhard Helms

Center for Bioinformatics, Saarland University, 66041 Saarbrücken, Germany
kunz@graphics.cs.uni-sb.de, volkhard.helms@bioinformatik.uni-saarland.de

Abstract: Proton transfer reactions play an important role in many areas of biology. We reimplemented a method introduced by Taraphder *et al.* to characterize putative proton transfer pathways in proteins along hydrogen bonded networks and could retrace their results on *cytochrome P450cam*. By combining this method with our established *Q-HOP* approach to compute proton transfer probabilities we present here to our best knowledge for the first time a general program package to characterize proton transfer pathways in biomolecules accounting for both side chain flexibility and transfer kinetics.

1 Introduction

Chemical reactions involving proton transfer are crucial in many areas of biology such as in enzyme catalysis and in bioenergetics, but also in other areas of chemistry and material science such as in hydrogen fuel cells. Our own interest is the identification of proton transfer pathways through integral membrane pumps that are the basis for establishing a membrane potential and the synthesis of ATP. As experimental techniques are facing fundamental difficulties in resolving these long-range pathways in biomolecules, there is great promise for developing computational methods that may step in. Establishing these pathways may guide further experimental studies or the design of proton pumps with tailored conduction properties. Recent experimental and computational work showed that proton conduction in proteins mostly proceeds along hydrogen-bonded chains involving amino acid side chains and buried water molecules. The protons migrate as structural defects via the well-known Grotthuss mechanism [Mar06]. Importantly, the proton(s) will distribute over all kinetically accessible titratable groups of the protein subject to their local pKa values [SW04]. The dynamics of proton transfer is determined by the energy barriers between different states, often involving reorientations of side chains and water molecules. Therefore, proton conduction is intrinsically a dynamic phenomenon that is not well understood on the basis of static crystal structures of proteins alone. In 2003, Taraphder *et al.* presented an algorithm to characterize putative proton transfer pathways in proteins along hydrogen bonded networks [TH03b] [TH03a]. This algorithm accounts for side chain flexibility and scores the detected pathways on the basis of a simple steric energy function. Unfortunately, their package is not publicly available. On the other hand,

we have previously developed a parametrized scheme termed *Q-HOP* to compute proton transfer probabilities per time interval during classical molecular dynamics simulations [LH01b] [LH01a]. Input parameters are the chemical nature of donor and acceptor atom as well as their distance. In 2006, Herzog *et al.* extended the parameter set to transfers between all titratable amino acids and involving water [HFHL06]. This parametrization allowed us for the first time to simulate the proton equilibrium between acetic acid and a water box from unbiased molecular dynamics simulations in good agreement with the experimental pKa value [GFSH07]. By combining *Q-HOP* with the Taraphder-Hummer algorithm, we present here for the first time a general program package to characterize proton transfer pathways in biomolecules accounting for both side chain flexibility and transfer kinetics. In the following, we introduce the Taraphder/Hummer (TH) algorithm, discuss our implementation, followed by results obtained on the protein *CYP450cam*.

2 Methods

While it is known that proton transfer in biomolecules is critically coupled to protein dynamics, this flexibility is only considered at the level of dihedral rotations of side chain residues in the Taraphder-Hummer approach. In principle, the search could even be restricted to searching conformational dihedral space according to one of the available rotamer libraries [DK94]. However, some important conformations may be missed in this way. Therefore, we generate side chain conformations by systematic step-wise scanning of the side chain dihedral angles whereof the titratable amino acid residues have 1 (Thr, Cys), 2 (Asp, Tyr, His), 3 (Glu), 4 (Lys), or even 5 (Arg).

2.1 Taraphder/Hummer algorithm

The algorithm uses as input the three-dimensional structure of a macromolecule, such as a protein crystal structure from the Protein Data Bank [BWF$^+$00]. In the first step, the algorithm determines all titratable atoms of the macromolecule (Asp: OD1, OD2; Glu: OE1, OE2; His: ND1, NE2; Cys:SG; Tyr:OH; Ser:OG; Thr:OG, Lys:NZ; Arg:NH1, NH2, NE; HOH:OW; HEM: O1A, O2A, O1D, O2D), computes their pairwise distances, and outputs a list of clusters containing groups of titratable atoms whose distances are shorter than a predefined threshold. Here, we used 0.35 nm as in [TH03b]. The results can be conveniently represented as a connectivity matrix. Unfortunately, the clusters are typically quite small and do not yield, for example, connected pathways from a protein's surface to its active site. The second step considers, therefore, whether alternative side chain rotamers may be able to connect titratable atoms of adjacent static clusters obtained in the first step. In the third step, Dijkstra's algorithm is applied to find a connected pathway of lowest cost where cost is determined as the sum of the Boltzmann energy-weighted rotameric states (see eq. (2) below). To allow for an efficient computation of steric energies, the potential

u_{ij} between two heavy atoms i and j depending on their distance is computed as

$$u(r_{ij}) = k_B T \left(\frac{R_i + R_j - 2\delta}{r_{ij}} \right)^{12} \tag{1}$$

where R_i and R_j are the van der Waals radii of atoms i and j, and δ is a small correction distance of e.g. 0.02 nm to account for conformational flexibility of the atoms, e.g. according to backbone movements that are not considered explicitly. This expression is equivalent to the repulsive part of a Lennard Jones potential. For convenience, we considered all $1-4$ interactions of the respective residue and computed interactions with surrounding residues by introducing an atomic neighbor list. Comparing the energy of rotamer k with that of the static reference structure gives the energy difference ΔE_k. We did not implement the *Potential of Mean Force* method used by TH for positioning additional disordered water molecules. Then, an energy-weighted connectivity matrix is introduced to characterize the likelihood of establishing new connections between atoms i and j. This likelihood is computed as:

$$P_{ij} = \frac{\sum_k e^{-\beta \Delta E_k} f_{ij}^k}{\sum_k e^{-\beta \Delta E_k}} \tag{2}$$

The sum in the numerator runs over all conformers of the ensemble. In this sum, each state k is Boltzmann weighted by the energy difference ΔE_k of this state relative to the static structure ($\beta = 1/k_B T$ with the Boltzmann constant k_B and the temperature T). The connectivity parameter f_{ij}^k determines in dependence on a maximal distance whether the energy of the respective k-th conformer is considered or not in the summation. It takes on values of 1 if the distance between atoms i and j is within cutoff and 0 otherwise. The denominator provides a normalization by the summation over the energies of the whole ensemble. Subsequently, the connectivity matrix established in the first step will be filled by the newly found connections using these likelihoods as entries. In the final step, Dijkstra's algorithm is used to connect starting and end points of the proton by a lowest-cost pathway. Rotation of n rotatable bonds by m steps each, leads to a $(m \times n)$-dimensional search space. Each point in this space can be represented as an one-dimensional vector of length $(m \times n)$, where each element stands for a particular rotation of the corresponding dihedral angle. To simplify the pathway search, TH assumed that transitions are only allowed between neighboring nodes, i.e. that differ only in one coordinate that is either increased or decreased by the corresponding stepsize. The transition probability between nodes i and j is computed as

$$p(i \rightarrow j) = min\left\{1, \exp(-\beta \Delta E_{ij})\right\} \tag{3}$$

with ΔE_{ij} being the energy difference between nodes i and j. The transition rate along a pathway of mutually neighboring nodes is approximated as:

$$p(i_0 \rightarrow i_1 \rightarrow ... \rightarrow i_n) = \prod_{\alpha=0}^{n-1} p(i_\alpha \rightarrow i_{\alpha+1}) \equiv e^{-S} \tag{4}$$

where the action S is the sum of all energy increases. Application of the Dijkstra algorithm will provide the pathway of minimal S, i.e. the pathway involving the least uphill energy.

2.2 Implementation

Java3D is a package of Java classes to represent 3D graphics in applications and applets.[1] A scene is described by a scene graph which contains the composition of the scene as an arborescent structure allowing for a convenient handling of objects. A scene graph is set up by the viewing and the content branches. The viewing branch contains all parameters used for the presentation of the scene such as the position and orientation of a user in a universe (the highest organizational level in a virtual 3D world containing all objects) and the view on this universe. The content branch describes the contents of a scene as geometry, appearance, position and behavior of objects in a virtual world. This is managed by the following elements: A *BranchGroup* allows to append additional elements in the hierarchies below itself, a *TransformGroup* can influence subordinate nodes, a *Transform3D* performs the transformation on a partial scene graph and a *Shape3D* refers to the geometry and appearance of its object.[2]

2.2.1 Scene graph composition of QVADIS

While *Java3D* is compatible with different platforms and allows performing remote applications, it has certain deficits related to its performance concerning interaction speed or its ability to load large molecules. In 2003, Can *et al*. [CWWS03] introduced a method to tackle these problems. Their *FPV* method was used here as skeletal structure for the scene graph concept to share the main functionalities of loading and reading a PDB file and creating a static scene in which required objects can be inserted. *Java3D* easily allows rotating the different rotation axes of side chains of selected residues. Also it allows to apply an appearance to the respective atom objects to visualize what is going on. *QVADIS* uses the main scene (root TransformGroup) created by *FPV* providing the world coordinates as a reference for the coordinates of the other objects. For each side chain to be rotated, a new BranchGroup is created. The Cα as well as the other side chain atoms are assigned a TransformGroup. Each of these TransformGroups has methods to put and get coordinates for all its geometry. When a TransformGroup generates the origin of a rotation axis, it is assigned a RotateBehavior to allow for a user selected rotation of this particular axis. The automatic rotation of a side chain is controlled by the existence of a TransformGroup and a rotation axis that are related to the considered atom. A fictitious example for assigning TransformGroups within a side chain is shown in fig. 1.

[1] Sun Developer Network: http://java.sun.com/products/java-media/3D/

[2] Getting started with Java3D. Dennis J. Bouvier. http://java.sun.com/products/javamedia/3d/collateral

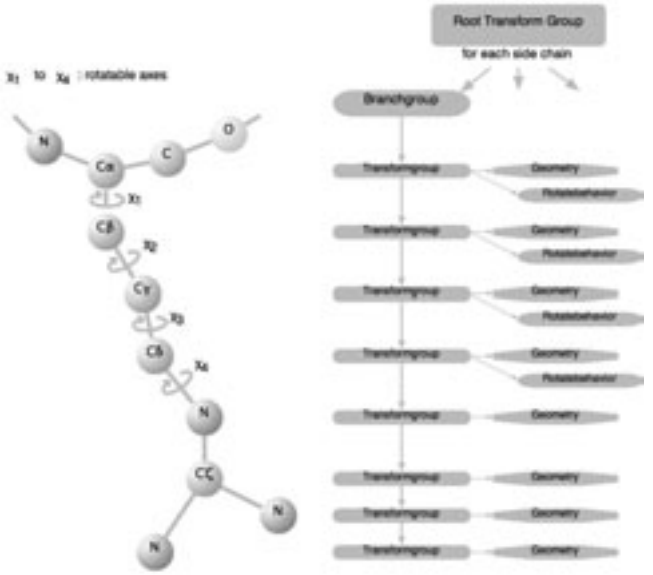

Figure 1: Scene graph in QVADIS for the side chain rotations of an Arginine residue. In this example, the rotation of the $\chi 5$ axis is not included.

2.2.2 Implementation of the Taraphder/Hummer algorithm

As exhaustive searching of conformational space to determine the best pathway would lead to a combinational explosion, a different approach is taken in *QVADIS* as explained below. Contacting titratable atoms are found by application of a distance criteria of 0.35 nm and are recorded in a connectivity matrix. A recursive algorithm combines all neighboring atoms to a list of clusters of differing size. Before being able to rotate side chain dihedral angles that may be involved in hydrogen bonded networks, the respective transform groups and the respective rotation axes need to be constructed. The stepsizes for rotation around particular dihedral angles may be controlled by external parameters as well as starting and end points of putative pathways and directives controlling graphical output. In case no start- and endpoints are known, TH suggested to search the side chain conformations of all those titratable residues along fragmented pathways. In *QVADIS*, this is implemented as automatic rotations of the involved side chains, computation of the newly constructed conformations, searching for neighbors, and update of the connectivity matrix according to eq. (2). In case one of the endpoints of the proton transfer pathway is known, the expensive update of the connectivity matrix is not required. Instead we perform a backward search starting at the endpoint and enlarge the cluster by subsequently including other residues or clusters found via suitable (low-energy) contacts of rotated side chains. All side chains of residues whose Cα atoms are in an appropiate distance to a searching group are rotated where searching denotes an already clustered residue or water molecule that looks out for other titratable residues or water molecules within a cutoff distance that can enlarge the

cluster. By doing so, we can omit calculating the steric energies, because these energies anyhow have to be computed in the following step to determine the transfer rates. As well we can leave out a complete axis rotation if a new connection has already been found. In this way one can fairly quickly obtain a list of residues and water molecules that are connected in a network. Most interesting are those pairs of residues that were not identified in the static clustering step but are found to connect neighboring static clusters. As the pathway of lowest cost will be found by the Dijkstra algorithm, transition rates are being computed in the next step.

2.2.3 Best path analysis with a modified Dijkstra graph

Storing all possible conformations of a molecule inevitably leads to an overflow. Hence, *QVADIS* uses a modified Dijkstra algorithm considering only those conformations that participate immediately in a proton transfer step. For these conformations, steric energies are calculated with residues in the surrounding region as well as transfer rates between two states. A conformation is represented as a vector with a length that is determined by the number of rotatable axes. Its elements define by which angle the rotation axes of the involved residues have to be rotated. Attributes of a conformation are its steric energy, the energy difference to the static reference structure, the calculated rates, the distance of the donor- and acceptor atoms as well as a sorting criteria for this conformation. In the following, we will illustrate our implementation of Dijkstra's algorithm. Details about the algorithm can be found in the appendix (A). In the beginning we need some prearrangements such as the initialization of sorted stacks where the created conformations can be sorted due to different criteria. These different criteria have become necessary because we can generate conformations by rotating in both senses of rotation. As we want to avoid calculating energies and rates of conformations that have already been visited before, we introduced search criteria by geometry and transition rate for conformations still to be processed and a combination of both for visited conformations. By doing so we ensure to visit geometrically identical conformations repeatedly only if they have a better transition rate, i.e. they can be reached from the starting structure more easily than before. A list of junctions between clusters, i.e. sites where a fragmented path may be completed, is processed. For each of these boundaries, we apply the algorithm sketched below. Starting from an initial conformation (i.e. the static structure) new conformations are created by successively changing each rotatable axis by one rotation step and then calculating the corresponding transition rates. According to Dijkstra's algorithm the best conformation w.r.t. transition rate is used to create new conformations. As in [TH03b] we allow only transitions between neighbored nodes i.e. conformations differing in none but in one rotation step. Conformations that result in new connections between donor and acceptor atoms by conformational rearrangement and have an appropriate donor-acceptor distance (0.245 - 0.35 nm) according to the Q-HOP criteria for the transfer rates are sorted into a stack due to a rate combining transition and transfer rate. For each of these good conformations a list of ancestors may be set up i.e. the progression on conformational changes from the starting structure to the resulting end structure.

2.2.4 Implementation of the Q-HOP MD method

The computation of proton transfer probabilities from the *Q-HOP* method is triggered if a distance criterion is fulfilled. Based on the distance between donor and acceptor atom, the corresponding atom type (O or N) as well as the residues involved, we can calculate the transfer probabilities using tabulated parameters. The parameter set introduced by Herzog et al. [HFHL06] is provided in a hashtable where the search key is composed of the string representation of the corresponding donor and acceptor residues.

3 Results

The correct functioning of QVADIS was tested on the same proton transfer system, cytochrome P450cam (PDB code 1DZ8), that was investigated by Taraphder and Hummer when they first presented their algorithm [TH03b]. Cytochrome P450 proteins are heme-containing monooxygenases that catalyze the stereospecific hydroxylation of aliphatic C-H bonds. During the catalytic cycle, two protons need to be transferred from the surface of the protein to its buried active site where they are used to form one molecule of water from dioxygen, while the other oxygen atom is inserted into the C-H bond. Test calculations showed that the steric energies computed by eq. (1) only differed by at most 10^{-2} kJ/mol when using a 0.5 nm cut-off radius and by at most 10^{-4} kJ/mol with a 1 nm cut-off. To increase computational efficiency, this variable may therefore be chosen rather small. Here, we used 1 nm. The angular step determines the size of the search space. We observed that a rather coarse spacing of 60 degrees was sufficient to compute the connectivity matrix. However, a much finer step size should be used during pathway analysis (which follows at a later stage) in order not to miss any favorable geometries, and also not to miss potential energy barriers that would turn this pathway kinetically infeasible. It turned out that the calculations could exactly reproduce the connectivity matrix found by TH. When neglecting crystallographic water molecules, static clustering yielded 285 possible donor and acceptor atoms belonging to 161 titratable amino acids and the dioxygen ligand. These atoms form 108 clusters of differing size. The dioxygen $O2^{418}$ and Thr^{252} are located in one cluster, see fig. 2. The neighboring cluster involving Asp^{251} is connected to the protein surface. Therefore, favorable connections at the cluster boundaries may either be found via direct connection of Thr^{252} and Asp^{251} or involving nearby water molecules 901 and 902.

3.1 Pathway analysis

Our model assumes that the proton may move freely among all nodes within each static cluster by structural diffusion along the hydrogen bonded network. Thereby, the search for proton transfer pathways reduces to the cluster boundaries where the cluster may be extended by side chains adopting different rotameric states. When internal water molecules are omitted at this stage, the starting cluster around $O2^{418}$ can only be enlarged by connect-

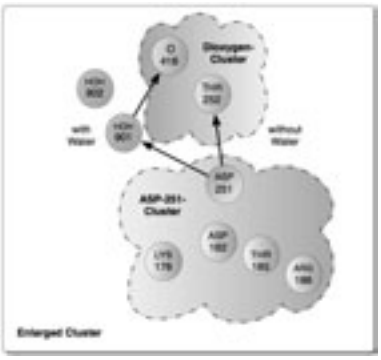

Figure 2: The two shaded clusters around the dioxygen molecule and around Asp^{251} need to be connected to form a continuous proton transfer pathway to the protein surface.

ing Asp^{251} and Thr^{252}. Fig. 3 (left) shows a pathway optimization involving rotations around $\chi 1$ and $\chi 2$ of Asp^{251} and around $\chi 3$ of Thr^{252} involving stepsizes of 10 degrees. The optimization finally leads to conformations with still relatively low steric energies (second-lowest panel) but short enough donor-acceptor distance (second-highest panel) so that *Q-HOP* computes high proton transfer probabilities (upper panel). As can be seen in the lowest panel, new conformations are generated by varying only one rotation angle at a time. The conformations are sorted according to the best energies due to eqs. (3) and (4). Tracing backwards from the final conformation generates a complete rotation pathway to the starting conformation (0,0,0) which reflects the side chain conformations of the crystal structure. As expected, the transition rate (eq. (4)) decreases with increasing steric energy relative to the starting conformation. The proton transfer rate only "kicks in" at relatively short distances below 0.3 nm. In the starting conformation, donor and acceptor atoms are at about 0.7 nm distance. After about 25 steps, a conformation (280,90,270) is reached with a very short donor-acceptor distance of 0.26 nm. The rotamer energy has increased to about 0.07 k_BT and is mostly driven by the rotation of $\chi 2$. In this conformation, both transition rate and transfer rate give high scores. As both rates reflect probabilities, they may be combined to yield a combined probability P_{comb} for conformations of low steric energy and high proton transfer efficiency. If internal water molecules are included for enlarging clusters, alternative pathways are found involving water 901. Fig. 3 (right) shows the best pathway connecting Asp^{251} and HOH^{901}. The optimization reaches a conformation with short donor-acceptor distance of 0.254 nm and a favorable steric energy of 0.03 k_BT.

4 Discussion

The proton transfer pathways found in this work are based on several assumptions of our current model. First of all, we assume that the proton can freely partition within each static cluster due to the local pK_a values of the amino acids. This leads to an enormous

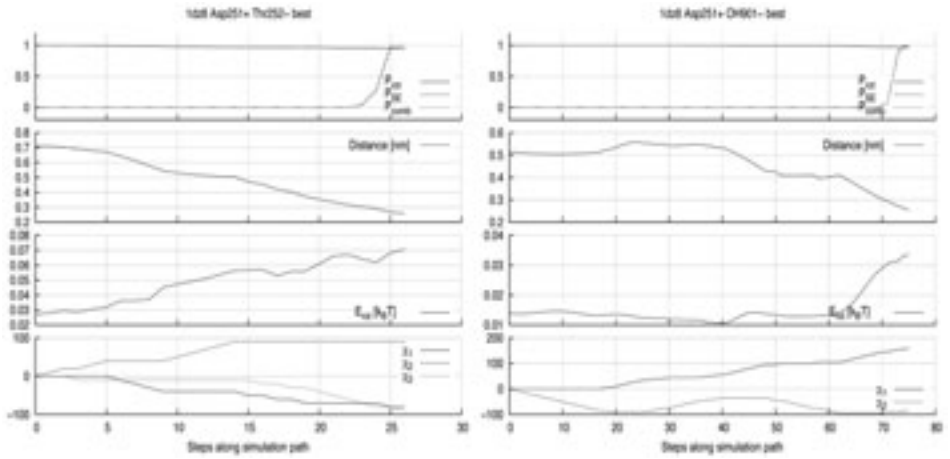

Figure 3: (left) Best connection between Asp^{251} and Thr^{252} according to P_{comb} (upper panel, dotted line): χ1=280, χ2=90, χ3=270. The stepsize is 10 degrees. (right) Best connection between Asp^{251} (donor) and HOH^{901} (acceptor) using 5 degree stepsize.

simplification of the search problem as the only degrees of freedom to be searched are the side chain dihedral angles of those residues at the cluster boundaries that were previously found in a coarse scan as putative candidates to enlarge the cluster. In this step, the protein backbone is kept fixed and only side chain rotations of the two contacting residues are exhaustively sampled. This latter approximation appears quite reasonable. The mobility of the buried water molecules was modeled implicitly by reducing their distance from potential donor and acceptor atoms by a given distance value. Occasionally, quite large energy barriers were encountered due to collisions of the rotated side chains with neighboring groups. In future work, we will test the effect of allowing for simultaneous relaxation of those colliding side chains with the central side chain. Also, we plan to combine this approach with electrostatic continuum calculations of local pK_a values to characterize the relative population of titratable residues in the local clusters. Concerning the accuracy of our results, the same static clusters and the same connectivity matrix were obtained as in [TH03b]. Dynamic enlargement of clusters starting from the distal oxygen to the protein surface gave several models that appear comparable to those of TH. When omitting internal water molecules, a pathway is found involving Asp^{251} and Thr^{252}. This is less favorable than the one involving water 901. Direct protonation from $O2^{418}$ is more favorable than via Thr^{252} suggesting that Thr^{252} may not directly participate in the PT pathway but rather acts as stabilizing partner [DMSS05]. Our energy values cannot be easily compared with those in [TH03b] mostly because TH rotate side chains one after the other whereas *QVADIS* rotates contacting side chains simultaneously. This may explain why we found an energy maximum of 0.07 k_BT for the transition between Asp^{251} and Thr^{252} whereas [TH03b] reported a value of 7 k_BT. When estimating the PT energetics, a future version of *QVADIS* will also include the electrostatic interaction with the environment.

In summary, we have introduced a novel software package *QVADIS* to discover proton transfer pathways in biomolecules and to score their transfer kinetics by combining a variant of a previously published algorithm by TH and of our *Q-HOP* methodology. As proton transfer reactions are crucial in many areas of biology and material sciences, this tool may assist researchers in obtaining a deeper understanding of the transfer processes and, eventually, to design macromolecules with desired transfer characteristics.

References

[BWF+00] 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. Ac. Res.*, 28:235–242, 2000.

[CWWS03] T. Can, Y. Wang, Y. Wang, and J. Su. FPV: Fast Protein Visualisation Using Java 3D. *Bioinformatics*, 19(8):913–922, 2003.

[DK94] R.L. Dunbrack and M. Karplus. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *Nature Struct. Biol.*, 1:334–340, 1994.

[DMSS05] I.G. Denisov, T.M. Makris, S.G. Sligar, and I. Schlichting. Structure and Chemistry of Cytochrome P450. *Chem. Rev*, 105:2253–2277, 2005.

[GFSH07] W. Gu, T. Frigato, T.P. Straatsma, and V. Helms. Dynamic Protonation Equilibrium of Solvated Acetic Acid. *Angewandte Chemie Int. Ed.*, 46(16):2939–2943, 2007.

[HFHL06] E. Herzog, T. Frigato, V. Helms, and C.R.D. Lancaster. Energy Barriers of Proton Transfer Reactions Between Amino Acid Side Chain Analogs and Water from ab initio Calculations. *J. Comput. Chem.*, 27:1534–1547, 2006.

[LH01a] M.A. Lill and V. Helms. Molecular dynamics simulation of proton transport with quantum mechanically derived proton hopping rates (Q-HOP MD). *J. Chem. Phys.*, 115:7993–8005, 2001.

[LH01b] M.A. Lill and V. Helms. Reaction rates for proton transfer over small barriers and connection to transition state theory. *J. Chem. Phys.*, 115:7985–7992, 2001.

[Mar06] D. Marx. Proton Transfer 200 Years after Grotthus: Insights from Ab Initio Simulations. *ChemPhysChem*, 7(9):1848–1870, 2006.

[SW04] C.N. Schutz and A. Warshel. Analyzing Free Energy Relationships for Proton Translocations in Enzymes: Carbonic Anhydrase Revisited. *J. Phys. Chem. B*, 108:2066–2075, 2004.

[TH03a] S. Taraphder and G. Hummer. Dynamic Proton Transfer Pathways in Proteins: Role of Sidechain Conformational Fluctuations. *Physica A*, 318:293–301, 2003.

[TH03b] S. Taraphder and G. Hummer. Protein Side-Chain Motion and Hydration in Proton-Transfer Pathways. Results for Cytochrome P450cam. *J. Am. Chem. Soc.*, 125:3931–3940, 2003.

A Pseudocodes to sketch the implementation of our best path analysis

Algorithm A.1: RUNDIJKSTRA()

```
for Residuepair (M1, M2) at cluster boundary
do { if both not rotatable
     then findNewConnections(M1, M2 , Startconformation )
     else while ToDoStack contains conformations
          do { K = fetch first element from toDoStack
               for each rotatable axis
                 do core algorithm
```

Algorithm A.2: CORE ALGORITHM()

```
Ecomp = K.getEnergy()
TransitionRate = K.getTransitionRate()
if this axis' angle + rotations step ≤ 360
then { Create new conformation newconf from K
       newconf.setAncestor(K)
       newconf.changeRotation(rotations step, axis)
       Econf = applyConformation(newconf, (M1, M2))
       newconf.setEnergy(Econf)
       deltaEI = Ecomp - Estatic
       deltaEJ = Econf - Estatic
       TransitionRate *= calculateTransProb(deltaEJ - deltaEI)
       newconf.setTransitionRate(TransitionRate)
       if newconf.isGoodConformation()
       then { combinedRate = TransitionRate * newconf.getPSE()
              newconf.setCombinedRate(combinedRate)
              if not Goodones.contains(newconf)
                then Goodones.add(newconf);
       process new conformation
Repeat for reverse sense of rotation
```

Algorithm A.3: APPLYCONFORMATION($Conformation, MGroupVector$)

```
for all MGroups
do { for each rotatable axis of MGroup
       do {apply angle given by Conformation on rotation axis
     ConformationEnergy += Energy of MGroup
     MGroup1 = Donor, MGroup2 = Acceptor
     findNewConnections(Donor, Acceptor, Conformation)
return (ConformationEnergy)
```

Algorithm A.4: PROCESS NEW CONFORMATION()

if $TransitionRate > 0.0$
then
- boolean searchdstack = false
- **if** (newconf not visited)
  - **then** searchdstack = true
  - **else**
    - **if** TransitionRate(visited conformation) $<$ newconf.getTransitionRate()
      - **then** searchdstack = true
      - **else** $Visited.add(newconf)$
- **if** $searchdstack$
  - **then**
    - **if not** geometrically identical conformation available
      - **then** DStackG.add(newconf), DStackT.add(newconf)
      - **else**
        - **if** TransitionRate(geometrically identical conformation) $<$ newconf.getTransitionRate()
          - **then** remove old conformation and substitute by newconf
          - **else** $Visited.add(newconf)$;

else $Visited.add(newconf)$

Algorithm A.5: FINDNEWCONNECTIONS($Donor, Acceptor, Conformation$)

mindist = maxvalue
newconnected = false
for all Acceptor atoms
do
- **for** all Donor atoms
- **do**
  - newdistance = acceptoratom.getDistance(donoratom)
  - **if** $newdistance < mindist$
  - **then**
    - mindist = newdistance
    - acceptortype = acceptoratom.getAtomType()
    - donortype = donoratom.getAtomType()
    - Conformation.setDistance(mindist)

if ($mindist > 2.44$ **and** $mindist < 3.5$)
then
- Conformation.isGoodconformation = true
- QHOPDFP.setDonorTyp(donortype)
- QHOPDFP.setAcceptorTyp(acceptortype)
- pSE =QHOPDFP.calulatePSE(donoratom, acceptoratom, mindist)
- Conformation.setPSE(pSE)