An Extended Dead-End Elimination Algorithm to Determine Gap-Free Lists of Low Energy States

EDDA KLOPPMANN, G. MATTHIAS ULLMANN, TORSTEN BECKER
Structural Biology/Bioinformatics, University of Bayreuth, Universitätsstr. 30, BGI, 95447 Bayreuth, Germany

Received 3 December 2006; Revised 2 February 2007; Accepted 13 March 2007
DOI 10.1002/jcc.20749
Published online 30 April 2007 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Proteins are flexible systems and commonly populate several functionally important states. To understand protein function, these states and their energies have to be identified. We introduce an algorithm that allows the determination of a gap-free list of the low energy states. This algorithm is based on the dead-end elimination (DEE) theorem and is termed X-DEE (extended DEE). X-DEE is applicable to discrete systems whose state energy can be formulated as pairwise interaction between sites and their intrinsic energies. In this article, the computational performance of X-DEE is analyzed and discussed. X-DEE is implemented to determine the lowest energy protonation states of proteins, a problem to which DEE has not been applied so far. We use X-DEE to calculate a list of low energy protonation states for two bacteriorhodopsin structures that represent the first proton transfer step of the bacteriorhodopsin photocycle.

© 2007 Wiley Periodicals, Inc. J Comput Chem 28: 2325–2335, 2007

Key words: protein design; homology modelling; global energy minimum; bacteriorhodopsin; protonation state; pH titration; X-DEE; dead-end elimination

Introduction

Proteins are the most versatile and complex among the macromolecules of a living organism. They are involved in molecular recognition, signal transduction, transport of substances, and in the catalysis of most metabolic processes. The understanding of protein function on the atomic level is thus of fundamental biological and medical interest. Ultimately, protein function depends on its three-dimensional structure. This structure depends on the physical properties of its constituents and is influenced by environmental factors such as temperature and pH. In accordance with the laws of thermodynamics, the state of lowest free energy is the most probable and thus of primary interest in structural research. Proteins, however, are complex systems with many degrees of freedom and the direct calculation of the lowest energy state is rarely possible.

In computational structure prediction, commonly the conformational space of proteins is discretized. In principle, this strategy allows an exhaustive search of the combinatorial space for the state of lowest energy. However, the size of the combinatorial space in general overstrains todays computer resources. Algorithms that deterministically find the state of lowest energy while circumventing an exhaustive search of the complete combinatorial space are therefore of great interest. In 1992, Desmet et al.[1] introduced such an algorithm termed dead-end elimination (DEE) for discrete systems that can be described in terms of pairwise interactions. Originally, DEE was applied to predict protein side chain positions in homology modelling. In this context, the protein can be seen as a set of residues, each of which may adopt a finite number of possible rotamers. More generally, one can describe such a system as a set of *sites* each having a finite number of possible *forms*. The DEE algorithm identifies high energy forms of sites that are incompatible with the state of lowest energy. These high energy forms are termed *dead-ends* and can be eliminated from further consideration. Thereby DEE reduces the combinatorial space to a point where a subsequent exhaustive search becomes feasible. DEE has been successfully applied in protein structure prediction,[1–7] protein design,[8–10] sequence alignment,[11] and also in the evaluation of protein solvent exposure.[12]

Proteins, however, are flexible systems that may adopt several functionally important states. To understand their mechanisms it is necessary to obtain a complete picture of the states that are accessible to the protein. Consequently, algorithms that are able to produce gap-free lists of low energy states, i.e., lists of states that are complete up to a given energy distance from the global energy minimum, are of general interest in structural biology. While Monte Carlo techniques allow to sample low energy states and generally provide an accurate description of thermal properties,[13] they do not allow to obtain gap-free lists of low-energy states for a given energy range. This is a major drawback if one investigates the kinetics of a working enzyme. States, not even populated to a measurable degree, may well

Correspondence to: T. Becker; e-mail: torsten.becker@uni-bayreuth.de

Contract/grant sponsor: Deutsche Forschungsgemeinschaft; contract/grant number: FOR 490 UL 174/4-3, UL 174/7-1

be functionally relevant.[14,15] Thus, an incomplete sample of states available to the system may lead to wrong conclusions about the mechanism. The DEE algorithm can be modified to exclude only those forms that are incompatible with states in a given energy distance from the global minimum.[3] The reduced combinatorial space is then searched by an A^* algorithm for states lying within the given energy range. This reduced combinatorial space, however, may still be very large and the search algorithm, though efficient, threatens to get stuck in near exhaustive searches. Here, we present an alternative algorithm, that uses the full power of DEE to reduce the combinatorial space. Subsequent DEE scans using so-called search keys can be used to unambiguously identify states according to their energetic order. Furthermore, this new algorithm allows to complete any given list of low energy states by identifying those low energy states that are missing in the list. The algorithm is termed X-DEE (extended DEE).

We implemented X-DEE to determine protonation states of a protein. Protonation state changes are crucial for many enzymatic reactions and other biochemical processes such as signal transduction, photosynthesis and proton transfer across membranes.[16–21] Even without explicit protonation state changes, the charge distribution of a protein is essential for its function. Thus, determining the energetically accessible protonation states of a protein is a crucial step towards understanding its function. The protonation state energy is routinely calculated from intrinsic energies of the sites and pairwise interaction between the protonatable residues.[13,22,23] The energy contribution of each protonatable residue depends on its protonation form. In proteins, not only the state of lowest energy but also the next higher protonation states are commonly significantly populated and often play a functional role. By generating a gap-free list of the low energy states, X-DEE provides a complete picture of functionally relevant protonation states.

We applied X-DEE to the light-driven proton pump bacteriorhodopsin (BR), an example of a protein where protonation state changes play a functionally important role. BR has been investigated extensively and its structure as well as its proton pumping mechanism is known in detail.[24–28] Figure 1 shows the structure of BR, the key residues involved in the proton transfer and the transfer events occurring after light absorption. The bR protonation state, the ground state, is characterized by a protonated retinal Schiff base and a deprotonated Asp85. Asp96, Asp115, and Arg82 were experimentally shown to be protonated, while Asp212 was shown to be unprotonated.[29,30] Two glutamate residues, Glu194 and Glu204, are part of the proton release complex which is protonated in the bR state.[31] After the first proton transfer step, BR is in the M1 protonation state characterized by a deprotonated retinal Schiff base and a protonated Asp85. No other key residue changes its protonation form. The next photocycle intermediate state is the M2 protonation state, where a proton has been released to the extracellular side. The M2 protonation state is characterized by an unprotonated retinal Schiff base, a protonated Asp85 and an unprotonated proton release complex. Subsequently, the retinal Schiff base is reprotonated from Asp96 close to the cytoplasmic site, which thereafter takes up a proton from the cytoplasm. Last, Asp85 deprotonates and the proton release complex is reprotonated.

In the present article, we describe the X-DEE algorithm that generates gap-free lists of low energy states. We implemented X-DEE to determine protonation states of proteins, a problem to which DEE has not been applied so far. We analyze the properties of the X-DEE algorithm and its computational performance. X-DEE is then used to calculate all low energy protonation states for two BR structures representing the bR and M1 photocycle intermediates. The energy distribution of the calculated low energy states is analyzed and implications for the proton transfer in BR are discussed.

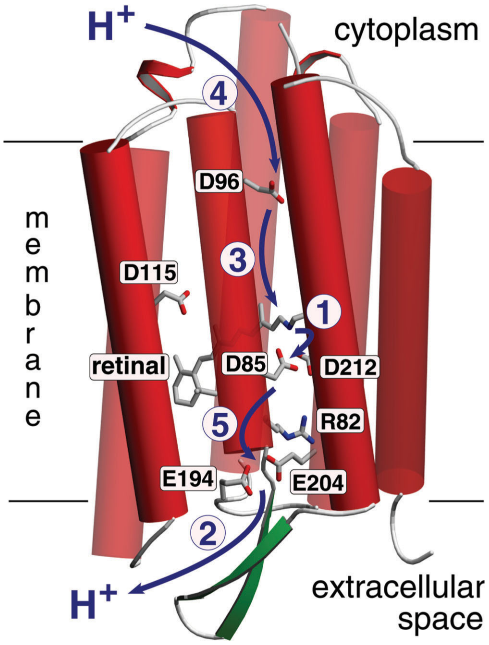

Figure 1. The structure of bacteriorhodopsin. BR is a seven-helix transmembrane protein that covalently binds its chromophore retinal to a lysinyl residue. Light absorption triggers BR's pumping cycle, during which a proton is transferred from the cytoplasm to the extracellular space. The key residues of proton transfer are shown explicitly and the five sequential transfer steps are indicated by the numbered arrows.

Theory

System

The DEE algorithm solves the combinatorial problem of identifying the global energy minimum for discrete systems whose state energy can be expressed in terms of intrinsic energies of individual *sites* and pairwise interactions between these sites. The sites can adopt discrete *forms* that differ in their contribution to the state energy. Each state of the system can be described by a vector $\vec{x}$ that specifies the form x of each site μ in this state: $\vec{x} = (x_1, \ldots, x_\mu, \ldots, x_N)$, where N is the number of sites. A general formulation of the state energy is given by:

$$E = \sum_{\mu}^{N} E_{\text{intr}}(x_\mu) + \frac{1}{2}\sum_{\mu}^{N}\sum_{\nu}^{N} I(x_\mu, x_\nu) \tag{1}$$

where $E_{\text{intr}}(x_\mu)$ is the intrinsic energy of site μ in form x and $I(x_\mu, x_\nu)$ is the interaction energy between sites μ and ν in their respective forms. Assuming an equal number of forms P for each site, the number of possible states M equals P^N, rendering an exhaustive search impossible even for rather small systems.

Dead-End Elimination

DEE circumvents an exhaustive search of the full combinatorial space by identifying forms of sites that cannot contribute to the state of lowest energy, so-called *dead-ends*, without explicitly calculating state energies.[1] These dead-ends can be excluded from further consideration, thereby significantly reducing the size of the combinatorial space.

The original DEE criterion of Desmet et al.[1] compares the energy of two forms of a site μ, d_μ and c_μ. Form d of site μ is defined as dead-end, if the state of lowest energy with d_μ is higher in energy than the state of highest energy that contains c_μ. In other words, d_μ is a dead-end, if all states with d_μ are higher in energy than all states that contain c_μ. Later, Goldstein published a more powerful criterion.[2] The Goldstein criterion identifies d_μ as dead-end, if the state energy can always be lowered by changing the form of site μ from d to c. That means, d_μ is a dead-end, if all states that contain d_μ are higher in energy than the corresponding states that contain c_μ. A graphic representation of the two schemes is shown in Figure 2.

According to the Goldstein DEE criterion, d_μ can be identified as dead-end if:

$$E_{\text{intr}}(d_\mu) - E_{\text{intr}}(c_\mu) + \sum_{\nu \neq \mu} \min_x [I(d_\mu, x_\nu) - I(c_\mu, x_\nu)] > 0 \quad (2)$$

where x_ν denote the form x of all sites $\nu \neq \mu$. The first two terms represent the difference in intrinsic energies E_{intr} between d_μ and c_μ. The third term accounts for the difference in interaction energy between the two forms, d and c of site μ and the forms x of all other sites ν. For each site ν, the minimum interaction energy is determined by searching over all possible forms x. Thus, eq. (2) yields a lower bound of the change in energy upon changing the form of site μ from d to c. If this lower bound is larger than zero, the state energy can always be lowered by changing d_μ to c_μ. Accordingly, d_μ is a dead-end and can be eliminated. The DEE criterion is successively applied to all forms of all sites. The computational cost to evaluate the DEE criterion is of the order N^2P^3, where N denotes the number of sites and P the number of forms per site.[4] This scaling behavior poses a huge advantage compared to the $O(P^N)$ scaling behavior of an exhaustive search of the combinatorial space. Commonly, the state of lowest energy is not directly determined by DEE, rather the size of the combinatorial space is reduced to a point, where a subsequent exhaustive search becomes feasible.

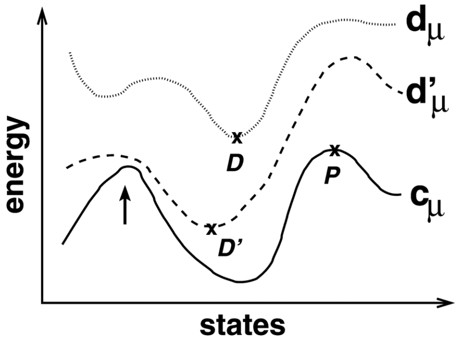

Figure 2. The DEE criterion. Desmet et al.[1] defined a DEE criterion that identifies d_μ as dead-end, if the state of lowest energy that contains d_μ (D) has a higher energy than the state of highest energy that contains c_μ (P). The Goldstein DEE criterion identifies those d_μ as dead-ends, for which the state energy can always be lowered by changing the form of site μ from d to c.[2] Thus, the Goldstein criterion additionally identifies d'_μ as dead-end, although the original Desmet criterion is not fulfilled (D' is lower in energy than P). Instead of calculating the curves, the DEE criterion of Goldstein considers the minimum energy difference between the states that contain d_μ and c_μ, indicated by the arrow.

The described DEE algorithm has been successful in the determination of the lowest energy state for various biophysical systems.[1–12]

However, for many systems a set of low energy states is functionally relevant. Based on DEE we developed a computational scheme termed X-DEE (extended DEE) that generates a gap-free list of low energy states.

X-DEE

The basic idea of the X-DEE algorithm is to exclude a list of states from the search space explored by DEE. If a gap-free list of k low energy states $\{\vec{x}_1, \cdots, \vec{x}_k\}$ is already known, the $(k+1)^{\text{th}}$ state can be found by restricting the search for the lowest energy state to the set of all states $\mathcal{M}$ excluding the set of already known states, i.e., by searching the state of lowest energy in $\mathcal{M} \setminus \{\vec{x}_1, \cdots, \vec{x}_k\}$. More generally, X-DEE can be used to restrict the search to a set $\mathcal{M} \setminus \mathcal{L}$ for any given list $\mathcal{L}$ of states. In case $\mathcal{L}$ is not gap-free, X-DEE will identify the state of lowest energy not included in $\mathcal{L}$ and can consequently complete $\mathcal{L}$, until a gap-free list of low energy states is obtained.

The DEE algorithm gains its computational efficiency from excluding whole sets of states from further consideration without explicitly calculating their energies. In particular, the identification of d_μ as dead-end allows to exclude all states that contain d_μ from further consideration, while none of the state energies have to be calculated explicitly. As a consequence, there is no straightforward way to exclude an arbitrary list of states $\mathcal{L}$ from the search space explored by DEE. However, a DEE search can be restricted to a specific type of subset of $\mathcal{M}$: keeping a number of sites fixed during a DEE search yields the state of lowest energy of a subset $\mathcal{S}$ of $\mathcal{M}$ characterized by the forms of those sites that were fixed. For example, applying DEE to the subset $\mathcal{S}$ of those states that have form h at site 1 will determine the state of lowest energy with form h at site 1. This subset $\mathcal{S}$ can be represented by a so-called *search key* $\vec{b}_\mathcal{S} = (h_1, *_2, \cdots, *_\mu, \cdots, *_N)$, where h is the specified form of site 1 and $*$ indicates that this site is undefined, i.e., will be determined during the DEE search. The idea is thus, to define search keys $\vec{b}_\mathcal{S} = (b_1, \cdots, b_\mu, \cdots, b_N)$ such that the subsets $\mathcal{S}$ represented by the individual search keys together represent $\mathcal{M} \setminus \mathcal{L}$. For each site μ of the system, these search keys have a component b_μ which is either fixed to a specific form or undefined. Determining the state of lowest energy of all subsets via the DEE algorithm then yields the desired state of lowest energy of $\mathcal{M} \setminus \mathcal{L}$. The main challenge of

X-DEE lies in creating a search basis $\mathcal{B}$, i.e., a set of search keys $\vec{b}_{\mathcal{S}}$, such that $\mathcal{L}$ is excluded from the search and the complete set $\mathcal{M} \setminus \mathcal{L}$ is searched. In the following section, a procedure to generate $\mathcal{B}$ is presented that fulfills both conditions.

Constructing a Search Basis $\mathcal{B}$ for $\mathcal{M} \setminus \mathcal{L}$

A schematic representation of the procedure CreateSearchBasis is shown in Figure 3. Given the list of states $\mathcal{L}$ to be excluded from the search, a search basis $\mathcal{B}$ is constructed as follows: Input of the recursive procedure CreateSearchBasis is a list of states $\mathcal{L}'$ and a list vector $\vec{t}'$ that is associated to $\mathcal{L}'$. At first, $\mathcal{L}'$ is equal to the complete list $\mathcal{L}$. The list vector $\vec{t}'$ contains an element for each site μ: $\vec{t}' = (t_1, \ldots, t_\mu, \ldots, t_{\mathrm{N}})$. This list vector keeps track of the sites that are already fixed to specific forms. At first, no site in $\vec{t}'$ is fixed, i.e., all sites are undefined, $\vec{t}' = (*_1, \cdots, *_\mu, \cdots, *_{\mathrm{N}})$. The algorithm proceeds in five steps:

1. It is checked whether the associated list vector $\vec{t}'$ contains at least one undefined site. Otherwise, the procedure returns.
2. Among the undefined sites, the algorithm searches for a site μ with forms that do not occur in any of the state vectors in $\mathcal{L}'$, i.e., a site μ with unused forms h. Once such a site μ is found, the algorithm proceeds to step 3. If no such site exists, the first undefined site in $\vec{t}\,'$ is chosen and the algorithm proceeds directly to step 4.
3. For each unused form h of site μ, a search key $\vec{b}$ is defined by copying the list vector $\vec{t}'$ to $\vec{b}$ and fixing site μ to form h in $\vec{b}$; $b_\mu = h$. Each search key $\vec{b}$ thus differs from the current list vector only at site μ. Fixing site μ to forms h not occurring in $\mathcal{L}'$, guarantees that the subset represented by $\vec{b}$ and $\mathcal{L}$ are disjoint, i.e., $\vec{b}$ represents a subset of $\mathcal{M} \setminus \mathcal{L}$. The search keys $\vec{b}$ are added to the search basis $\mathcal{B}$.
4. The vectors in $\mathcal{L}'$ are divided into sublists $\mathcal{L}'_{\mathrm{sub}}$ such that site μ has form g in all state vectors $\vec{x}$ in $\mathcal{L}'_{\mathrm{sub}}$, i.e., $x_\mu = g$ for all states in $\mathcal{L}'_{\mathrm{sub}}$. To each sublist $\mathcal{L}'_{\mathrm{sub}}$, a separate list vector $\vec{t}'_{\mathrm{sub}}$ is assigned by copying list vector $\vec{t}'$ to $\vec{t}'_{\mathrm{sub}}$ and fixing site μ to the form g common to all state vectors in $\mathcal{L}'_{\mathrm{sub}}$; $t_\mu = g$.
5. For each sublist $\mathcal{L}'_{\mathrm{sub}}$ and its list vector $\vec{t}'_{\mathrm{sub}}$, the steps 1–5 are executed again.

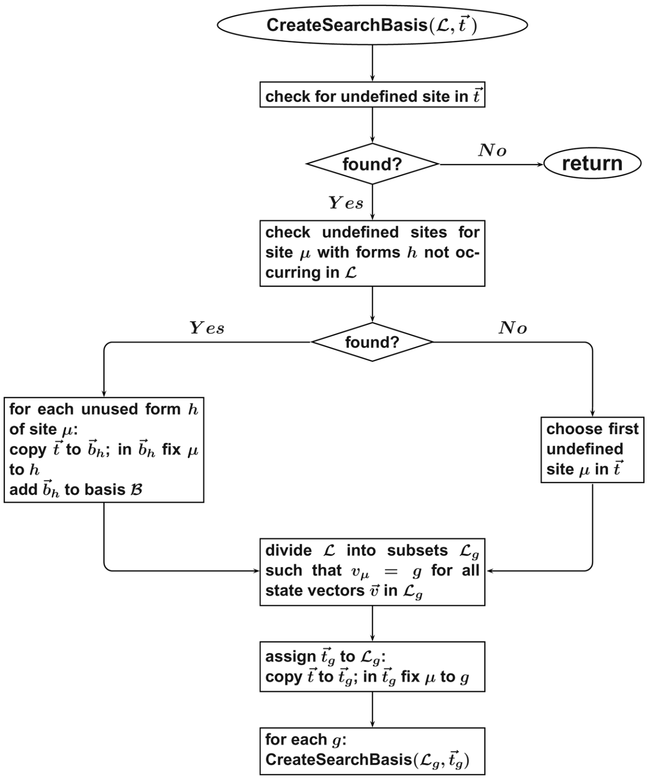
CreateSearchBasis($\mathcal{L}, \vec{t}$)
check for undefined site in $\vec{t}$
found?
No
return
Yes
check undefined sites for site μ with forms h not occurring in $\mathcal{L}$
Yes
found?
No
for each unused form h of site μ:
copy $\vec{t}$ to $\vec{b}_h$; in $\vec{b}_h$ fix μ to h
add $\vec{b}_h$ to basis $\mathcal{B}$
choose first undefined site μ in $\vec{t}$
divide $\mathcal{L}$ into subsets $\mathcal{L}_g$ such that $v_\mu = g$ for all state vectors $\vec{v}$ in $\mathcal{L}_g$
assign $\vec{t}_g$ to $\mathcal{L}_g$:
copy $\vec{t}$ to $\vec{t}_g$; in $\vec{t}_g$ fix μ to g
for each g:
CreateSearchBasis($\mathcal{L}_g, \vec{t}_g$)

Figure 3. Flowchart of the recursive procedure CreateSearchBasis. Input of CreateSearchBasis is a list of states $\mathcal{L}$ and an associated list vector $\vec{t}$. With each recursion, $\mathcal{L}$ is divided into sublists $\mathcal{L}_{\mathrm{sub}}$ and one additional site is fixed in the associated list vectors $\vec{t}_{\mathrm{sub}}$. CreateSearch-Basis terminates when all sites of a list vector $\vec{t}_{\mathrm{sub}}$ are fixed. With each recursion, search keys $\vec{b}_{\mathcal{S}}$ can be generated that differ from the list vector $\vec{t}_{\mathrm{sub}}$ in the form of site μ. The search keys are added to the search basis $\mathcal{B}$. In total, CreateSearchBasis generates a set of search keys $\vec{b}_{\mathcal{S}}$ characterizing subsets $\mathcal{S}$ whose union represent $\mathcal{M} \setminus \mathcal{L}$.

The recursive procedure CreateSearchBasis thus starts with a list $\mathcal{L}'$ and an associated list vector $\vec{t}'$ and ends up with a set of sublists $\mathcal{L}'_{\mathrm{sub}}$ with associated list vectors $\vec{t}'_{\mathrm{sub}}$. CreateSearchBasis terminates for a given sublist, if all sites are fixed in the associated list vector. If step 3 is executed, new search keys are added to the search basis $\mathcal{B}$. An example of creating a search basis is given in Figure 4.

All search keys in $\mathcal{B}$ are subjected to a DEE search yielding the states of lowest energies of the represented subsets $\mathcal{S}$. These states include the state of lowest energy of $\mathcal{M} \setminus \mathcal{L}$.

Completeness of the Search Basis $\mathcal{B}$

For the search basis $\mathcal{B}$, it remains to be shown that (i) all subsets of states $\mathcal{S}$ represented by the search keys are subsets of $\mathcal{M} \setminus \mathcal{L}$ and that (ii) the union of all subsets $\mathcal{S}$ represent the complete set $\mathcal{M} \setminus \mathcal{L}$.

i. Each search key is derived from a list vector $\vec{t}'$ which is associated with a list $\mathcal{L}'$. By construction, a search key derived from $\vec{t}'$ at most searches the subset $\mathcal{L}'$ of $\mathcal{L}$. However, each search key differs at a given site μ from all state vectors in $\mathcal{L}'$ and thus, the set $\mathcal{S}$ represented by this search key and $\mathcal{L}'$ are disjoint. It follows that each search key represents a subset $\mathcal{S}$ of $\mathcal{M} \setminus \mathcal{L}$.

ii. To prove that the complete set $\mathcal{M} \setminus \mathcal{L}$ is represented by the search basis $\mathcal{B}$, it is sufficient to show that for any state $\vec{x}$ in $\mathcal{M} \setminus \mathcal{L}$, there is a subset $\mathcal{S}$ of $\mathcal{M} \setminus \mathcal{L}$ represented by a search key $\vec{b}_{\mathrm{S}}$ such that $\vec{x} \in \mathcal{S}$. After the first recursion of CreateSearchBasis, there are two possibilities: either a search key is generated with only one fixed site, which adopts the same form as in $\vec{x}$, or a sublist $\mathcal{L}_{\mathrm{sub}}$ and an associated list vector $\vec{t}_{\mathrm{sub}}$ exist such that $\vec{t}_{\mathrm{sub}}$ has its fixed site in common with $\vec{x}$. In the first case, the search key represents the subset $\mathcal{S}$ which includes $\vec{x}$. Hence, $\vec{x}$ is represented by $\mathcal{B}$. In the second case, the algorithm proceeds further. (This second case can occur for two reasons: either a site μ was found with unused forms but $\vec{x}$ has a used form at site μ, or no site with an unused form was found and an undefined site was chosen.) After the next recursion, either an adequate search key is constructed or a sublist exists with an associated list vector having its two

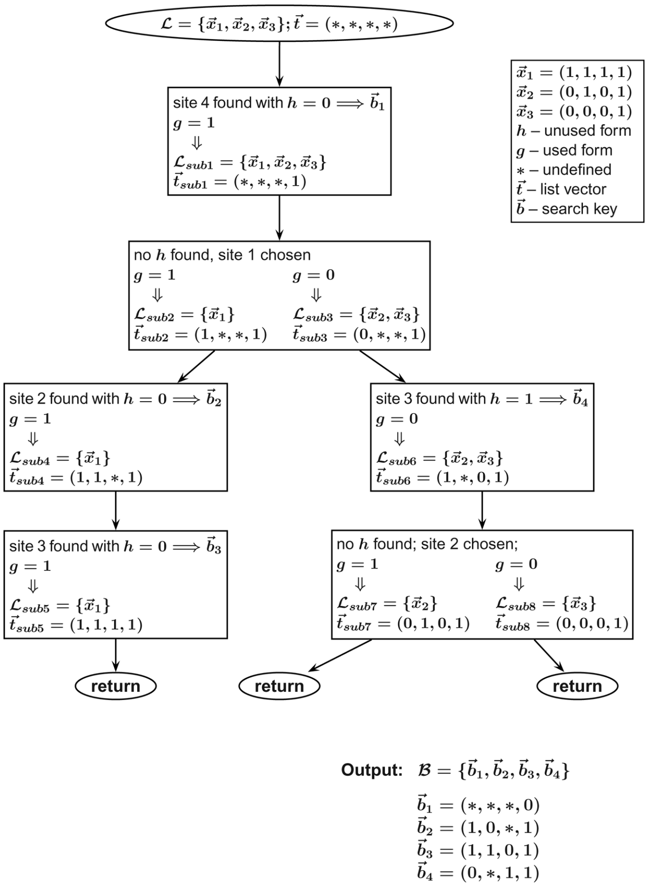
$\mathcal{L} = \{\vec{x}_1, \vec{x}_2, \vec{x}_3\}; \vec{t} = (*, *, *, *)$
$\vec{x}_1 = (1,1,1,1)$
$\vec{x}_2 = (0,1,0,1)$
$\vec{x}_3 = (0,0,0,1)$
h – unused form
g – used form
$*$ – undefined
$\vec{t}$ – list vector
$\vec{b}$ – search key
site 4 found with $h = 0 \Longrightarrow \vec{b}_1$
$g = 1$
$\Downarrow$
$\mathcal{L}_{sub1} = \{\vec{x}_1, \vec{x}_2, \vec{x}_3\}$
$\vec{t}_{sub1} = (*, *, *, 1)$
no h found, site 1 chosen
$g = 1$
$\Downarrow$
$\mathcal{L}_{sub2} = \{\vec{x}_1\}$
$\vec{t}_{sub2} = (1, *, *, 1)$
$g = 0$
$\Downarrow$
$\mathcal{L}_{sub3} = \{\vec{x}_2, \vec{x}_3\}$
$\vec{t}_{sub3} = (0, *, *, 1)$
site 2 found with $h = 0 \Longrightarrow \vec{b}_2$
$g = 1$
$\Downarrow$
$\mathcal{L}_{sub4} = \{\vec{x}_1\}$
$\vec{t}_{sub4} = (1, 1, *, 1)$
site 3 found with $h = 1 \Longrightarrow \vec{b}_4$
$g = 0$
$\Downarrow$
$\mathcal{L}_{sub6} = \{\vec{x}_2, \vec{x}_3\}$
$\vec{t}_{sub6} = (1, *, 0, 1)$
site 3 found with $h = 0 \Longrightarrow \vec{b}_3$
$g = 1$
$\Downarrow$
$\mathcal{L}_{sub5} = \{\vec{x}_1\}$
$\vec{t}_{sub5} = (1, 1, 1, 1)$
no h found; site 2 chosen;
$g = 1$
$\Downarrow$
$\mathcal{L}_{sub7} = \{\vec{x}_2\}$
$\vec{t}_{sub7} = (0, 1, 0, 1)$
$g = 0$
$\Downarrow$
$\mathcal{L}_{sub8} = \{\vec{x}_3\}$
$\vec{t}_{sub8} = (0, 0, 0, 1)$
return
return
return
Output: $\mathcal{B} = \{\vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_4\}$
$\vec{b}_1 = (*, *, *, 0)$
$\vec{b}_2 = (1, 0, *, 1)$
$\vec{b}_3 = (1, 1, 0, 1)$
$\vec{b}_4 = (0, *, 1, 1)$

Figure 4. Example for the generation of a search basis $\mathcal{B}$. The list $\mathcal{L}$ contains three states: $\{\vec{x}_1, \vec{x}_2, \vec{x}_3\}$. The system has four sites that can adopt the forms 1 and 0. An unused form h is a form that does not appear in $\mathcal{L}$ for a given site μ. In the list vectors $\vec{t}$ and the search keys $\vec{b}$, sites can be either fixed to a form, 1 or 0, or undefined, $*$. The output is the search basis $\mathcal{B}$.

fixed sites in common with $\vec{x}$. Proceeding further, either a search key $\vec{b}_{\mathcal{S}}$ is generated such that $\vec{x} \in \mathcal{S}$, or finally a sublist $\mathcal{L}_{\text{sub}}$ of $\mathcal{L}$ exists with an associated list vector $\vec{t}_{\text{sub}}$ which has only one undefined site and is identical to $\vec{x}$ at all other $(N-1)$ sites. Since $\vec{x}$ is assumed to be in $\mathcal{M} \setminus \mathcal{L}$, it follows that $\vec{x}$ has to differ from all vectors in $\mathcal{L}_{\text{sub}}$ at the site not fixed in $\vec{t}_{\text{sub}}$. Consequently, the form that $\vec{x}$ adopts at this undefined site is unused and a search key $\vec{b}_{\mathcal{S}}$ is created that represents a subset $\mathcal{S}$ of $\mathcal{M} \setminus \mathcal{L}$ which contains $\vec{x}$. In this case, all sites are fixed in $\vec{b}_{\mathcal{S}}$ and the subset $\mathcal{S}$ contains only the state vector $\vec{x}$.

Thus by construction, all states in $\mathcal{L}$ are excluded from $\mathcal{B}$ and the union of the subsets $\mathcal{S}$, characterized by the search keys in $\mathcal{B}$, represent the complete set $\mathcal{M} \setminus \mathcal{L}$.

Application of X-DEE to the Determination of Protonation States

Proteins contain protonatable residues whose charged state depends on their interaction with the protein environment. These protonatable residues are treated as sites and each site can adopt two forms, protonated and unprotonated. The protonation state energy G_{n} of a protein with N protonatable sites depends on the protonation form of each protonatable site x_μ and its intrinsic pK_{a}, pK_{intr}, on the interaction between each pair of protonatable sites $W_{\mu\nu}$ and on the pH-value:[13]

$$G_{\text{n}} = \sum_{\mu=1}^{N} \left(\left(x_\mu^n - x_\mu^\circ\right) RT \ln 10 (\text{pH} - \text{p}K_{\text{intr},\mu}) \right) + \frac{1}{2} \sum_{\mu}^{N} \sum_{\nu}^{N} \left(W_{\mu\nu} \left(x_\mu^n - x_\mu^\circ\right) \left(x_\nu^n - x_\nu^\circ\right) \right) \quad (3)$$

The first term of eq. (3) represents the intrinsic protonation energy and the second term the interaction energy between sites. The value of x_μ^n is 1 if site μ is protonated and 0 if site μ is unprotonated in protonation state n. The reference protonation form of the site μ is given by x_μ°. R is the gas constant and T the absolute temperature. The intrinsic pK_{a}, pK_{intr}, and the interaction energy matrix $W_{\mu\nu}$ can be calculated from the linearized Poisson-Boltzmann equation as described in detail elsewhere.[13,22,32]

In analogy to eq. (2), d_μ is identified as dead-end if:

$$(d_\mu - c_\mu) RT \ln 10 \; (\text{p}H - \text{p}K_{\text{intr},\mu}) + \sum_{\nu \neq \mu} \min_{x} [W_{\mu\nu} (d_\mu - c_\mu) \left(x_\nu - x_\nu^\circ\right)] > 0 \quad (4)$$

Those d_μ fulfilling eq. (4) are excluded from further consideration. Since the sites adopt only two possible forms, identifying one form as a dead-end already fixes this site to the remaining form $f = 1 - d$. Equation (4) can be made computationally more efficient, if the sum in the second term is only taken over unfixed site. Since the protonation forms of the fixed sites influence the pK_{intr} of the unfixed sites, these must be corrected during the DEE search:

$$\text{p}K_{\text{intr},\mu}^{\text{cor}} = \text{p}K_{\text{intr},\mu} - \frac{1}{RT \ln 10} \sum_{i}^{N_{\text{fixed}}} W_{\mu i} \left(f_i - x_i^\circ\right) \quad (5)$$

where the value of f_{i} is either 1 or 0 depending on whether site i is fixed in the protonated or unprotonated form. The corrected p$K_{\text{intr},\mu}$ of the unfixed sites obtained from eq. (5) includes the interaction with the fixed sites as a constant energy contribution.

Computational Details

Structure Preparation

Calculations were performed on high resolution X-ray structures of bacteriorhodopsin. Two structures were chosen that represent the bR intermediate (PDB-entry: 1c3w[33]) and the M1 intermediate (PDB-entry: 1kg8[34]).

The crystal structures are resolved from residue Thr5 to Gly231. The termini were blocked with methyl groups. The coordinates of the missing EF-loop, which is highly flexible, were taken from a reference structure (PDB-entry: 1qhj[35]). The structures were prepared following the protocol given in ref. 36. For all structures, hydrogen atoms were generated using the HBUILD routine of the CHARMM force field. Their positions were optimized with the steepest descent algorithm followed by the conjugate gradient and Newton-Raphson algorithm until a tolerance gradient of 10^{-7} was fulfilled. All non-hydrogen atom coordinates were kept fixed during the minimization. The hydrophobic core of the membrane around the proteins is modelled by a ring of uncharged dummy atoms with a radius of 1.5 Å.[36]

Continuum Electrostatic Calculations

The intrinsic pK_a, pK_intr, and the interaction energy matrix $W_{\mu\nu}$ were calculated from the linearized Poisson-Boltzmann equation using the program multiflex of MEAD.[32] Partial atomic charges were taken from the CHARMM22 parameter set.[37] The charges for the retinal Schiff base were derived from a density functional calculation as described previously.[36] Atomic radii were set to 1.7 Å for carbon atoms, 1.0 Å for hydrogen atoms, 1.5 Å for oxygen atoms, 1.55 Å for nitrogen atoms, and 1.8 Å for sulfur atoms. The linearized Poisson-Boltzmann equation was numerically solved using the finite difference method. For the finite difference lattices, two focusing levels were used, first a coarse grid with a spacing of 1.0 Å, followed by a fine grid with a spacing of 0.25 Å. The dielectric boundary between the protein and the solvent was derived using a probe sphere with a radius of 1.4 Å and the atomic radii as given above. All calculations were performed using dielectric constants of 80.0 for the solvent and 4.0 for the protein and the membrane environment, the temperature was set to 300 K and the ionic strength to 0.1 M.

Models

The protonatable residues arginine, aspartate, glutamate, lysine, and tyrosine and the retinal Schiff base are defined as sites. These protonatable groups can bind one proton and thus can adopt $P = 2$ forms, protonated ($x = 1$) and unprotonated ($x = 0$). The two BR structures, bR and M1, each contain $N = 39$ sites leading to a total of $M = P^N \approx 5 \times 10^{11}$ states. The bR and M1 structure will be referred to as bR system and M1 system, respectively.

Additionally, systems small enough to allow an exhaustive search of the combinatorial space were derived from the bR and M1 structure. As sites were chosen: Asp36, Asp38, Lys40, Lys41, Glu74, Arg82, Asp85, Asp96, Asp102, Asp104, Asp115, Lys129, Arg164, Glu164, Glu166, Arg175, Glu194, Glu204, Asp212, Arg225, and the retinal Schiff base. In total, $N = 21$ sites were chosen leading to $M = P^N \approx 2{,}000{,}000$ states. All other sites were fixed to their standard protonation form at pH $= 7.0$. The fixed sites are treated as a constant energy contribution using eq. (5). These systems will be referred to as small bR system and small M1 system.

Results and Discussion

The X-DEE algorithm which we present in this article allows to determine gap-free lists of low energy states for molecular systems that can be treated by the DEE algorithm. A gap-free list is obtained by subsequently excluding the determined low energy states from the search space. In this section, we first analyze the computational performance of the X-DEE algorithm. As a first application, we then used X-DEE to determine protonation states of the proton pump bacteriorhodopsin (BR). We analyze the calculated low energy states with regard to the convergence of thermodynamic properties in dependence on the number of states determined by X-DEE. The energy distribution of the states of two BR structures, bR and M1, are discussed in detail with respect to the proton transfer of BR.

Statistics of the Search Basis $\mathcal{B}$

X-DEE divides the set $\mathcal{M} \setminus \mathcal{L}$ into suitable subsets $\mathcal{S}$ each characterized by a so-called search key $\vec{b}_\mathcal{S}$ (Figs. 3 and 4). $\mathcal{M}$ is the complete set of states of the system and $\mathcal{L}$ is a list of states to be excluded from the search for low energy states. Together, the search keys $\vec{b}_\mathcal{S}$ form the search basis $\mathcal{B}$. Each subset $\mathcal{S}$ represented by the respective search key $\vec{b}_\mathcal{S}$ is subjected to a separate DEE search. Since the DEE search is the most time consuming operation, the computational performance of X-DEE depends on the number of search keys and their properties. The search basis $\mathcal{B}$ is analyzed in terms of the absolute number of keys, the number of undefined sites per key and the number of new keys per state added to $\mathcal{L}$.

Total Number of Search Keys

In Figures 5a and 5b, the total number of search keys is plotted in dependence on the number of states in $\mathcal{L}$ and on the number of sites N, respectively. These graphs show the dependence of the number of search keys on (i) the number of states in $\mathcal{L}$, (ii) the number of forms P, and (iii) the number of sites N.

i The graph in Figure 5a shows that the total number of search keys depends approximately linearly on the number of states in $\mathcal{L}$. The number of states in $\mathcal{L}$ influences the number of search keys in two different ways: on the one hand, each additional state in $\mathcal{L}$ increases the number of states to be excluded from the search and thereby tends to increase the number of generated keys. On the other hand, each additional state in $\mathcal{L}$ decreases the search space $\mathcal{M} \setminus \mathcal{L}$ and thereby tends to decrease the number of generated keys. Ultimately, the number of search keys will decrease with the number of states in $\mathcal{L}$. However, as long as $\mathcal{L}$ is small compared to $\mathcal{M} \setminus \mathcal{L}$, an approximately linear increase of the total number of search keys can be observed (Fig. 5a).

ii The graph in Figure 5a shows the total number of search keys in dependence on the number of states in $\mathcal{L}$ for systems with $N = 21$ sites and a number of forms P ranging from 2 to 10. For each additional form, one additional key can be generated per site. Thus, increasing the number of forms by 1 increases the slope of the curve by approximately N.

iii The graph in Figure 5b depicts the total number of search keys in dependence on the number of sites N for a number of forms P ranging from 2 to 10. The search keys were generated for lists of 1000 random states. The number of recursion levels of CreateSearchBasis is equal to the number of sites. Thus, the number of generated keys increases approximately linearly with N.

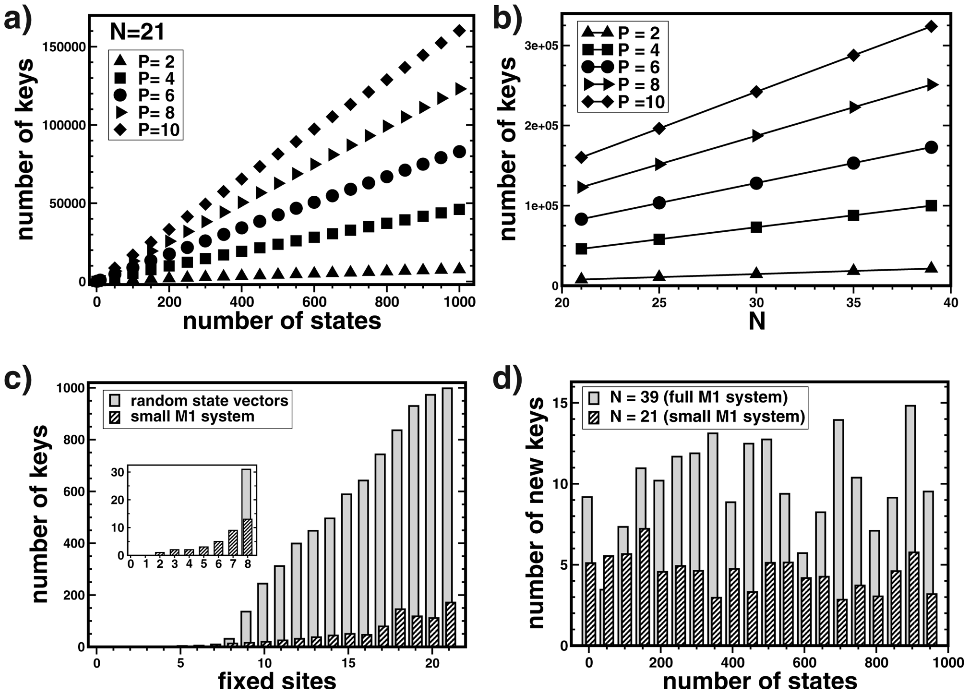

Figure 5. Statistics of the search basis. (a) Total number of search keys in dependence on the number of states in $\mathcal{L}$ plotted for $N = 21$ sites and numbers of forms P ranging from 2 to 10. (b) Total number of search keys plotted in dependence on the number of sites N for numbers of forms P ranging from 2 to 10. The search bases were generated for lists of 1000 random states. (c) Distribution of search keys with regard to the number of fixed sites. The search basis was generated for two systems with $N = 21$ sites and $P = 2$ forms and for lists of 1000 states. One list contained random state vectors, and the other contained 1000 low energy states of the small M1 system. The insert shows a magnification of the range of fixed sites from 0 to 8 (x-axis). (d) The average number of new keys in dependence on the number of states in $\mathcal{L}$ for systems with $P = 2$ forms and $N = 21$ and 39 sites, i.e., the small and full M1 system, respectively. The average was taken over 50 consecutive states.

From Figures 5a and 5b it can be seen that the number of search keys increases linearly with the size of the system. While the total number of keys can be relatively high, we will show in the next paragraphs that the computational cost of X-DEE does not rise as quickly as the number of keys.

Properties of the Search Keys

The computational cost of a DEE search depends on the size of the subset $\mathcal{S}$ that has to be searched. The size of $\mathcal{S}$ is directly correlated to the number of undefined sites in the search key $\vec{b}_{\mathcal{S}}$. In other words, the more sites are fixed in $\vec{b}_{\mathcal{S}}$, the faster is the DEE search due to the scaling behavior of the Goldstein algorithm.[4] In Figure 5c, the number of search keys is plotted in dependence on the number of fixed sites per key for a system with $N = 21$ sites and $P = 2$ forms. The search keys were generated for two lists: a list of 1000 random states and the gap-free list of 1000 low energy states of the small M1 system. As can be observed, the majority of search keys have a large number of fixed sites. Each recursion level of CreateSearchBasis potentially increases the number of sublists $\mathcal{L}_{\text{sub}}$ for which search keys can be generated. The higher the recursion level, the more sites will be fixed in the generated keys. Thus, with each recursion level more keys can be generated and more sites will be fixed in the search keys rendering DEE searches for these keys inexpensive.

Furthermore in Figure 5c, it can be seen, that significantly more search keys are generated for a list of random states than for a list of low energy states. This behavior is due to the fact that, in contrast to random states, consecutive low energy states resemble each other. Lists of similar states lead to fewer sublists $\mathcal{L}_{\text{sub}}$ during early recursion levels, since less used forms will exist for a given site in a given sublist $\mathcal{L}_{\text{sub}}$. During the early recursion levels, this smaller number of sublists leads to the generation of more keys for lists of low energy states (see inset in Fig. 5c). Since the final number of sublists $\mathcal{L}_{\text{sub}}$ is equal to the number of states in $\mathcal{L}$, during later recursion levels, there will be a larger number of subdivisions for a list of low energy states. This implies that for the list of low energy states, significantly less search keys are generated than for lists of random states making X-DEE significantly more efficient for the former.

Number of New Keys for Each New State in $\mathcal{L}$

X-DEE can be used to build up a list of low energy states one after the other. Then, each $\mathcal{L} \cup \{\vec{x}_{\text{new}}\}$ differs from $\mathcal{L}$ only by the new state $\vec{x}_{\text{new}}$ and the search keys representing $\mathcal{M} \setminus \mathcal{L} \cup \{\vec{x}_{\text{new}}\}$ will be mostly identical to those keys representing $\mathcal{M} \setminus \mathcal{L}$. To obtain the next state, only subsets represented by new keys have to be subjected to a DEE search. Therefore, the computational cost to determine the next state will be proportional to the number of new keys necessary to characterize $\mathcal{M} \setminus \mathcal{L} \cup \{\vec{x}_{\text{new}}\}$.

In Figure 5d, the number of new keys is plotted in dependence on the number of states in $\mathcal{L}$ for systems with $P = 2$ forms and $N = 21$ and 39 sites (small and full M1 systems). The number of new keys is averaged over 50 consecutive states. The number of new keys for every new state in $\mathcal{L}$ remains approximately constant. Peaks in the number of new keys can be observed that amount to a maximum of 112 and 253 new keys for the small and the full M1 system, respectively. These peaks occur, if the new state that was added to $\mathcal{L}$ originated from a search key that represents a large subset $\mathcal{S}$ of $\mathcal{M} \setminus \mathcal{L}$, i.e., if the state was found by a search key with a large number of unfixed sites. This large subset $\mathcal{S}$, previously represented by only one search key, now has to be split up into many smaller subsets, each represented by a new search key. The frequency and also the amplitude of the peaks decrease with the number of states in $\mathcal{L}$, since search keys with mostly undefined sites become less probable. On average, for the small and the full M1 systems, 5 and 10 new keys are generated per state added to $\mathcal{L}$, respectively. The computational cost to determine an additional state is on average constant.

Convergence of the Partition Function

Thermodynamic properties are efficiently calculated using Monte Carlo methods. These methods achieve a convergence of thermodynamic properties by sampling a large fraction of the phase space biased towards the low energy regime.[38] While Monte Carlo methods can provide a set of low energy states, there is no guarantee that this set is gap-free, i.e., that no states with lower energy exist. X-DEE, in contrast, was developed to provide a complete description of the lowest energy regime, i.e., a gap-free list of low energy states. However, since the thermodynamic properties of a system depend mainly on the states of lowest energy, a gap-free list of low energy states may additionally be used to derive thermodynamic properties.

Thermodynamic properties depend on the partition function Z which is the sum of the Boltzmann factors over all possible states of the system:

$$Z = \sum_{\mathrm{i}}^{M} \exp(-\beta G_{\mathrm{i}}) \tag{6}$$

where M is the total number of states, G_{i} is the state energy and $\beta = 1/k_{\mathrm{B}}T$; k_{B} is the Boltzmann constant and T the absolute temperature. The exact value of Z can only be obtained, if all state energies of the system are known. Then, the exact population P of each state n can be calculated as:

$$P_{\mathrm{n}} = \frac{\exp(-\beta G_{\mathrm{n}})}{Z} \tag{7}$$

The calculation of all state energies is, however, not feasible for complex systems. We approximate the partition function as:

$$Z_{\text{approx}} = \sum_{\mathrm{i}}^{M_{XD}} \exp(-\beta G_{\mathrm{i}}) \tag{8}$$

where the sum goes over all M_{XD} states that were determined by X-DEE. We calculate the approximate population P of each state n as:

$$P_{\mathrm{n}} = \frac{\exp(-\beta G_{\mathrm{n}})}{Z_{\text{approx}}} \tag{9}$$

Since all low energy states are known, the convergence of the partition function can be analyzed with respect to the number of low energy states that have to be considered.

Lists of low energy states are calculated for the full bR and M1 systems and for the small bR and M1 systems. For the small systems the total number of states is approximately 2 million rendering the calculation of all state energies and thus of the exact partition function possible.

Figures 6a and 6b show the convergence of the partition function for all systems in dependence on the number of considered states. In Figure 6a, the exact partition function is indicated by dotted lines for the small systems. As can be seen in Figures 6a and 6b, the approximated partition function Z_{approx} converges quickly with the number of considered states. Likewise, the population of the lowest energy state P_{lowest} converges quickly to the exact population for the small systems as shown in Figure 6c. Figure 6d shows the convergence behavior of the population of the lowest energy state P_{lowest} for the full systems. For these systems, a convergence of the approximated partition function and the population of the lowest energy state to a constant value can be observed when approximately 50 low energy states are considered. Z_{approx} does not change anymore between the 1500th and the 3000th state. Both the full and the small M1 systems show a slower convergence than the respective bR systems, indicating a larger density of states in the low energy range for the M1 system as will be discussed in more detail in the next section.

Analysis of the Low Energy States

In Figure 7, the energy distribution of the low energy states is shown for the bR and M1 system. All states in an energy range of 15 $k_{\mathrm{B}}T_{300\,\mathrm{K}}$ are depicted. As can be seen, the density of states is significantly higher for the M1 system than for the bR system. For the bR system approximately 200 states lie in the 15 $k_{\mathrm{B}}T_{300\,\mathrm{K}}$ range, while the M1 system has approximately 700 states in the same energy range. An indication of this behavior was already given by the slower convergence of the partition function for the M1 system (Fig. 6b). Whether a system is well represented by the lowest energy state or by several low energy states can only be decided, after a list of low energy states was calculated. Interestingly, for the bR system, the first two states are closer in energy than for the M1 system (see inset in Fig. 7).

As shown in Figure 6, the population of each state can be calculated to a good approximation from relatively few low energy states.

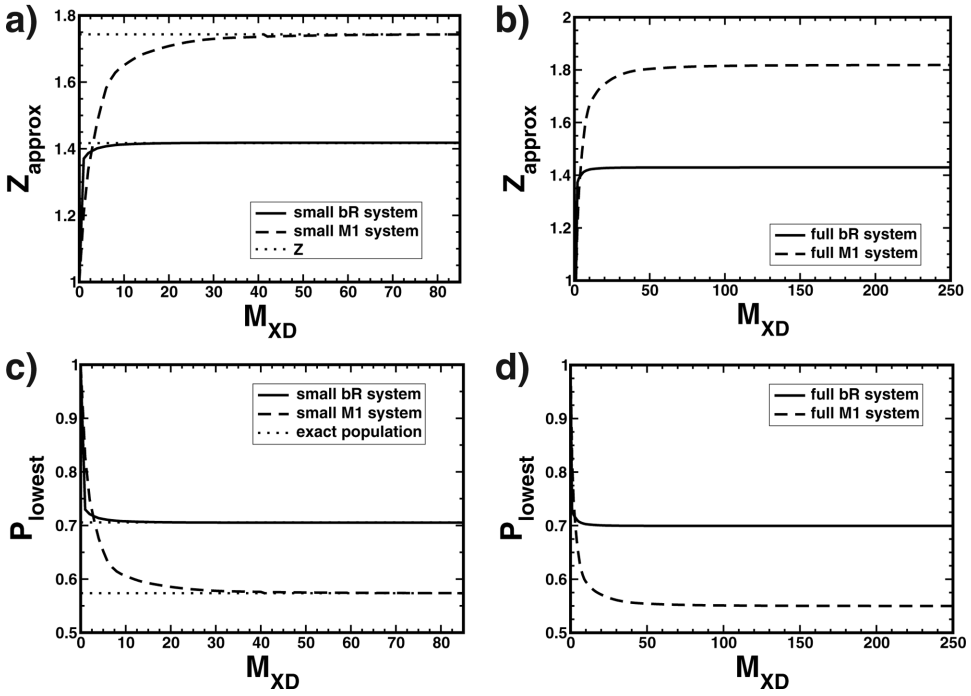

Figure 6. Convergence of Z_{approx} and the population of the lowest energy state P_{lowest}. Z_{approx} [eq. (8)] is plotted in dependence on the number of states M_{XD} considered to approximate the partition function: (a) small systems, and (b) full systems. The dotted lines indicate Z calculated from all states for the small systems. The population of the lowest energy state P_{lowest} is plotted in dependence on the number of states considered to approximate the partition function: (c) small systems, and (d) full systems. For the small systems, the dotted lines indicate the exact population of the lowest energy state calculated from all states. All calculations were performed at pH $= 7.0$.

Thus, for the full bR and M1 system, the population for each low energy state P_{n} [eq. (9)] was calculated from the converged partition function. The calculations were performed for different pH-values. We define the cumulative population P_{tot} as the sum of populations up to a given number of states K:

$$P_{\text{tot}} = \sum_{n=1}^{K} P_{\text{n}} = \sum_{n=1}^{K} \frac{\exp(-\beta G_{\text{n}})}{Z_{\text{approx}}} \tag{10}$$

The graphs in Figure 8 show P_{tot} as a function of K. It can be seen that the cumulative population converges rapidly to 1.0 for most pH-values showing that relatively few low energy states are significantly populated. Additionally, the contribution of each state to P_{tot} can be deduced. The number of significantly populated states depends on pH. At pH-values of $5.0, 6.0$, and 9.0, more states are significantly populated and thus, the convergence of P_{tot} is slower when compared to the convergence at pH $= 7.0$. Low energy states of similar energy and thus, similar population result from the titration behavior of the protonatable residues. For example, aspartate and glutamate residues commonly titrate at low pH. At high pH, lysine and arginine residues usually titrate. States that only differ in the protonation form of these residues are expected to have similar state energies in the respective pH range. Most other states are not significantly populated. Thus, at physiological pH, the bR structure is well represented by the lowest energy state. For a good representation of the M1 system, however, 10 or even more states should be considered.

Energy Spectrum of the Protonation States of the BR Photocycle

The energy spectrum obtained by X-DEE provides a picture of the energy distribution of functionally important states. The calculations were performed on two BR structures, the bR and M1 system, that represent the photocycle intermediates before and after the first proton transfer step. To differentiate between the bR and M1 systems (BR structures) and the [bR], [M1], and [M2] protonation states, the latter are enclosed in squared brackets. The [bR] protonation state is characterized by a protonated retinal Schiff base and a deprotonated Asp85. Furthermore, Asp96, Asp115, Arg82, and the proton release complex are protonated, and Asp212 is unprotonated. The [M1] protonation state differs from the [bR] protonation state by a deprotonated retinal Schiff base and a protonated Asp85. The [M2] protonation state differs from the [M1] protonation state by a deprotonated proton release complex. The discussed key residues of the proton transfer are shown explicitly in Figure 1.

Figure 9 shows the energy of the protonation states for the bR and the M1 system. The states that represent the lowest [bR], [M1], and [M2] protonation states are highlighted. As can be seen, for the bR system the state of lowest energy is the [bR] protonation state, and the [M1] protonation state is the third state. The energy difference between these states is approximately 2.5 kcal/mol. The

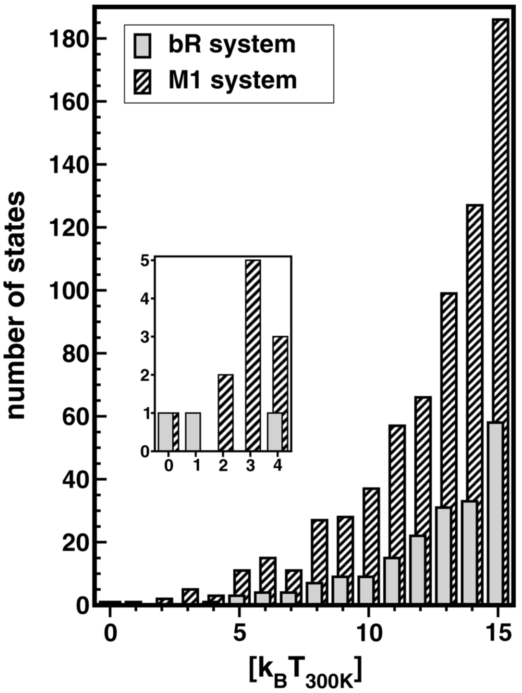

Figure 7. Distribution of state energies. All low energy states in an energy range of 15 $k_BT_{300\,K}$ are considered for the bR and M1 systems. The state energies were calculated at pH = 7.0. The insert shows a magnification of the $k_BT_{300\,K}$ range from 0 to 4 (x-axis).

[M2] protonation state differs from the state of lowest energy by 10.8 kcal/mol. The second state is a [bR] protonation state that differs from the lowest energy state in the protonation form of an arginine residue that is exposed to the bulk water and close to the membrane. This arginine does not interact with the key residues of proton transfer. For the M1 system, the lowest energy state is the [M1] protonation state. The [bR] protonation state is the seventh state differing from the lowest energy state by less than 2 kcal/mol. The [M2] protonation state is about 3.5 kcal/mol higher in energy than the state of lowest energy. Thus, for the M1 system the energy difference between [M1] and [M2] state is significantly smaller than for the bR system. The second state of the M1 system has a protonated retinal Schiff base and a protonated Asp85, i.e., constituting a state that does not correspond to a photocycle intermediate protonation state. The following three states are [M1] protonation states where surface residues change their protonation. The sixth state has a deprotonated proton release complex, while both retinal Schiff base and Asp85 are protonated. Like the second state, this state thus does not correspond to a photocycle intermediate protonation state.

The states of lowest energy of the bR and M1 systems agree with their experimental classification. For the bR system the next new intermediate protonation state is [M1], which follows the [bR] protonation state during the photocycle. For the M1 system, the next two new intermediate protonation states are [bR], corresponding to the foregoing state during the photocycle, and the [M2] state that corresponds to the following state during the photocycle. The energy differences between the lowest [bR], [M1], and [M2] protonation

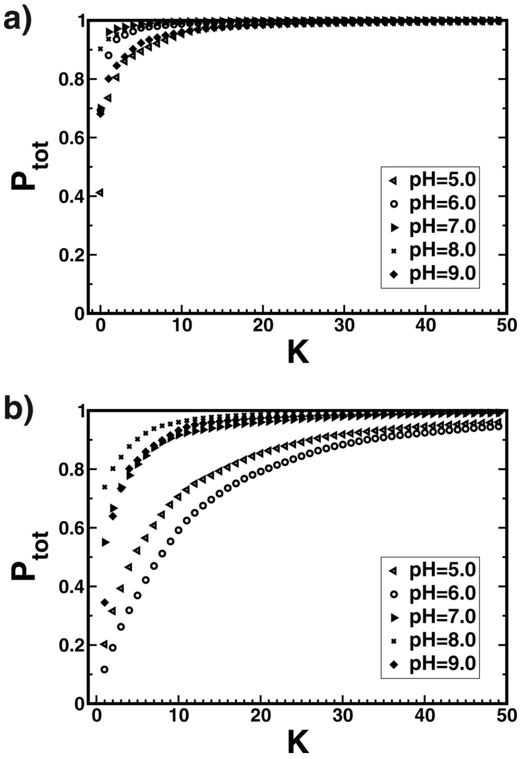

Figure 8. Population of states. Cumulative population P_{tot} as a function of K [eq. (10)]. (a) bR system and (b) M1 system. The data are shown for different pH-values.

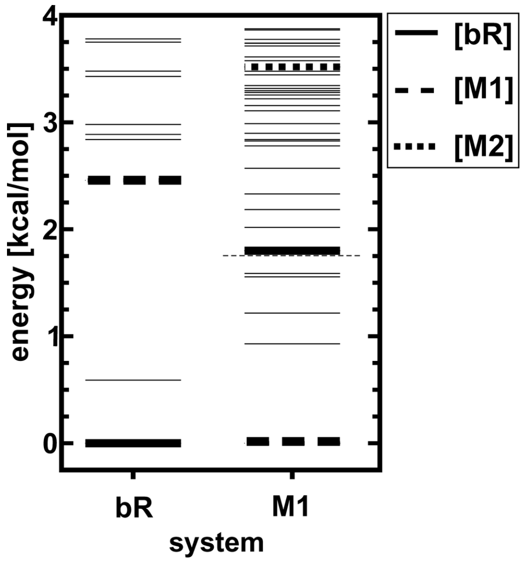

Figure 9. Energy spectrum of protonation states of the BR photocycle. For the bR and the M1 systems, all state energies are plotted up to 4 kcal/mol. The lowest [bR], [M1], and [M2] protonation states are highlighted. The [bR], [M1], and [M2] protonation states reflect the first two proton transfer steps of the BR photocycle. The [bR] protonation state is characterized by a protonated retinal Schiff base and a deprotonated Asp85, the [M1] protonation state by a deprotonated retinal Schiff base and a protonated Asp85, and the [M2] state has in addition lost a proton in the proton release group. The thin dashed line indicates the sixth state of the M1 system.

states are relatively small. Interestingly, for both systems, these lowest functionally important states differ only in the discussed key residues of proton transfer. This observation implies that protonation changes of the key residues during the first step of the photocycle are decoupled from protonation changes of the other protonatable residues. In this respect, the first proton transfer step of the BR photocycle might be a rather special case. Other transfer steps may show a more complex behavior, since protonation changes of functionally relevant groups can be coupled to changes in other groups that have not yet been considered before. It will be interesting to analyze whether the behavior of transfer steps both in the BR photocycle and in other systems is indeed more complex. Since X-DEE is able to determine gap-free lists of low energy states, it is well suited for analyzing the energetics of charge transfer processes.

Conclusion

This article introduces the X-DEE algorithm that generates a gap-free list of low energy states. X-DEE can be applied to all systems that can be treated with the DEE algorithm. The basic idea of X-DEE is to exclude a given list of states from the search space that is explored by DEE. X-DEE determines the state of lowest energy of all states not in this list. Consecutively adding new states to the list, X-DEE can be used to build up a gap-free list of low energy states. Alternatively, X-DEE can be used to complete any list of low energy states by eliminating gaps. Such lists can for instance be generated by Monte Carlo sampling. The computational cost of X-DEE depends linearly on the size of the system and the number of states to be excluded from the search. For lists of low energy states that are build up one after the other, we have shown that the computational cost to determine an additional state remains on average constant.

In the present work, X-DEE is implemented to generate complete lists of low energy protonation states. We applied X-DEE to determine low energy protonation states of two structures of the proton pump bacteriorhodopsin, a bR and an M1 structure. For both structures, the calculated state of lowest energy agrees with experimental data. The list of low energy states allows an analysis of all energetically accessible states. From this analysis the energy difference between functionally important states could be obtained.

X-DEE will be useful, whenever a set of low energy states is of interest. In addition to protonation state predictions, X-DEE can be applied to obtain a complete list of the lowest energy states in homology modelling and X-ray refinement. A particular interesting application is enzyme design. A good catalyst should not only stabilize the reactant but also the transition state. X-DEE can be used to optimize the energy difference between these states and thus can be a powerful tool in enzyme design. In addition, X-DEE provides useful starting information for the theoretical investigation of kinetic properties. By providing a gap-free list of low energy states, the complete pool of possible states is available, i.e., states that may be occupied by the system. The system kinetics can then be analyzed by determining the transitions between these states.

Acknowledgment

We thank Prof. Donald Bashford for providing his program MEAD.

References

1. Desmet, J.; de Maeyer, M.; Hazes, B.; Lasters, I. Nature 1992, 356, 539.
2. Goldstein, R. F. Biophys J 1994, 66, 1335.
3. Leach, A. R.; Lemon, A. P. Proteins 1998, 33, 227.
4. Pierce, N. A.; Spriet, J. A.; Desmet, J.; Mayo, S. L. J Comput Chem 2000, 21, 999.
5. Hellings, M.; de Maeyer, M.; Verheyden, S.; Hao, Q.; van Damme, E. J.; Peumans, W. J.; Engelborghs, Y. Biophys J 2003, 85, 1894.
6. Adcock, S. A. J Comput Chem 2004, 25, 16.
7. Stiebritz, M. T.; Muller, Y. A. Acta Crystallogr D Biol Crystallogr 2006, 62, 648.
8. Looger, L. L.; Hellinga, H. W. J Mol Biol 2001, 307, 429.
9. Shah, P. S.; Hom, G. K.; Mayo, S. L. J Comput Chem 2004, 25, 1797.
10. Georgiev, I.; Lilien, R. H.; Donald, B. R. Bioinformatics 2006, 22, e174.
11. Lukashin, A. V.; Rosa, J. J. Bioinformatics 1999, 11, 947.
12. Zhang, N.; Zeng, C.; Wingreen, N. S. Proteins 2004, 57, 565.
13. Ullmann, G. M.; Knapp, E.-W. Eur Biophys J 1999, 28, 533.
14. Chen, I.; Mathis, P.; Koepke, J.; Michel, H. Biochemistry 2000, 39, 3502.
15. Becker, T.; Ullmann, R. T.; Ullmann, G. M. J Phys Chem B 2007, 777, 2957.
16. Nicholls, D. G.; Ferguson, S. J. Bioenergetics 3; Academic Press: London, 2002.
17. Mulkidjanian, A. Y. FEBS Lett 1999, 463, 199.
18. Gunner, M. R.; Mao, J.; Song, Y.; Kim, J. Biochim Biophys Acta 2006, 1757, 942.
19. Onufriev, A.; Smondryev, A.; Bashford, D. J Mol Biol 2003, 332, 1183.
20. Kloppmann, E.; Becker, T.; Ullmann, G. M. Proteins 2005, 61, 953.
21. Bombarda, E.; Becker, T.; Ullmann, G. M. J Am Chem Soc 2006, 128, 12129.
22. Honig, B.; Nicholls, A. Science 1995, 268, 1144.
23. Schaefer, M.; Karplus, M. J Phys Chem 1996, 100, 1578.
24. Stoeckenius, W. Prot Sci 1999, 8, 447.
25. Haupts, U.; Tittor, J.; Oesterhelt, D. Annu Rev Biophys Biomol Struct 1999, 28, 367.
26. Heberle, J.; Fitter, J.; Sass, H. J.; Büldt, G. Biophys Chem 2000, 85, 229.
27. Lanyi, J. K.; Luecke, H. Curr Opin Struct Biol 2001, 11, 415.
28. Neutze, R.; Pebay-Peyroula, E.; Edman, K.; Royant, A.; Navarro, J.; Landau, E. M. Biochim Biophys Acta 2002, 1565, 144.
29. Heßling, B.; Souvignier, G.; Gerwert, K. Biophys J 1993, 65, 1929.
30. Zscherp, C.; Schlesinger, R.; Tittor, J.; Oesterhelt, D.; Heberle, J. Proc Natl Acad Sci USA 1999, 96, 5498.
31. Garczarek, F.; Brown, L. S.; Lanyi, J. K.; Gerwert, K. Proc Natl Acad Sci USA 2005, 102, 3633.
32. Bashford, D. In Scientific computing in object-oriented parallel environments; Ishikawa, Y.; Oldehoeft, R. R.; Reynders, J. V. W.; Tholburn, M. Eds.; Springer: Berlin, 1997, pp. 233.
33. Luecke, H.; Schobert, B.; Richter, H.-T.; Cartailler, J.-P.; Lanyi, J. K. J Mol Biol 1999, 291, 899.
34. Facciotti, M. T.; Rouhani, S.; Burkard, F. T.; Betancourt, F. M.; Downing, K. H.; Rose, R. B.; McDermott, G.; Glaeser, R. M. Biophys J 2001, 81, 3442.
35. Belrhali, H.; Nollert, P.; Royant, A.; Menzel, C.; Rosenbusch, J. P.; Landau, E. M.; Pebay-Peyroula, E. Structure 1999, 7, 909.
36. Calimet, N.; Ullmann, G. M. J Mol Biol 2004, 339, 571.
37. MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Jr.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Prodhom, D. T. N. B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M. J Phys Chem B 1998, 102, 3586.
38. Landau, D. P.; Binder, K. A Guide to Monte Carlo Simulations in Statistical Physics; University Press: Cambridge, 2005.