Calculating Mean First Passage Times from Markov Models of Proteins

Christian H. Jensen Dmitry Nerukh Robert C. Glen

May 22, 2007

Abstract

Assuming that the dynamics of peptides and proteins can be described by Markov transitions between different configurational states, we present a method which can calculate the mean first passage time (MFPT) between *sets* of initial *and final* states. The method is described in detail and differences between this and a method commonly employed [1] are explained. It is shown that the proposed method is (i) more general in allowing sets of final states, (ii) significantly faster, (iii) more accurate since it does not involve the calculation of infinite summations. Particular attention is given to the biologically important case of multiple final states.

1 Introduction

There are many methods which seek to simulate the folding of a peptide or protein. They range from the very course grained approaches like the HP model [2] to models with atomic detail like the Molecular Dynamics approach [3]. While the course grained method gives results which can be useful as guidelines when designing proteins, they do not describe exactly how a specific protein folds. To describe how a specific protein in atomistic detail folds, a model like Molecular Dynamics is needed. The problem with the latter is that for system sizes of biochemical interest the computational task of making a single Molecular Dynamics simulation which shows a complete folding process is infeasible. However, recently a method which combines several Molecular Dynamics simulations by using clustering and a Markov model for the state transitions has been proposed [1]. Using this method it is possible to reconstruct the overall dynamics of a peptide from thousands of individual simulations. This can be done by first clustering the configurational space into discrete states, and then count the number of transitions between the different states from all the simulations. The Markov model can be described by a state vector v and a transition matrix T. Given that the system is described by a state vector v_t at time t the state vector at time $t+1$ can be calculated as $v_{t+1} = Tv_t$.

One of the fundamental problems that arises in the analysis of such simulations is the definition of the average folding times in cases when several states assigned by the algorithm can be considered as "final". From the bio-chemical point of view the native states of proteins are defined by their ability to perform biological functions. The latter is often assigned to a "functional core" of the molecule while the rest of the molecule is allowed to "breath" producing configurations that differ sometimes very significantly [4]. Clustering algorithms produce many states described above and formal coarse graining can not be used here because of the high risk of loosing important details in the functionally relevant parts of the molecule. Thus, the most natural solution of the problem lies in assigning a set of final states rather than trying to artificially lump states together.

In this paper, we present a method for calculating the average folding time from a Markov model. This method generalises to calculating the folding time between multiple initial and final states. This is different from the method [1] previously and most commonly used (see, for example, [5, 6]) and we investigate cases where multiple final states lead to significant differences in the folding times. For degenerate cases, the differences turn out to be critical, however, the differences are also shown to be very substantial (up to ∼70%) for a Markov model constructed from a Molecular Dynamics simulation.

2 Theory

To calculate the average transition time of a Markov model we need to define initial and final states. Each of these can either be one state or a set of states. Assuming that we have a set of initial states I and a set of final states F the average transition time can be written as:

$$t_{IF} = \sum_{n=1}^{\infty} n P_{IF}(n) \tag{1}$$

Here $P_{IF}(n)$ is the probability for all paths of length n which start in I and end on F. We assume that the Markov process is described by a transition matrix T and that there is a total of N states. The first problem in the calculation is to find an expression for $P_{IF}(n)$. Below we introduce $\tilde{T}$,L,o and v.

- From the transition matrix T remove the rows and columns for all states in F to form a new matrix $\tilde{T}$. This new matrix will have a dimension of $(N-d) \times (N-d)$, where d is the number of states in F.

- Form matrix L which is of dimension $d \times (N-d)$ and holds the matrix elements of T that give the probabilities for entering F from all other states.

- Form row vector o which is of dimension $1 \times d$ and holds ones in all places.

- Form vector v of dimension $(N-d) \times 1$. The elements of v must describe the initial distribution of states in I. If each starting state is equally likely then their elements must be equal. For the states not in I the initial value in v must be zero. The total sum of all elements in v must be 1.

Using the quantities given above $P_{IF}(n)$ can be written as $oL\tilde{T}^{n-1}v$ (an explanation for this is given in Appendix A). Let us assume that $\tilde{T}$ has eigenvectors e_i with corresponding eigenvalues λ_i. We then expand v in this basis. This gives $v = \sum_i \alpha_i e_i$. The average transition time (1) can then be written as:

$$\begin{aligned} t_{IF} &= \sum_{n=1}^{\infty} nP_{IF}(n) = \sum_{n=1}^{\infty} noL\tilde{T}^{n-1}v \qquad (2) \\ &= \sum_{n=1}^{\infty} noL\tilde{T}^{n-1} \sum_i \alpha_i e_i = \sum_{n=1}^{\infty} \sum_i noL\alpha_i \lambda_i^{n-1} e_i \qquad (3) \\ &= \sum_i \left(\sum_{n=1}^{\infty} n\lambda_i^{n-1} \right) \alpha_i oLe_i = \sum_i \frac{\alpha_i}{(1-\lambda_i)^2} oLe_i \qquad (4) \end{aligned}$$

The method proposed differs from the commonly used method [1] in (i) it allows sets of *final* states, while the latter one only permits a single final state, (ii) it does not involve the calculation of infinite summations, thus reducing the calculation time significantly, (iii) for the same reason our method is more accurate since the calculation of the eigenvectors and eigenvalues is a well developed area of applied mathematics.

3 Results

In this section we apply the method described to three examples. Firstly we investigate two artificial degenerate cases which describe extreme folding situations. One case is when only one folding path exists and the other extreme is when transitions between all states exist so that all paths are possible.

The situation with only one folding path can be described by the transition matrix (5). There are a total of five states in this matrix. Labeling with numbers from 1 to 5 we consider a situation where the initial state is 1 and the set of final state is 4,5. Using the method given in the literature [1] the average folding times from state 1 to 4 and from state 1 to 5 can be found. They can then be compared with the time of reaching the set of states 4 and 5. The average folding time from state 1 to 4 is 45.0 time steps and the average folding time from state 1 to 5 is 80.0 time steps. Thus, it is critical which state to consider as a ”final” one even though they are the neighbouring states on the folding path.

Using the method proposed in this work, the average folding from state 1 to the set of states 4 and 5 is found to be 45.0 time steps. This is the same as the average folding time between states 1 and 4. The reason why this result is correct is that there is only one folding path between states 1 and 5. This means that it is not possible to reach state 5 from state 1 without first passing through state 4. Hence, only the time taken to arrive to state 4 which is of importance defines the folding time to the set 4,5.

$$\begin{bmatrix} 0.8000 & 0.1000 & 0.0000 & 0.0000 & 0.0000 \\ 0.2000 & 0.8000 & 0.1000 & 0.0000 & 0.0000 \\ 0.0000 & 0.1000 & 0.8000 & 0.1000 & 0.0000 \\ 0.0000 & 0.0000 & 0.1000 & 0.8000 & 0.2000 \\ 0.0000 & 0.0000 & 0.0000 & 0.1000 & 0.8000 \end{bmatrix} \qquad (5)$$

The other limiting case, the situation with multiple folding paths can be represented by the transition matrix (6). Again there is a total of five states in the matrix and we take state 1 to be the initial state and states 4 and 5 to be the final states. By using the method from [1] the calculated average folding time from state 1 to 4 is 20.0 time steps and from state 1 to 5 is also 20.0 time steps. Our method, however, results in 10.0 time steps for the 4,5 set. The reason why this result is correct is because compared to a single final state the probability of entering two final states from all other states is doubled. Since this probability is doubled in each time step the average folding time is halved.

$$\begin{bmatrix} 0.8000 & 0.0500 & 0.0500 & 0.0500 & 0.0500 \\ 0.0500 & 0.8000 & 0.0500 & 0.0500 & 0.0500 \\ 0.0500 & 0.0500 & 0.8000 & 0.0500 & 0.0500 \\ 0.0500 & 0.0500 & 0.0500 & 0.8000 & 0.0500 \\ 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.8000 \end{bmatrix} \tag{6}$$

Thus, in both limiting cases it is fundamentally important to consider multiple final states as *sets* of states. Considering them individually leads to significantly different results in the mean passages times.

The cases discussed above are constructed examples. To test a realistic situation a transition matrix was obtained from a Molecular Dynamics simulation of the four residue peptide VPAL solvated in 874 water molecules. The simulation box was 3.0x3.0x3.0Å. The system was equilibrated before it was sampled for 500ns. The force field was 53a6 [7, 8, 9] and the temperature was held at 300K during the simulation. The Molecular Dynamics trajectory was then clustered in configurational space and the number of transitions between these states was counted. From this the transition matrix (7) was calculated.

As in the previous cases we take state 1 to be the initial state and states 4 and 5 to be the final states. Following the method from [1] the average folding time from state 1 to 4 was calculated to be 277.7 ps and 384.7 ps for the 1 to 5 transition. The average folding time from state 1 to the set 4,5 obtained with our method is 220.0 ps. The difference demonstrates that not only limiting cases can produce substantial differences in calculated average folding times.

$$\begin{bmatrix} 0.9940 & 0.0080 & 0.0018 & 0.0220 & 0.0001 \\ 0.0023 & 0.9802 & 0.0000 & 0.0000 & 0.0232 \\ 0.0016 & 0.0003 & 0.9973 & 0.0073 & 0.0006 \\ 0.0020 & 0.0000 & 0.0007 & 0.9505 & 0.0152 \\ 0.0000 & 0.0115 & 0.0000 & 0.0201 & 0.9610 \end{bmatrix} \tag{7}$$

From the matrix we can see that it is not of the form of either of the matrices 5 or 6. The new matrix can be described as a combination of the two. Therefore, the difference in average folding time can also be understood by a combination of a limited number of folding paths (5) and an increase in the probability at each time step for reaching the final states (6). For larger peptide and protein systems possibly involving thousands of states the difference in the calculated average folding times may be less significant, however, for a given example this is difficult to quantify without calculating the correct average folding time given by Equation 1.

4 Conclusions

In this work we present an efficient method for obtaining the average folding time between sets of states. By applying the method to two constructed degenerate transition matrices we demonstrate that the results are fundamentally different from the individual transitions to the states in the final set. The method was also applied to a transition matrix of a four residue peptide obtained from a Molecular Dynamics simulation. Our method predicted an average folding time of 220.0 ps to a set of two states whereas the commonly used method from [1] resulted in 277.7 and 384.7 ps for the states individually. It should be stressed that taking the average of the times for the individual states is incorrect. The only solution in this situation would be joining together all final states by forming a single one and then applying the method [1]. This, however, would involve recalculation of the transition matrix and that would result in significant computational overheads for realistic molecules having thousands of states.

5 Acknowledgments

The work is supported by Unilever and the European Commission (EC Contract Number 012835 - EMBIO). The authors thank Dr Maxim Fedorov for useful comments on the interpretation of the results.

References

[1] G. Jayachandran, V. Vishal, and V. S. Pande. Using massively parallel simulation and markovian models to study protein folding: Examining the dynamics of the villin headpiece. *Journal of Chemical Physics*, 124(16):164902, 2006.

[2] K. F. Lau and K. A. Dill. A lattice statistical-mechanics model of the conformational and sequence-spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989.

[3] S. A. Adcock and J. A. McCammon. Molecular dynamics: Survey of methods for simulating the activity of proteins. *Chemical Reviews*, 106(5):1589–1615, 2006.

[4] Alexei V. Finkelstein and Oleg Ptitsyn. *Protein Physics, A Course of Lectures.* Academic Press, 2002.

[5] Sergei V. Krivov and Martin Karplus. Hidden complexity of free energy surfaces for peptide (protein) folding. *PNAS*, 101(41):14766–14770, 2004.

[6] Peter M. Kasson, Nicholas W. Kelley, Nina Singhal, Marija Vrljic, Axel T. Brunger, and Vijay S. Pande. Ensemble molecular dynamics yields submillisecond kinetics and intermediates of membrane fusion. *PNAS*, 103(32):11916–11921, 2006.

[7] B. Hess and N. F. A. van der Vegt. Hydration thermodynamic properties of amino acid analogues: A systematic comparison of biomolecular force fields

and water models. *Journal of Physical Chemistry B*, 110(35):17616–17626, 2006.

[8] C. Oostenbrink, T. A. Soares, N. F. A. van der Vegt, and W. F. van Gunsteren. Validation of the 53a6 gromos force field. *European Biophysics Journal with Biophysics Letters*, 34(4):273–284, 2005.

[9] C. Oostenbrink, A. Villa, A. E. Mark, and W. F. Van Gunsteren. A biomolecular force field based on the free enthalpy of hydration and solvation: The gromos force-field parameter sets 53a5 and 53a6. *Journal of Computational Chemistry*, 25(13):1656–1676, 2004.

A Calculation of $P_{IF}(n)$

To illustrate how $P_{IF}(n)$ is calculated let us consider a three state system. Let the initial state be 1 and the final state 3. The transition matrix for the system is given as:

$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

First we form the matrices $\tilde{T}$, L, o and v:

$$\tilde{T} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad L = \begin{bmatrix} a_{31} & a_{32} \end{bmatrix}, \quad o = \begin{bmatrix} 1 \end{bmatrix} \quad v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

For $n = 1$ we have:

$$P_{13}(1) = oL\tilde{T}^0 v = a_{31}$$

Since $P_{13}(1)$ is the probability to go from state 1 to state 3 in one step there is only one possible path, 1-3. The probability for this is simply a_{31}. For $n = 2$ we have:

$$P_{13}(2) = oL\tilde{T}^1 v = a_{31}a_{11} + a_{32}a_{21}$$

There are two possible paths 1-1-3 and 1-2-3. The probability for each of these is $a_{31}a_{11}$ and $a_{32}a_{21}$ respectively. The sum, therefore, gives the total probability. For $n = 3$ we have:

$$P_{13}(3) = oL\tilde{T}^2 v = a_{31}a_{11}a_{11} + a_{31}a_{12}a_{21} + a_{32}a_{21}a_{11} + a_{32}a_{22}a_{21}$$

In this case there are four possible paths from state 1 to 3. These are 1-1-1-3, 1-2-1-3, 1-1-2-3 and 1-2-2-3. $P_{13}(3)$ is the sum of the probabilities for each of these paths.