A bayesian multibody multinomial model as a rapid method of measuring protein stability

A bioinformatical study in protein stability prediction

Here I explore a method for rapid prediction of protein stability upon change of residue composition. A bayesian probabilistic knowledge-based state of the art multibody multinomial model, MuMu, was investigated as a mean of predicting protein structure stability of mutated proteins in real-time. By using a readily available method for predictions of protein side-chains it was possible to make a large number of mutations on the protein 1STN. Probabilistic structure stability of mutations were calculated by an implementation of MuMu in a high-troughput python script. Probabilistic data was statistically compared to experimental ΔΔG values by means of correlation by the Pearson product-moment correlation coefficient. It was revealed that the product of the MuMu model is moderately correlated with experimental ΔΔG values. The MuMu model may in fact be used as a method for rapid prediction of protein stability and it is hypothesized that it could be a tool for predicting sites for mutations for protein stabilization in the field of protein design.

Introduction

Protein stability is a balance of forces, which determines if a protein will be in a native folded state, N, or a denatured unfolded state, U. The stability of a folded protein is often measured as the thermodynamic conformational stability and is measured as the change of Gibbs free energy, ΔG (Murphy, Kenneth P., 2001). Measures of free energy are based on the equilibrium between protein folding states that in turn is based on the ratio of the forward and the reverse rate constant of a fold (Park, C. et al., 2004). A negative ΔG is observed, if the folded protein is more stable than the unfolded protein. Gibbs free energy is based on observations from experimental data that are obtained via different methods, e.g. absorbance, fluorescence, circular dichroism, NMR, activity assays, etc. The actual structure of a stable folded protein is often determined by X-ray crystallography (Murphy, Kenneth P., 2001). All above mentioned methods require time-consuming laboratory work.

The growing field of protein stability prediction have resulted in an expansion of computer-based models for predicting energy of proteins. Protein structure prediction models essentially seek to minimize the energy to get the most favorable structure. Such models tries to approach the ideal structure offered by x-ray crystallography (Qian, B. et al., 2007). However these models face problems that sacrifices accuracy for speed.

An obstacle is the search through the three-dimensional conformational space in a protein. A high resolution complicates the task of prediction, as the task is growing exponentially with the resolution (Voigt et al., 2000). In other words; the more factors, contacts, that are taken into account during protein prediction, the higher the number of degrees of freedom is.

A widely used simplification of three-dimensional conformational space is found by looking only at the side chain formations of residues (rotamers). This minimizes the number of degrees of freedom but degreases accuracy (Dunbrack, 2002).

Structure prediction methods can be grouped into two groups: First, most commonly used are physical-based potentials (PBP) that are based on statistical data of the fundamental forces between atoms in proteins (Potapov, 2009). Second are knowledge-based potentials (KBP), based on probabilistic data derived from protein databases.

For small changes in proteins, e.g. mutations, structure prediction can be used for predicting the stability of the mutated protein. This has greatly used in protein design including enzymes for specific catalytic properties. For practical purposes it has been shown that active catalysts can be made from computational designed protein models (David Baker 2010).

For medical purposes it has been shown that a *de novo* design of a protease inhibitor have successfully been able to target HIV-1 for HIV treatment (Bellows, et. al. 2010).

The major drawback of protein structure prediction is that it is often slow and takes a lot of computer-power if you want to capture the complete protein with all its bodies (Qian, B. et al., 2007). However, studies in alternative methods have shown that it is possible to formulate a KBP probabilistic model that does not need a rotamer library (Harder T, et. al. 2010) and recently it has been shown that it is possible to formulate a model for higher order interactions, that does not suffer from an arbitrary upper limit of contacts (Johansson, K, et. al. 2013). A fast probabilistic model for a higher order of contacts may provide a method for rapid measure of protein stability. Here I explore the possibilities of utilizing such a method. First, I will briefly describe the general theory of a multibody multinomial model, MuMu formulated by Johansson, K, et. al. 2013.

MuMu

Consider an amino acid, A_i, at position i, in a given sequence. It will be conditional on both the structure, X, and the all other amino acids, $A_{\sim i}$, in the sequence and have the probability distribution: $P(A_i \mid A_{\sim i}, X)$. A joint distribution for the entire sequence will be the product of all conditional probabilities of each amino acid A_i in the sequence A:

$$P(\mathrm{A} \mid \mathrm{X}) \approx \prod_i P(A_i \mid A_{\sim i}, X).$$

The conditions, $A_{\sim i}, X$, describe the environment, E_i, of the amino acid, A_i. The environment is a composition of biomolecular forces that have an impact on the probability of a given amino acid. The environment is represented by the first shell of neighbors, C_i, the total neighbor count, N_i, the visible volume, V_i, and the backbone angles Φ_i and Ψ_i. Not surprisingly, the first shell of neighbors is conditional on the total neighbor count. By the product rule we can obtain the joint probability, $P(A_i, E_i)$:

$$P(A_i \mid E_i) = \frac{P(A_i, E_i)}{P(E_i)}.$$

The joint probability can expanded by introducing a hidden variable, *H:*

$$P(A_i, C_i, N_i, V_i, \Phi_i, \Psi_i)$$
$$= \sum_{h \in H} P(A_i \mid h) P(C_i \mid N_i, h) P(N_i \mid h) P(V_i \mid h) P(\Phi_i, \Psi_i \mid h) P(h)$$

The probability distribution formulated here is the MuMu model that is a Bayesian network, figure 1. The model is readily available in Phaistos, a framework for Monte Carlo simulations of protein structure (Boomsma, W. et. al. 2013).

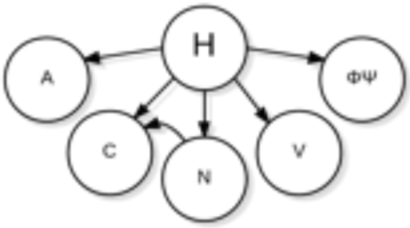

Figure 1. A Bayesian Network representation of the MuMu model. All observed variables are conditioned on the hidden parent, H.

The MuMu model has been trained on high resolution protein structures exceeding 3.8 Å all of which are not membrane proteins, disordered, small, or protein complexes. The model is therefore most accurate for proteins that does not contain any of those features.

I present a method for using a multibody multinomial model, *MuMu*, for rapid prediction of protein stability upon the introduction of one or more mutations, that could potentially be used for real-time applications. The data will be compared to experimental ΔΔG values as a mean of validating the robustness of the MuMu model.

Materials & method

The idea of using the MuMu model for a rapid method of predicting protein stability is based on the model's ability to obtain the probabilities of each amino acid in a protein, $P(A_i \mid A_{\sim i}, X)$, in a small amount of time. The output of MuMu from the Phaistos framework, upon receiving a pdb files, is a logarithmic value of the probability for each amino acid in the protein sequence. The logarithmic joint probability can be expressed as the sum of all logarithmic conditional probabilities:

$$\log(P(\mathrm{A} \mid \mathrm{X})) \approx \sum_i \log(P(A_i \mid A_{\sim i}, X)).$$

When a protein is subjected to mutagenesis, a new joint probability can be calculated. The change in joint probability will indicate the stability of a mutation:

$$\Delta \log(P(\mathrm{A} \mid \mathrm{X})) =$$
$$-\log(P(\mathrm{A}_{mut} \mid \mathrm{X}_{mut})) + \log(P(\mathrm{A}_{WT} \mid \mathrm{X}_{WT}))$$

This allows for a stability analysis that could potentially be used on thousands of protein structures in relatively short time.

The main challenge is to obtain target structures.

A set of 95 experimentally obtained ΔΔG values from mutated versions of a the wild-type protein 1STN was obtained from Meeker, K, et. al. 1996. The reference protein structure 1STN was downloaded from the protein data bank (http://www.rcsb.org/).

Mutagenesis

95 pdb files with mutations corresponding to mutations from Meeker, K, et. al. 1996 were constructed with a high-throughput python script that uses Scwrl4 for predicting side-chains of proteins by using data from a library of rotamers (Krivov, G. G., 2009).

The python script for handling mutations, named box_mutations.py, (see appendix) is designed to take a list as an input with the wild type sequence and wanted mutations in a column. A file containing the wild type amino acids in small letters and the mutation as a capital letter is generated for each mutant. These files along with the wild type file are then sent to Scwrl4. Scwrl4 recognizes the capital letter as a mutation and matches it to the protein structure. The new side-chain is predicted and a new protein structure file for each mutation is generated and outputted with its appropriate identifiable name.

Probabilities

All 95 mutant structures and the wild type protein structure were processed by the python script box_mumu.py (see appendix). The script works as a high-throughput method for piping data into MuMu from the Phaistos package. The output from the MuMu analysis is a series of logarithmic probabilities for each amino acid. All probabilities are summed for each protein and a difference in logarithmic probability of mutants compared to the wild type protein, $\Delta \log(P(\mathrm{A} \mid \mathrm{X}))$, is calculated.

All data are outputted as a data file for analysis.

Correlation

The data output was analyzed by using R statistics. A simple plot between the experimental ΔΔG values and $\Delta \log(P(\mathrm{A} \mid \mathrm{X}))$ was made. To test the correlation between the samples, a Pearson product-moment correlation coefficient was calculated. The coefficient was validated by a test of significance.

Results

The correlation between the experimental ΔΔG values and $\Delta \log(P(\mathrm{A} \mid \mathrm{X}))$ is depicted as a scatter plot, figure 2.

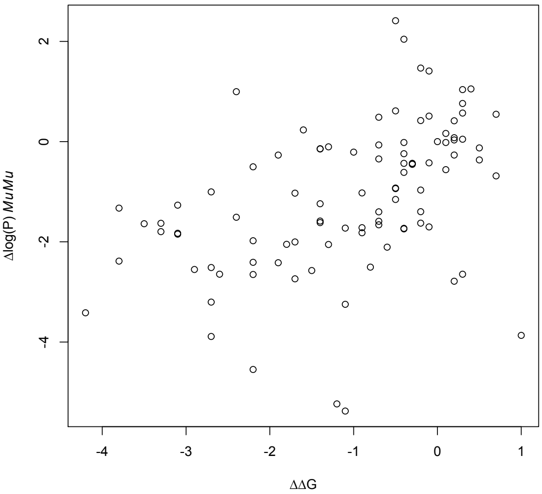

Figure 2. Data is not perfectly centered around a diagonal line. y-axis shows the change in lcogaritmich probability upon mutation, Δlog(P(A | X)). x-axis shows the change in change of free energy upon mutation.

It seems as though there is a correlation between ΔΔG values and $\Delta \log(P(\mathrm{A} \mid \mathrm{X}))$, however it is not completely clear. It looks like there is a tendency for the data points to follow a positive diagonal line, which indicates a positive correlation. The values of $\Delta \log(P(\mathrm{A} \mid \mathrm{X}))$ are negative for stabilizing mutations and positive for destabilizing mutations. In accordance, the more negative the ΔΔG values are, the more stable the mutation is and vice versa. The positive slope of the data points follow this definition. However, the data points spread indicates some roughness. To further analyze the relationship between $\Delta \log(P(\mathrm{A} \mid \mathrm{X}))$ and ΔΔG, a Pearson product-moment correlation coefficient was calculated with the program R statistics, table1.

	log(P) ***MuMu***	**ΔΔG**
log(P) ***MuMu***	1.0	0.45
ΔΔG	0.45	1.0

Table 1. Pearson correlation.

n = 95, df = 93	**log(P)** ***MuMu***	**ΔΔG**
log(P) ***MuMu***	0.0	0.0001
ΔΔG	0.0001	0.0

Table 2. Pearson correlation test.

The Pearson correlation shows that there is a correlation between $\Delta \log(P(\mathrm{A}\,|\,\mathrm{X}))$ and ΔΔG of r = 0.45, table 1. The hypothesis for the Pearson correlation test is defined as:

A null hypothesis

H(0): p>0.001: *there is no relationship between the two samples.*

Alternative hypothesis

H(a): p<0.001: *there is a relationship between the samples.*

The low p-value of 0.0001, table 2, indicates that there is ample evidence to reject the null hypothesis at a 0.1% confidence interval. However the value of 0.45 only indicates a modest correlation (Taylor, R. 1990).

Conclusion

The MuMu model offers a fast method for analyzing probabilities of residues within proteins. All probabilities were obtained within seconds of program execution. It has been showed here that MuMu can be easily applied as a model in a high-throughput method for rapid protein stability prediction.

The probabilistic stability change of 95 mutated versions of 1STN were compared to 95 experimentally obtained ΔΔGs. The comparison is visualized as a plot, figure 2, and displays that the two datasets tend to correlate. This was further explored by Pearson correlation that showed a correlation of 0.45. A 0.1% significance level was chosen objectively based on the amount of samples. The result indicates that there is a moderate correlation (Taylor, R. 1990).

The reference data were chosen based on the complete list of ΔΔG values that made an excellent basis for comparison. Only one protein was used as reference for the purpose of this project. This provides a one-sided approach that requires more validation. To further validate the results obtained here, it would be desirable to apply the method on a broader spectrum of proteins.

The approach of using the MuMu model as a high-throughput method gives the possibility for more advanced studies of protein stability. Potentially MuMu could be used as a real-time application, for finding residues where a mutation would be beneficial for the general stability of a protein. As an example; all sites of a protein could be mutated to an alanine and an increase in stability would be based on the maximization of $\Delta \log(P(\mathrm{A}\,|\,\mathrm{X}))$. Potentially all positions of a protein could be mutated with all types of residues. This offers a method that could be used for protein design applications.

Learning

With a bachelors degree in biochemistry, my goal has not been to further my biochemical understanding. However, my focus has been on using my newly acquired knowledge in the field of bioinformatics. In this project I have expanded my skills in python programming. I have been able to successfully write scripts for piping data into MuMu and Scwrl4 and correctly handle the output. I have learned to use knowledge-based methods as a way of describing protein stability. I have understood the purpose of a Bayesian network and I have generally evolved my skills in working with the shell of an unix-based system.

I believe I will apply some of these newly learned skills on future projects.

Literature

Boomsma, W., Frellsen, J., Harder, T., Bottaro, S., Johansson, K. E., Tian, P., … Hamelryck, T. (2013). PHAISTOS: a framework for Markov chain Monte Carlo simulation and inference of protein structure. Journal of Computational Chemistry, 34(19), 1697–705. doi:10.1002/jcc.23292

Potapov, V., Cohen, M., & Schreiber, G. (2009). Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. Protein Engineering, Design & Selection : PEDS, 22(9), 553–60. doi:10.1093/protein/gzp030

Murphy, Kenneth P. Protein Structure, Stability, and Folding. Totowa, NJ: Human, 2001. Print.

Park, C., & Marqusee, S. (2004). Analysis of the stability of multimeric proteins by effective ⌬, 2553–2558. doi:10.1110/ps.04811004.1

Roland L. Dunbrack Jr, Chase, F., & Avenue, B. (2002). Rotamer libraries in the 21 st century, 431–440.

Voigt, C. a, Gordon, D. B., & Mayo, S. L. (2000). Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. Journal of Molecular Biology, 299(3), 789–803. doi:10.1006/jmbi.2000.3758

Qian, B., Raman, S., Das, R., Bradley, P., McCoy, A. J., Read, R. J., & Baker, D. (2007). High-resolution structure prediction and the crystallographic phase problem. Nature, 450(7167), 259–64. doi:10.1038/nature06249

Krivov, G. G., Shapovalov, M. V, & Dunbrack, R. L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. Proteins, 77(4), 778–95. doi:10.1002/prot.22488

Harder, T., Boomsma, W., Paluszewski, M., Frellsen, J., Johansson, K. E., & Hamelryck, T. (2010). Beyond rotamers: a generative, probabilistic model of side chains in proteins. BMC Bioinformatics, 11, 306. doi:10.1186/1471-2105-11-306

Johansson, K. E., & Hamelryck, T. (2013). A simple probabilistic model of multibody interactions in proteins. Proteins, 81(8), 1340–50. doi:10.1002/prot.24277

Meeker, a K., Garcia-Moreno, B., & Shortle, D. (1996). Contributions of the ionizable amino acids to the stability of staphylococcal nuclease. Biochemistry, 35(20), 6443–9. doi:10.1021/bi960171

Taylor, R. (1990). Interpretation of the Correlation Coefficient: A Basic Review. Journal of Diagnostic Medical Sonography, 6(1), 35–39. doi:10.1177/875647939000600106

Links & programs

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

The protein data bank: http://www.rcsb.org/

Special thanks

My supervisor Thomas Hammelryck for supervising.
Kristoffer Johansson for providing articles and guidance.
Wouter Boomsma for hours spent setting me up with Phaistos.

Appendix

All files associated this project are delivered as an external file.