

Inverse Protein Folding of Ubiquitin on the Illinois Bio-Grid

Jonathan Gemmell¹, David Sigfredo Angulo¹, Tobin Sosnick², Karl Freed²,
Abhishek Jha², Andrés Colubri², Joseph DeBartolo², David Kendall², Gregor von Laszewski³

¹DePaul University
School of Computer Science, Telecommunications and Information Systems
Chicago, IL

²The University of Chicago
Department of Biochemistry and Molecular Biology, Department of Chemistry
Chicago, IL

³Argonne National Laboratory
Mathematics and Computer Science Division
Argonne, IL

jonathan.gemmell@gmail.com

Abstract

The Inverse Protein Folding problem, also called Protein Design, spans the boundaries of both the computer and biological sciences. The problem consists of determining a sequence of amino acids to compose a protein that will, due to their combined bio-chemical properties, fold into a predetermined three-dimensional structure. This three-dimensional structure, or conformation, determines the effect of the protein on its environment. Hence, success in the Inverse Protein Folding problem would allow scientists the ability to redesign existing proteins with additional or enhanced functionality or even to design new proteins with novel functionality. Nevertheless, while the impact of a solution to the Inverse Protein Folding is apparent, the computational challenges are extraordinary, since the search space explodes exponentially as the length of the protein increases. We demonstrate a Monte-Carlo model for sampling the search space coupled with energy functions for evaluating conformations. These energy functions are based on statistical propensities mined from the Protein Data Bank. We further demonstrate a framework based on the Illinois Bio-Grid Toolkit which utilizes grid technologies to give the capabilities of engaging massive amounts of computational power to the problem.

1 Introduction

The ramifications of a time efficient method able to solve the Inverse Protein Folding problem would be extraordinary, possibly resulting in significant medical advancements and a deeper understanding of molecular biology. Many recent papers have shown encouraging results [1] [8]. However, the computational time required to solve even the most modest of problems is vast and

has been proven to be NP-Hard [3] [14] [2].

The sequence of amino acids in a protein is called its primary structure. Architectural substructures like alpha-helices and beta-sheets are commonly referred to as secondary structures, while the global conformation is known as tertiary structure. Proteins with identical primary sequences will assume identical three-dimensional structures within the cell. This fact cannot be overstated, as it demonstrates that it is the sequence of amino acids and their collective bio-chemical properties that determine the stable three-dimensional conformation of a protein. [2].

The goal of Inverse Protein Folding is to determine a sequence of amino acids that would due to their collective bio-chemical properties, fold into the proposed three-dimensional conformation. A sequence of amino acids composing a protein is rarely as short as 50 amino acids, and more often about 300 amino acids long. Further, there are 20 distinct amino acids that may exist at any position in the amino acid chain. Hence, for even the smallest sequences, a brute force search examining every possible sequence of amino acids would require the evaluation of 20^{50} sequences, larger proteins requiring 20^{300} evaluations. A thorough brute force search is clearly computationally infeasible. Any advancement in the Inverse Problem Folding problem will likely contain computer models utilizing biologically relevant data and advanced search space algorithms [4].

2 Previous Work

Among the techniques that have been previously considered are algorithmic approaches that randomly sample the search space. While not examining the entire search space these approaches may find a solution that is sufficient, even if it is not optimal. Other algorithmic approaches may be deterministic, cutting out large swathes of the search space that can be proven not to contain the ideal solution. Such approaches however, require an exponential running time. Still other options include coupling one of the above approaches with the computational power of a computer cluster or even a computational grid.

Monte-Carlo simulations [12] are a powerful method used to examine search spaces that might be too large to thoroughly examine. Many researchers have experimented with this technique. Early experiments were performed by Hellinga and Richards [28], in which a limited number of amino acids in the protein were marked for mutation. However, the approach was limited to small proteins or subunits of proteins. Further, their experiments were used, not to design a protein, but instead to converge on the wild type sequence.

Klepeis and Floudas [1] have also experimented with the Monte-Carlo approach. Their approach is interesting due to their procedure of first creating thousands of sequences through Monte-Carlo experiments and then using Protein Folding simulations to choose the most appropriate sequence. They met with success in designing a therapeutic peptide, Compstatin, used to moderate the auto-immune mediated damage of organs during transplantation and in various inflammatory diseases. The redesigned peptide was seven times more efficacious and stable than the original peptide. Still, the method was limited to the smallest proteins, constrained by the computationally expensive Protein Folding element.

Genetic algorithms are another non-deterministic method worth considering [5] [13] [18]. The first attempts to design proteins using Genetic algorithms were conducted by Jones [6] [21]; however, none of the designed proteins were ever synthesized. Other experiments were

conducted by Desjarlais and Handle [6] [19] [20]. Their approach focused on the repacking of the hydrophobic core and results suggested that the core amino acids are likely responsible for the overall stability of the protein while the non-core amino acids play a significant role in determining the tertiary structure. The results were encouraging, demonstrating that it was possible to repack the core alone with novel sequences. Unfortunately, the potential sequences were limited by the small rotamer library used in the experiment.

While Genetic algorithms remain a powerful tool to search large computational terrains, it has been noted that they are insufficient to find an optimum solution [22]. Desjarlais, himself, argues [23] that non-deterministic algorithms such as Dead End Elimination are the superior method for finding the global minimum energy. Further, it has been suggested that Genetic algorithms are problematic. Protein models are highly coupled entities, and the crossover disruption that occurs as a subsequence of amino acids is replaced is fundamentally inevitable, creating major disturbances in the viability of the structure [19].

Branch and Bound algorithms are effective in discovering the best solution by cutting out large branches of the search space and thus eliminating large amounts of computational work [24]. Wernisch, Hery, and Wodak [25] [26] [27] have successfully repacked the cores of several proteins, including the c-CRK SH3 domain and the B1 domain of protein G and Ubiquitin. Further, they have designed entire sequences of small proteins. However, even with the computational power of a small 24 node cluster, they have only been able to design proteins of about 60 amino acids in length. Still this is a great accomplishment, and further illustrates the computational challenge and the usefulness of cluster or grid technologies.

Similar to Branch and Bound is the Dead End Elimination Theorem [5] [6]. It too, tries to cut computational time by eliminating portions of the search space from consideration. If the problem being considered is combinatoric in nature then the subunits can be considered individually. Moreover, if particular subunits can be proven not to belong to the ideal solution, they can be removed from consideration all together. Dead End Elimination was originally used by Desmet, Maeyer, Hazes and Lasters [9] to solve for side chain optimization, selecting particularly favorable side chain positions, or rotamers, and eliminating unlikely ones.

Mayo and his team adapted the Dead End Elimination technique into the sequence search itself [5]. Their technique involves first dividing the conformation into architectural sub-units: core, boundary, and surface and solving for each subunit individually thus drastically reducing the running time [4]. Consider a protein with two hundred amino acids. Using a brute force method, 20^{200} conformations would need to be examined. However, if the protein is divided into three subunits with sizes fifty, seventy and eighty amino acids, then the total number of conformations to be examined is drastically reduced to $20^{50} + 20^{70} + 20^{80}$. Moreover, Mayo and his team divided the search for the backbone sequence and side chains positioning into separate problems. The first versions were slow and limited to proteins about 80 amino acids long, but further advancements mostly based on heuristics allow for the design of proteins up to 200 amino acids long [5] [7] [8] [10] [11].

Even with an excellent searching algorithm the sheer size of the search space is formidable. Techniques in grid computing [15] [16] allow the programmer to spread the computational load across a myriad of computer resources. At its simplest, the work can be spread across a few connected computers in the same room that compose a cluster, all with identical hardware and operating systems. At its most complex, the work can be spread across hundreds of thousands of processors on different continents using a wide variety of hardware and operating systems.

Recent evidence shows that the reign of Moore's law, which states that computational power of new processors increases two-fold every eighteen months, is coming to an end [17]. Hence grid technologies may be the only realistic method offering the means to tackle the heavy computation required by the Inverse Protein Folding problem in a reasonable time.

Any strategy for finding a solution to the Inverse Protein Folding problem must take into account the computational requirements as well the relevant biological information. Non-deterministic methods can be employed to search for a sequence of amino acids that would fold into the sought after three-dimensional shape, but while it may find a plausible solution, it may never find the most stable sequence. Deterministic methods may be used, but even after limiting the search space the computational time required is still quite large. The biological representation can be broken down into less exact but more manageable units, perhaps by representing only the backbone of the protein and not the side chains themselves, or perhaps by breaking the protein in separate distinct subunits and solving for each subunit separately. But the loss of information may do more to hamper the search for a solution than it does to help. Further, even the most efficient solution will remain computationally expensive and will likely benefit from distributed computing. The proposed method relies on the lessons learned from these previous efforts.

3 Proposed Method

This paper proposes an alternative method for finding a solution to the Inverse Protein Folding problem, borrowing certain procedures from previous efforts while introducing new strategies. Given a three-dimensional conformation, a set of probabilities is generated for each position based on the likelihood of the twenty amino acids to fill that position in the sequence. This set of probabilities is then used when selecting amino acids for the sequence during a Monte-Carlo simulation. Initially an entire sequence is generated, randomly assigning amino acids to positions in the sequence based upon the probabilities previously determined. Then, at each step of the simulation the sequence is altered. At every step, the sequence is evaluated by energy functions, scoring the likelihood that the sequence would fold into the desired three-dimensional conformation. This process is repeated hundreds of times, each time creating a new potential sequence. Finally, after hundreds of simulations, the potential sequences are ranked, and a small subset is selected for further evaluation. Each step of the process is described in greater detail below.

3.1 Computing Amino Acid Probabilities

The first step of the procedure is to compute, using the input structure and a library of known structures, the probabilities for particular amino acids to be present at certain positions in the sequence. To this end, a series of variables are calculated for each position in the amino acid chain. These variables are the phi and psi angles and an environmental variable. Phi and psi angles are common throughout protein science. They are the dihedral angles around which the protein's backbone is able to rotate. Tertiary structure can be described by merely the phi and psi angles for each amino acid.

The environmental variable proposed for this strategy is the number of beta carbons located within a certain radius of the amino acid being examined and is used to quantify the depth of the amino acid in the protein. It is common for proteins to have a core of hydrophobic amino acids,

repelled from water, while the exterior amino acids are hydrophilic and attracted to water. For each position in the sequence, the phi and psi angles and environmental variable as well as those of the neighboring amino acids are combined into a 9-tuple. They are compared to every amino acid in a library of known structures. If an amino acid is found in the library matching all nine variables, within a certain degree of error, it is recorded as a potential amino acid for that position. When every amino acid in the structural library has been compared to every 9-tuple, a file is generated of amino acid probabilities for each position in the amino acid chain.

3.2 Energy Functions

Two energy functions were used to evaluate potential sequences. The first is DOPER (Discrete Optimized Energy, reduced), a backbone only energy function evaluating the statistical propensities for amino acids to be particular distances from one another. It is a reduced form of DOPE, an all atom energy function [31]. The second energy function used was IBG_Probabilities, an energy function based on the probabilities discussed above.

For DOPER, prior to running a simulation a set of statistics is generated by mining data from the PDB library [29]. For every atom in every amino acid a series of scores is generated based on the observed occurrences in nature for atoms of one amino acid to be a particular distance from the atoms of another amino acid. This information is of great benefit when designing a protein sequence, since it permits the scoring of a sequence based on whether nature has produced similar occurrences of amino acids in relation to one another. During the evaluation of a potential sequence, every atom in the sequence must be compared to every other atom in the sequence. Since, in this reduced implementation, every amino acid has five atoms in its backbone, a carbon, nitrogen, oxygen, alpha carbon, and beta carbon, a sequence with n amino acids will have to make about $(5n)^2$ comparisons.

Similarly, IBG_Probabilities uses data mined from the PDB. The probabilities generated for the amino acid as described above are used to score the sequence based on the likelihood of an amino acid to take a particular position in the sequence. Positions with highly probable amino acids are scored better than positions with less likely amino acids. The result of IBG_Probabilities is the sum of the scores for every position over the entire sequence.

3.3 Monte-Carlo Routine

The next step in the proposed strategy is to select an initial sequence on which the simulation will build. There exist a few possibilities for selecting the initial sequence. When redesigning a known protein, it is possible to reuse the amino acid sequence known to conform to the given three-dimensional shape. However, this may not be appropriate when searching for unique sequences not related to the known sequence. Moreover, if a unique three-dimensional structure is being designed, no such sequence would exist. Further, since hundreds or even thousands of simulations are run for each experiment, the value of beginning each simulation with the same sequence is questionable, if the goal is to examine the search space as thoroughly as possible.

Another possibility is to use the probabilities generated for the amino acids, selecting for each position the most likely amino acid. Hence the best amino acid, based on the trimer statistics is selected for each position. But again, beginning every simulation with the same sequence is inappropriate if the goal is to examine a broad swatch of the search space.

A third possibility is to generate a sequence by randomly choosing amino acids weighted by the amino acid probabilities generated. While particular circumstances may require one of the above methods, in general this method seems most appropriate, since every simulation will begin with a unique sequence, more broadly evaluating the computational terrain. Further using the calculated probabilities rather than merely a purely random sequence will weigh the sequence to more likely areas of the search space.

As in any Monte-Carlo simulation, modifying and reevaluating the model is of significant importance. At each step in the proposed simulation, a random position on the amino acid chain is chosen to be modified. Using the probabilities for the amino acid sequence, a new amino acid is chosen for that position at random. The new sequence is then applied to the structure and scored via the weighted sum of the DOPER and IBG_Probabilities energy functions. The score of the original structure is compared to the score of the structure after modification. If the new sequence is ranked better than the original the modification is accepted and the algorithm continues on to the next step. However, if the new sequence is ranked poorer than the original, the modification is reversed and the original sequence is passed on to the next step in the simulation. Thus every newly accepted sequence is an improvement on the last. Eventually, the sequence will converge to one that can no longer be improved, by a single point mutation.

Other options include modifying more than one amino acid per step, perhaps examining a larger portion of the search space, if less thoroughly. Further, it is possible to begin by altering many amino acids per step, and as the simulation runs, the number of amino acids being changed per turn is reduced, until as the simulation converges the number of amino acids being modified drops to one in order to find the local minima that cannot be improved by a single point mutation.

Not only is the computational terrain being examined vast, but there exists a large number of deep local energy minima. Consequently, once a sequence is discovered in a local minimum, it is difficult for the simulation to escape this minimum and search other portions of the terrain. To overcome this, the metropolis technique is used. As stated before when the sequence is perturbed such that a new sequence is created with a better energy score than the previous sequence, this new sequence is maintained and used in the next step of the simulation. However, with the metropolis technique, if the new sequence has a worse score, there remains a possibility that the new sequence will still be retained. A random number is generated and if it is less than a predetermined limit, the new sequence is accepted regardless of its score. Consequently, if the simulation is stuck in a local minimum, the metropolis technique provides a means by which the simulation can escape, to examine other portions of the terrain.

3.4 Selecting Winners

The algorithm proposed merely samples the search space, opting to discover hundreds or even thousands of local minima rather than exhaustively searching for the global minimum. Consequently, at the end of the simulation, a large number of potential sequences will have been generated. These sequences are all potential primary amino acids structures for the given three-dimensional structure. They may or may not fold into the desired structure. While all these sequences represent the best discovered sequence for a particular portion of the search space, when compared among themselves, it is likely that certain sequences will, according to the DOPER and IBG_Probabilities energy functions, be more likely to conform to the proposed

tertiary structure. Consequently, after an experiment containing many simulations, the results are ranked and only the best sequences are evaluated further as potential sequences for the given structure.

4 Experiments and Results

The experiment created 200 amino acid sequences after running the Monte-Carlo experiment for 20,000 iterations for each sequence. One amino acid was modified per iteration and the metropolis variable was set to 2%. The IBG_Toolkit was used to conduct the experiment. The top ten sequences were evaluated further.

Conclusive evidence that the procedure generates amino acid sequences that will fold into the submitted three-dimensional structure can only be obtained by creating the sequence in the laboratory and experimentally determining the tertiary structure by either X-ray crystallography or NMR spectroscopy. Nevertheless, evidence does exist suggesting that the sequences being generated are relevant. For this experiment, the top ten sequences, based upon their combined DOPER and IBG_Probabilities score, were submitted to an independent secondary structure prediction program, NNPREDPREDICT [30], which takes as input a sequence of amino acids and predicts the secondary structure of the protein.

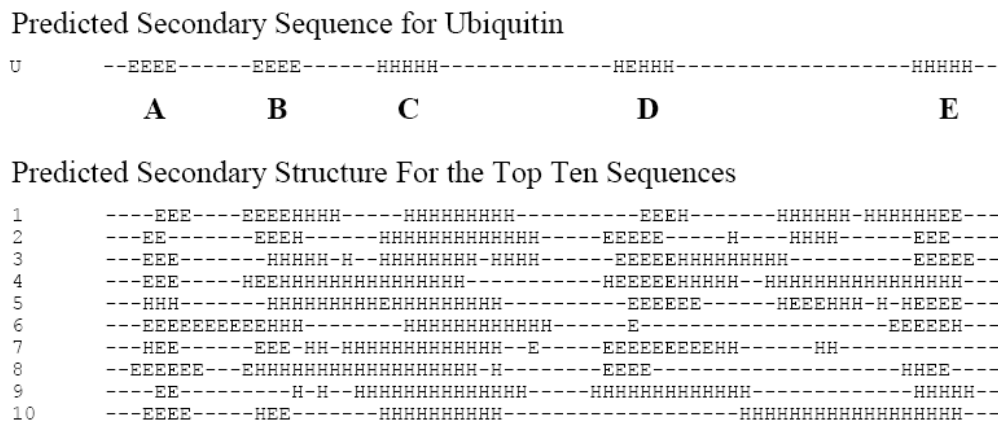


Figure 1. Secondary Structure Predictions

Notice that the secondary prediction for the original sequence of Ubiquitin has an extended segment from position 3 to 7, marked as A (Figure 1). Likewise, nine of the proposed sequences were predicted by NNPREDPREDICT to have an extended secondary structure in this area. The next secondary structure element predicted was another extended region, labeled as B. Half of the proposed sequences were suggested to have an extended secondary structure in this area and many of the predictions suggest the similar helical structure. The portion of the predicted secondary structure for Ubiquitin labeled C is a helix. All of the top ten sequences were predicted by NNPREDPREDICT to also have a helix in this area. They are of different sizes, but all correspond very well to this area. Further, the fourth secondary structure predicted by NNPREDPREDICT for the Ubiquitin sequence, marked by D, is an extend helix. Nine if the top ten sequences show a mix of helices and extensions. Finally, the last secondary structure predicted, marked E, is a helix. Five of the proposed sequences ranked in the top ten, place a helix in this

area.

In general, many of the substructures present in the proposed three-dimensional conformation are predicted to be in the generated sequences. A few of the structures like the extended region A and the helix region C were present in nine out of ten sequences. Other portions of Ubiquitin's secondary structure were found in the secondary structure predictions with fewer occurrences. The worst results were found when designing the final helix region E, when five of the top ten sequences were found to contain a similar helix. However, it does appear that portions of the proposed sequences are predicted to have structures not found in Ubiquitin, particularly in the large turn regions between regions C and D and regions D and E.

While this evidence can not conclusively prove that the proposed strategy generates sequences that will fold into the given three-dimensional conformation, it does suggest the approach will result in sequences with similarity in secondary structure and hence possibly tertiary structure. Final proof can only be attained by creating the sequences in a laboratory.

5 Conclusions

Much of the previous work discussed above use energy functions requiring complex bio-physical calculations. On the other hand, the DOPER and IBG_Probabilities calculations are based only on the statistical probabilities mined from the PDB. The running time is quite small in comparison. For a completely new sequence the running time is $O(n^2)$ since it must do an atom to atom comparison. Yet for a sequence modified by only one amino the running time drops to $O(n)$ since the only calculations that need to be recomputed are for the atoms in the modified amino acid in relation to the other atoms in the sequence. Similarly, this strategy is true for the IBG_Probabilities energy function in which the running time is $O(n)$ for a completely new sequence and $O(c)$ for a modified sequence. The speed of the energy function coupled with the reduced model of solving the Inverse Protein Folding problem on the side chain-only representation further reduces the running time when compared to other efforts.

Compared to other strategies this approach is able to examine far more sequences in the same period of time. Moreover, the use a non-deterministic model, while not exhaustively searching the entire computational terrain, randomly samples the search space for potential solutions. While there is no guarantee that the global minimum will be discovered, at the very least many local minima will be found. Further it should be noticed that the Inverse Protein Folding problem may have many solutions, since multiple sequences can fold into identical conformations. Therefore, a local minimum may be sufficient. Furthermore, the amino acid probabilities generated for the sequence help to steer the Monte-Carlo simulation to sequences more likely to hold a valid solution.

In sum, despite the reduced model, the selection of a statistically based energy function over the computational expensive molecular dynamic algorithms and the use of a nondeterministic approach, the initial results are intriguing. The first tests have resulted in sequences that independent programs predict to have secondary structures quite similar to the proposed three-dimensional conformation. Most importantly these results have been achieved in hours, instead of the weeks or even months required by other approaches, suggesting that this strategy may be a realistic approach to tackle the Inverse Protein Folding problem in a time relevant to researchers

6 Future Work

Even though the strategy, as implemented in the IBG toolkit, has shown interesting results, a great deal of work remains to be done. Testing new algorithms that more accurately score protein conformations may improve the result of the program. Also, methods to more accurately predict the likelihood of an amino acid to fill a position in the main chain needs to be investigated. Since the IBG_Toolkit was designed with grid computing in mind, an experiment testing the program across grid architectures to determine the relative speed up needs to be completed. Moreover, experiments with alternative protein structures need to be satisfied, so that the validity of the strategy can be verified across multiple structures. Finally, conclusive proof for the strategy can only be attained by actually building the protein in the lab, and determining its three-dimensional conformation through experimental means.

Acknowledgements

This work was supported in part by the National Science Foundation under Grant No. 0353989.

References

- [1] J. L. Klepeis and C. A. Floudas, "In Silico Protein Design: A Combinatorial and Global Optimization Approach," *SIAM News*, Vol 31, num. 1, 2004 January/February
- [2] M. Miller "Levinthal's Paradox", University Chemical Laboratories, Retrieved 1/06 from brian.ch.cam.ac.uk/~mark/levinthal/levinthal.html
- [3] N. A. Pierce and E. Winfree, "Protein Design is NP-Hard," *Protein Engineering*, vol. 15, no. 10, 2002, pp. 779-782
- [4] D. B. Gordon, G. K. Hom, S. L. Mayo, and N. A. Pierce, "Exact Rotamer Optimization for Protein Design," *Wiley Periodicals, Inc*, 2002
- [5] Koehl, "Exploring Protein Sequence Space and Structure Space: A Review of Computational Search Techniques," Retrieved 06/2005 from www.cs.ucdavis.edu
- [6] W. F. DeGrado, "Protein Design: Enhanced proteins from scratch," *Science*, vol. 278, no. 5335, 1997 October, pp. 80-81
- [7] D. B. Gordon and S. L. Mayo, "Radical Performance Enhancements for Combinatorial Optimization Algorithms Based on the Dead End Elimination Theorem," *Journal of Computational Chemistry*, vol. 19, 1998, pp. 1505-14
- [8] F. Offredi, F. Dubail, P. Kischel, K. Sarinski, A. S. Stern, C. Van de Weerd, J. C. Hoch, C. Proserpi, J. M. Francois, S. L. Mayo, and J. A. Martial, "De Novo Backbone and Sequence Design of an Idealized alpha/beta-barrel Protein: Evidence of Stable Tertiary Structure," *Journal Molecular Biology*, vol 325, 2003, pp. 163-174
- [9] J. Desmet, M. D. Maeyer, B. Hazes and I. Lasters, "The Dead End Elimination theorem and its Uses in Protein Side-Chain Positioning," *Nature*, vol. 356, 1992, pp. 539-542
- [10] S. A. Marshall and S. L. Mayo, "Achieving Stability and Conformational Specificity in Designed Proteins via Binary Patterning," *Academic Press*, vol. 305, 2001, pp. 619-631
- [11] D. N. Bolson and S. L. Mayo, "Enzyme-like proteins by Computational Design," *PNAS*, vol. 98, no. 25, 2001, pp. 14274-14279
- [12] S. M. Ross, *Simulation, Third Edition*, 2002, Academic Press
- [13] G. Jones, "Genetic and Evolutionary Algorithms," University of Sheffield, UK, Retrieved

08/2005 from <http://www.wiley.co.uk/ecc/samples/sample10.pdf>

[14] P. Berman, B. DasGupta, D. Mubayi, R. Sloan, G. Turan, Y. Zhang, "The Protein Design Sequence Problem in Canonical Model on 2D and 3D Lattices," Retrieved 08/2005 from <http://www.math.uic.edu/~mubayi/papers/cpm-2004-finalversion.pdf>

[15] I. Foster, "The Grid: Computing Without Bounds," *Scientific America*, 2003, April

[16] M. Kaufmann, *the Grid: Blueprint for a new Computing Infrastructure, 2nd edition*, 2004, Morgan Kaufmann Publishers

[17] I. Toumi, "The Lives and Death of Moore's Law," *First Monday*, Retrieved 08/2005 from www.firstmonday.org/issues/issue7_11/toumi/

[18] D. T. Jones, "De Novo Protein Design Using Pairwise Potentials and a Genetic Algorithm," *Protein Science*, vol. 3, 1994, pp. 567-574

[19] J. R. Desjarlais and T. M. Handel, "De Novo design of the hydrophobic cores of proteins," *Protein Science*, vol. 4, 1995, pp. 2006-2018

[20] G. A. Lazar, J. R. Desjarlais and T. M. Handel, "De Novo design of the hydrophobic cores Ubiquitin," *Protein Science*, vol. 6, 1997, pp. 1167-1178

[21] D. T. Jones, "De Novo Protein Design of the Hydrophobic Cores of Proteins," *Protein Science*, vol. 3, 1994, pp. 567-574

[22] T. Hiroyasu, M. Miki, T. Iwahashi, and Y. Okamoto, "Dual Individual Distributed Genetic Algorithm for Minimizing the Energy of Protein Tertiary Structure," Retrieved 08/2005 from <http://mikilab.doshisha.ac.jp/dia/research/protein/thesis/src/2003/sice2003.pdf>

[23] J. R. Desjarlais and N. D. Clarke, "Computer Search Algorithms in Protein Modification and Design," *Current Opinion in Structural Biology*, vol. 8, 1998, pp. 471-475

[24] S. Russell and P. Norwig, *Artificial Intelligence, Second Edition*, 2003, Prentice Hall

[25] L. Wernisch, S. Hery and S. Wodak, "Automatic Protein Design with all Atom Force-Fields by Exact and Heuristic Optimization," *Journal of Molecular Biology*, vol 301, 2000, pp. 713-736

[26] Service de Conformation des Macromolécules Biologiques et de Bioinformatique, Retrieved 08/2005 from http://www.scmbb.ulb.ac.be/bioinformatics/sequence_design.html

[27] A. Jaramillo, L. Wernisch, S. Hery and S. Wodak, "Automatic Procedures for Protein Design," *Combinatorial Chemistry and High Throughput Screening*, vol 4, 2001, pp. 643-659

[28] H. W. Hellinga and F. M. Richards, "Optimal Sequence Selection in Proteins or known structures by Simulated Evolution," *Biochemistry*, Vol 91, 1994, pp. 5803-5807

[29] RSCB, PDB: Protein Data Bank, Retrieved 06/2005 from <http://www.rcsb.org/pdb>

[30] D. Kneller, NN-PREDICT: Protein Secondary Structure Prediction, Retrieved 06/2005 from www.cmpharm.ucsf.edu/~nomi/nnpredict.html

[31] A. Jha, A. Colubri, K. Freed, T. Sosnick, "Statistical coil model of the unfolded state: resolving the reconciliation problem," *PNAS*, vol. 102, 2005 pp.13099-104