Classifying protein-like sequences in arbitrary lattice protein models using LatPack

Martin Mann[†,*], Daniel Maticzka[†], Rhodri Saunders[‡] and Rolf Backofen[†]

[†]Bioinformatics, Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee 106, 79110 Freiburg, Germany

{mmann,maticzkd,backofen}@informatik.uni-freiburg.de

[‡]Department of Statistics, University of Oxford, 1 South Parks Road, Oxford, England, OX1 3TG

saunders@stats.ox.ac.uk

Abstract

Knowledge of a protein's 3-dimensional native structure is vital in determining its chemical properties and functionality. However, experimental methods to determine structure are very costly and time-consuming. Computational approaches, such as folding simulations and structure prediction algorithms, are quicker and cheaper but lack consistent accuracy. This currently restricts extensive computational studies to abstract protein models. It is thus essential that simplifications induced by the models do not negate scientific value. Key to this is the use of thoroughly defined *protein-like* sequences. In such cases abstract models can allow for the investigation of important biological questions.

Here we present a procedure to generate and classify *protein-like* sequence data sets. Our LatPack tools, and the approach in general, are applicable to arbitrary lattice protein models. Identification is based on thermodynamic and kinetic features. Further LatPack can incorporate the sequential assembly of proteins by addressing co-translational folding.

We demonstrate the approach in the widely used, unrestricted 3D-cubic HP-model. The resulting sequence set is the first large data set for this model exhibiting the protein-like properties required. Our data and tools are freely available and can be used to investigate protein-related problems. Furthermore our data sets can serve as the first benchmark sequence sets for folding algorithms that have traditionally only been tested on random sequences.

Introduction

Proteins have evolved to adopt a unique or very few functional *native* structures. In contrast, random amino acid sequences generally form non-functional *random coils*. This prompts one of the major biological questions: *"What are the features of proteins that enable the unerring folding into their functional native structures rather than just producing random coils?"*

To address such questions comparative studies of protein sequence and structure space are necessary to identify underlying properties. Due to extreme computational complexity and limited knowledge of aspects governing protein folding it is not currently feasible to investigate the folding process of real proteins via full simulations nor to calculate their native structure directly. Thus abstract protein models have been defined to focus on and elucidate certain features of proteins and protein folding.

By reducing complexity, protein models are computationally accessible but induce a major problem: *One has to identify protein-like sequences!* Real protein sequences are usually not applicable due to model restrictions in sequence/structure space or simplified energy functions. Thus a biological protein sequence is not guaranteed to show protein-like (*in vivo*) behavior when ported into the model.

Therefore, an *independent classification/definition of protein-like sequences* has to be calculated for each protein model! Identified protein-like sequences must posses a (unique) stable native structure and, more importantly, be able to fold to this structure within a short (biologically relevant) time interval. Thus thermodynamic and kinetic properties have to be used. Without such a data set the study of the initially formulated question is inhibited by the unvalidated data underlying it. Here we introduce such a classification scheme of protein-likeness, essential for computationally accessible, biologically relevant models!

Our procedure is applicable to widely used lattice protein models. These models restrict the placement of atoms to nodes of an underlying 2- or 3-dimensional lattice (e.g. 3D-cubic) and, usually, use just a few monomers to represent a single amino acid. For instance, the widely used HP-model (Lau and Dill, 1989) represents each amino acid with only a single monomer, which is (H)ydrophic or (P)olar, in the lattice. The HP energy function focuses on hydrophobic interactions (by maximizing HH-contacts) that are known to be a driving force in the folding process (Chan and Dill, 1990; Dill *et al.*,

*Corresponding author

1995). Though this yields a very rough protein representation, the associated problems, such as optimal structure prediction or sequence design, are still computationally demanding (NP-complete) (Garey and Johnson, 1990; Unger and Moult, 1993; Berger and Leighton, 1998; Berman *et al.*, 2004). Other models also often utilize a contact based energy function; but all levels of interaction detail can be found from the HP- to the HPNX-model (Wolfinger *et al.*, 2006) up to full 20 amino acid contact potentials provided by the Miyazawa-Jerningmatrix (MJ) (Miyazawa and Jernigan, 1996). All share the lattice discretization that allows for an enumeration of the whole structure space, obviously not applicable to real protein structures in free 3D-space. Thus exhaustive studies of folding pathways (Steinhöfel *et al.*, 2007), energy landscape features (Wolfinger *et al.*, 2006), general structural properties (Jacob and Unger, 2007) or protein design (Gupta *et al.*, 2005) and evolution (Irback and Troein, 2002) are applicable.

Our procedure can be applied to any of these models, independently from the used energy function or the underlying lattice. It is directly capable of work on any protein model due to the general applicability of the used LatPack tools (see methods or reference LatPack-home). Quintessentially, we use a three-step classification system. First, thermodynamic features are checked to ensure a stable native structure (Crippen and Chhajer, 2002). Next, the filtered sequences are tested if they can adopt their native structure in a short time interval. Thus a *good*/*bad* folder classification is achieved (Jacob and Unger, 2007). Only *good* folders are considered in the final step: sequential folding with the ability to only traverse low energy barriers (Huard *et al.*, 2006). This final step considers the sequential assembly of proteins and therefore the occurrence of co-translational folding during elongation or membrane transports (Fedorov and Baldwin, 1997; Kolb *et al.*, 2000; Deane *et al.*, 2007). Co-translational folding is assumed to restrict the accessible parts of the energy landscape during folding and hence to guide the process to the native structure (Levinthal, 1968; Govindarajan and Goldstein, 1998). The resulting *protein-like* sequences can thus be used to address the initial question.

Our approach is exemplified in the widely studied HP-model (Wolfinger *et al.*, 2004; Coluzza and Frenkel, 2007; Jacob and Unger, 2007). We use the LatPack-tools package (LatPack-home); a collection of programs and approaches to enable folding studies in the field of lattice proteins with arbitrary energy functions. The package is tailored to be as flexible as possible while ensuring high performance, essential for the computationally demanding tasks. So it is possible to perform the necessary kinetic folding simulations (LatFold) as well as sequential/co-translational folding studies (LatSeF). The tools are described in more detail in the methods section.

Based on our classification we provide a large set of *protein-like*, *good* and *bad* folding sequences for the 3D-cubic HP-model. The data set is freely available, see materials section.

In addition to the applicability of this data set to address relevant biological questions it serves as the first well defined benchmark sequence set for folding algorithms (Steinhöfel *et al.*, 2007). So far new methods have usually been tested on random sequences that, with high-probability, will not show protein-like behavior. Since the approach and the used LatPack tools are applicable to arbitrary lattice protein models it opens the selection of such *data sets for any lattice protein model!*

Results and Discussion

In the following we will demonstrate our strategy to classify protein-like sequences in simplified lattice protein models based on folding properties. We utilize the HP-model, but the strategy is directly applicable to arbitrary lattice protein models. The LatPack tools applied are described more thoroughly in the methods section. The free package together with manuals is available from

`http://www.bioinf.uni-freiburg.de/Software/`

The HP-model in the unrestricted 3D-cubic lattice was chosen due to its prevalence in previous protein studies (Jacob *et al.*, 2007; Steinhöfel *et al.*, 2007; Wolfinger *et al.*, 2006; Jacob and Unger, 2007; Thachuk *et al.*, 2007) and the abundance of reasonable sequence sets. Often the used benchmark sets consider degeneracy only and thus (with the exception of (Jacob and Unger, 2007)) do not reflect a reasonable protein-likeness definition based on kinetic properties. Furthermore, they usually consist of a few sequences only. Here we implement a generic, transparent and reproducible definition with the aim of producing a large benchmark set for use in future studies.

The classification is mainly achieved using folding simulations. For global folding, where the whole fold space is explored, we utilize the Pull-move set (Lesh *et al.*, 2003). This set is often used (Thachuk *et al.*, 2007) and has been shown to yield realistic folding times (Steinhöfel *et al.*, 2007). We address the problem of correct folding temperatures essential for reasonable Monte-Carlo simulations (see methods). The outcome of our procedure is a data set consisting of *protein-like* sequences, *good* and *bad* folders that is freely accessible at (SeqData-URL).

`http://www.bioinf.uni-freiburg.de/Data/`

Non-degenerate native structure

In the HP-model, protein-likeness is usually defined via thermodynamic properties only. The simplified energy function yields a very high degeneracy for most of the sequences, i.e. they can adopt thousands or millions of optimal structures (Mann *et al.*, 2008). Such sequences have low thermodynamic stability and are very unlikely to fold into a single native structure. Therefore, a common way to select protein-like sequences is to request that degeneracy = 1, i.e. a non-degenerate, unique ground state (Crippen and Chhajer, 2002; Jacob and Unger, 2007). Such a single energetically minimal structure is assumed to be the native structure of the sequence. This is discussed in more detail in the materials section.

In the first classification step of our approach we search for *non-degenerate* sequences. Thus to classify a sequence as protein-like we assume, as the minimal requirement, that such a unique native structure exists. Using the CPSP-tools (Mann *et al.*, 2008) we observed that only about 0.01 percent of all sequences fulfill this property in the unrestricted 3D-cubic HP-model (data not shown). Thus only a small fraction of sequences are considered in the next, kinetic based, classification step.

For illustration, we derived a random non-exhaustive set of 10,500 *non-degerate* HP-sequences of length 27 (in the 3D-cubic lattice) using the CPSP-tools. This sequence set will be used in the following to demonstrate the whole classification approach.

Determination of the optimal folding temperature

Protein folding is a kinetic process and therefore highly temperature depending. When modelling this process by Metropolis Monte-Carlo (MC) simulations (as done in the next classification step), this dependency is reflected by the folding temperature T used in the Metropolis criterion to calculate the Boltzmann weight $e^{-\frac{E}{k_B T}}$ of a given structure with energy E. Due to the coarse grained energy function, the Boltzmann factor k_B cannot be applied! Furthermore the optimal folding temperature T_f, where the native structure of a protein is adopted best and is stable too, is unknown and has to be determined for each protein model independently.

It is sufficient to determine the product kT_f instead of T_f and k independently. This is achieved by a screening with folding simulations (using LatFold) over different values of kT for a non-redundant set of non-degenerate sequences, because we are only interested in their folding behavior and not in random sequence folding. We define kT_f as the value where the folding simulations spend most of the time in the native state.

We expect that the screening shows a very low ratio for low kT and that the folding simulations are usually frozen in local minima (non-native structures) of the energy landscape. For high kT, a randomized traversal of the landscape is expected, resulting in a few native fold hits and a high variety in the adopted energies. At kT_f the simulation should hit the native structure at high rate and stay there for long periods. To exemplify the process we use Monte-Carlo folding simulations based on Pull moves (Lesh *et al.*, 2003) (see materials).

Figure 1 shows representative screening simulation trajectories (energy runs) that exemplify the expected behavior for different kT values in the HP-model. For very low kT the simulation is immediately trapped while for high values a random behavior is observed. Only in the 3rd plot at kT_f, the energy of the *single* native structure is reached, kept and recovered (if left) over long time series.

To prepare the folding based classification of sequences in the next section, we have performed a kT screening for a subset of the underlying sequence set (see materials section). Therefore, a non-redundant set of 50 sequences were selected at random from the pool of 10,500 non-degenerate sequences from the first classification step. For each sequence at every kT-value screened, 1000 folding simulations with 10,000 steps were done and the native structure ratio averaged. This way we could determine the kT_f^i for each screened sequence S_i. To derive a general kT_f we averaged over all gained kT_f^i.

The resulting kT_f for the non-degenerate HP-sequences of length 27 in the unrestricted 3D-cubic lattice using Pull-Moves is $kT_f \sim 0.3$ (in detail 0.285). We observe a very low variance of 0.006, supporting the low sample size. Independent tests revealed the same kT_f characteristics for the resulting classified groups (data not shown). Thus the kT_f choice seems to be invariant to the specific sequence set used in this model. In general a higher sample size should be used if the kT values show a higher variance.

Our determined kT_f is close to the folding temperature ($kT = 0.5$) for sequences of length 25 in the 2D-square lattice as used in (Jacob and Unger, 2007); however, it is unclear how the authors decided on this value.

Identifying *good* and *bad* Folders

Given the optimal kT_f value, we are now in the position to classify lattice protein sequences based on kinetic properties. The goal of the second classification step is to separate our *non-degenerate* sequences in two sets: *good and bad folders*, dependent on their kinetic properties. Good folders are assumed to be the more protein-like sequences due to the ability to fold into their native structure very fast. On the opposite the bad folders represent ran-

dom protein sequences that are able form a random coil but no stable functional native structure (Mazzoni and Casetti, 2006). Such a classification opens new studies to investigate the common properties of good vs. bad folders; perhaps facilitating indentification of the properties that allow for folding into a unique native state (Jacob and Unger, 2007). This property is often assumed to correlate with a feature of the energy landscape, the *folding funnel*. For good folders such a funnel is assumed to cover large parts of the landscape and drives the folding process downwards to the native fold (Wolynes *et al.*, 1995; Klemm *et al.*, 2008).

We are going to do a *good/bad* folder classification for the large non-redundant set of non-degenerate HP sequences of length 27 from the first classification step. For each sequence we perform a series of 1000 very short folding simulations with 1000 steps at the given value kT_f using LatFold to allow for reasonable statistics and a high parallisation of the computations. The choice of 1000 steps was based on prelimary tests (data not shown) and has to be adopted for each protein model and length. We stop a simulation early if the native structure is adopted. Therefore, we are able to measure how often a sequence is able to adopt its native conformation in a given short time interval. This is of importance due to the relatively short folding time of proteins *in vivo*. Once the native structure is reached we assume it is kept because we are simulating at the optimal folding temperature (kT_f used).

A histogram on the "success rates" of the sequences is given in Figure 2. For each sequence the number of successfull runs that found the native structure out of the 1000 runs was determined (bins of the histogram). The label of each bin gives the lower bound on the interval the bin covers to allow for a logarithmic view. The observed number of hits lies in the range 0 to 125, thus the sequence with the highest success rate found its native structure in 12.5% of the short runs. These are the best candidates for *good* folders. The number of sequences not able to fold into its native structure within the given simulation time is about 10%. Furthermore it becomes visible that a low number of hits is a common feature (about 70% show 1-9 successful runs). The wide range of hit-ratios allows for an arbitrary classification of sequences as done in the following.

Based on the collected data on the 10,500 sequences we can set two thresholds. h_{bad} marks the maximal number of hits to mark a sequence still as a *bad* folder and h_{good} the minimal hit number for a sequence to be classified as a *good* folder.

For our data, we set $h_{bad} = 1$ and $h_{good} = 10$ to split the data set to gain a large set of good folders for the last classification step.

Based on this setting we get 3 classes of sequences: 2163 *bad* folder, 2447 *good* folder and 5890 "in-between" not classified *non-degenerate* sequences with a hit rate in (h_{bad}, h_{good}) excluding the limits.

Sequential folding properties

As discussed at the beginning, proteins are assembled in a sequential manner at the ribosome. Thus it is very likely and in some instances has been experimentally verified that the protein begins to fold before release from the ribosome (Frydman *et al.*, 1994; Nicola *et al.*, 1999; Kolb *et al.*, 2000; Kolb, 2001). Our current classification does not consider this co-translational scenario and assumes global folding of the whole protein as occurring e.g. after unfolding of the structure due to heat shock or other environmental changes.

To integrate co-transational folding ability into our classification we revisit our set of *good* folders. We want to further partition this set based on the ability to fold co-translationally, assuming that this feature describes an additional fundamental property of proteins.

For all 2447 *good* folders, we run a sequential folding simulation using LatSeF. To prevent sequences from becoming trapped in shallow, local energy minima we allow sequences to overcome small energy barriers in the co-translational folding pathway. We perform simulations at varying maximal energy barriers $\Delta E \in \{0, 1, 2\}$. For each sequence we check if and on which energy threshold the native structure is reachable.

We classify a *good* folder as *protein-like* if it is able to adopt its native structure with sequential folding traversing a maximal energy barrier ΔE of 2.

Based on this classification we end up with 605 *protein-like* sequences. A first screen on sequence features did not revealed significant differences to one of the other sequence classes.

Conclusion

The selection of protein-like sequences is an important problem in simplified protein models. The identification of protein-like sequences opens the door for studies on folding kinetics, sequence evolution and docking experiments. Currently, within abstract but computationally accessible lattice protein models often only thermodynamic criterias are considered in selection. Alternatively, random sequences used.

We introduce a classification scheme that incorporates both the thermodynamic features and kinetic properties of sequences. A *protein-like* sequence has to be able to adopt its unique native structure in a short simulation time. Furthermore, we consider the sequential assembly of proteins and so include co-translational folding. Here each sequence is checked to see if its native structure can

be adopted sequentially if only small energy barriers are allowed to be overcome.

This classification scheme was applied to a non-exhaustive set of 10,500 random non-degenerate sequences of length 27 in the 3D-cubic HP-model. We end up with 4 sequence sets that are available online (see reference SeqData-URL):

- 605 *protein-like* sequences
- 1842 *good* folders
- 2163 *bad* folders
- 5890 unclassified *non-degenerate* sequences

This data set is the first classification based on thermodynamic and kinetic features that respects the sequential production of proteins as well. It can therefore form the basis for validated studies in abstract models.

Though only demonstrated here for short sequence lengths in the simple 3D HP-model, the whole classification approach is applicable to any arbitrary lattice protein model using a contact or even distance based energy function. The used tools LatFold and LatSeF are able to perform the necessary folding simulations for any of these models. Thus the classification can be done for any sequence set and model of interest using our freely available LatPack tools.

Materials and Methods

In this section we give detailed information on our sequence data and the tools utilised from the LatPack package.

Non-degenerate HP sequences

As stated above it is an essential feature of a protein-like sequence to have a thermodynamically stable native conformation. This results in the common assumption, in simplified protein models, that the structure of minimal energy corresponds to the native fold (Crippen and Chhajer, 2002; Jacob and Unger, 2007).

Unfortunately, the simple energy function in the HP-model (Lau and Dill, 1989) in tandem with the discretization of structure space (due to the lattice) induces a high degeneracy of the model. Thus a high number of sequences have thousands or millions of structures with minimal energy. In order to allow for a stable native structure, such sequences cannot be considered as protein-like. Therefore, we are interested in sequences with a very low degeneracy or, even better, non-degenerate sequences. In earlier research it was felt, that such non-degenerate sequences with a unique minimal energy structure would not exist (Shakhnovich, 1996). Due to the high computational complexity required to calculate even a single optimal structure (NP-complete) (Berger and Leighton, 1998) it was not thought possible to determine efficiently the degeneracy of a sequence, i.e. all optimal structures. Using the new Constraint-based Protein Structure Prediction (CPSP) approach of Backofen and Will (Backofen and Will, 2006) it was shown in (Mann *et al.*, 2008) that such structures exist and can be detected with very low time consumption.

We have used the CPSP-tools (Mann *et al.*, 2008; CPSP-home) to calculate a random non-exhaustive set of 10,500 non-degenerate HP sequences of length 27. These sequences have a unique minimal energy structure in the unrestricted 3D-cubic lattice. The whole set of sequences is available, see reference (SeqData-URL).

The problem of a high average degeneracy is common in lattice protein models. It results mainly from the discretization of sequence and structure space. Thus an approach for the calculation of a sequence's degeneracy would be needed for each model, as the CPSP-approach for the HP-model. Currently, only for the HPNX-model (Renner and Bornberg-Bauer, 1997; Wolfinger *et al.*, 2006) does such an approach exists - an extension of the CPSP-approach (Backofen and Will, 1998, 2006).

In some cases, the restriction to non-degenerate sequences might be too severe and also sequences with a low degeneracy are of interest. The CPSP-tools, used for the degeneracy classification in the HP-model, support identification of these sequences too (Mann *et al.*, 2008). The number of sequences grows exponentially with rising degeneracy in the HP-model. Nevertheless, the modularity of our classification approach is well suited to incorporate such customisations.

LatFold - global folding simulations

LatFold enables global folding simulations of lattice proteins. The folding path is emulated via the common Monte-Carlo (MC) simulation using a Metropolis criterion (Jacob and Unger, 2007; Thachuk *et al.*, 2007). It is therefore an iterative procedure that at each step takes the current structure and, utilizing a move set, indentifies a random neighbor in the energy landscape (discussed later). The Metropolis criterion is used to determine if the neighboring structure is accepted. If rejected, the simulation keeps the current structure for this step. Thus if the neighboring structure has lower energy it is always accepted. If not it is adopted with probability $e^{-\frac{\Delta E}{kT}}$ while ΔE is the energy difference between the neighbored structure and the current one. kT is protein model specific and has to be calculated as discussed above. A similar method was successfully applied to RNA models (Flamm *et al.*, 2000) and reflected realistic folding features.

The program is applicable to lattice protein models with arbitrary contact or distance based energy functions and is consequently very general. Furthermore the energy function can be chosen independently from the lattice used. We are currently supporting the unrestricted simple 2D-square, 3D-cubic and the highly complex 3D-Face-Centered-Cubic (FCC) lattice. The latter was shown to allow for high precision real protein structure presentations (Park and Levitt, 1995). Park and Levitt achieved a coordinate root mean square deviation of 1.78 Å, whereas a deviation of 2.84 Å was obtained in the 3D-cubic lattice. An extension of LatFold to other lattice models is easily possible (see BIU library (BIU-home)).

The applied neighborhood generation within LatFold utilizes two different ergodic so called *move sets*. These are generic definitions of rules to apply (small) structural changes within a given protein conformation to generate structures neighbored in the energy landscape. Thus an iterative application of such *moves* models structural changes over time, i.e. folding. The ergodicity ensures that all structures can be transformed into each other via a sequence of moves. The Pivot-moves (Madras and Sokal, 1988) yield relatively strong structural changes (Wolfinger *et al.*, 2006) by rotating huge parts of the structure. In contrast, the application of Pull-moves (Lesh *et al.*, 2003) results in more local changes of the structure. It has been shown that this move set is able to reproduce realistic folding times (Steinhöfel *et al.*, 2007) and is therefore well suited for our folding based classification (Jacob and Unger, 2007). To our knowledge there is no other ergodic move set for lattice protein models. The non-ergodic local moves (Madras and Sokal, 1987) are not used due to the partitioning of the accessible energy landscape into independent ergodicity classes. Their number is growing exponentially while each ergodicity class gets exponentially small.

The current implementation of LatFold is based on the free and open-source BIU (BIU-home) and ELL (ELL-home) C++ programming libraries to allow for highest performance and modularity and is applicable to any sequence length.

It would be of great interest to enable the application of LatFold to side chain lattice protein models too, but to our knowledge this is currently not possible due to the absence of a suitable ergodic move set for such structure models. In general, Pivot-moves should be applicable. Unfortunately, most of the moves will produce overlapping and therefore invalid structures, leading to a large computational overhead critical in folding simulations.

Another possible extension is the usage of a *rejection-less* MC-simulation that was successfully applied to RNA folding simulations (Flamm *et al.*, 2000) and offers lower runtimes when a high rejection rate is present (low kT values). Furthermore, such simulations yield more accurate time scalings.

ELL - the energy landscape library

To investigate the folding process of biopolymers the generic concept of energy landscapes is often applied (Wales, 2004; Wolfinger *et al.*, 2006; Flamm and Hofacker, 2008). It is defined by a triple $(\mathcal{X}, E, \mathcal{N})$ where $\mathcal{X}$ is the set of structures a sequence can adopt, its structure space, $E : \mathcal{X} \to \mathcal{R}$ is an energy function and $\mathcal{N} : \mathcal{X} \to \mathcal{X}^*$ the neighborhood relation, e.g. defined by a move set.

Studying energy landscapes can give insights into the folding process and kinetic properties of a molecule (Mazzoni and Casetti, 2006). It has, therefore, been an area of great study and many algorithms have been designed to address the problem (Hoffmann and Sibani, 1988; Flamm *et al.*, 2002; Wolfinger *et al.*, 2004). The energy landscape library (ELL) (Mann *et al.*, 2007) was developed to serve as a platform to implement such algorithms independently from a concrete energy landscape model as e.g. HP lattice proteins.

Our LatFold program performs a Metropolis MC-simulation and, thus, utilizes the modularity and functionality of the ELL. Consequently, the program is easy to extend while still ensuring high performance, necessary for exhaustive high-throughput studies that are possible and needed for lattice protein models.

The ELL offers a flexible framework to define move sets and, hence, the neighborhood relation $\mathcal{N}$ that defines the energy landscape. It is, thus, easy to extend LatFold if new ergodic move sets, e.g. for side chain models, are developed.

LatSeF - sequential folding

LatSeF implements a greedy, heuristic chain-growth approach similar to the procedure applied in (Bornberg-Bauer, 1997). The monomers are placed successively on lattice positions such that the structure forms a self-avoiding walk. For each length all possible structure extensions with one monomer are generated and evaluated. The energetically best structures are considered in the next extension iteration.

Due to the lattice restrictions and the constraint of self-avoidance, the procedure may end in non-extensible structures during the iteration and fail. A fast way to overcome this problem is to check the extensibility of the last monomer after its placement. This check is not ensuring extensibility to the whole length but seems to be sufficient for most sequences (data not shown). Only extensible structures are considered further and evaluated later.

The algorithm in detail:

Algorithm 1 LatSeF - chain growth algorithm

```
1:  S = S_1, ..., S_n                                   ▷ protein sequence
2:  E(S, P)                                             ▷ energy function
3:  N                                                   ▷ lattice's neighboring vectors
4:  ΔE                                                  ▷ maximal energy barrier to overcome
5:  B ← {L_1 = (0, 0, 0)}                               ▷ best structures of last iteration
                                                        ▷ initialized by placing the first monomer to (0, 0, 0)
6:  C ← ∅                                               ▷ structures generated in current iteration
7:  for i = 2 ... n do
8:      for all L ∈ B do                                ▷ L has length (i − 1)
9:          for all v⃗ ∈ N do
10:             if L_(i−1) + v⃗ ∉ L_1, ..., L_(i−1) then   ▷ selfavoidingness
11:                 if L_1, ..., L_(i−1), (L_(i−1) + v⃗) is extensible then
12:                     C ← C ∪ {(L_1, ..., L_(i−1), L_(i−1) + v⃗)}
                                                        ▷ store extension
13:                 end if
14:             end if
15:         end for
16:     end for
17:     minE ← minimal energy of all elements in C
18:     B ← {c | c ∈ C and E(S_1...i, c) ≤ (minE + ΔE)}   ▷ all best
19:     C ← ∅                                           ▷ reset structure storage
20: end for
21: report best placement L ∈ B with minimal energy E(S, L)
```

A main feature of the algorithm is its ability to overcome energy barriers in the co-translational folding path. This is done by not only considering the energetically *best* structures from the last iteration for elongation but all structures within an energy interval of ΔE of the minimal energy found for the current chain length. Thus, the method is not trapped by local minima and the sequential folding can escape over low energy barriers. This extension is essential for longer chain lengths or complex 3D lattices. Nevertheless, the interval should be choosen thoughtfully because of its direct influence on the memory consumption of the program. The higher the allowed energy difference the more sequences are stored for consideration in the next iteration. Their number typically grows exponentially, depending on the degeneracy of the protein model, such that even small intervals might lead to a high memory requirement for long sequences.

A further, but here not applied, feature is the consideration of side chain lattice models. LatSeF is the first tool that allows for sequential folding including side chain monomers. As discussed above, once a suitable ergodic move set / neighborhood relations for side chain models is available, we can directly apply the whole presented classification approach on these models.

Last, but not least, the strength of LatSeF is its applicability to any contact or distance based energy function (thus also a full potential as the MJ-matrix (Miyazawa and Jernigan, 1996)) and the possibility to use high complex 3D-lattices as the discussed Face-Centered-Cubic (FCC) lattice.

The implementation is based on the BIU C++ programming library (BIU-home).

Acknowledgements

We thank Dr. Sebastian Will for his helpful comments on the manuscript. Martin Mann is supported by the EU project EMBIO (EC contract number 012835).

References

Backofen, R. and Will, S., 1998. Structure prediction in an HP-type lattice with an extended alphabet. In *Proc of German Conference on Bioinformatics (GCB'98)*.

Backofen, R. and Will, S., 2006. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. *J Constraints* 11, 5 – 30.

Berger, B. and Leighton, T., 1998. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *J Comp Biol* 5, 27–40.

Berman, P., DasGupta, B., Mubayi, D., Sloan, R., Turán, G., and Zhang, Y., 2004. The protein sequence design problem in canonical model on 2D and 3D lattices. In *Combinatorial Pattern Matching*, volume 3109, 244–253. Springer.

BIU-home, 2007. BIU : Bioinformatics utilities. Available as an open-source library from `http://www.bioinf.uni-freiburg.de/sw/biu/`.

Bornberg-Bauer, E., 1997. Chain growth algorithms for HP-type lattice proteins. In *RECOMB'97*, 47–55.

Chan, H. S. and Dill, K. A., 1990. Origins of structure in globular proteins. *Proc Natl Acad Sci USA* 87, 6388–92.

Coluzza, I. and Frenkel, D., 2007. Monte carlo study of substrate-induced folding and refolding of lattice proteins. *Biophys J* 92, 1150–1156.

CPSP-home, 2008. CPSP-tools : Constraint-based protein structure prediction. Available as an open-source package from `http://www.bioinf.uni-freiburg.de/sw/cpsp/`.

Crippen, G. M. and Chhajer, M., 2002. Lattice models of protein folding permitting disordered native states. *J. Chem. Phys.* 116, 2261.

Deane, C. M., Dong, M., Huard, F. P., Lance, B. K., and Wood, G. R., 2007. Cotranslational protein folding–fact or fiction? *Bioinformatics* 23, i142–8.

Dill, K. A., Bromberg, S., Yue, K., Fiebig, K. M., Yee, D. P., Thomas, P. D., and Chan, H. S., 1995. Principles of protein folding–a perspective from simple exact models. *Protein Sci* 4, 561–602.

ELL-home, 2007. ELL : Energy landscape library. Available as an open-source library from `http://www.bioinf.uni-freiburg.de/sw/ell/`.

Fedorov, A. N. and Baldwin, T. O., 1997. Cotranslational protein folding. *J Biol Chem* 272, 32715–8.

Flamm, C., Fontana, W., Hofacker, I. L., and Schuster, P., 2000. RNA folding at elementary step resolution. *RNA* 6, 325–38.

Flamm, C. and Hofacker, I. L., 2008. Beyond energy minimization: approaches to the kinetic folding of RNA. *Chemical Monthly* 139, 447–457.

Flamm, C., Hofacker, I. L., Stadler, P. F., and Wolfinger, M. T., 2002. Barrier trees of degenerate landscapes. *Z.Phys.Chem* 216, 155–173.

Frydman, J., Nimmesgern, E., Ohtsuka, K., and Hartl, F. U., 1994. Folding of nascent polypeptide chains in a high molecular mass assembly with molecular chaperones. *Nature* 370, 111–117.

Garey, M. R. and Johnson, D. S., 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Govindarajan, S. and Goldstein, R. A., 1998. On the thermodynamic hypothesis of protein folding. *Proc Natl Acad Sci USA* 95, 5545–9.

Gupta, A., Manuch, J., and Stacho, L., 2005. Structure-approximating inverse protein folding problem in the 2D HP model. *J Comp Biol* 12, 1328–1345.

Hoffmann, K. H. and Sibani, P., 1988. Diffusion in hierarchies. *Physical Review A* 38, 4261–4270.

Huard, F. P. E., Deane, C. M., and Wood, G. R., 2006. Modelling sequential protein folding under kinetic control. *Bioinformatics* 22, e203–210.

Irback, A. and Troein, C., 2002. Enumerating designing sequences in the HP model. *Journal of Biological Physics* 28, 1–15.

Jacob, E., Horovitz, A., and Unger, R., 2007. Different mechanistic requirements for prokaryotic and eukaryotic chaperonins: a lattice study. *Bioinformatics* 23, i240–i248.

Jacob, E. and Unger, R., 2007. A tale of two tails: why are terminal residues of proteins exposed? *Bioinformatics* 23, 225–230.

Klemm, K., Flamm, C., and Stadler, P. F., 2008. Funnels in energy landscapes. *The European Physical Journal B* 63, 387–391.

Kolb, V. A., 2001. [cotranslational protein folding]. *Mol Biol (Mosk)* 35, 682–90.

Kolb, V. A., Makeyev, E. V., and Spirin, A. S., 2000. Co-translational folding of an eukaryotic multidomain protein in a prokaryotic translation system. *J Biol Chem* 275, 16597–601.

LatPack-home, 2008. LatPack : Lattice protein folding package. Available as an open-source package from `http://www.bioinf.uni-freiburg.de/Software/`.

Lau, K. F. and Dill, K. A., 1989. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules* 22, 3986–3997.

Lesh, N., Mitzenmacher, M., and Whitesides, S., 2003. A complete and effective move set for simplified protein folding. In *Proceedings of the seventh annual international conference on Research in computational molecular biology (RECOMB'03)*, 188–195.

Levinthal, C., 1968. Are there pathways for protein folding? *Extrait du Journal de Chimie Physique* 65.

Madras, N. and Sokal, A. D., 1987. Nonergodicity of local, length-conserving Monte Carlo algorithms for the self-avoiding walk. *Journal of Statistical Physics* 47, 573–595.

Madras, N. and Sokal, A. D., 1988. The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics* 50, 109–186.

Mann, M., Will, S., and Backofen, R., 2007. The energy landscape library - a platform for generic algorithms. In *Proc. of BIRD'07*, volume 217, 83–86. OGC.

Mann, M., Will, S., and Backofen, R., 2008. CPSP-tools - exact and complete algorithms for high-throughput 3D lattice protein studies. *BMC Bioinformatics* 9, 230.

Mazzoni, L. N. and Casetti, L., 2006. Curvature of the energy landscape and folding of model proteins. *Physical Review Letters* 97, 218104.

Miyazawa, S. and Jernigan, R. L., 1996. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *J Mol Biol* 256, 623–44.

Nicola, A. V., Chen, W., and Helenius, A., 1999. Co-translational folding of an alphavirus capsid protein in the cytosol of living cells. *Nat Cell Biol* 1, 341–5.

Park, B. H. and Levitt, M., 1995. The complexity and accuracy of discrete state models of protein structure. *J Mol Biol* 249, 493–507.

Renner, A. and Bornberg-Bauer, E., 1997. Exploring the fitness landscapes of lattice proteins. *Pac Symp Biocomput.* 361–372.

SeqData-URL, 2008. Classified set of protein-like, good/bad folding and non-degenerate HP-sequences of length 27 in the unrestricted 3D-cubic HP-model. Available from `http://www.bioinf.uni-freiburg.de/Data/`.

Shakhnovich, E. I., 1996. Modeling protein folding: the beauty and power of simplicity. *Fold Des.* 1, R50–54.

Steinhöfel, K., Skaliotis, A., and Albrecht, A. A., 2007. Stochastic protein folding simulation in the d-dimensional HP-model. In *Proceedings of the 1st Conference on BioInformatics Research and Development*, 381–394. Springer.

Thachuk, C., Shmygelska, A., and Hoos, H. H., 2007. A replica exchange Monte Carlo algorithm for protein folding in the HP model. *BMC Bioinformatics* 8, 342.

Unger, R. and Moult, J., 1993. Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. *Bull Math Biol* 55, 1183–1198.

Wales, D. J., 2004. *Energy Landscapes*. Cambridge University Press, Cambridge.

Wolfinger, M., Will, S., Hofacker, I., Backofen, R., and Stadler, P., 2006. Exploring the lower part of discrete polymer model energy landscapes. *Europhysics Letters* 74, 725–732.

Wolfinger, M. T., Flamm, W. A. S.-S. C., Hofacker, I. L., and Stadler, P. F., 2004. Exact folding dynamics of RNA secondary structures. *J.Phys.A: Math.Gen.* 37, 4731–4741.

Wolynes, P. G., Onuchic, J. N., and Thirumalai, D., 1995. Navigating the folding routes. *Science* 267, 1619 – 1620.

Figure Legends

Figure 1

Folding simulations for a kT series to identify kT_f (3rd plot). The dotted green line marks the energy of the unique native structure, i.e. if it is reached the native structure is adopted.

Figure 2

Histogram of the sequence numbers based on successful runs that end in the native structure out of 1000 runs with maximal 1000 simulation steps per sequence (green bars). The red bar on the left represents the number of sequences that did not found their native structure within the given simulation length.

Figures

Figure 1

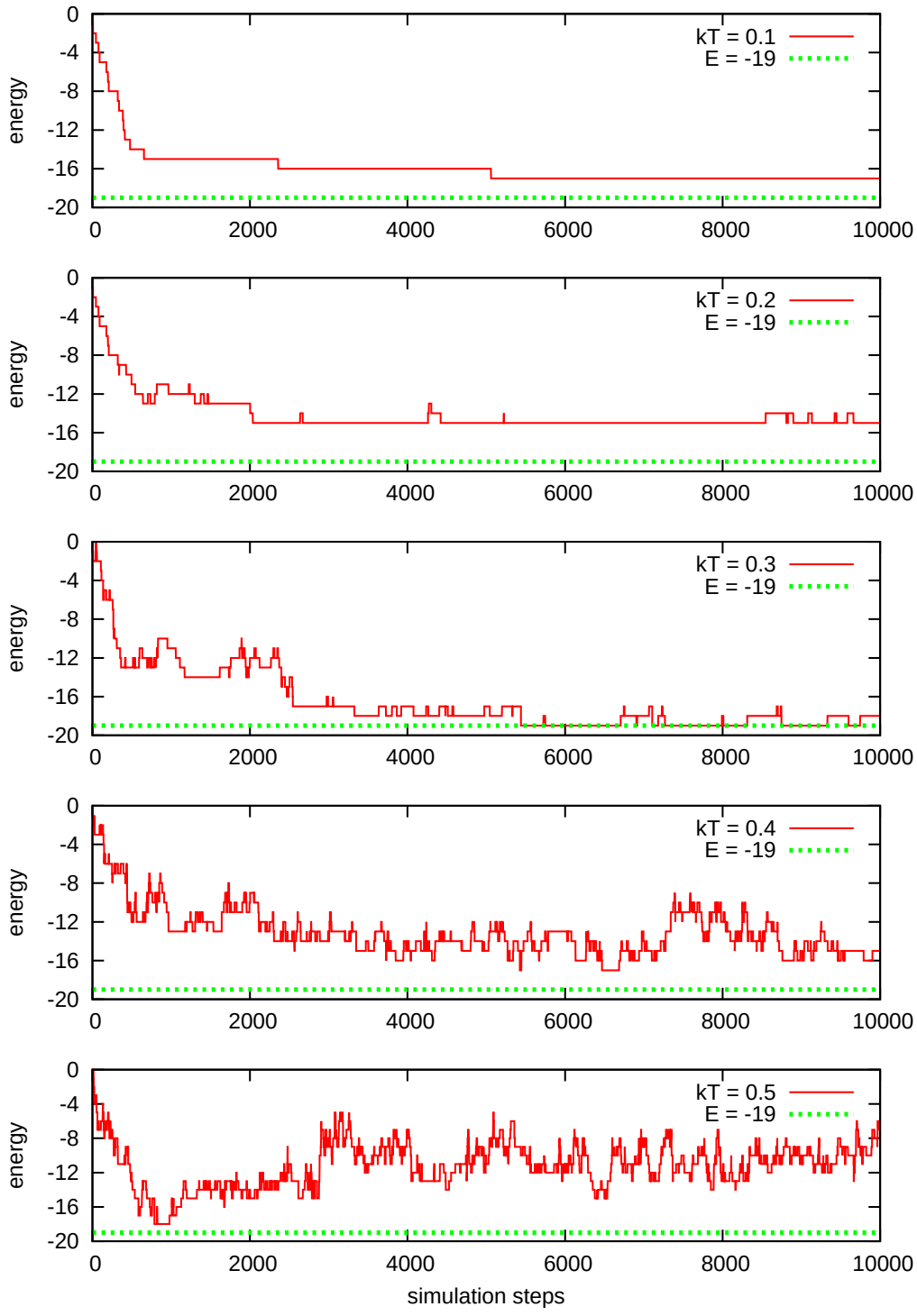
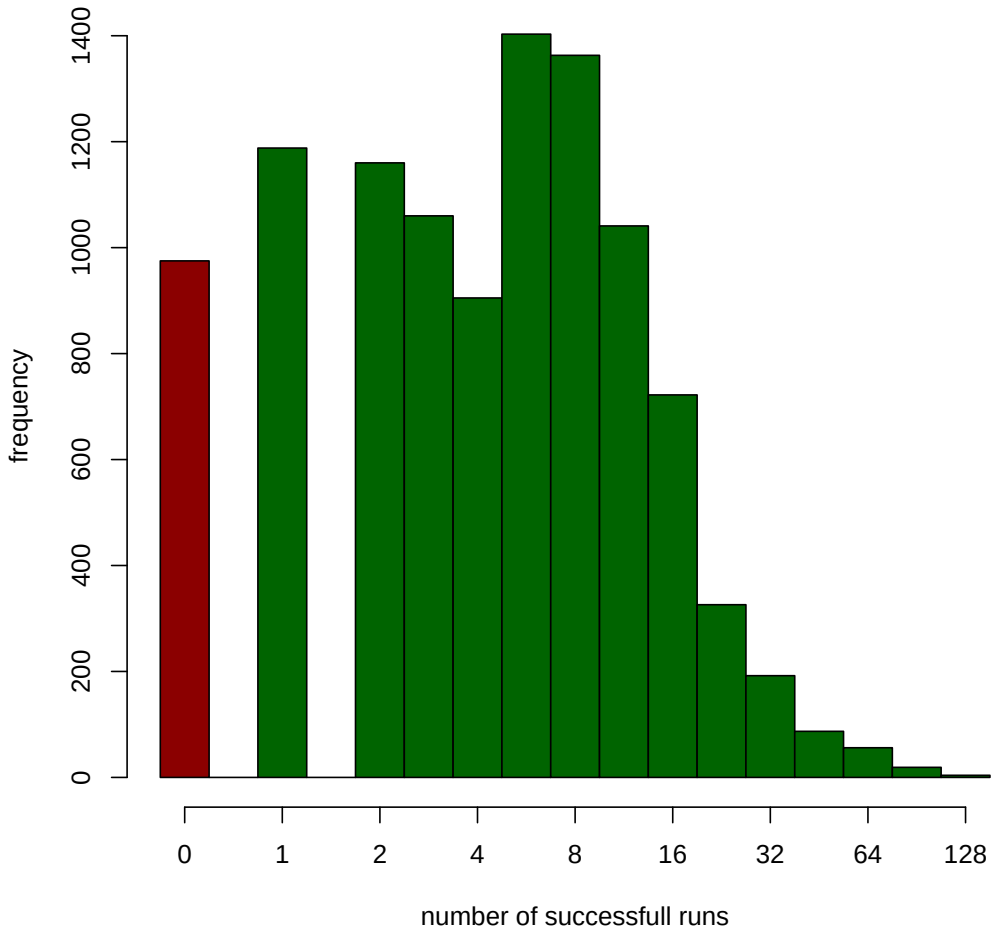


Figure 2


frequency
0
200
400
600
800
1000
1200
1400
0
1
2
4
8
16
32
64
128
number of successfull runs