How to recognise different types of *shape*
from quite a long way away

Random Structures and Evolution of Biopolymers

DISSERTATION

eingereicht von

Mag. rer. nat. Erich Bornberg - Bauer

zur Erlangung des akademischen Grades
Doctor rerum naturalium
an der Formal- und Naturwissenschaftlichen Fakultät
der Universität Wien

May 1995

Acknowledgements

Work in hand was accomplished between November 1993 and May 1995 at the "Institut für Theoretische Chemie, Abteilung Theoretische Biochemie". It is a pleasure for me, to thank Prof. Peter Schuster for offering to work on an interesting subject, introduction to scientific work and providing excellent facilities.

I want to thank all members on the institute for their help and support:

Peter Stadler provided numerous useful comments and help with programming and designing some data structures. Several numerical results (SAW lengths, cSAWs and Collapsing SAWs) and programming utilities are outcome from joint work with Alexander Renner. Some ideas and computations on RNA are result from a cooperative and productive joint work with Manfred Tacker. Walter Fontana provided some general programming utilities and the landscape routines as well as many valuable comments, Ivo Hofacker frequently helped to speed up programming process.

All other colleges and members on the institute also supported me whenever needed and contributed to a creative and productive working atmosphere. Financial support by "Fonds zur Förderung der Wissenschaftlichen Forschung, Projekt P-8526-MOB" is greatfully acknowledged.

I want to thank Paul Stolorz for the data as they were used to compare the results from the chain growth algorithm to ground state energies.

Abstract

We use two different approaches to investigate the folding of biopolymers. In both cases, RNA secondary structures and lattice proteins, folding is viewed as a map assigning a simplified structural representation to sequences and hence as an example of genotype to phenotype mapping.

We compute large ensembles of RNA secondary structures by means of various computer algorithms. Structures are sorted and ranked. Frequencies of structures are highly non-uniform: we find relatively few common and many rare ones, as expressed by a generalised form of Zipf's law.
The number of neutral neighbours are computed as well as the length of neutral paths. The probability to find networks of neutral mutants percolating the whole sequence space depends strongly on the probability to find neutral neighbours which in turn depends on the overall numbers of admissible structures.
As it is known that any random structure can be found in a close vicinity in sequence space to any other [89] this implies a new perspective on the efficiency of evolution by point mutations: from any random sequence it is not far to any desired structure, and yet most mutations will preserve the structure.

These investigations suggest, that an accurate prediction of the complete spatial structure is not always necessary to characterise the sequence to structure mapping of biopolymers. We designed a deterministic sequential folding procedure and applied it to a strongly simplified model of lattice proteins that has been put forward by Dill and Lau [54]. Applications to sequences with known, non degenerate ground states show a logarithmic dependency of the frequency of successful predictions on the search depth that is applied at the end of the chain. Depending on how pronounced kinetic traps are, thus it is possible to find ground-states by sequential folding. Increasing the search-depth leads to better results mainly because of long range hydrophobic interactions.
To perform all calculations on various lattices and to compare a number of folding strategies we developed a toolkit for computation and comparison of structures of lattice polymers. The package can be easily extended to any regular lattice. Various methods to characterise and compare structures as well as to detect secondary structure elements, distance measures and grafik utilities, are also introduced.

To both systems we also apply the concept of landscapes and combinatory maps. We use correlation lengths as a measure for the ruggedness of a landscape, which in turn is crucial for the performance of an evolutionary strategy. Although results from lattice proteins can

only be viewed as preliminary, we find interesting similarities to the case of RNA secondary structures: correlation lengths are small compared to the diameter of the sequence space, structure correlations are smaller than energy correlations, higher for low dimensional lattices and scale linearly with the chain lengths for all parameter sets, lattices and alphabets.
Due to the enormous number of possible structures and the somewhat ambiguous notion of structural coarse graining in the case of lattice proteins, the concepts of neutrality and structure enumerations can not be accomplished so easily which leaves space for further attempts.

Zusammenfassung

Zwei grundsätzlich unterschiedliche Ansätze werden benutzt, um Strukturstatistiken von Biopolymeren zu untersuchen. In beiden Fällen, dem Falten von RNA Sekundärstrukturen sowie von Gitterproteinen, betrachten wir den Vorgang der Faltung als eine Abbildung, welche der Sequenz eine vereinfachte Struktur zuordnet und folglich als ein Beispiel der Abbildung eines Genotyps auf einen Phenotyp.
Mit Hilfe unterschiedlicher Faltungsalgorithmen berechnen wir große Ensembles von RNA Sekundärstrukturen. Diese werden sortiert und nach Häufigkeiten geordnet. Strukturhäufigkeiten sind sehr ungleich verteilt: wir finden wenig häufige und sehr viel seltene, was durch ein generalisiertes Zipf'sches Gesetz charakterisiert werden kann.
Die Anzahl neutraler Nachbarn werden berechnet, ebenso wie die Länge neutraler Pfade. Die Wahrscheinlichkeit, Netzwerke neutraler Mutanten, welche durch den gesamten Sequenzraum perkoliern, zu finden, hängt stark ab von der Wahrscheinlichkeit, neutrale Nachbarn an jedem Punkt zu finden, was wiederum von der Anzahl der zugänglichen Strukturen abhängt.
Da auch jede Zufallsstruktur in enger Nachbarschaft im Sequenzraum zu jeder Anderen liegt [89], impliziert dies eine neue Perspektive über die Effizienz evolutionärer Suchstrategien mittels Punktmutationen: von jeder Zufallsstruktur ist es nahe zu jeder Anderen und dennoch können im selben System Perioden phänotypischer Stagnation auftreten da viele Mutationen die Struktur unverändert lassen.
Diese Untersuchungen legen Nahe, daß eine exakte Vorhersage der räumlichen Struktur nicht unbedingt notwendig ist, um das evolutionäre Verhalten von Ensembles von Biopolymeren zu beschreiben. Wir entwickelten eine deterministische, sequentielle Vorhersagestrategie, die das Kettenwachstum von Gitterproteinen beschreibt und wendeten sie auf das Modell von Lau & Dill [54] an.
Der Vergleich mit Strukturen, deren Grundzustand bekannt ist, zeigt eine logarithmische Abhängigkeit der Häufigkeit einer exakten Vorhersage von der Suchtiefe. Offensichtlich ist eine sequentielle Faltung, in Abhängigkeit von der Ausprägung der lokalen Minima während das Faltens imstande, den Grundzustand zu finden.
Um all diese Berechnungen auf Sequenz und Strukturräumen verschiedener Größe durchführen zu können, entwickelten wir ein Programmpaket, welches dem Benutzer ermöglicht, auf vielen verschieden Gittern zu arbeiten und Strukturen auf mehrere Arten zu kodieren, zu charakterisieren, grafisch darzustellen und zu vergleichen.
Wir wenden das Konzept der Landschaften und kombinatorischen Abbildungen auf beide Modelle an. Wir berechnen Korrelationslängen, welche als ein Maß für die "ruggedness" einer

Landschaft gibt, welche ihrerseits die Erfolgsträchtigkeit von evolutionären Suchstrategien beschreibt. Obwohl die Ergebnise im Falle der Gitterproteine nur als vorläufig anzusehen sind, konnten wir bemerkenswerte Ähnlichkeiten zum RNA Fall feststellen: Energiekorrelationen sind länger als Strukturkorrelationen, beide sind kurz im Vegleich zum Durchmesser des Sequenzraumes und skalieren linear mit der Kettenlänge. Höher dimensionale Gitter zerstören die Strukturkorrelationen, nicht jedoch die Energiekorrelation.
Wegen der enormen Anzahl realisierbarer Strukturen und der weniger klaren Möglichkeit, eine Grobauflösung zu definieren, konnten die Konzepte der Enumerierung und der Neutralität nicht ad hoc auf Gitterproteine angewandt werden, was Platz läßt für weitere Anstrengungen.

Contents

List of Figures

List of Tables

1 Introduction

The influence of mutations on the structure and thereby on phenotype and functionality of biopolymers is of paramount interest for the understanding of both natural and artificial evolution. On one hand we could learn about the development of life. On the other hand it would be very important to predict the influence of mutations to develop new drugs. Most interest is focussed on proteins as they accomplish most of the enzymatic tasks in contemporary organisms. They are therefore the potential target to all methods of biotechnology. The interest in RNA arises mainly from the fact, that this biopolymer may have stood at the beginning of prebiotic evolution [76, 36]. RNAs are also much simpler to handle by computational means. Therefore they serve well as a - however strongly simplified - model to study evolution and learn about possibilities and requirements for applied molecular evolution.
In all our investigations we present here, we are interested in the development of function of biopolymers. The basic assumption is that in the course of evolution many mutations occur which do not alter the function of the molecule. These mutations are preserved. Mutations that destroy the function of biopolymers are strongly selected against. It is assumed that function depends on structure and therefore structure is conserved in evolution.

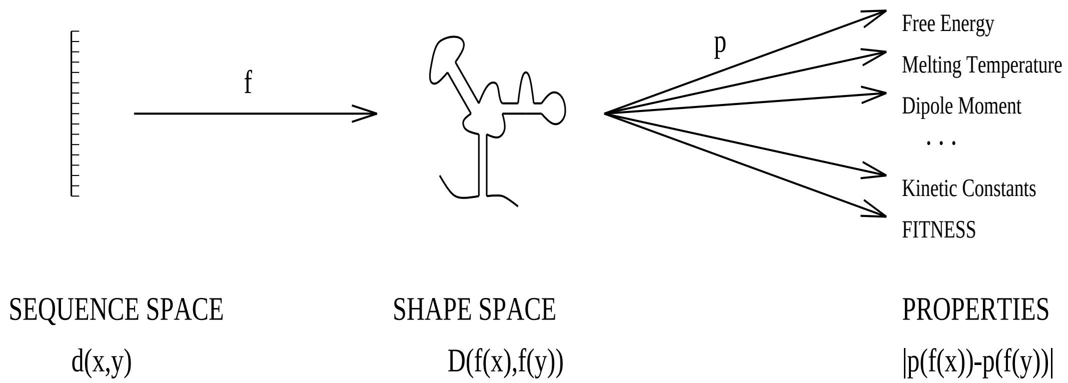


Figure 1: Biopolymers folding from a given sequence (left) into a structure. The structure exhibits several features that are the characteristic for the biological role

1.1 Concepts of Evolution

When Charles Darwin presented his theories in his famous essay on "The Origin of Species", [18] he was the first to recognise the principles of variation and selection as the basis of evolution.
The minimal requirements for Darwinian adaptation are easily stated:

- A population of objects that are capable of replication.
- A sufficiently large number of variants of those objects.
- Occasional variations which are inheritable.
- Restricted proliferation which is constrained by limited resources.

Still many problems seemed to be insurmountable: Can we assume a universal principle underlying evolution, so that the results of individual diversification still might look very different? Can we offer reasonable concepts by the means of biophysical, thermo-dynamical and statistical laws only? How could something, as complex as creatures arise from molecular structures by chance as the major driving force? Still today it seems amazing that chance might play such a predominant role in the development of life. But Darwin's catch-phrase "survival of the fittest" also caused many misunderstandings. Defining the fittest as the one with the most offsprings reduces to the well known (and well cited) tautology of "survival of the survival" which is, obviously, an a posteriori explanation.
In the third decade of this century Sewall Wright and others created a mathematical basis for Darwin's theories [115]. They introduced the concept of fitness-landscapes and interpreted Darwinian evolution as a complex optimisation problem of individuals in a population. Each genotype a unique fitness value is assigned to. The underlying molecular basis was discovered in the decades afterwards. By the means of this concept it is possible to calculate in principle the fitness value of a mutant a priory. The parameters of the fitness function which is to optimise are a chosen set of properties having an influence on the fertility of the individual.
The process of evolution then can be viewed at as a walk on a multidimensional fitness landscape where a local (or global) fitness maximum can be climbed. Moving downhill is permitted only with low probability. The adaptive walk is the most simple model of evolution and will be used on several occasions in this work. Rechenberg [84] presented twenty years ago this and other more sophisticated optimisation algorithms based on the mutation-selection principle, which have been applied to problems of mechanical engineering with great success. Developing models for evolution in nature one has to face enormous problems in finding appropriate functions, which map the genotype to the phenotype and which define a fitness for the phenotype.

1.2 Biopolymers and Prebiotic Evolution

Still the question, if proteins, polynucleotides or some other molecules can be viewed at as the starting point of prebiotic evolution is unsolved and provides us with a problem that is similar to the well known hen and egg paradox: polynucleotides need proteins for replication, proteins

can't do without polynucleotides as their encoding counterpart. How could evolution start on a system, that is based on two components that demand each other?
It is the general assumption that life on earth started with chemical processes taking place in the "soup" of organic, energy rich molecules in the primeval seas. Nucleic acids and amino acids have been synthesised in the laboratory under conditions thought to exist on the primitive earth [66, 75]. In 1971 Manfred Eigen demonstrated [23] that the only prerequisite for Darwinian evolution is an open system far from thermodynamic equilibrium with replication living on limited resources. Another necessary aspect for adaptation to take place is that information (embodied in the genotype) must be conserved. Such a conservation of acquired abilities was made possible by template directed replication. Self replicating RNAs are thought to be the first replicators of this type. The information is stored in the primary sequence. The ingredients for Darwinian evolution are provided by error prone copying of these molecules in an energy rich system. It is assumed, that RNA can be replicated by RNA itself, the search for the first RNA polymerase made entirely from RNA is very intensive [1]. Thus RNA alone is sufficient for providing the molecular basis of evolution.
Further hints that the first Darwinian evolving entities were made from RNA are that in todays cells many key processes are still catalysed by RNA including protein and RNA synthesis [53]. DNA can be viewed as an RNA derivative that is chemically stabilised by removal of the $2'$-OH group. Furthermore RNA is used as a primer by DNA polymerase whereas RNA polymerase do not need any primer at all. All these data suggest that RNA was the information carrier historical evolution on earth started with.
Some investigation, that also strongly inspired the approaches in this work have already been carried out by computer experiments. Walter Fontana et al. developed a computer model of RNA evolution in a chemical tank reactor [31, 30]. They could show that adaptation in this system is a very difficult task because the RNA landscape is an example of a rugged landscape (see chapter 3) with many local optima. Adapting molecules very easily get trapped. They also observed long periods of constance before a sudden change towards better fitness values was observed.

1.3 Irrational Design: Applied Molecular Evolution

Several new methods have been developed recently, which exploit the mechanisms of Darwinian evolution and the features of large ensembles of biopolymers to generate proteins and nucleic acids with novel properties. These methods might well lead to a partial substitution of rational designing of molecules [8]. As selection acts on the phenotype whereas mutation generates new genotypes a straightforward application of in vitro evolution is possible only in the case of

nucleic acids where genotype and phenotype are embodied in the same molecule.
In vitro experiments can be characterised by the following steps:

1. Generation of an ensemble of random sequences
2. Selection of molecules with useful properties
3. Amplification of the selected molecules with the introduction of new mutations

which is an analogy to Darwinian evolution:

1. Generation of mutations
2. Evaluation, depending on a fitness criterion
3. Selective error prone, amplification according to the fitness criterion

In laboratory experiments the number of quasi random RNAs is about 10^{13} to 10^{15} in the start population [2, 1]. DNA can be converted into RNA with T7 RNA polymerase. The opposite reaction is performed through the use of reverse transcriptase. Thus there is inter-conversion between RNA and DNA possible at every step.
Selection depends on the characteristics of the desired property. RNAs can be selected with respect to their ability to bind dyes [26] or to catalyse specific reactions, for example phosphoester transfer [2] or ligation [1]. Amplification is performed by transcription of DNA by T7 RNA polymerase which amplifies cDNA about 1000fold [10] or by polymerase chain reaction (PCR).
The PCR reaction under mutagenic conditions has been used for the generation of mutations. Starting from a pool of 10^{15} random RNAs David Bartel and Jack Szostak succeeded in finding a novel RNA ligase with reaction rates 7 million times faster than the uncatalysed reaction [1, 45]. This achievement is very promising, because it is the first step to find an RNA replicase made entirely from RNA, a key enzyme in the RNA world.
For proteins the situation is somewhat more involved because there is a difference between the genotypic (DNA) and phenotypic level (protein) and there is no simple and practical possibility for generating the corresponding DNA from selection experiments. A way out of this dilemma is the use of catalytic antibodies. It has been shown that second class antibodies (antibodies directed against the binding site of antigen binding antibodies) can mimic antigen [3] and theoretical models led to the concept of a shape space [80]. 10^8 antibodies should saturate this shape space and therefore provide a universal set of enzymes.

1.4 Rational Design: Predicting the Structure from the Sequence

It is a very simple task to translate the DNA sequence in the corresponding RNA sequence or the protein sequence. It is an enormous problem however, which is not yet solved, to deduce the tertiary structure of the protein or RNA from the primary sequences. But this three dimensional structure is an essential prerequisite to understand the processes going on in an organism - a protein or RNA is biologically active only when it assumes its correct form, incorrectly folded molecules are in general biologically inactive.
One of the most challenging questions in structural biology therefore is the folding problem of biopolymers: can we predict the structure, i.e. the spatial conformation simply from the knowledge of the sequence or, can we at least give answers to the question about how strong the influence of mutations in a sequence is on the change of structures.
We can classify the different goals and approaches as follows:

- A prediction of the influence, small changes (i.e. a few point mutations) in the sequence have on the structure.
- A complete prediction of the overall structure from the knowledge of the sequence (and solvent) only, termed "the great challenge" (Results will be announced in "spot the looney" [16] .)
- The ultimate goal: a complete prediction of the functionality and the activity such as interaction of enzymes and substrates.

Computer calculations, based on heuristic or exact prediction techniques provide us valuable answers about some aspects like the dynamics of short periods, a detailed description of the problem will be given in the next section, the answer is 42.

1.5 Why Lattice Polymers ?

A frequently used method in science is to simplify the problem by reducing the problem to a few parameters, that are considered to be crucial. One method is, to use fixed torsion angles and bond lengths, that is representing structures on a lattice, preferable a regular one. Another simplification can be achieved by reducing the spectrum of interactions inside a protein to one force that is assumed to be the major driving force in structure formation, the hydrophobic interaction.
Interestingly lattice protein models have been used frequently very recently for simulations of protein folding and re-folding [15, 87]. Our concept of combining the model of LPs and the concept of landscapes offers a way out and will be explained in the following sections. To apply

the concept of landscapes, it is not necessary to know the ground state, i.e. the minimum free energy conformation. As soon as we dispose of an algorithm that is fast, yields structures according to an energetic criterion and are *relatively* robust towards mutations, we need not find the global minimum.

1.6 Organisation of this Work

First we give a short description of structure formation and description of biopolymers in general and of RNA secondary structures and folding algorithms in special. We continue with a description of the folding problem of proteins. Then we explain the concept of landscapes and it's implications to this work more thoroughly. Next we do the same for proteins and close the section with a short comparison of computational methods and conceptual differences and similarities.

Then we present some of the calculations we performed to describe the sequence to structure mapping for RNA secondary structures.

Then a survey of lattice polymers is followed by a description of several methods to characterise and compare structures. We continue with the description of a novel chain growth algorithm, being the core of this thesis. We investigate the features of folded structures and compare them to known data of ground states from [99].

A package, as developed for this work is shortly described. Next are the computations for self avoiding walks and lattice proteins.

In the appendix, among some data tables we finally provide the manual for the toolkit. (It was designed for this work in cooperation with [86].)

In order to protect the wisdom in this work we close with the killing joke [16].

I: General Background

2 Structures and Folding of Biopolymers

Biopolymer sequences are objects of combinatorial complexity. The enormous number of possible sequences is a characteristic feature and the ultimate basis of the variability we observe in biology. The structure is imposed directly and only by the sequence.
Available resources for an exact forecast are still far out of reach and methods anything but reliable.
For any prediction strategy there is a tradeoff between accuracy and the amount of conformational space that can be searched. Computational simulations of real biophysical landscapes have to tackle the very difficult problem of transforming the genotype into the phenotype. At present no reliable algorithms exist which predict the tertiary structure from the primary sequence for both, RNA and proteins.
Some methods to investigate biopolymer folding, as well as their limitations and advantages are discussed in the following chapters.

2.1 The Folding Problem for Biopolymers

A usual way in science, coping with a problem which is too large to be solved in one step is to break it down into smaller parts that can be solved more easily: Fortunately this can be achieved easily in the case of RNA

1. Folding of the primary sequence into the secondary structure consisting of (almost) all base pairs.

2. Formation of the spatial structure by folding the quasi-planar secondary structure into a three dimensional object.

3. Further simplification can be achieved by using strongly simplified energy rules.

In the case of proteins we will use the lattice approximation.
It has been shown that the secondary structure model is not a mere theoretical construct but somehow reflects reality: secondary structure elements are conserved in evolution, for example the clover leaf structure of tRNAs class I introns [9, 56]. This conservation throughout evolution is even exploited to determine the secondary structure of RNAs by means of the phylogenetic method. But there are also many technical reasons to investigate ensembles of RNA with their

secondary structures. In contrast to spatial predictions, secondary structures can, within narrow limits, be calculated by theoretical means as there are a lot of well established and reliable methods.
Compared to spatial information the storage requirements are very small for secondary structure. Large ensembles can be studied by computations. Computer simulations make it feasible to investigate more than some single peaks in a landscape. This is the major restriction for experimental investigations, based on gene technological methods. Randomly distributed sequences coincide with respect to their distribution of secondary structural elements with natural examples of RNAs, where secondary structure does not play an important role like in m-RNA.
Due to the fact that random strings are not the result of any optimisation processes – such as evolution is – they can serve as a good reference for further investigations on natural or other artificial strings with certain sequences, outstanding structural features and their behaviour towards mutations.
Unfortunately an analogy to the secondary structure model is far more difficult to apply to proteins – the concept of a secondary structure being more vague because contrary to RNA where a base pair is defined in a all or nothing way (either base i is paired with base j or it is not) there exist a continuum of interactions with no clear cut-off value of the binding energies. The only advantage we have in the case of proteins is the - relatively - large number of known structures from X-ray analysis that are often used to develop heuristic-based prediction strategies. In the case of RNA, tRNA is probably the best examined example, as quite a few structures have been analysed by X-ray crystallography [51].

2.2 Kinetic Folding and Local Minima

In order to predict the functionality of biopolymers, most interest is focussed on determining the structure that corresponds to a stable energetic minimum, preferable the global minimum which is often supposed to fall together with the native state.
Due to thermodynamic fluctuations biopolymers do not exist in one, well defined state only. For reasonably rigid molecules, characterised by the fact that there is a unique local (and hence global) minimum in the part of conformational space accessible to the molecules, this determines the kinetics of the molecules up to temperatures sufficiently high to break some of the chemical bonds, thus altering the nature of the molecules. Therefore, the geometry defined by the minimum of the potential energy surface, referred to as the *stable state* of the molecule, gives a correct geometric description of the average shape of rigid molecules.

However, proteins, as other polymers, are not rigid but can be easily twisted along the bonds of the backbone. Therefore the potential energy surface becomes very complicated and exhibits a large number of local minima. Random excitations allow the molecule to escape from the neighbourhood of one local minimum and reach the neighbourhood of another one. Local minima of the potential are *meta-stable states*, and the occasional escapes to other local minima are referred to as *state transitions*.
The frequency of transitions depends on the temperature and on the energy barrier along the energetically most favourable path between two adjacent local minima. Any such path has its highest point on the energy surface at a saddle point called a *transition state*. The energy difference ΔE between the energies at a local minimum and at a transition state is called the *activation energy* for the transition. The *Arrhenius law* predicts the mean transition frequency k as

$$k = \frac{RT}{h} \exp\left(-\frac{\Delta E}{RT}\right)^{\ddagger} \tag{1}$$

The exponential term shows that transitions become vastly more difficult and more rare as the activation energy increases, while at higher temperatures, transitions become easier.
One may consider two adjacent local minima with potential energies E_1 and E_2 and the corresponding transition state with energy E_{12}. (See Figure 2 for a cross section of the energy surface along the reaction coordinate.) Suppose that $E_1 < E_2$. The activation energy $\Delta E_{1\to 2} = E_{12} - E_1$ needed to cross from state 1 to state 2 is larger than the activation energy $\Delta E_{2\to 1} = E_{12} - E_2$ needed to cross from state 2 to state 1. Using the Arrhenius law we find that the corresponding transition frequencies satisfy

$$\frac{k_{2\to 1}}{k_{1\to 2}} = \exp\left(-\frac{E_2 - E_1}{k_B T}\right)$$

independent of the transition state energy. In particular, unless both metastable states have nearly the same potential energy, transitions from the state with the higher energy to that with the lower energy are *much* more frequent than transitions to a higher energy level.

This implies that over sufficiently long time scales, a molecule spends most of its time in the deepest valley near the global minimum of the potential, once it has attained this conformation. It also shows that in a collection of many (independent) molecules of the same kind, most molecules are in a conformation close to the state with absolutely smallest potential energy. The overall distribution of states occupancy is given by the Boltzmann distribution. This is the reason why it is commonly expected that the geometry defined by the *global* minimum of the potential energy surface is the correct geometry describing the conformation observed in folded proteins. However, if the energy barrier of a meta-stable state is sufficiently high,

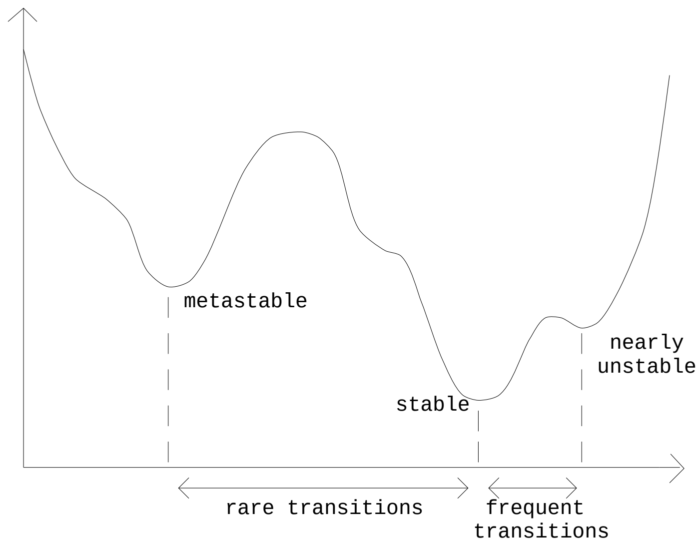


Figure 2: A meta-stable state with high energy barrier

transition frequencies may be so low that within a biologically meaningful time the meta-stable state behaves like the stable state given by the global minimum.

In a molecular system (such as a protein) where the atoms are highly mobile, the potential energy surface has a complicated shape. Therefore it seems impossible to attain the ideal of studying all features of the dynamics; in practice one must be content with the exploration of sample paths through the state space by means of so-called *molecular dynamics calculations* Most state of the art simulations are so-called molecular dynamics, simulating short 'time trajectories by the changing the configuration and evaluating the structure with force-field calculations. These are mainly based on a set of equations for each type of bonds. Atoms far apart are subject to non-bonded interactions: If both atoms carry partial charges, there is the long range, slowly decaying *electrostatic (Coulomb)* interaction, and for all pairs of atoms there is the short range, fast decaying *van der Waals* interaction. The Coulomb interaction is modified by polarisation effects due to the presence of solvent, and this is modelled by a *dielectric constant* D. Hydrogen bonds and non-bonded interactions are also relevant for the interaction of the molecule with the atoms of the solvent (water). However, in many studies, solvent effects are ignored (or at best accounted for implicitly by the choice of the dielectric constants in the model).

The most important force that stabilises proteins is considered to be the hydrophobic potential [14, 83, 96], though no-one seems to be able to give an exact description. Most probably the "hydrophobic effect" is caused by influence of the molecule on the solvent structure. Con-

sequently in real proteins, residues that are termed hydrophobic can be found in the core in aqueous solution and polar ones are rather located on the outside.
Following above mentioned findings and several other concepts (for reviews see [37]) we can outline the following concept on the folding of proteins under kinetically controlled constraints: biopolymers obviously exist most of their lifetime in the global or, at least a *very* low local energetic minimum structure. Most likely there are directed pathways leading the newly synthesised biopolymers to a subspace in the huge shape space, that includes the final state. This process may also be assisted and directed by other molecules such as chaperones and chaperonins that, in case of proteins, prevent the molecules from folding and associating too early. The final state can then be obtained by state transitions within a reasonable time scale.

2.3 A Short Detour to RNA

In the last two decades it has become easy to sequence nucleic acids and impressive attempts are being made nowadays to sequence the whole genome of eucaryotic organisms, even the human genome with its approximate three billion base pairs. Apart from the severe ethical problems which are associated with this extension of the human knowledge and the questions of accessibility and property of information there remains the, comparatively minor, scientific problem of exploiting the information present in the four letter code of the sequences.
RNA is a very suitable molecule for evolutionary models because it combines several important features:

- Genotype and Phenotype are different features of the same molecule
- The phenotype can be computed from the genotype
- The origin of life is, most probably, tightly associated with RNA (see above).
- Evolution experiments are performed with RNA in the test tube

2.3.1 Components and Structure

An RNA molecule is a biopolymer built up from different ribonucleotides in a sequence, linked together by covalent phosphate bonds that form a backbone. These nucleotides contain a sugar, *ribose* and, linked to it's 1'-OH group, one of the nucleotides that are derivates either of purine or pyrimidine. The nucleotides namely are the purine derivates **A**denosine and **G**uanosine and the pyrimidine derivates **U**ridine and **C**ytidine respectively. The phosphate group is attached to the 5' position of the riboside, which itself is connected to a another phosphate group, attached to the 3' position of the ribose of it's neighbour.
The *symbolset* simply denotes the volume of nucleotides the RNA is composed of. The natural symbolset, termed *biophysical* (BP) is composed of **AUGC**, simplified symbol-sets that could stand for archaic examples in a prebiotic world and represent the two extremes of binding strength are **AU** and **GC**. In DNA **T** appears instead of **U** and *d-desoxyribose* instead of ribose.
Three hydrogen bonds between **G≡C** cause a much stronger bond than between **A=U**, a substantially even weaker binding strength can be observed between **G-U**. Artificial symbol-sets have been used for the investigations of intrinsic statistical features in earlier projects of our group [31, 28, 32, 29].
ABCD (synonymous to **ACKX**) and **ABCDEF** are symbol-sets that have been used to investigate the change of special features with increasing number of possible matching base-pairs

(2 and 3 respectively) without the complications of different base pair strength an the additional non-standard **G-U** interactions as in the biophysical set. Here the binding strength is assumed the same between all pairs, usually the one of **G≡C**. In natural occurring RNAs, for example in tRNA several different bases, that have been posttranscriptionally modified can be found.

2.3.2 The Secondary Structure Model

There are several reasons for considering the secondary structure as a crude first approximation to the spatial structure of the RNA.

- Conventional base pairing and base pair stacking cover the major part of the free energy of folding.
- Secondary structures are used successfully in the interpretation of RNA function and reactivity.
- Secondary structures are conserved in evolutionary phylogeny.

Folding can conceptually be divided into two steps, both following the minimum free energy criterion: 1)Folding into the secondary structure and then 2)Formation of the spatial tertiary structure by folding the quasi planar secondary structure into a three dimensional object. *RNA secondary structures* are defined as the list of Watson-Crick base-pairs that minimise the free energy. An example of the planar graph is shown in Fig. 3.

In contrast to proteins the secondary structures largely determines the tertiary structure. In the case of RNA secondary structures are from now on simply termed as *structures*.
The secondary structure model is not a mere theoretical construct. It's biological relevance has been proved by the conservation of secondary structural elements during evolution [9, 56]. Elements of secondary structure such as multi-loops are the simplest examples in biological entities that, once established, can hardly vanish by further mutations and hence stand for the unidirectional run of evolution. Hence RNA is the simplest known model, where it is able to determine how structural and functional similarity are related. Several examples are known from experimental data, where during the evolution functional properties that are closely related to structural features are highly conserved. The CUUCGG motif appears quite often in structural hairpins that apparently occur in termination sites [108]. 5s rRNA secondary structures appear to have a Y shape [93]. tRNAs appear to have a clover leaf secondary structure. Here the same shape can be observed even in the case of different underlying sequences.

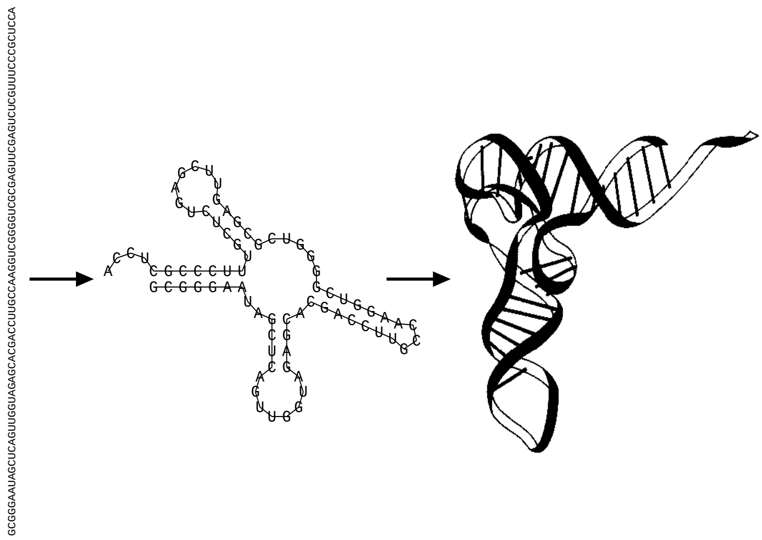


Figure 3: RNA folding from a given sequence (left) can be conceptually partitioned into two steps: formation of the secondary structure (middle) and from there the spatial structure (right). The secondary structure can be viewed as a planar graph with a unique pairing matrix. Shown example is Phe-tRNA, it consists of four stacks, three of them closed by hairpinloops (H), one large multiloop (M) and one free end (E) with 4 nucleotides. Denoted secondary structure can be symbolised by fine graining (see text for eplanation): ....((((((((((((.......))))).....(((((((...))))))).((((........))))..))))))) and coarse graining (where only loops are depicted and stacks are omitted): E(H(M(H)(H)(H))).

Structure is essential for enzymatic interaction and modulated by slight differences, mainly in the length of the variable loop. It has to be stressed here that tRNAs are not a good example for theoretical analysis on randomised ensembles as they contain a lot of posttranscriptional modified bases.

RNA secondary structure of viruses is known to influence the replication velocity along the strand and the overall stability against hydrolytic degradation.

The thermodynamic stability correlates with statistical significance [56].

In the case of RNA molecules, and if one is willing to accept base pairing patterns as a suitable (coarse grained) description of the structures, one can actually compute the structure of minimum free energy for (in principle) arbitrary sequences [120, 119, 65, 43]. These algorithms are based on a simple thermodynamic model of RNA (secondary) structures, for which the majority of parameters has been measured directly on small oligonucleotides [35]. The simplicity of the energy model and the relatively small number of contributions in a given sequence allow this approach to be applied successfully.

A secondary structure can be conceptually decomposed into:

1. Double-helical **S**tacks, the only elements with stabilising contribution to the overall energy.
2. **H**airpin loops, **M**ultiloops, **B**ulge loops, **I**nterior loops and
3. External elements such as **J**oints and **F**ree ends.

2.3.3 Structure Representations and Structure Distances

The notion of secondary structure is but one of a spectrum of possible levels of resolution that can be used to define shape. Atomic coordinates as well as relative spatial orientation of the structural elements are discarded, only the topology of structure elements is retained.
Secondary structure graphs can be mapped one to one to strings and trees. In this contribution we use two levels of structural representation:
Fine graining: The string representation is obtained by denoting each unpaired base by a dot, each upstream paired base by an open parenthesis, and each downstream paired base by a closing parenthesis. It is equivalent to the mountain representation and tree representation [29].
Coarse graining is a symbolic notation where all distinguishable elements of secondary structure from (2) and (3) (or (2) only) are symbolised in hierarchical order. It is therefore possible to focus on the major structural features of the RNA molecule. These are the easiest and most efficient representations for string comparison and sorting by computational means so that structure comparison and alignment can be achieved in a similar manner as for sequences.
While the energy difference between two structures can be obtained by simple subtraction we need well defined measures to compare distances between sequences and their corresponding structures. For any scalar quantity $F(x)$ that can be assigned to a sequence x (or its structure) we can define *Local Optima*: points x such that function $F(x)$ for all neighbours (i.e. one error mutants) y one has $F(x) \geq F(y)$. Series of random searches of point mutations (prohibiting back-mutations) towards better solutions of $F(y)$ are called *adaptive walks*. The number of consecutive steps until the series is trapped at a local optimum is called the *walk length*, and is an important measure of the performance of optimisation processes.

2.3.4 Neutrality

Neutral neighbours are one or two error mutants that fold into the same structure as some reference sequence. Obviously it depends on where the mutation is located in regard to the

structure: in a stack 2 simultaneous base exchanges are required to maintain the structure, in an unpaired region a single exchange can be sufficient. Consequently we have two different paths that lead to the same 3 - error mutant and a large number of possibilities for each single step in this simple network.
Neutral nets are sets of connected neutral neighbours. *Neutral paths* are subsets with monotonously increasing Hamming distances from a reference sequence.

2.3.5 Folding Algorithms

Several different strategies have been developed to predict RNA structure. For a detailed survey see [105]. As these attempts and methods are a good example for computing of biopolymers structures and several computational optimisation experiments in general we give a short description of the main features:

Phylogenetic Comparison: is the method of choice for determining secondary structure provided that there are enough RNA sequences from different organisms available. It is assumed that function depend on structure and therefore structure should be conserved in evolution. Based on the assumptions that in the course of evolution mutations occur which do not alter the function of the molecule and are preserved methods have been developed to reconstruct from covariances of nucleotides the secondary structure [78] . The predictive capability of the method increases with the availability of homologous sequences and so the playground of phylogenetic comparison is the 16S rRNA and the 23S rRNA molecule [42], where hundreds of rRNAs in different species have already been sequenced. Improved methodology makes it possible to study higher order correlations and to predict pseudo-knots and other tertiary interactions [41].
Similar methods have been applied to obtain value functions to check the quality of a proposed structure for a given sequence [7, 100].

Minimal Free Energy Folding: algorithms find the secondary structure with the lowest free energy (surprise, surprise). These folding algorithms are the most frequently used, although experimental data are not accurate and RNAs very often run into a kinetic trap during folding.

Experimental data is available for bulges, interior loops and hairpin loops but not for multi loops. Different parameter sets give different energy values, for a detailed comparison see [105].

- Combinatorial Algorithms: The easiest possibility to obtain the structure with the minimum free energy is to calculate all secondary structures and choose the best. Unfortunately the number of secondary structures grows with at least 1.8^{ν} for large ν with

A,U,G,C being randomly distributed with equal probability [119]. The number of possible helixes though increases with ν^2, ν being the chain length of the sequence and thus the number of possible helix combinations grow with 2^m, $m = \nu^2$. (This problem is $NP-complete$, in other words, no search procedure exists which can, in the worst case, solve it in less than exponential time.) This method can in principle predict pseudo-knots as well as tertiary interactions. Above mentioned objections do not justify such an approach.

- Dynamic Programming: The general strategy of these algorithms is to determine first the optimal structure of the smallest sub-fragments of a given sequence and then calculate the optimal structures of the larger fragments by adding together optimal sub-fragments.

  Crucial to these algorithms are some assumptions:

  - Unique decomposition of the problem is possible The free energy of a secondary structure is the sum of the free energies of the substructures.
  - No pseudo-knots
  - No interaction between substructures is allowed.
  - It is assumed that the main energy contribution of the secondary structure stems from the Watson Crick base pairs and the **GU** base pair.

  The following algorithms were used in this work:

  - Maximum Matching (MM) The first successful application of the dynamical programming algorithm to the prediction problem of RNA secondary structures dates back to 1978 [73]. Ruth Nussinov presented a computer method which maximised the number of base pairs in a secondary structure with the restriction of a minimal loop size of 3. In our implementation of the programming algorithm it is possible to tune the minimal stack length as well. Maximising of base pairs is a very unphysical way of RNA folding because no loop destabilising energies are taken into account. But it is very fast and it was the starting point for the developing of more realistic algorithms. We used the MM algorithm for reasons of comparison because it represents a folding algorithm without any energy parameters at all. It can be viewed as an abstraction of the mapping from the sequence to structure, tuning the minimum stack and loop size can be viewed as a simulation of the thermodynamic requirements to form a stable stack or loop respectively. As there are no differences in the energies of AU and GC base pairs, any differences in statistics of structure formation must result from the size of the sequence space and the stickiness.

– Minimum Free Energy (MFE) Folding:

The dynamic programming folding algorithm used in this work has been developed by Zuker and Sankoff [119, 44]. A program package, the `Vienna RNA Package`, written in the `C` programming language running on `UNIX` systems is available which includes most source codes [43]. It is available via anonymous FTP from `ftp.itc.univie.ac.at`.

It's basic idea is to determine the optimal substructures first and than to calculate the overall value of the free energy for the molecule by adding up the stored energies from the optimal sub-fragments. Other interactions than those of the nearest neighbour are not taken into account as optimal substructures would not lead to an optimal overall folding.

The runtime is cubic with n which means a limitation for the calculations that are based on an extensive use on large ensembles of random strings as used for the present contribution. Experimental values are applied in units of 0.1kcal to avoid floating point operations and can thereby accelerate the algorithm. Still a limit can be reached very quickly on extensive usage and CPU - time be exhausted.

Apart from the rules listed above, the following constraints are imposed on base-pairing:

1. *There are at least 3 intervening bases between base-paired ribonucleotides.* Hence the backbone cannot fold back too sharply on itself and, if $i : j$ form a base pair where $j > i + 3$. Smaller hair-pins have not been observed in natural sequences and are prohibited in the computations for the folding algorithm by assigning large destabilising values.
2. *The graph is unknotted and planar.* Structures are termed pseudo-knots if $h < i < j < k$ and $i : k$ and $h : j$ respectively form base-pairs. The problem of pseudo-knots is relegated to tertiary structure representation and therefore of no relevance to secondary structure. This means a major restriction to the equivalence of secondary and tertiary structure as pseudo-knots form a common motif in natural RNA folding [82]. But energy assignment is difficult as no experimental data are available on how knots affect the molecule's stability and the algorithm would be substantially slowed down.
3. *The secondary structure can be uniquely decomposed into substructures* which contribute additively to thermodynamic and kinetic properties of the complete molecule. This allows the application of a recursive, dynamic algorithm, which first constructs the structure of substrings and then builds up the entire struc-

ture. If the structure of sequence **I** is optimal, i.e. $f(I)$ is minimal, any substring $\mathbf{I}_{ij}$ is optimal too, provided that (s_i, s_j) form a pair.

4. *Only complementary bases can form base pairs in the Crick-Watson scheme.* The only non Watson-Crick base pairs that have been taken into account optionally are **G-U**.
5. *The free energy data have been derived from melting studies* on small chains and are available for bulges, interior loops and hairpin loops [35, 44, 81, 102, 103].
6. *In the case of multi-loops a linear ansatz has been used:*

$$\Delta G^{\circ}_{37}\mathrm{ML} = a + bn + ch \tag{2}$$

where n is the number of unpaired nucleotides in the loop, h the number of helixes and a, b, c are adjustable parameters, the values of which were determined by comparison with experimentally examined structures and the prediction of the folding algorithm.

Other Algorithms:

- Equilibrium Ensembles:

  Ensembles of RNA molecules show a Boltzmann distribution of secondary structures. The equilibrium partition function contains all information about this equilibrium distribution.

  The full equilibrium partition function can be calculated recursively with the dynamic programming approach with ν^3 time requirement [65]. The temperature dependence of the partition function gives information about the melting temperature of the RNA [4, 105]. A matrix display of the probabilities of base pairing between different positions is very useful for assessing base pair slippage, alternate secondary structures and the stability of the dominant secondary structure. It is in general not possible, though, to reconstruct the minimal free energy structure from the base pairing probabilities.

- Deterministic Kinetic Folding:

  These algorithms try to simulate the actual folding of RNA during transcription assuming that the biologically active RNA structure is not the minimal free energy structure. The earliest attempt to solve the kinetic folding problem was made by Martinez in 1984 [64]. His approach was first to calculate all possible stacks compatible with a given sequence and then generate the secondary structure by sequentially incorporating the stacks with the highest equilibrium constants. Sequential growth of RNA secondary structure was based on the local initiation of folding in some regions, called "nucleation centres". It is assumed that the secondary structure obtained does not refold.

- Nondeterministic Kinetic Algorithms: The folding process is described by a Markov process which is modelled by a Monte Carlo method. It is argued that the folding process depends on the rate constant of helix formation and on the rate constant of helix decay and not on the equilibrium constant as in the Martinez approach [67].

  The great advantage of this method is that it gives a Boltzmann weighted equilibrium ensemble of secondary structures. The disadvantage is that it is very time consuming and that many calculations have to be performed for folding a single molecule.

- Sequential folding [74] was proposed to consider step-by-step process of secondary structure growth and reorganisation during sequential growth of the molecular chain with fixed steps of lengthening. At each step the structure was sought for by global minimisation of free energy. This concept resembles the model that will be introduced for lattice proteins in section 6.

2.4 A Detour to Proteins

2.4.1 Primary Structure

The proteins in living cells contain 20 different residues (and some rare derivates), with side chains having 1-18 atoms. Their properties, such as hydrophobicity, polarisation, size etc. are the reason why proteins attain unique structures and function. The residues are usually abbreviated with three identifying letters of the corresponding amino acid:

$$\mathcal{AA} = \{\text{Ala,Arg,Asn,Asp,Cys,Gln,Glu,Gly,His,Ile,}$$

$$\text{Leu,Lys,Met,Phe,Pro,Ser,Thr,Trp,Tyr,Val}\}.$$

A summary of some important properties of amino acids and the one letter code is given in the appendix. Under the influence of RNA containing the genetic information coding for the amino acid sequence, amino acids polymerize in a specific sequence to a chain with the structure as given in Figure 4. Bonds joining two residues (called *peptide bonds*) hydrolise (i.e., break under consumption of a water molecule) in a sufficiently acid environment, and this can be used to determine the precise sequence of residues in a given protein.

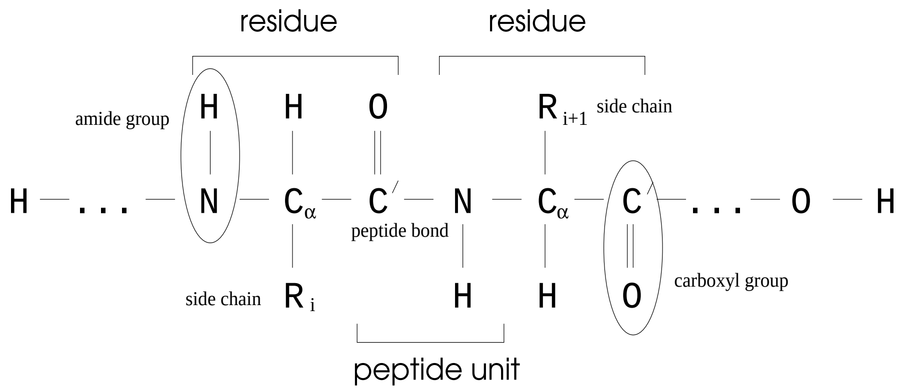


Figure 4: The chemical structure of a protein: a list of amino acids, linked by peptide bonds.

The repeating $-\mathrm{NC}_\alpha\mathrm{C}'-$ chain of a protein is called its *backbone*. Inter atomic forces bend and twist the chain in a way characteristic for each protein. They cause the protein molecule to curl up into a specific three-dimensional geometric configuration called the *folded state* of the protein. This configuration and the chemically active groups on the surface of the folded protein determine its interaction and selectivity.

The smallest known peptides of biophysical relevance, hormones, have about $10-100$ residues (though e.g. solitude glycine molecules play an important role as a neurotransmitter),

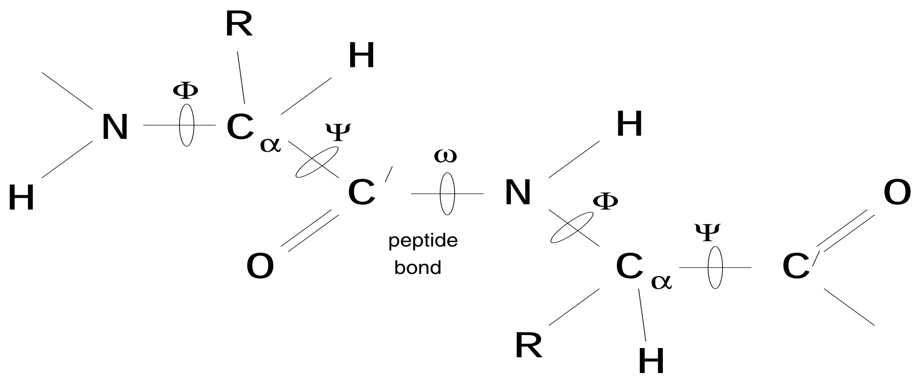


Figure 5: Backbone dihedral angles of a protein: there are two angles of torsion, the peptide bond itself is fixed.

typical globular proteins about $100 - 500$; fibrous proteins may have more than 3000 residues. One of the smallest proteins, BPTI (bovine pancreatic trypsin inhibitor), with 58 residues and 580 atoms only, has become a well-studied model protein from both the computational and the experimental point of view; very accurate data for the crystal structure are available. Another small protein that has found considerable attention is Crambin (with 46 residues). It has already been subjected to investigations on the lattice approximation [15].

2.4.2 Levels of Structure Formation

Most essential for structure formation is the fact that 2 angles between each two residues are flexible, the peptide bond itself however is rather fixed (in the range of $170^o - 190^o$) because of the mesomerie (between $R_2CH - CO - R$ and $R_2C = C(OH) - R$). A full set of bond lengths, bond angles and dihedral angles already fixes the geometry of a molecule. However, the geometry is quite sensitive to small changes in the angles, and, to reduce the sensitivity, it is useful to specify in addition a number of so-called *out-of-plane bending* or *improper torsion angles*, which are defined in a similar way for any tetrahedron formed by an atom j with three adjacent atoms i, k, l. Under biological conditions, the bond lengths and bond angles are fairly rigid, and the dihedral angles along the backbone (usually labelled as in fig. 5) determine the main features of the final geometric shape of the folded protein.

Unlike the RNA case the definitions are rather vague on that point, especially between secondary and tertiary structure. *Secondary structure* is not a clearly defined state, but regions along the sequence with nearly the same geometry, i.e. angles of torsion and spatial relation are termed α helixes, parallel β-sheets, anti-parallel β-sheets and ω-turns. Frequent arrangements

of secondary structural elements, such as the $\alpha_2\beta_2$ tetramer of the GABA receptor are termed *folds* if they fall into the same class, i.e. their RMS (root mean square distance) is small. For each pair of amino acids there are several regions of torsional angles that are forbidden or preferred. These can be illustrated in a so-called Ramachandran plot (see fig. 6).
The *tertiary structure* stands for the overall geometry, the *quaternary structure* for the orientation of independently subunits without peptide bond between each other.
Domains are areas within the molecule that fold independently but may contain several different patterns of secondary structural elements. A similar description of "contiguous regions" that can be easily distinguished from the rest of the molecule and arise from a simple nucleation event has been given by Wetlaufer [114] and the size has been suggested to lie in a range between 40 and 150 residues.

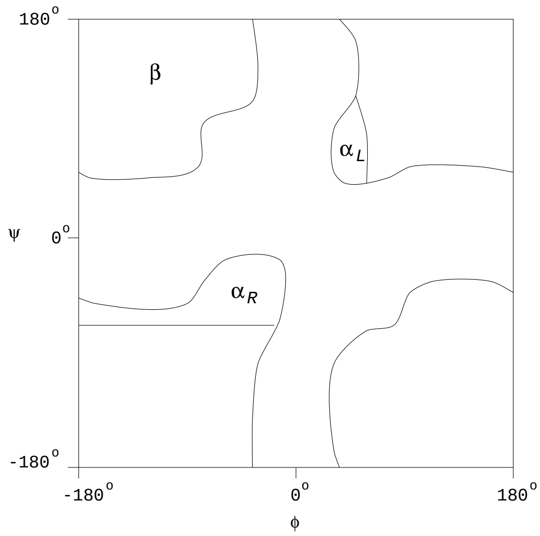


Figure 6: A Ramachandran plot. The region centered near $(\phi, \psi) = 0$ is forbidden; the regions designated α_L, α_R and β correspond to left-handed helixes, right-handed helixes and sheets, respectively.

2.4.3 The Protein Folding Problem

As mentioned above, the structure prediction of proteins is of paramount interest. The term "protein folding problem" itself has become a terminus technicus that is implicitly synonymous with associations of an - abstract - ultimate goal of all science. It therefore attracts also the interest of a great number of scientists from other fields, culminating in numerous interdisciplinary collaborations, projects and obscure obsessions.

The first experiment by Anfinsen proved that a denatured protein can fold back to the

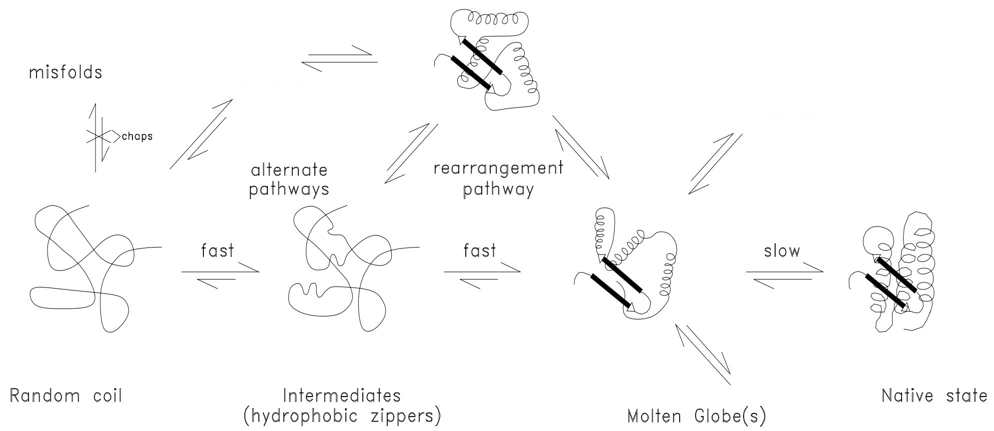


Figure 7: The Folding of Proteins can be illustrated by the scheme: A given structure, e.g. the random coil of a denatured protein or a newly processed chains starts the formation of secondary structure elements by local collapse (e.g. hydrophobic interactions may cause the formation of an α-helix) to a population of "Molten-Globe" states with similar energies. Most of the final secondary structure is already formed. The Molten Globe states are relatively compact, however contains still a lot of cavities with water and much accessible surface. From there follows a - relatively slow, and therefore time determining step for the overall process - reformation to the final native state, probably ground state.

native state under the right condition [101]. Mercaptoethanol links to the sulphur of cysteins instead of a cys-cys disulfide bond if the protein is denatured by urea. The enzymatic activity can be (at least partly) regained if the both components are removed by dialysis. When re-oxidation took place before removing the urea, only a minimal enzymatic activity is measured. It corresponds to the proportion of statistically correct linked disulfide bonds out of the number that is possible from combinatorial reasons. It was therefore assumed, that proteins attained their global minimum in a sufficiently short period of time from the random coil state and without any further help.

Unfortunately an exact forecast from the sequence only is not possible. This is due to uncertainties about the adequate formulation of the problem on one hand, to insufficiencies of the methods such as molecular dynamics and to computational limitations. An extensive review on these aspects has been given recently [72].

Even simplified models, such as the lattice approximation has been proven to be NP complete very recently [33, 109]. This fact simply states, that the computational effort for a method to find the global minimum in any case is equivalent to other problems that are supposed to be

unsolvable in polynomial time. Typically the effort increases exponentially with the size of the problem.
NP completeness however only denotes a worst - case behaviour and, depending on the problem it might well be that a useful solution, possibly even the global minimum can be found within reasonable computational time. As proteins, however, find their minimum in finite time, there must be some other methods that are applicable.
Very often therefore the goal is rather to understand *how* the folding process works. Levinthal's paradox is the most striking finding that of molecular dynamics properties: to find the lowest energy state of a protein, by simple trial and error of all possible configurations would take even small proteins astronomic time scales [57]. Actually proteins do not fold by a random search. In fact the ground state is found within a relatively short time, i.e. seconds or less. Levinthal proposed the existence of a directed folding pathway. Several assumptions to circumvent the difficulties expressed by the formulation of Levinthal's paradox have been formulated recently. Zwanzig [121] offered a reasonable explanation very shortly, showing how a small bias that propagates a folded state along the chain dramatically increases the overall search speed.

The first step is looked as the collapse of hydrophobic regions into the interior of the molecule [21]. Followed by the formation of stable secondary structures, that are stabilised by long range interactions such as hydrogen bonds and finally the formation of covalent interactions. Subsequent folding seems to occur through a limited number of pathways involving distinct intermediates (molten globules, see following paragraph). Those intermediates seem to exist at various states of similar energy, quickly re-folding each other. The rate determining state occurs at a very late state, shortly before getting to the stable native state. At this point the molecule gets more compact by expelling H_2O. In fact it has not been proven yet, if the native protein conformation really is the MFE one [110].
Much attention nowadays is payed to the so called *Molten Globe* state, that is thought to be attained quickly after the formation of the first structural elements and following the above mentioned pathway. Random, or non-natural sequences also fold to these states and then, unlike "folding sequences", remain there. Folding sequences, however, start a relatively time consuming search to their final state. The molten globe can be characterised as follows:

- a high number of short versus long range interactions [114]
- a high content of secondary structure [50]
- a strong correlation between the native contact map and interaction parameters [38]
- a high number of low-energy states with near native conformations

- a lot of cavities with water and much accessible surface.

It should, however not remain unmentioned, that nearly each of these steps is doubted to be correct by someone's findings. An example, worth to be mentioned is the molecular dynamic studies by Mark and Spencer [63] on Hen Egg White Lysozyme: thereby the re-folding procedure indicated an initial collapse of the chain first and the formation of secondary structure later on during the folding process.
Guo et al. [40] report that the folded states of proteins are not necessarily the maximal compact ones, the same ambiguity appears in discussions about the role of hydrophobic interactions. Furthermore there is evidence, that several globular proteins have nearly no secondary structure at all [46].

3 Landscapes and Complex Combinatory Maps

3.1 Fitness Landscapes

A very useful method of thinking in evolutionary terms was introduced in the early 1930s by Sewall Wright. He coined the term fitness landscape [115]. By fitness landscape he defined a mapping which assigns to each genotype a fitness value. Evolution is understood in this terminology as a complex optimisation process on the fitness landscape.

Unfortunately it is until now not possible to calculate the fitness values for species. Point wise characterisation of landscapes can be done experimentally with gene technological methods for both, RNA and proteins. A global characterisation is feasible only by computer simulations.

The concept of landscapes represents a powerful instrument to study a wide field of optimisation problems. By application of this concept to the theories of early evolution it is possible to calculate in principle the fitness value of a mutant in a set of biopolymers a priory. Landscapes are not the biological equivalent to physical potential energy surfaces. They are deformed in response to environmental changes [49]. Furthermore fitness is not a property of an individual genotype but is dependent on the population size of other genotypes - fitness is an ensemble property [91] and the target of selection is the entire quasi-species. In the case of coevolution two or more genotypes cooperate to obtain higher fitness values [97].

3.1.1 The Sequence Space of Biopolymers

The configuration space of RNA molecules is the total of all possible sequences of a given length where each node in the representing graph stands for one sequence. The configuration space for a two letter alphabet can be represented by a hypercube of the dimension n.

The number of possible sequences at a given error class can be calculated by the means of the following formula, the extended binomial distribution, where κ is the number of letters in the symbol-set, h the error class, the hamming-distance from a reference string, n the string length:

$$N_I(h) = (\kappa - 1)^d \binom{n}{h} \tag{3}$$

For sequences with $n > 20$ these functions have a sharp maximum at

$$\overline{h} = \frac{\kappa - 1}{\kappa} n \tag{4}$$

Consequently for two randomly chosen sequences will most likely have this very hamming-distance. The configuration space for **GC**-sequences of chain length n is a hypercube of dimension n and shown in the unavoidable figure 8

For $n = 100$ and $\kappa = 4$ (i.e. biophysical) $n_I = 1.6 \text{ x } 10^{60}$, which is far beyond any number of species in any population. The number of possible sequences is 'hyper-astronomical'.

As κ^n gives the total number of possible sequences in sequence space the probability $p(h)$ that two randomly chosen sequences have Hamming distance h is given:

$$p_I(h) = (\kappa - 1)^d \binom{n}{h} \kappa^{-n} \tag{5}$$

they are of very high dimension ($25 < n < 500$ for RNA molecules in test tube experiments, $250 < n < 400$ for viroids, and $3500 < n < 20\,000$ for (most) RNA viruses), but there are only few points on each coordinate axis. For the binary alphabet we have the hypercube of dimension n, see figure for the example with $n = 4$. The distribution of sequences around an arbitrary reference sequence is described by where κ is the number of letters in the alphabet and d is the index of the error class, i.e., the Hamming distance to the reference sequence. $p(d)$ increases exponentially for $d \ll n$, in sharp contrast to the quadratic or cubic increase on 2D or 3D lattices, respectively.

The configuration space, the total of all possible sequences at given n and κ, is termed the *sequence space*. More generally spoken there can be $\vec{x} = (x_1, x_2, ...x_n)$ in an n-dimensional space. The sequence space for a biopolymer landscape is therefore the space of all vectors that can be uniquely assigned to one sequence and therefore of dimension n.
A natural metric can be defined by the *Hamming Distance*, the minimum number of point mutations to get from one sequence to the other. It was first proposed in information theoretical problems.

3.1.2 Characterisation of Landscapes

In mathematical terms a landscape can be defined as a triple (X, d, f) where X is a finite set, $d : X \times X \to \mathbb{R}_o^+$ is a metric and $f : X \to V$ is a function which maps into a vector space with scalar product.
Neighbourhoods and distances on X can be defined by the means of the metric. The choice of the metric is not unique and therefore essential for the properties of the landscape, some metrics can heavily distort the statistical properties. Together with it's metric X is called the *configuration space.*
For all landscapes we refer to, X is assumed to be finite and the configuration space X has to be discrete therefore. V is in R, f is also called the *cost function* and used to evaluate the fitness of a species. (X, d_X) is for polymers the configuration space (or sequence space).

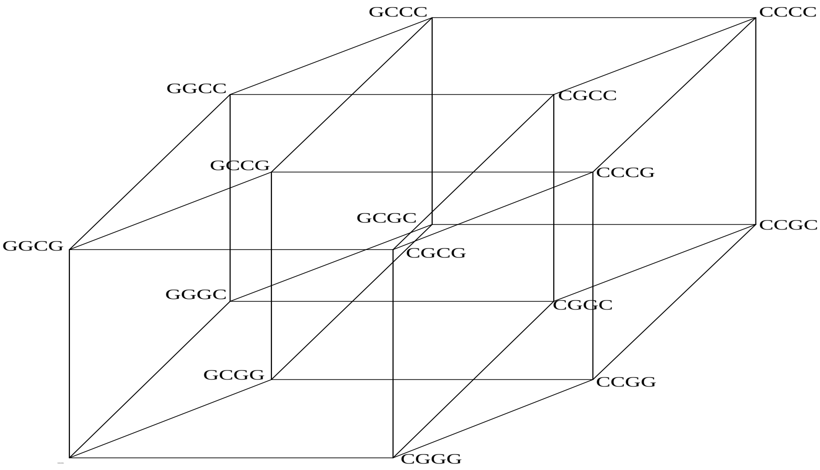


Figure 8: The configuration space of **GC**-sequences of chain length n is a hypercube of dimension n.

A suitable method to characterise landscapes is the autocorrelation function

$$\rho(h) = 1 - \frac{\langle D^2(f(x), f(y))\rangle_{h(x,y)=h}}{\langle D^2(f(x), f(y))\rangle_{random}} \tag{6}$$

where for example

$$D(f(x), f(y)) = \Delta G(x) - \Delta G(y) \tag{7}$$

is the difference in free energies for two sequences x and y. As analytical solutions are not available for most landscapes, we use large statistical ensembles of computationally folded RNA molecules to compute $\rho(h)$. This expression can be viewed as a measure of the average similarity of energies or structures etc. as a function of the Hamming distance h of the underlying sequences.

A useful measure for the ruggedness of a landscape is the correlation length $l = \rho(h) = 1/e$ which in turn is crucial for the performance of an evolution strategy. It is in the order of the average distance between two local optima and therefore a characteristic measure for the complexity of an optimisation problem. Together with other statistical features of RNA free energy landscapes they were extensively explored in our group [4, 28, 29, 32, 106]. We find energy correlation lengths that are longer than structure correlation lengths, both are very short compared to the diameter of the sequence space.

One of the most important characteristics of a landscape is the number $\mathcal{N}$ of local optima. A crude estimate is, that (for Gaussian landscapes) there is one local optimum within a ball with the radius of roughly the correlation length ℓ. While the number of local optima itself increases

Uncorrelated Landscapes:	$\rho(x) = \delta_{0x}$
Extremely rugged Landscapes:	$\rho(x)$ discontinuous for $x = 0$
Generic Landscapes:	$\rho(x) = 1 - \frac{1}{\mu}\lvert x\rvert + \ldots$
Nontrivial Fractal Landscapes:	$\rho(x) = 1 - \lvert x\rvert^{\alpha} + \ldots$

Table 1: Classification of Landscapes with $x = \frac{d}{\nu}$, adapted from [113]

exponentially the probability to hit a local optimum at random, decreases exponentially with n on correlated landscapes. A related statement is that the length of a gradient walk should be approximately the correlation length, while adaptive walks should be longer by a factor that depends on the structure of the configuration space only. Numerical evidence for the Nk model and RNA landscapes [90, 29] is consistent with this statement.
Local optima are obstacles for an optimisation algorithm. It is a very reasonable conjecture, therefore, that optimisation by a general algorithm should be easier if there are fewer local optima, or probably equivalently, if the nearest neighbour correlation is larger.
The correlation length is a statistical measure of the hardness of optimisation problems. The structure correlation length allows for a similar interpretation. The shorter the correlation length, the more likely is a structural change occurring as a consequence of mutation. The correlation length thus measures stability against mutation.

3.1.3 Classification of Landscapes

Analogous landscapes appear in many different fields in physics, theoretical biology and computer science. They can be classified [112, 32] according to their autocorrelation function near the origin (see table 1).

In biological and physical applications most landscapes and *CM*s are generic (or rugged) landscapes. These are characterised by exponentially decaying autocorrelation functions.
Prominent examples of such rugged landscapes are:

- Sherrington Kirkpatrick Spinglasses [52, 79]
- The Travelling Salesman Problem [55, 98]
- N-k model [48, 47, 49]
- Energy Landscapes of Proteins/RNAs [34, 28, 32]
- Fitness Landscapes [31, 30]

- Affinity Landscapes of Antibodies [59]
- Melting Kinetic Constants Landscapes [106]

Both, numerical experiments and analytical calculations show that the correlation functions $r(s)$ for model landscapes are (at least very close to) decaying exponentials. A real extensive summary with many more or less useful references is given in [90].

3.1.4 Evaluation of the Phenotype

To use the landscape concept for investigations of evolutionary behaviour of biopolymers we have to assign a 'fittnes value' to the sequence. A natural measure is every phenotypic parameter. Structures are non scalar objects, and there is no "structure landscape". Nevertheless, RNA secondary structures form a metric space Y by means of one of distance measures.

3.2 Complex Combinatory Landscapes and Maps

Assigning to each RNA sequence an energy value results in the RNA energy landscape the statistical investigation of which have been the focus of detailed investigations [28, 32].

A *Combinatory map* is a quintuple $(X, d_X; Y, d_Y; f)$ where X and Y are sets endowed with metrics d_X and d_Y respectively, and f is a map $X \to Y$.

In this definition the above defined landscape is a combinatory map with Y being the real numbers and $d_Y(a,b) = \mid a - b \mid$. A very important combinatory map is generated by mapping to each RNA sequence a secondary structure. This complex map of secondary structures was characterised in [32].

If the property to be assigned to the sequence space is a scalar quantity, like the free energy, the corresponding mapping Ψ into real numbers

$$\Psi : (X, d_h) \longrightarrow \mathbb{R} \tag{8}$$

is also termed a *complex combinatorial landscape* (**CCL**).
Landscapes of this class are generalisations of mappings from genotype into fitness values.
For free energy landscapes we can define the distance:

$$d(S_i, S_j) = d_f(i,j) = |F(S_i) - F(S_j)| \tag{9}$$

The free energy landscape is defined by the Hamming distance as the distance measure in sequence space and the differences in the corresponding free energy values that are determined by the folding algorithm. A different landscape is gained for each length and chosen symbol-set

[28, 32, 29, 4].

Another space of combinatorial complexity is the shape space. It is formed by the space of secondary structures S_k or trees T_k respectively with the tree distance $d_t(k,l)$ as the metric on it. As RNA folding can also be understood as a mapping Φ from one space, the sequence space X into another, the shape space Y:

$$\Phi : (X, d_h) \Longrightarrow (Y, d_t) \tag{10}$$

A mapping of this type is also termed a *complex combinatorial map* (**CCM**). Landscapes of this class are generalisations of mappings from genotype into phenotype.

3.3 Density Surfaces

A two dimensional probability density $P(., h)$ can be used to gain insight into the details of RNA folding without dealing with CCMs explicitly. A *tree density surface*, for example, expresses the joint probability of two RNA sequences of chain length n, $\mathbf{I}_i$ and $\mathbf{I}_j$, having Hamming distance $d(I_i, I_j) = d_h(i,j) = h$, as well as having a tree distance $d(T_i, T_j) = d_t(i,j) = t$ between their structures. The surface $P(t,h)$ is biased along the Hamming distance (h) axis with the distribution from eqn. 5. By considering eqn. 11 this bias can be compensated, $P(t,h)$ can be computed sampling the tree distances of pairs of sequences with Hamming distance h and tree distance t.

The conditional probability to find a distance d of a mapped value, that can be the tree distance d_t or the free energy distance d_f as outlined in the next section, given that two sequences have hamming distance h is given

$$p(d|h) = \frac{p(d,h)}{p(h)} \approx \frac{n(d,h)}{\sum_{d=0}^{d_{max}} n(d,h)} \tag{11}$$

where $n(d,h)$ denotes the number of sequences which have hamming distance h in a sample and d_{max} is the maximal distance. The total sample-size N is given by

$$N = \sum_{d=0}^{d_{max}} \sum_{h=0}^{n} n(d,h) \tag{12}$$

and from the definition of conditional probabilities follows that

$$\sum_{d=0}^{d_{max}} p(d|h) = 1 \tag{13}$$

for each h so that for sampling techniques like uniform sampling the bias of the binomial distribution can be corrected directly.

Several sampling techniques have been developed to investigate the overall statistics of RNA landscapes [32, 29]. The occurrence of the certain energy distances d_f under the condition that the two underlying sequences have hamming distance h is counted. The same procedure can be used for tree distances [29].

We have to stress here that for the sake of easier computation and comparison and to avoid a complexer conception of sequence space all these models are based on the assumption of a constant string length for single mutational events. Calculations for all features are independently repeated for increasing chain-lengths by steps of ten. Data points are close enough then to interpolate a continuous behaviour.

Density surfaces of RNA binary sequences appear to be roughly symmetric. This is amazing since the complementary sequence is in general not the sequence at $h = n$ and complementary sequences do not necessarily have the same free energies and structures. With four letter alphabets the symmetry is lost.
Inspections in the surrounding of $h = 0$ shows that large values of tree distances - coming close to the maximum - are common at small h already. Since this depends neither on n, nor on the alphabet or the structure distance measure it can be supposed that almost all typical sequences already appear at small error classes.

3.4 Optimisation on Rugged Landscapes

Optimisation on rugged landscapes is a non trivial problem. It is in general not possible to find analytically the global minimum. The performance of optimisation algorithms depend crucially on the structure of the underlying landscape.

Optimisation algorithms, which are universal (no information of the landscape is *a priory* necessary) are:

- Neural Networks: Originally used as models for the dynamics of the nervous system. Today they are used in certain optimisation problems and in pattern recognition.
- Genetic Algorithms: These algorithms apply the methods of biological evolution to optimisation problems.
- Simulated Annealing: This method is an improvement of Monte Carlo optimisation. [52, 30]
- Grand Deluge: The acceptance rate is a step function instead of the exponential function used in simulated annealing. [22]

II: RNA Secondary Structures

4 RNA: Neutral Mutations and Distributions of Structures

In the following we will describe several experiments that were designed to investigate the degree of neutrality around secondary structures of RNAs. We do this with various algorithms to determine, if findings can be assumed to be generic properties of the Sequence Space mapping and also present some statistical results by application of the landscape approach. We will also investigate the influence of the size of both spaces and the isotropy of the mapping.

4.1 Structure Distributions

We fold large ensembles of RNA sequences for several lengths with various algorithms and sort structures according to their frequencies. The most frequent is assigned rank 1, the next 2 and so forth.
For all structure representations and algorithm the ranking yields a distribution that follows a generalised Zipf law:

$$f(r) = a(r+b)^{-c} \tag{14}$$

where r is the rank and $f(r)$ is the frequency of the corresponding structure. a is a suitable normalisation constant, b can be interpreted as the number of "very frequent" structures. The parameter b is a measure for the very abundant structures.

Zipf's first law was found by Zipf in 1949 [116] when analysing literary texts. It states that "if one takes the words making up an extended body of text and rank them by frequency of occurrence, then the rank r multiplied by it's frequency of occurrence $f(r)$ will be approximately a constant". The above formulation of Zipf's first law was proposed by Mandelbrot [62].

4.1.1 Coarse Graining:

For longer chains we had to restrict our computations to sampling with coarse graining representation. Using fine graining would result in so many structures that we could hardly observe any structure twice. From the data in table 2 we see, that there are at least an order of magnitude more frequent structures for the **GC** alphabet than for the **GCAU** alphabet for all algorithms.

We are, thus, dealing with relatively few common shapes and many rare ones. The distinction can be made by the fraction of (no. of possible sequences/ no. of observed structures):

	AUGC		GC	
	b	c	b	c
MFE(F)	70	1.72	997	2.66
MFE(S)	36	1.45	938	2.57
KIN	33	2.40	312	3.74
MM(4,3)/(3,3)	59	2.58	126	0.74
MM (5,3)	56	3.20	1038	4.70

Table 2: Fits to Zipf's law for n = 100: b reflects the amount of very abundant structures. MM(43) was used for the AUGC alphabet, MM(33) for the GC alphabet.

more abundant structures are termed common, other rare (P. Schuster, personal communication). This is consistent with the fact that the ratio of structures to sequences is much larger for two letter alphabets.
The results for samples of 10000 strings of length $n = 100$ and a coarse grained resolution are shown in fig. 9. There we can see, that this holds true for all algorithms and symbol-sets. Parameters for fits are listed in table 2.

It is also interesting to note that the tail of the distribution is steeper for the kinetic algorithm than for the minimum free energy algorithm. This indicates that many rare minimum free energy structures are not accessible at all for kinetic folding which complies with the results from [105]: although qualitative features of structure statistics were robust, with respect to the chosen algorithm, structures generated with the KIN algorithm had fewer base pairs, larger mean loop and mean stack sizes. This is also reflected by the approximation of the distributions with the MM algorithm and different structural constraints. Forcing longer stacks forms the shape of the distribution much like the MFE first and, at even stronger lower bounds like the KIN. Results for the biophysical alphabet (AUGC) and the binary alphabet (GC) are shown in fig. 10 and 11.

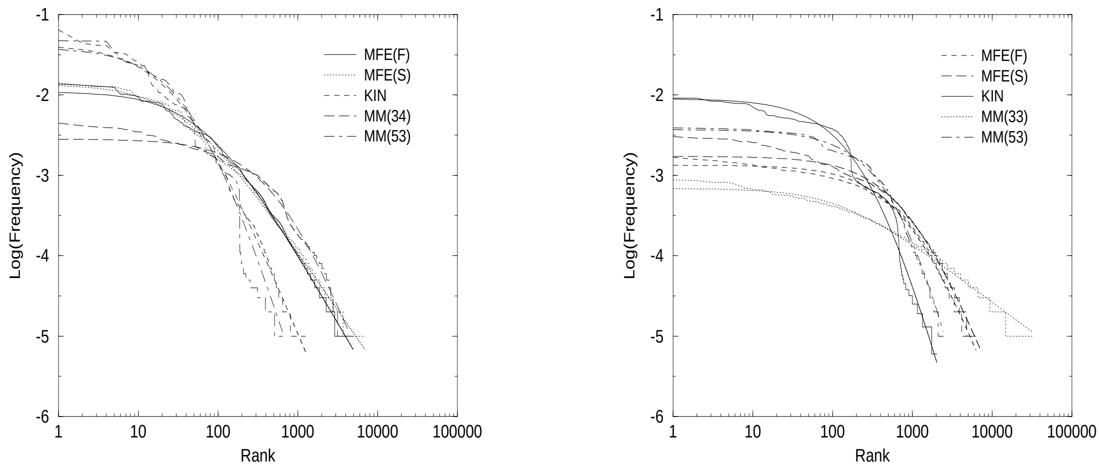


Figure 9: The distribution of coarse grained RNA secondary structures follows Zipf's law: shapes are ranked by their frequencies, corresponding fits are plotted in the same line style. a) (left) refers to AUGC alphabet, b) (right) to GC alphabet. (KIN and MFE(S) data from [105])

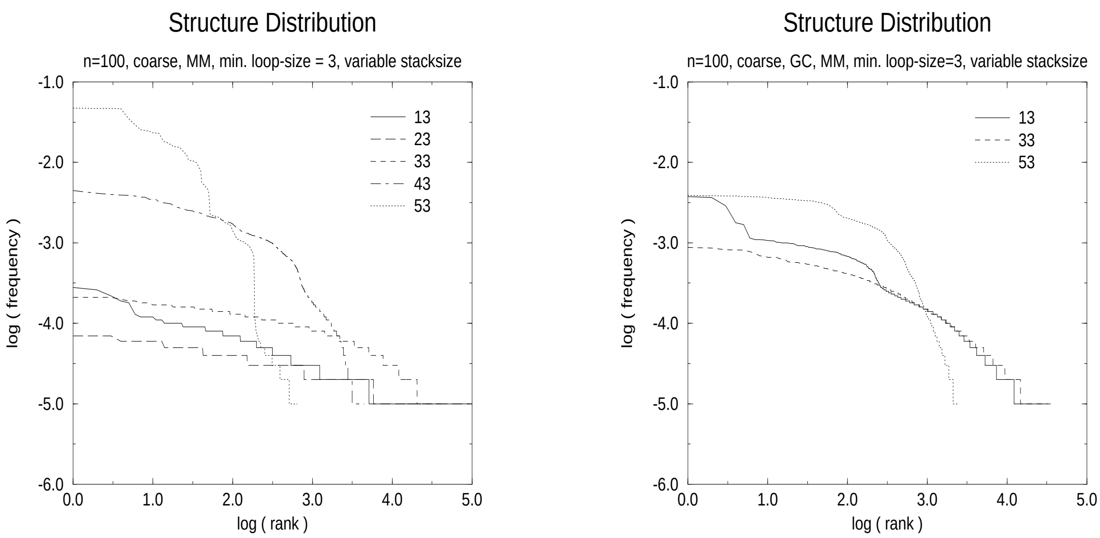


Figure 10: Structure distributions for coarse grained structures and chainlength $n = 100$ folded with MM and variable stack-size. A sample of 10^5 random sequences was folded and structures were sorted, counted and ranked according to their frequency. a) (left) refers to AUGC alphabet, b) (right) to GC alphabet.

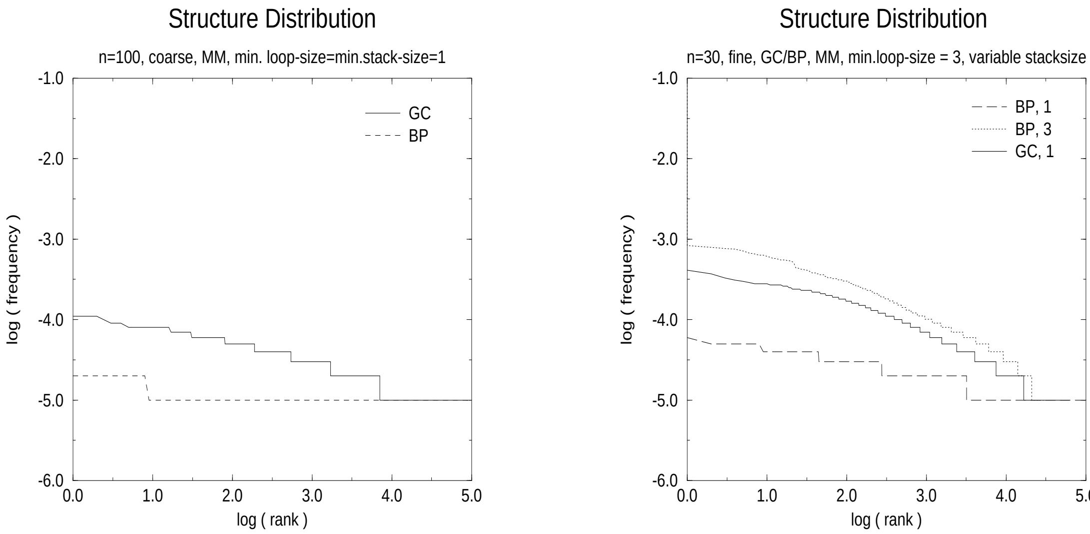


Figure 11: The distribution of structures for fine grained and coarse grained structures generated with the Maximum Matching algorithm. a) (left) Coarse grained Structures for n=100, for the AUGC (=BP) and the GC alphabet without any restrictions on the structures. b) (right) Fine grained structures for n=30, for the same alphabets and with min. stack-size =1 or 3.

4.1.2 Fine Graining

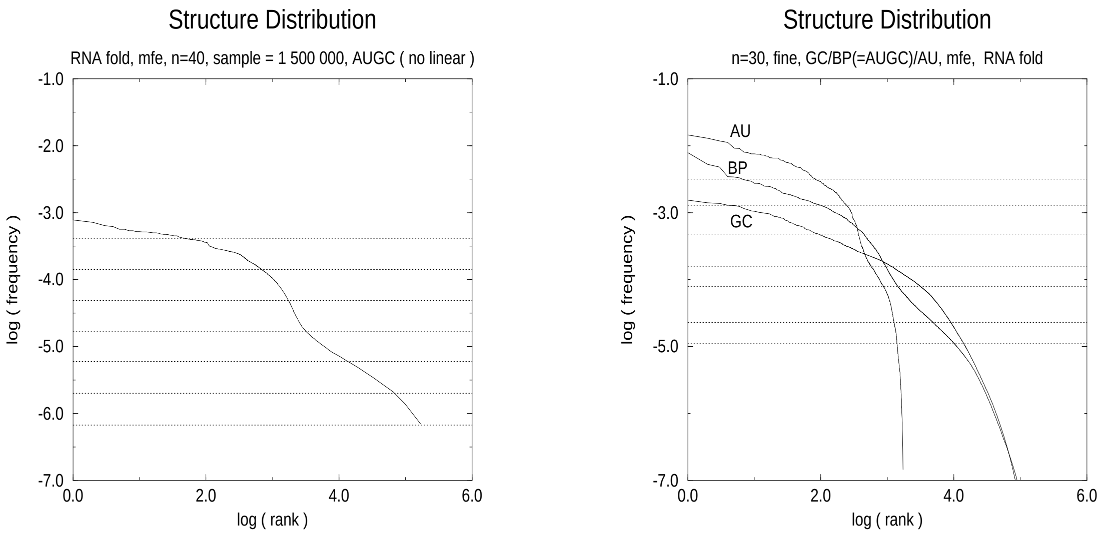


Figure 12: Zipf's law for the MFE algorithm and all three alphabets. The distribution of structures for n=30 and n=40. The frequencies of subsamples, as used in the following sections are indicated in dotted, horizontal lines.

In the case of fine graining structures we encountered the limits of computability. So we calculated only large ensembles of short chains. Results for $n = 30$ and all three alphabets and for $n = 40$ for the AUGC are shown in fig. 12. Due to the weaker H-bonds AU sequences are less likely to fold successfully to a stable structure, consequently there are a smaller number of structures and a steeper descent as the stable ones will make up a higher fraction. The opposite is, of coarse true for GC sequences. In any case however the distributions are following Zipf's law.

The $n = 40$ sample for AUGC consisted of 1500000 structures. 40 is still pretty short compared to natural RNAs, but a fairly close approximation for longer RNAs, in the sense that the average size of structural elements is already converged to the values of much longer structures [29]. As is well known, 30 is an even smaller number, but we sorted structures for the AU and GC alphabet as well. Due to the weaker H-bonds between A and U there are many non folding events so that frequent structures make up a larger fraction. In the case of the stronger GC-bonds shorter stacks and therefore more different structures can be realised.

4.2 Neutral Neighbours

To estimate the sphere of neutral mutants around any typical structure we computed the distribution of frequencies with which random structures have a certain number of neutral neighbours.

4.2.1 Maximum Matching

Using the MM algorithm we computed neutral one-error mutants for different minimal loop and stack sizes, chain lengths n, folding algorithms and alphabets. Results for n=100 and the MM algorithm are summarised in fig. 13 Restricting the number of possible structures leads to higher values of neutral neighbours.

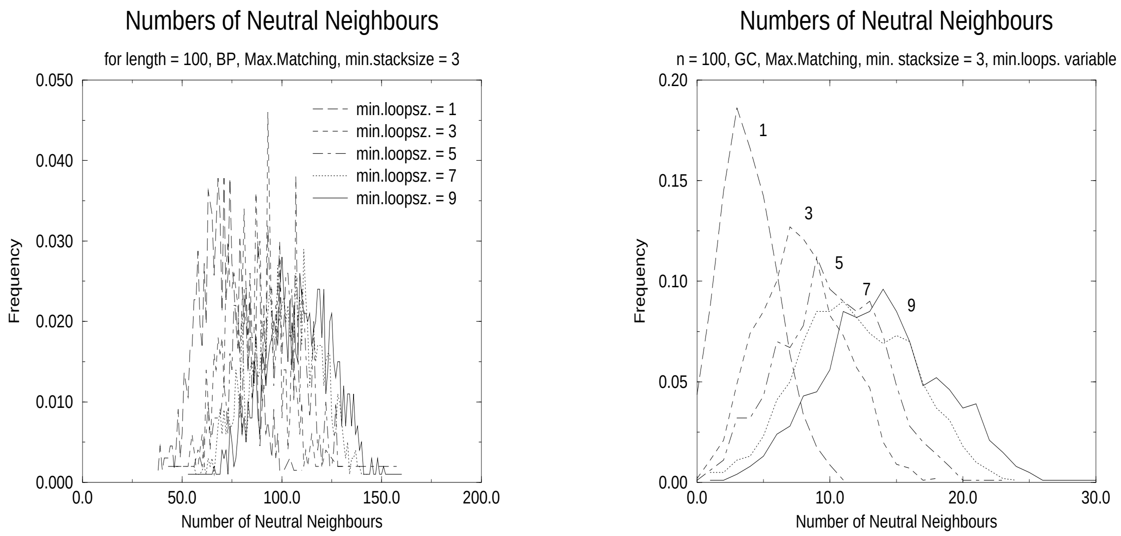


Figure 13: The distribution of structures with a certain number of neutral neighbours at varying minimal loop size for the AUGC (left) and the GC (right) alphabet. Stack size is kept constant at 3.

The influence of limiting stack-size is much greater than for loops: two mutations are required if the structure is to be maintained (except for an AU to GU substitution in the AUGC alphabet) and the number of affected nucleotides is doubled compared to limited loop-size. The chance to find a neutral neighbour of a successful folded reference string therefore becomes very high. Recalling the results from the last section we are illuminated now: in the limit of long stacks successful folds become rare events and only very few structures are accessible. Again the size of the alphabet seems to play a dominant role on the distribution. Results are shown in figs. 13 to 14 for some distributions of one-error mutants for imposing the min. loop size, the min. stack-size and finally the finale grande, both at the same time 15.

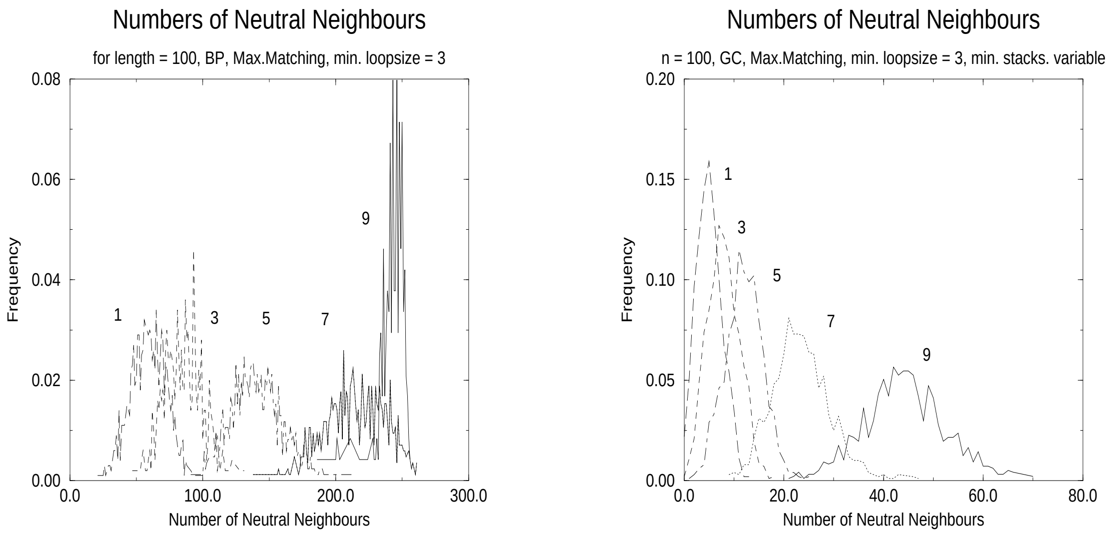


Figure 14: The Distribution of neutral neighbours at varying minimal stack size for the AUGC (left) and the GC (right) alphabet. Loop size is kept constant at 3.

As we have a high probability to choose a very common structure, we can expect a large number of structures with many neutral neighbours and a relatively high probability to start and therefore end on a large neutral net. If this holds true, there should be many long neutral paths. We are scared. The results are also summarised in a date table in appendix 14.

4.2.2 Dependency on the probability of the structure

We also calculated the average number of neutral one and two error mutants for structures with different probability by choosing subsamples from the large sample as shown in fig. 12.

7 subsamples, each consisting of 250 structures with the same probability were picked. We then calculated the number of neutral 1 and 2 - error neighbours as well as the number of neutral mutants, randomly occuring in Sequence Space. A uniform sampling procedure was used to adjust the bias arising from the binomial distribution.

As there are more paired than unpaired bases in an average structure [29] and there are several ways to exchange one base pair with another in the biophysical alphabet the frequency is much higher for 2 - error mutants. The slight decrease of neutral neighbours for very frequent structures is probably due to fact, that these structures, consisting of very simple components can be easily inter-converted. At larger hamming distances the correlation is heavily destroyed and the number of possible sequences increases drastically due to the binomial distribution [32, 29]. This is why the average frequency to find neutral neighbours in Sequence Space is so

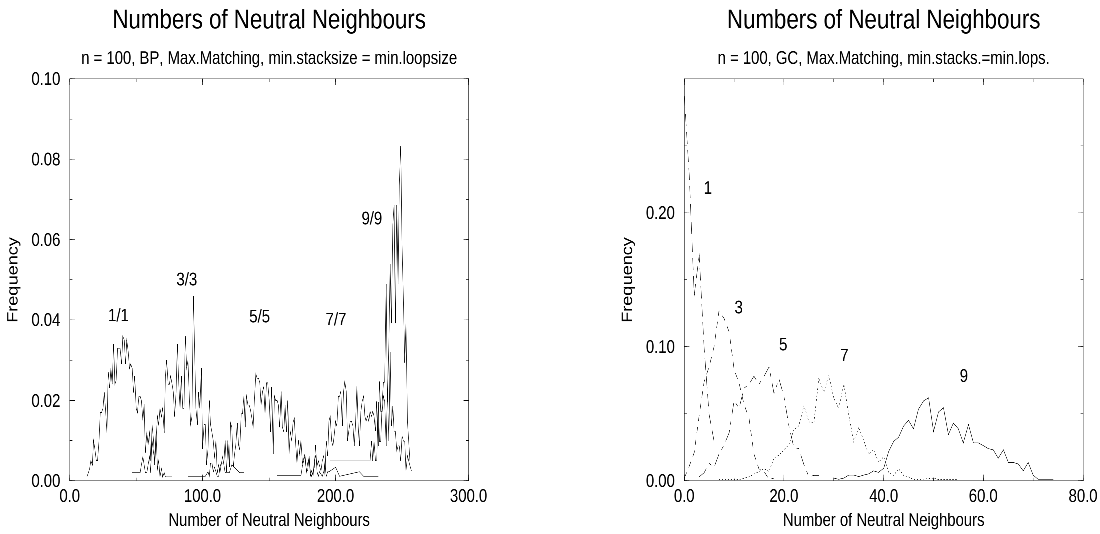


Figure 15: The Distribution of neutral neighbours at varying minimal loop and stack size for the AUGC (left) and the GC (right) alphabet

small (fig. 16). The frequency scales roughly linearly with the logarithm of the probability of the structure.

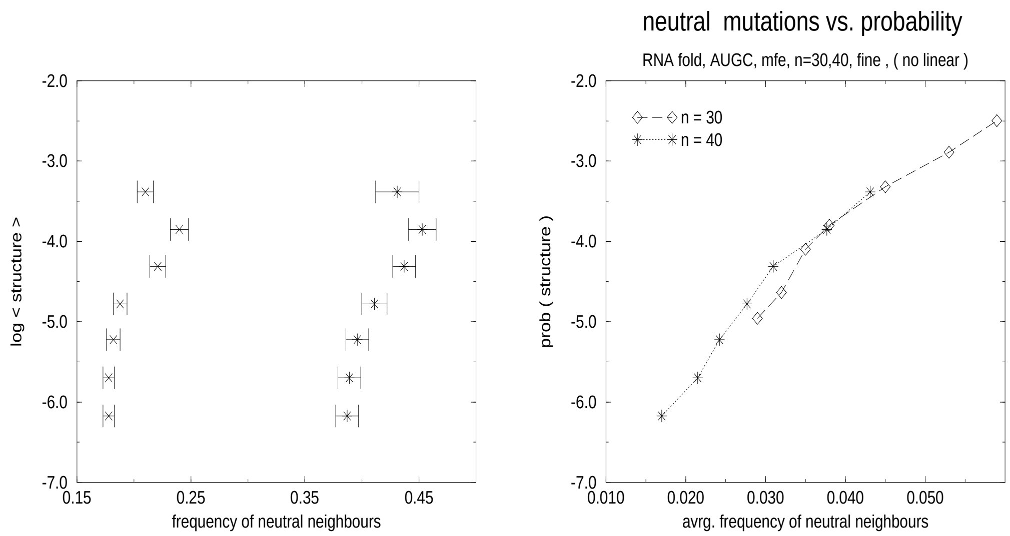


Figure 16: a) Probabilities to find neutral 1 ($\times$) and 2 ($*$) -error mutants for structures of a certain frequency (see also fig. 12). b): Probabilities to find neutral neighbours for structures of a from the same samples randomly in Sequence Space.

4.3 Neutral Paths

We performed computer simulations to assess the extension of these networks. It was investigated how far one can walk through sequence space without changing the structure by the following procedure:

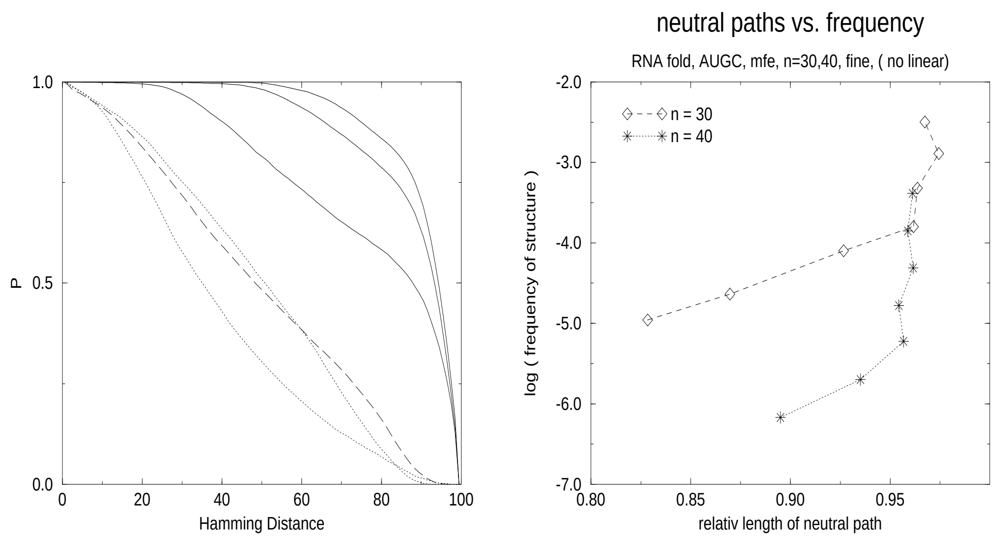


Figure 17: a)The probability P for a neutral path to reach at least a Hamming Distance h from a random starting point. Solid lines refer to AUGC alphabet, dotted lines to GC. (Data from [105])
b) The relative length of neutral paths ($< h > /n$ for the samples as in fig. 12 depends on the frequency of the underlying structure. ($\diamond$ refers to $n = 30$ and $*$ to $n = 40$)

1. Start with a (random) sequence and fold to it's secondary structure.
2. Look for neutral neighbours with $h = 1$ or 2.
3. If we find one, repeat (2) with this new sequence. Back-mutations to (1) are prohibited.
4. If the structure is different, remember the current Hamming distance and start a new path.
5. Repeat steps (1) - (4) for a large number of starting sequences.
6. For each Hamming distance count the incidents of (4).

The results are shown in fig. 17 a for chain length 100 and both alphabets and were supported from [105].
It is clearly seen that for the **GCAU** alphabet most neutral paths cross the whole sequence

space with MFE(F) and KIN showing the same behaviour. Neutral networks for the maximum matching algorithm are smaller than for the other algorithms. Neutral networks for the **GC** alphabet are smaller. This is due to the fact that combinatorial reasons allow only one sequence with Hamming distance ν from the starting sequence whereas for the **GCAU** alphabet much more (4ν) sequences with Hamming distance ν exist. In this case the KIN and MFE(F) algorithms show the same behaviour whereas the maximum matching algorithm yields smaller networks.

freq.(str.)	log	1err	±	2err	±	nnnb	dev	path	±	diam.
n=40, mfe, AUGC										
0.00000067	-6.17	0.387	0.010	0.178	0.005	0.017	*	35.81	0.56	0.895
0.00000200	-5.69	0.389	0.010	0.178	0.005	0.021	*	37.40	0.31	0.935
0.00000600	-5.22	0.396	0.010	0.182	0.006	0.024	5.361	38.27	0.20	0.957
0.00001667	-4.77	0.411	0.011	0.188	0.006	0.028	5.785	38.17	0.19	0.954
0.00004867	-4.30	0.437	0.010	0.221	0.007	0.031	6.187	38.46	0.15	0.961
0.00014067	-3.85	0.453	0.012	0.240	0.008	0.038	6.946	38.35	0.14	0.959
0.00041467	-3.38	0.431	0.019	0.210	0.007	0.043	7.395	38.45	0.11	0.961
0.02963733	-1.52	0.297	0.017	0.299	0.009	0.031	6.113	38.11	0.16	0.953
n=30, mfe, AUGC										
0.000011	-4.96	*	*	*	*	0.029	*	24.85	*	0.828
0.000023	-4.64	*	*	*	*	0.032	*	26.09	*	0.870
0.000080	-4.1	*	*	*	*	0.035	*	27.8	*	0.927
0.000159	-3.8	*	*	*	*	0.038	*	28.85	*	0.962
0.000478	-3.32	*	*	*	*	0.045	*	28.91	*	0.964
0.001286	-2.89	*	*	*	*	0.053	*	29.23	*	0.974
0.003186	-2.5	*	*	*	*	0.059	*	29.02	*	0.967

Table 3: Average number of Neutral Neighbours and Average Length of Neutral Paths (data for $n = 30$ from [105]): 25 out of all sequences folding into a structure with a frequency *freq(str.)* and the corresponding logarithm *log()* were chosen. Their average number of neutral neighbours $nnnb$ (with their standarddeviation dev) was counted and the average $path$ pathlength for 30 paths with each sequence calculated. $diam$ denotes the normalized pathlength $diam = path/n$. (*: data not available)

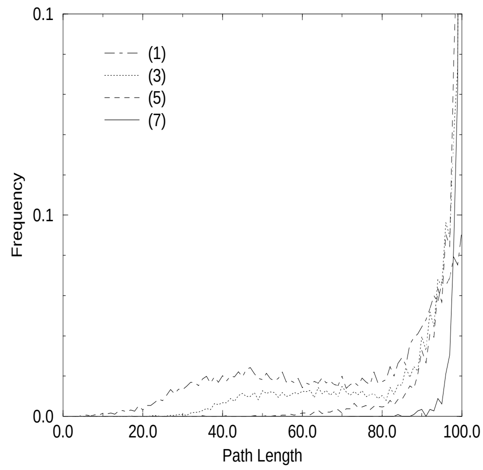
4.3.1 Dependency on the probability of the structure

We picked 7 subsamples with 250 structures that occurred at different probabilities from the sample from fig. 12. We performed 30 neutral path calculations for each structure and averaged the walk lengths.

Results are shown in fig. 17: the more frequent a chosen structure is, the longer the neutral path we can expect. Even for rare structures, however, we find networks that are surprisingly large. Due to the sampling procedure (not in any case of a terminating mutational event there is a backtracking procedure) the results of 95% have to be viewed as a lower bound. There appears to be a percolation threshold for the number of neutral neighbours and the corresponding probability of the structure, above that neutral nets become connected. Structures with a higher probability have a nearly 100constant while systematically inverting the underlying sequence. The dependence of the walk length on the probability of the structure is shown in fig. 12 for the case of $\nu = 30$ and $\nu = 40$ and the **GCAU** alphabet. The size of the neutral network depends on the frequency, however only up to a certain threshold value from where the values remain constant. Even for very rare structures (probability of 10^{-5}) most networks percolate through more than $\frac{5}{6}$ of sequence space. This analysis shows clearly that only more frequent structures percolate, while neutral networks of rare sequences exhibit significantly shorter paths. (This is consistent with analytical calculations from random graphs [85].)

4.3.2 Maximum Matching

Next we investigate the influence of restricting the accessible Shape Space by using the MM algorithm with different constraints of the minimum loop and stack size as initial conditions.



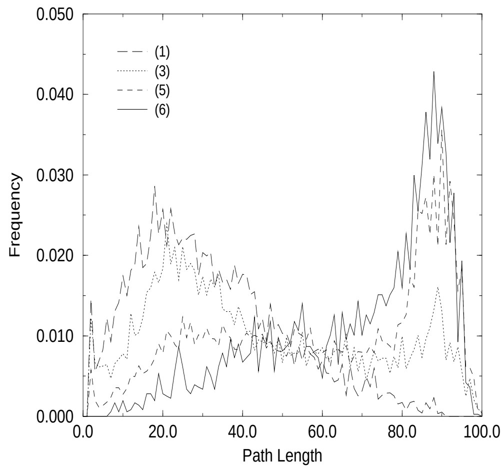


Figure 18: Neutral Paths for MM and $n = 100$. the frequencies of Hamming Distances at which Neutral Paths stop: The minimum loop size is 3 in any case, the stack size is tuned (1, 3, 5, 6 or 7 as indicated). a) (left) AUGC alphabet: many neutral paths percolate through more than 4/5 of Sequence Space. b) (right) GC alphabet. Larger stacks imply longer paths.

The influence of limiting stack-size is much greater than for loops: in general two mutations are required if the structure is to be maintained (except for a GU that can be replaced by a

AU and a GC and vice versa) and the number of affected nucleotides is doubled compared to limited loop-size. Results for varying the minimum stack-size are shown in fig. 18. In the case of the *AUGC* alphabet long paths are very likely even without any restrictions. In the case of *GC*, however we need to impose strong restrictions to achieve long neutral paths. In the limit of long stacks successful folds become rare events and only very few structures are accessible. The chance to find a neutral neighbour of a successful folded reference string therefore becomes very high. Changing the minimal allowed stack size allows a fine tuning of the number of structures accessible. The higher the value of the minimal stack size the lower the number of secondary structures accessible. For the four letter alphabet nearly all neutral networks extend through more than $\frac{4}{5}$ of sequence space with the higher values for the minimal stack size having larger neutral nets. The same is true for the two letter alphabet where for small values of the minimal stack size a distribution of neutral paths analogous to the **GC** alphabet folded with the kinetic algorithm is observed whereas for larger minimal stack sizes the distribution is very much like the **AU** alphabet.

The folding of RNAs with MM with different parameters of the minimal stack size shows that the ratio of structures to sequences determines the extension of neutral nets. This is very clearly seen in the mean walk length as a function of the minimal stack size.

Again we conclude (uh, deja vu...) that the the characteristical features of the sequence to structure space mapping depends rather on the size of sequence and shape space than on folding parameters or algorithms.

4.4 Landscapes

For comparison with other algorithms [32] we calculated the lengths of adaptive walks to local optima (i.e. maximum number of base-pairs in the case of MM) as well as energy and structure correlation lengths for the MM with various minimum stack length and loop size. Increasing the limits for stack sizes leads to shorter energy correlations, the influence on the structure correlations is not so obvious. Results are shown in fig. 19.

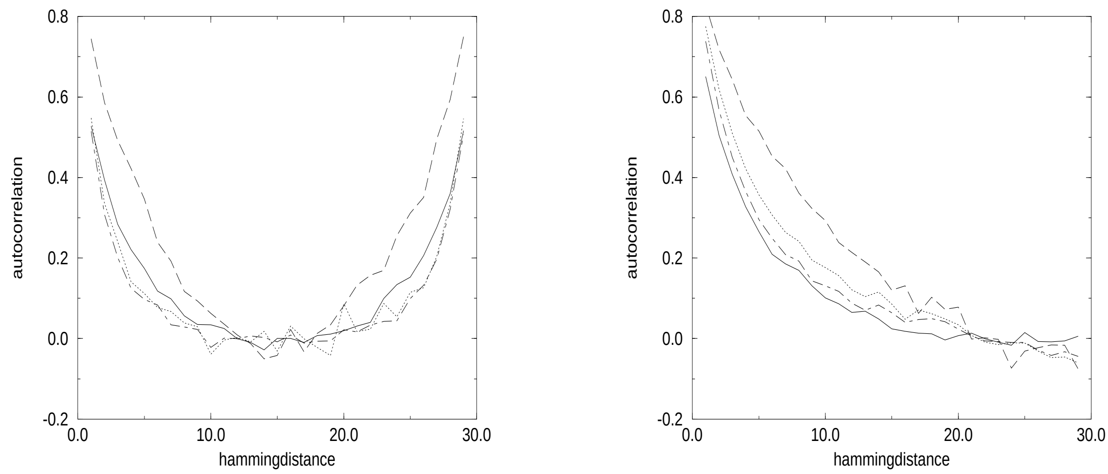


Figure 19: RNA landscapes: Autocorrelation function for sequences of the biophysical (AUGC) alphabet (left) and the binary (GC) alphabet (right) length $n = 30$, folded with the MM algorithm with min. stack-size 5 (dot dashed, dotted) or 1 (solid, long dashed) for free energy landscapes (long dashed, dotted) and secondary structure landscapes (solid, dot dashed).

As with "realistic" folding algorithms, correlation lengths are larger for the AUGC alphabet than for the GC alphabet, structure correlation lengths are smaller than those for the energy, both are small compared to the diameter of the sequence space. The only striking differences with the MM algorithm we find for the AUGC landscape: correlation lengths are particularly small compared to other algorithms since the ground state is highly degenerate, i.e. there are many unrelated secondary structures with the maximum number of base pairs. For larger stack size we find fewer structures that are scattered in shape space, so that after a short increase values are slightly decreased at stronger limitations. This is also indicated by very short adaptive walk lengths for small loop and stack-size restrictions. For very drastic restrictions there are many non-folding events so that values for the average number of formed base pairs is of coarse much smaller.

Binary sequences at $h = n$ fold into the same structure, thus yielding the maximum correlation and a symmetric correlation function. Some boring tables can be seen in the appendix.

4.5 Discussion

Together with the findings from [89] about the Shape Space covering radius, our results imply a new perspective on the evolutionary behaviour of biopolymers and the possibility to explore shape spaces by point mutations.

(i) There are many more sequences than structures, hence many sequences fold into the same structure [43].

(ii) The distribution of the number of sequences folding into the same structure follows a Zipf-like distribution, i.e., there are few very common structures and many different very rare structures.

(iii) The sequences folding into a common structure are distributed randomly through-out sequence space. No clustering is visible [89].

(iv) The sequences folding into a common structure form extremely extended *neutral networks*, i.e., there are pathways consisting of sequences that fold into same structure which extend through all sequence space.

(v) The distance from a random sequence to a sequence that folds into a desired structure is short compared to the maximum distance in sequence space [43].

(vi) The length of a neutral path depends on the density of the structure, i.e. on the frequency and on the number of neutral neighbours. There is, however a threshold value, from where nets are percolating through sequence space.

As a consequence evolutionary adaptation of RNA should be much easier than commonly assumed: From any given structure it is not far to any desired structure and the function can be fine-tuned by mutating the sequence without ever changing the structure
We emphasise that the qualitative aspects of the Sequence Space to Shape Space mapping depend neither on the folding algorithm (MFE, KIN, MM) or the parameter set used to calculate the structures nor on the distance measure used for comparing the secondary structures (based on tree editing, alignment, or the number of identical base pairs) [105], they are robust properties. We postulate the number of neutral neighbours and size and isotropy of Sequence Space and Shape Space to be the crucial parameters for the existence of neutral nets. The existence of a "threshold value" of neutral neighbours for percolation has also been pointed out by analytical calculations based on random graphs models (C. Reidys, personal communication).

This stimulates further considerations about the interpretation of the concept of fitness landscapes: the picture of a single entity with a maximum fitness value has to be replaced by populations with a given fitness on neutral nets. Applied to the concept of quasi-species [24, 31] this means that one has to distinguish between a genotypic and a phenotypic error-threshold. Populations, starting from a homogeneous distribution and, without adaptational pressure might travel through through sequence space along neutral nets that stands for the current fitness maximum. In case of need, quick adaption of single entities that are relatively remote from the consensus sequence (or falling onto a nearby structure on a different net) may be achieved by jumping to another (neutral) net with higher fitness value. Thus there are not single peaks in the fitness landscape but rather extended nets where each one is close to any other somewhere in Sequence Space.

If one is willing to accept the secondary structure concept and simplifications from the folding algorithms, RNAs appear to be a perfect playground to study these aspects of evolutionary dynamics. Our approach intentionally omitted any dynamic aspects such as the application to quasi-species simulations [31], which opens the door for further studies to concentrate on this interesting aspects.

The major shortcoming of the above-mentioned study is that it is restricted to RNA molecules and their 2D secondary structures. Those who want to overcome this karma should proceed with the next chapter.

III: Protein Lattice Structures

5 Lattice Polymers and Self-Avoiding Walks

In this chapter we will introduce some methods to investigate and characterise self-avoiding walks (SAWs) and lattice polymers (LPs). A detailed description of transformational matrices and their application is given in the appendix.

5.1 Definition and Characterisations

5.1.1 Lattices

A regular Lattice $\mathcal{L}$ can be characterised by the set of basis-vectors of equal length, the rules to move on the lattice are defined by lattice constants: the dimension d, the number of nearest neighbours and of possible moves, the length of regular steps. Basis-vectors for the FCC lattice are $[0,1,1],[1,0,1],....[0,-1,-1],....[-1,-1,0]$ which allows 12 combinations for all permutations and sign combinations. TET, SC, BCC can be viewed as a subset, HEX and SQ in turn as a subset of SC lattice. All lattice parameters of importance to calculate structures on lattices are listed in table 4. By using transformation matrices all structures can be encoded by the means of relative directions. Self Avoiding Walks (SAWs) are defined as non-intersecting growing chains (i.e. random walks) on a regular lattice [61].

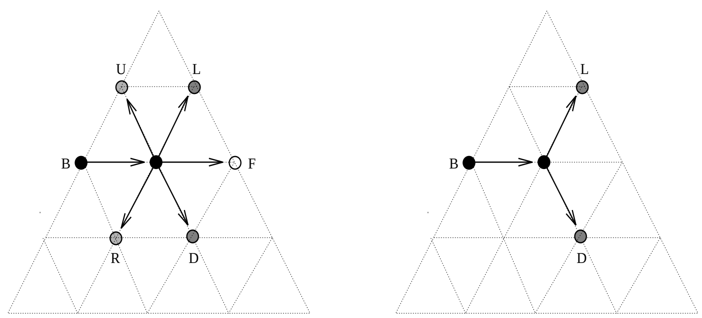


Figure 20: Relative Moves of the TRI (left) and the HEX (right) lattice

5.1.2 Polymer folding on Lattices

Little is known about the actual behaviour of polymer folding/unfolding and several algorithms such as the pivot algorithm [61] or the slithering snake have been proposed. In the sense of

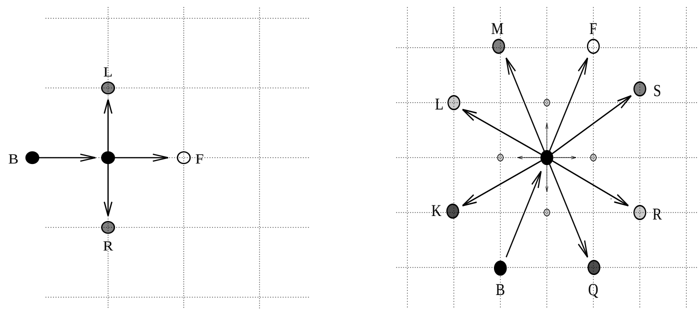


Figure 21: Relative moves of the SQ (left) and the KM (right) lattice

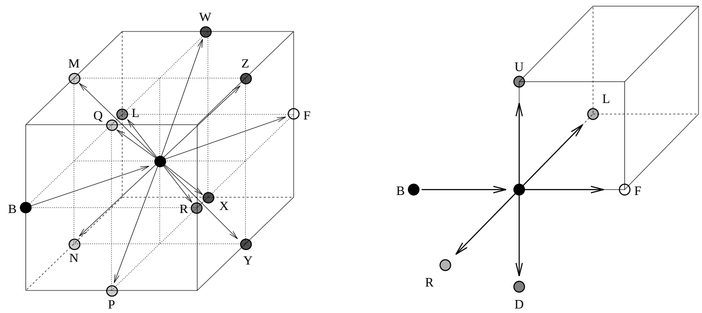


Figure 22: Relative moves of the FCC (left) and the SC (right) lattice

encoding LPs in relative move notion, the pivot algorithm can be interpreted as a point mutation in the string of relative moves. Obviously, mutations that do not yield a SAW have to be rejected and one has to consider, that not all possible moves, even if they result in a SAW are physically accessible. On the other hand several models have been developed, that permit temporary to expand to a 4th dimension, thus enabling a chain to overcome barriers that are artifacts from the chain rigidity and lead to better conformations (W. VanGunsteren, FEBS course Ascona, 1994).

In our implementation only regular lattices and the dimension 2 or 3, bonds of equal length are considered. In [39] the SC was reported to favour α-helixes, TET to fit well for β-structures. For realistic modelling of C_α chains however, they were reported to be the worst fits, describing the structure well down to 3-4 angstrom. A reasonable compromise between computational demands and accuracy is the 3DKM lattice, characterised by the basis-vectors $[2,1,0],[2,0,1],.....[-2,1,0],.....[0,-1,-2]$ with an average accuracy down to 1-2 angstrom and a set of 23 moves. (only the Ultra$[3,1,1]$ lattice, a combination of several possible moves such

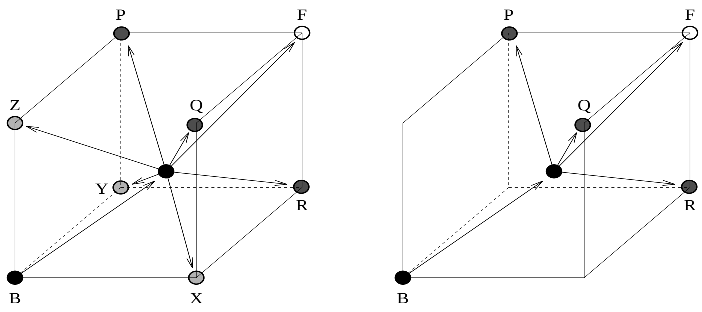


Figure 23: Relative moves of the BCC (left) and the TET (right) lattice.

Lattice	t	s	N	d	m	α	pβ	apβ	dim.	walklen.		
HEX	5	1	6	6	2	6	1	1	2	70.9 ± 0.2	2.5 10^3	50
SQ	3	1	4	4	3	4	1	1	2	71.7 ± 0.2	2.5 10^3	50
TRI	3	1	6	6	5	4	1	1	2	77.9 ± 0.7	3 10^3	54
KM	3	1	4	8	7	4	1	1	2-3	3212.0± 10	5.9 10^6	2400
TET	3	2	8	8	3	4	2	2	3	381.0 ± 1.3	11.6 10^4	340
SC	3	1	6	6	5	4-5	1	1	3	3997.0 ± 38	14.5 10^6	3800
BCC	3	1	8	8	7	4	1	1	3	22447.0±296	4.4 10^8	21.200
FCC	3	1	12	12	11	4	1	1	3	34963 ±1024	10.5 10^8	32.000
TDKM	3	1	6	24	23	4	1	1	3	> 2000000		
Prots.	3	1				3.6	2	2	3			

Table 4: Lattice Constants for all lattices as used: t = turnsize, s = number of steps, N = number of Neighbours, d = directions, m = number of possible moves, α the size for one repititive unit of an α-helix, pβ a parallel β sheet and an apβ antiparallel β sheet, the dimension and the average walklength of a gSAW [86].

as $[3, 1, 0], [3, 0, 0], [2, 2, 0] etc.$ was reported to be superior.) The excluded volume effect is counted out by the number of possible neighbours, so that lengths of SAWs exceed computational resources, leaving enough place to introduce side-chains on this geometry.

5.1.3 Characteristic Measures

We can easily implement methods to calculate characteristic measures, independently from the chosen lattice $\mathcal{L}$:
The mean square end to end

$$e_s = \|x\| = \left(\sum_{i=1}^{d}(x_i)^2\right)^{1/2} \tag{15}$$

The most common measure for compactness, the *radius of gyration* for a structure $\mathcal{A}$ with length n defined as the mean square distance from the center of mass can be expressed as

$$g^2(\mathcal{A}) = \frac{\sqrt{\sum_{j>i} r_{ij}^2}}{(n+1)} \tag{16}$$

The masses of all residues are considered equal.
Another measure of compactness is the number of contacts. This measure is especially easy to implement on regular lattices, it was proposed to be directly correlated to the radius of gyration [5].
Another possibility has been suggested in [114]: the compactness of a region is the surface / volume ratio A/V, normalised by dividing by A_0/V of a sphere of equal volume.

5.1.4 Compact Polymers

For above mentioned reasons, compactness is an important property of polymers. A measure in the notion of contacts has been proposed by Dill. When $C_S(\mathcal{A}, \mathcal{L})$ stands for the number of contacts in a structure of length n on a lattice $\mathcal{L}$ and $C_{max}(n, \mathcal{L})$ the maximum number of contacts on a structure of the same length n:

$$Q = \frac{C_S(n, \mathcal{L})}{C_{max}(n, \mathcal{L})} \tag{17}$$

For maximum compact *SAWs (MCS)* the number of contacts can be explicitly approximated for structures without energy functions. For the lattices SQ (d=2) and SC (d=3) and chain length $n \to \infty$ the formulation is intuitive (when $n_c = \lfloor n^{1/d} \rfloor$ and $\lfloor x \rfloor$ denotes the the greatest integer less or equal to x) a summation of the compact block minus a surface term and the contacts from the chain bonds.

$$C_{max}(n, \mathcal{L}) \simeq n\frac{\kappa}{2} - d(n_c)^{d-1} - (n-1) \tag{18}$$

The last term is, obviously, influenced by coordination number and size of the surface and therefore lattice-dependent.

On a square lattice the number of contacts in a maximum compact state was estimated by Chan & Dill: [12]

$$C_{Max}(n,\mathcal{L})) = \begin{cases} (n-2m) & \text{if } \sqrt{n+\frac{1}{4}}-1/2 \leq m < \sqrt{n} \\ (n-2m+1) & \text{if } \sqrt{n} \leq m < \sqrt{n+\frac{1}{4}}+1/2 \end{cases}$$

$$= n - 2\sqrt{n} + \mathcal{O}(1) \qquad (19)$$

The number of compact structures has been estimated by the following expression ([6]):

$$\#(S_c(x,\mathcal{L})) \sim \beta n^{(\gamma-1)} (\frac{z}{e})^n f^{(n-n/d)} \qquad (20)$$

where z is the number of nearest neighbours, d the lattice dimension. For a mean field treatment [77] reported $f = 1$ the surface fugacity, and $\gamma = \ln(E)$ a correction term and $b \sim \mathcal{O}(1)$ a non-universal amplitude

5.2 Structures

Great attention has to be paid to the characterisation of structures and distances in-between. Several methods that have already been incorporated into the package are introduced in the next section.

In a very general model like this one, the analogy of secondary structure notion to real proteins should not be overestimated. Each occupied bead represents a whole residue, different sizes of side chains and sterical hindrance are not considered. From that point of view, a chain of, say 10 amino acids length could possibly correspond to a real sequence that is much longer. One should therefore focus rather on the essence of a secondary structure: the order in space, basically recognisable as repitivity. For that reason we can also investigate the occurrence of any repitititive unit within an overall shape. It is therefore not so important to find structural elements that are a geometric analogy, but to distinguish between regularities that are intrinsic properties of the model.

Two principal features of structures must be accessible to a clear and non-ambiguous application of the approach as described in the RNA case, so that we can apply the landscape approach and investigate structure landscapes:

- enumeration
- reasonable measures of distance, i.e. measures that are easy to handle and capture features that are essential for the addressed questions

It is however not always necessary to operate with folded states. A Density Of States algorithm (DOS) algo of Stolorz [99] e.g. approximates the distribution of low energy states without weighting them. Increasing the accuracy however again increases computational efforts exponentially.

5.2.1 Representation of Structures

1. Given the absolute direction, i.e. the vector (or a symbolic code for all possible vectors) of the previous move, structures of LPs on a given Lattice can be encoded as a string of relative moves.

2. The Distance Matrix *DM* is symmetric and contains the distances between two residues, where the nearest neighbours in the lattice are set to 1 by convention.

$$d_{ij} = ||\Delta(x_i, x_j)|| \tag{21}$$

   where $||.||$ denotes the euklidian norm. Not considering the chirality, all information about the geometry, however not about chemical bonds can be retrieved from the Distance Matrix.

3. We can now easily characterise structures by their contact matrices *CM*. Given a lattice $\mathcal{L}$ the entries simply state if there is a nearest neighbour contact between the residues i and j:

$$c_{ij} = \begin{cases} 1 & \text{if} \quad d_{ij} = 1 \\ 0 & \text{else} \end{cases} \tag{22}$$

   if $|j - i| > 1$.
   The number of contacts can now be easily retrieved by summing up all entries of the $\mathcal{CM}$. Considering real proteins, regions of secondary structural elements can be identified by characteristic patterns in the DM. The case of discrete values, which is characteristic difference for LPs, the contact matrix is the proper analogy.

4. A coarse grained distance between structures can be defined in analogy to the RNA case by the exchange of an element of secondary structure with another. This, however, makes only sense, if most of the structure forms secondary structural elements.

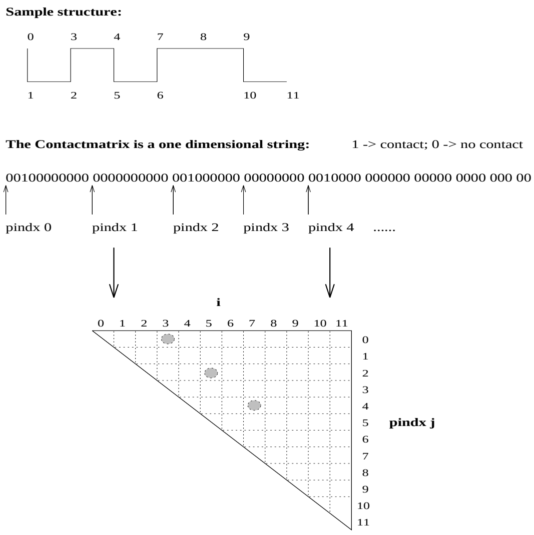


Figure 24: Coding the contact matrix: the CM is stored in a one-dimensional array and can be equivocally encoded and addressed by the means of a well-defined index array pindx.

5.2.2 Elements of Secondary Structure

A definition, based on the analogy with real proteins has been given by Dill [11].
It is essentially based on repetitive units that can be found in the contact matrix, which in turn is based on the notion of nearest neighbours (see above). Characteristic measures of elements in natural proteins and Lattice constants of LPs are summarised in table 4.
A general definition works as follows (t is the minimal turn-size, s the step-size):
For an anti-parallel-beta-sheet (*APBS*) we have:

$$c_{ij} = c_{i+s,j-s} = ... = c_{i+ps,j-ps} = 1 \tag{23}$$

when $j > i$ upstream in the chain, p the length of (i.e. the number of contacts in) the APBS.
For a parallel-beta-sheet (PBS) we have:

$$c_{ij} = c_{i+s,j+s} = ... = c_{i+ps,j+ps} = 1 \tag{24}$$

with the same conditions as above and the constraint that $j > i + p + t$ to bend back. In both cases $p > 1$, so that 3 contacts are required as the minimum to define a unit.
In the case of the α-helix analogy, however, the definition is rather ambiguous. The definition is essentially based on the size of a turn, for SC e.g. $t = 3$ (which is the minimal number to be added to a position on a given lattice to yield a turn).

$$c_{ij} = c_{i+a,j+a} = \ldots = c_{i+pa,j+pa} = 1 \tag{25}$$

with $j = (i + \mathrm{turnsize})$, a the height of the helix and $j > i < i + a$ and $p > 1$. α-helixes therefore appear as repetitive patterns with some of the contacts, corresponding to the turns, close to the diagonal.

In general repetitive structures can be defined as follows:

$$\begin{aligned} c_{ij} = &\quad c_{i+u,j+u} = \ldots = c_{i+pu,j+pu} = 1 \\ c_{kl} = &\quad c_{k+u,l+u} = \ldots = c_{k+pu,l+pu} = 1 \\ c_{mn} = &\quad \ldots. \end{aligned} \tag{26}$$

so that a subset where $i, j, k, l < i + u, j + u, k + u, l + u < \ldots < i + pu, j + pu$ and at least $i \neq k$ or $j \neq l$, where u is the length of the repetitive structure and $p \geq 2$ the number of units required to define a repetitive structure .

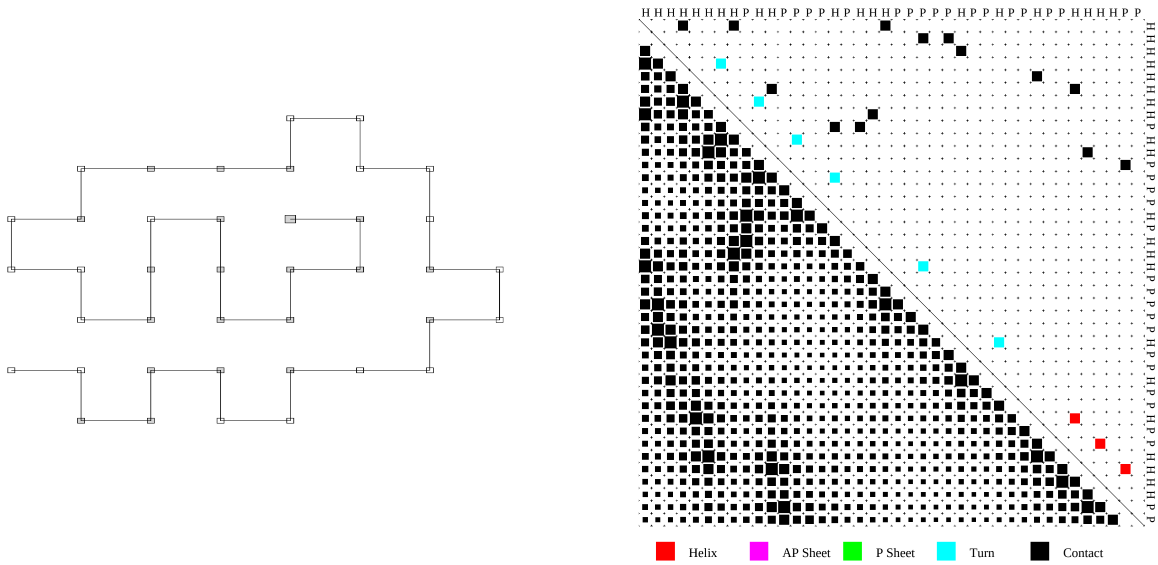


Figure 25: Structural Elements on the SQ lattice: The structure of length n=42 with a short α helix at the end and a short β strand close to the begin as well as one unoccupied (solvent) position in the interior as well as the corresponding CM and DM. (folded with the CGA algorithm (see next section), and parameters: eset=2, alphabet=HP, a=1.9 and m=2. "H"s are shaded grey, "P"s white.

Contact Maps contain entries in an upper (or lower) triangular matrix if two residues are topological neighbours, i.e. they are adjacent on the lattice but not along the sequence. Secondary structural elements appear in the CM as following:

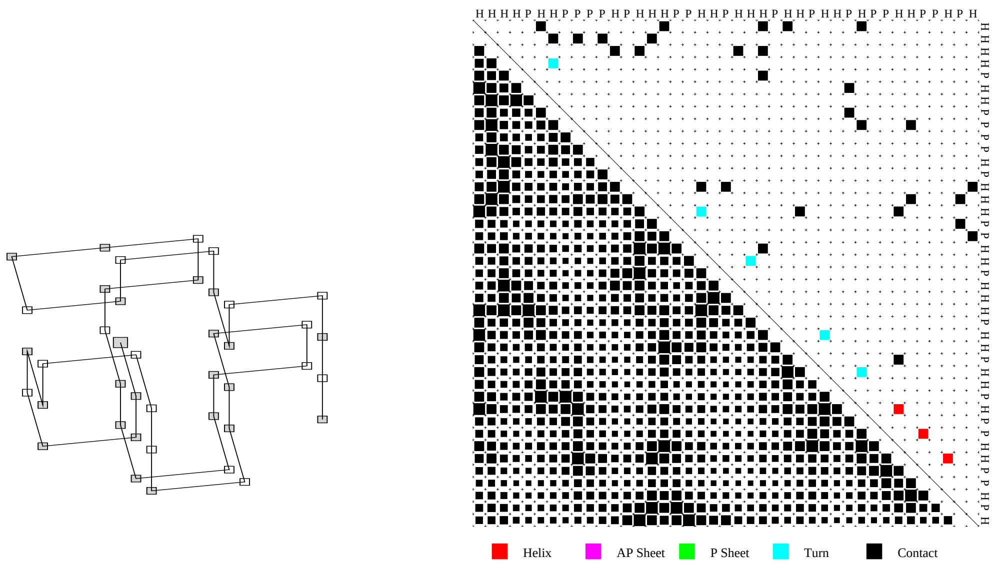


Figure 26: Structural Elements on the SC lattice: the same parameters as for the SQ-example were used.

- α-helical conformations appear as bands parallel and near the diagonal
- parallel arrangements are parallel but far from the diagonal
- anti-parallel arrangements (α-helixes, β-sheets) of residues appear as bands perpendicular to the diagonal.
- turns are single contacts next to the diagonal

In our graphic routines simple contacts that are remote along the chain are also indicated.

5.3 Structure Distances

To define Structure Distances is essential for statistical investigation on generic properties of the mapping from sequence space to shape space which in turn is crucial to learn about molecular evolutionary adaptation without respect to the method the structure is achieved.

5.3.1 Comparing Relative Moves

The most trivial structure representation for comparison is the string of relative moves.

- The number (of pairs) of identical directions is $(n - h(\mathcal{A}, \mathcal{B}))$ when n is the length of the two structures $\mathcal{A}$ and $\mathcal{B}$ in the notion of relative moves, h is the hamming-distance between two strings of equal length.

- Structures of relative moves can be aligned using standard dynamic programming procedures defining gap-penalties and edit costs for the exchange of directions. Exchange is calculated by an edit-table (see section E for each basic symmetry operation. The edit cost is based on the angle, a transformation from one to the other move affords and the class this move falls to. Gap-penalties correspond to the value of the maximum exchange edit costs.

The distribution of moves in such classes has been calculated for SAWs very recently [86].

5.3.2 Comparing Contact Matrices

Another versatile data structure for comparison is the contact matrix.

- The number of identical contacts is defined as

$$N(\mathcal{A}, \mathcal{B}) = \mid \{i, j \in \mathbb{N}, i < j | c_{ij}(\mathcal{A}) = c_{ij}(\mathcal{B}) = 1\} \mid \tag{27}$$

- Defining costs for exchanges and Insertions/Deletions of bonds, depending on the absolute and relative position in the string a score matrix can be generated and, once more, subjected a dynamic alignment procedure.

  1. First *CM*s (and *BM*s) are converted into lists of contacts (bonds) (one list, containing all contacts for each residue) and aligned in the following way: each bond-list of one structure is compared with each of the other, all entries with all others.
  2. In the following the contact lists l_{ai} and l_{bj} for each $i, j = 0....n$ residues in the two structures $\mathcal{A}, \mathcal{B}$ composed of the residues s_{ai}, s_{bj} are compared (where n denotes the length of the structures, κ the maximum coordination number and κ and l are the corresponding entries to be compared in the two lists).
     We define

$$z = (n - 1)(\kappa - 2) \tag{28}$$

a dependency of the type of bonds that are exchanged

$$q = q(s_{Ai}, s_{l^l_{bj}}, s_{Bj}, s_{l^k_{aj}}) \tag{29}$$

and ε a parameter that tunes, if the structure comparison focusses rather on local (e.g. $\varepsilon = \frac{1}{2}$) or global comparison. The exchange of two bonds depends on the distance difference of the exchanged positions, and the type of contact.

$$\text{ex.} = \delta(l^k_{Ai} \neq 0, l^k_{Bj} \neq 0) = q\frac{2}{z}(\mid l^k_{Ai} - l^l_{Bj} \mid)^{\varepsilon} c(\mathcal{L}) \tag{30}$$

(c considers a lattice dependent exchange constant (e.g. it is impossible on a square lattice to align bonds that are shifted ONE position) that corresponds to the size of the minimum turn size.)

For an Insertion (Deletion) the maximal exchange values are assumed:

$$\text{del./ins.} = \delta(l^k_{Ai}) = \delta(l^k_{Ai} \neq 0, 0) = q\frac{1}{z}(\mid i - l^l_{Bj} \mid)^{\varepsilon} c(\mathcal{L}) \tag{31}$$

3. The distance between two lists l_{ai} and l_{bj} is defined as the minimum of all possible combinations of distances and it's entries

$$D(l_{Ai}, l_{Bj}) = \min_{\forall k \neq l} \sum_{k,l=1}^{\kappa-1} \delta(l^k_{Ai}, l^k_{Bj}) \tag{32}$$

The overall distance between lists L can then be evaluated by application of equn. 46 to a score matrix filled up with these expressions and a distance dependent (non linear) gap - penalty g:

$$\Delta CL(A_i, B_j) = S_{ij} = D_{ij} + \ \max \ldots + g(i,j) \ldots \tag{33}$$

and by usage of a recursive dynamic algorithm as usual, where D_{ij} is dependent on $q(s_{Ai}, s_{Bj})$ and / or $|i - j|$.

An example for two structures on a square lattice is depicted in fig **??**.

5.3.3 Comparing Distance Matrices

The most common data structure used to compare "real proteins" is the distance matrix DM.

- The distance of two DMs in it's simplest form has been formulated as the RMS distance of corresponding entries of two structures [117].

$$\Delta DM(\mathcal{A},\mathcal{B}) = \frac{1}{n-1}\left(\sum (d(a_{ij}) - d(b_{ij}))^2\right)^{1/2} \tag{34}$$

where $1 \leq i < j \leq n$. As this does not discriminate between local and global similarities the usage of a cutoff-distance has been suggested in [117] (and refs. therein) so that local pattern similarities are measured.

Another possibility we introduce here is the application of the distance dependency within the chain

$$\Delta DM(\mathcal{A},\mathcal{B}) = \frac{1}{n-1}\left(\sum (d(a_{ij}) - d(b_{ij}))|i-j|^{\varepsilon})^2\right)^{1/2} \tag{35}$$

where ε stands for the focus on short-range or long-range conservation of structures if $\varepsilon < 0$ or $\varepsilon \geq 0$ respectively.

- Another method would be a standard clustering method using Wards Criterion [27]. Pairs of Distance measures are sorted according to their corresponding euklidian distance. Starting with the smallest distances, nodes are added stepwise and clustered, building rooted trees that can be compared then by standard tree editing operations [94].

- Again we can use a dynamic alignment of a score matrix (eqn. 46) on the entries of the differences of corresponding entries of two distance matrices of two structures. Those entries can either be generated by using the difference only

$$D_{ij} = a(|d_{ij}(\mathcal{A}) - d_{kl}(\mathcal{B})| + b)^{-1} \tag{36}$$

or, by applying the dynamic alignment procedure to generate entries of S_{ij} from

$$S_{ik} = max\{a(|d_{ij}(\mathcal{A}) - d_{kl}(\mathcal{B})| + b)^{-1}\} \tag{37}$$

so to produce a best equivalence of positions between the two environments compared [107]. S_{ij} then becomes the maximum score obtained by applying the dynamic algorithm operating on the rectangular matrix defined by all values of j of $\mathcal{A}$ and l over $\mathcal{B}$.

- Based on the RNA analogy a structure distance between structures, encoded by their elements of secondary structure can be formulated.

5.4 Sequences

5.4.1 Alphabets

Alphabets naturally have a tremendous influence on the structure formation. The most important properties of amino-acids are listed in appendix 15 . In all our models we do not account for different sizes of the residues such leaving space for further work.
Dill's HP - alphabet is the simplest possible one. It distinguishes between hydrophobic and hydrophilic residues only, solvent is not regarded explicitly.

Hydrophobic (**"H"**) amino acids are:

`A, I, L, M, F, W, Y, V, C`

Hydrophilic (**"P"**) are:

`G, P, S, T, N, Q, D, E, K, R, H`

5.4.2 Sequence Distances

Sequence distances can be defined for any sequence of coarse such as LPs. For the Sequence Space the natural measure is the Hamming distance, defined as the minimum number of positions to be exchanged to convert one string into another.
Other measures, however, appear promising when one considers a greater variability of bond combinatorics and each residue is able to bind to more than one other at the same time. Applying a dynamic alignment procedure using entries of a Dayhoff Matrix is a common method. Simplified exchange matrices for *LP*s are depicted in chapter E.

5.5 Energy Functions

Potentials from bond angles, lengths, electrostatic etc. are neglected. A general form of the energy function can be formulated as follows:

$$E(A, \mathcal{A}) = \sum_{i}^{n} \sum_{j \geq i+1}^{n} E(s_i, s_j) d_{ij}^{\alpha} f(s_i, s_j, |i-j|) \tag{38}$$

so that contributions are considered for topological neighbours only. E is the listing of pairing energies listed in a matrix, depending on the nature of the residues s_i, s_j only. f is a bond-dependent function, as different types of bond may have different distance e dependency.
In particular, Dill's potential considered the simplest case only for c instead of DM entries d and only one pairing force, stabilising the overall structure so that $d_{ij} = c_{ij}$ and $f(...) = 1$. (This refers to a cutoff a = 1 in our model.) In our implementation contributions can be considered up to a certain cutoff distance:

$$d_{ij}^{\alpha} = 0 \text{ if } d_{ij} > \text{ cutoff} \tag{39}$$

An interesting approach is Crippen's empiric potential considering 4 different classes of amino acids with respect to the distances along the chain [17]. There are non-residue specific interactions for each separation $(j - j)$ =3, $(j - i)$ =4, and a class for $(j - i)$ = 5-7. For residues more than 7 residues apart 4 classes of interaction are used.

In general it is useful to distinguish between contributions that are specific for the conformation and a separate term reflecting the contribution of an average over all conformation.

5.6 Lattice Proteins

In this section I will shortly discuss the different types of Lattice Proteins as well as their advantages and shortcomings.

5.6.1 HP-Type Lattice Proteins

This type of proteins have been developed by Lau and Dill [54]. The model is as simple as can be:

- Each residue is put on one node on a lattice
- The chain has to satisfy the conditions of a SAW
- The alphabet consists of two residues only: hydrophobic residues have a stabilising contribution when they are topological neighbours, i.e. nearest neighbours but not along the chain.
- Hydrophilic residues and solvent positions are without influence to the overall energy.
- Structures are the SAWs that have the minimum free energy, i.e. the maximum number of HH contacts.

These assumptions are reasonable, as the hydrophobic effect, which is commonly assumed to be the major driving force for structure formation is the one and only force. The excluded volume is regarded and computations can be done with integers as angles, bond length and the potential are discrete values. Structures can be easily visualised and are very intuitive.
Disadvantages are the lack of discrimination between amino acids of different polarity and size. Consequently other types of interactions and packing effects are not regarded. Unfortunately the number of possible configurations is still too large to permit exhaustive enumeration for large ensembles of random structures. Furthermore there is a high probability, especially for short chains that there are degenerate ground-states.

The simplicity and conceptual clearness of the model has inspired several other groups to use it for investigations:
Lau & Dill [54] performed exhaustive simulations of random chains on a quadratic lattice. They investigated the affection of "protein" native structures under the influence of single and double-site mutations. Replacing hydrophobic elements in the core has a destabilising effect, the surface positions are less sensitive. Neutral mutations are most likely, destabilising less

and stabilising mutations are least likely for single site and double-site mutations each. Most mutations with influence on the structure formation are in the core, which is consistent to natural observations. Sequences with too many (i.e. > 70%) H's have many different compact structures of equal lowest energy and are thus degenerate.
An interesting approach concerning neutral evolution was done by Lipman and Wilbur [58]: They defined as an *Equivalence class* the maximal set of sequences which all map to the same structure and as a *Path family* the maximal subset whose sequences form a continuous single step - network and as a *Mutational network* a maximal set of sequences which have been connected by single mutational steps.
With increasing chain length a larger fraction of possible structures become linked to a single network. As the network size is drastically reduced if neutral mutations are not allowed, neutral mutations are important on an arbitrarily mutational path within the network. The necessity of neutral mutations increases with chain-length Determining the non-neutral neighbour differences gives the same results: a single neutral mutation has a significant probability of changing the possible phenotypes one mutation away.

5.6.2 Other Lattice Proteins

Random Link Chains types: An interesting approach is the use of sequences with a random distribution of links instead of a generic potential function simulating attracting or repulsing forces between residues. [88, 87]. A recent study based on a MC optimisation of a 3x3x3 cube revealed a behaviour of quick formation of a compact state and a slow re-folding process to something that could possibly refer to a ground state. One consequence, observed in [92] is the lack of similarity between the ground state and the first “excited” state. The pronounced energy gap was postulated to be of greater importance for a successful fold to the ground state than the molten globe state.
Beside of an explanation of the driving force for the collapsing process, the model also lacked a reasonable explanation of how such structural changes work and what corresponds to something like secondary structural elements.
Molecular Modelling on Lattices: These models are basically investigated by Skolnik, Sikorsky and coworkers [95] (and references therein). In their studies they investigated the folding pathways by the means of MC-simulations over a discrete conformational space. X-ray data are taken as input or, at least for comparison to results and trajectories of folding / unfolding. Side chains are usually omitted.
Among others they used a 210 lattice for bond-simulation with a HP symbol-set for the sequence and proved that in finding a native-like state the underlying lattice is not crucial for a

correct prediction of the folding pathway, they are characteristic of the underlying **topology** only.

Stochastic Optimisation of Lattice Polymers: Genetic Algorithms have been applied very recently to the protein folding problem in general and to the lattice simulations in particular [110]. Unger and Moult claim to mimic the folding pathway and thereby reducing the computational requirements. The idea is based on the "Pathway hypothesis": an algorithm must mimic the pathway to be successful. The early folding of units is kinetically determined. GAs have the tendency to preserve local favourable conformational features through successive generations. A pattern used to describe a feature, common to many current solutions, will preserve more likely, be herited and later on resembled to the overall structure. So GAs might in this case rather be looked as procedures, sampling points in the conformational space that are on the way to the (global) optimum instead of a population of structures.

6 A Greedy Chain Growth Algorithm

6.1 Motivation

Several considerations, some of them already suggested above, tempted us to design a "kinetic" folding rule to lattice polymers:

- Proteins first fold on the ribosome: finding the right quadrant in shape space might well be directed by a sequential folding on the ribosome [70, 25]. The real folding process is still not fully understood, there are however several factors induced in vivo, such as chaperones and chaperonines: they attach non covalently to the growing chain, such preventing misfolding and attachment to other particles that are present in the cell.
  So we are, in this work concerned solely with unaided peptides excreted directly from ribosomes into solvent. The idea is illustrated in fig. 27.

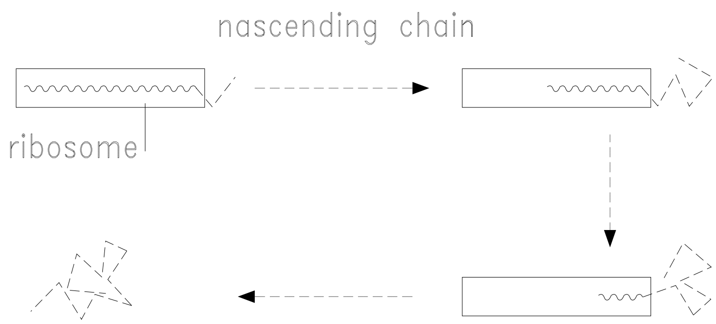


Figure 27: Unguided folding of a growing peptide on a ribosome: the growing protein chain is extruded to the lumen, subsequently solvated and folds into an initial configuration.

- Molecular dynamic studies: Some recent works from VanGunsteren's group with molecular dynamics on Hen-Egg-White Lysozyme [63] claim, that folding follows a scheme of initial collapse and later on the formation of secondary structure. Also there has been some efforts of (strange) people, folding stepwise proteins by molecular dynamics [19] thus yielding satisfactory results for short peptides (namely delta-sleep-inducing-peptide with 9 residues chrrrr...).

- Statistic investigations from RNA secondary structures: correlation-lengths from RNA kinetic folding indicated that it might well play no role to characterise the energy and the structure landscape, as long one is only interested in statistical properties especially in the near neighbourhood of any typical (i.e. random) structure. Even low-energy results kinetic folds should therefore be a reasonable approximation to characterise the energy and, maybe even the structure landscape.

- RNA folding algorithms: Some models for RNA also work that way [67, 68, 105].
- As finding the ground state is NP-complete: (see above) and therefore not applicable for statistical investigations,

So for both, a technical reason and experimental interest we decided to design a greedy chain-growth algorithm that works basically in $\mathcal{O}(n)$ and, hopefully finds at least a very good solution. Our expectations are that, at least a few solutions might correspond to the molten globe state and in some cases the ground state is found. Furthermore the size of the molecule is limited, as there is only one nucleation center (at the end) and once a hydrophobic core is formed, no rearrangement from inside to outside and vice versa can take place. The size of folding domains, termed "continuous regions" that arise from such nucleation events has been estimated to lie in a range between 40 and 150 residues [114]. (That should be enough motivation. If not, I can't help it. And I don't care: cause I'm a lumberjack and ... [16].)

6.2 Description

The algorithm is essentially based on a grown SAW assumption under the constraints of energy potentials: The algorithm is a "pseudo - kinetic" (i.e. without regarding association or dissociation constants) chain growth algorithm:

- The first residue is positioned on an arbitrarily chosen point on a lattice defining the starting point. The very first move is always defined "F".
- Next the list of possible move is tried and the energy calculated with respect to the overall configuration(so far). As dihedral angles are usually not 180 degrees, we start with a curve, e.g. "FLR" is systematically tried . (However, which move is tried first is a parameter that can be defined during the input.) Other residues as predetermined by the sequence are added subsequently without consideration of all the following residues.
- At each step the best of the possible configurations is evaluated (or: at each step the best overall value for all possible configurations of the last $m = 2, 3, 4...$ residues is evaluated): The influences of the neighbourhood are retrieved from an "energy value" matrix and systematically evaluated for the all neighbours within a certain (adjustable) cutoff distance. The proper choice for the next move is evaluated by backtracking the structure and summing up the energy values from the possible neighbours:

At the k-th step assume a configuration as a list of relative moves

$$\mathcal{A}_{1,k} = (a_1, a_2, \ldots a_k) \tag{40}$$

on a given lattice $\mathcal{L}$ with $a_i \in \mathcal{R}_{\mathcal{L}} = \{r_1, r_2, \ldots r_\kappa\}$ and the energy, that corresponds to the configuration so far

$$E_k := E(\mathcal{A}_{1,k}) \tag{41}$$

Next we generate all $i \leq l \leq \kappa^m$ possible configurations for the search depth m

$$\mathcal{A}^l_{k+1,k+m} = (a^l_{k+1}, a^l_{k+2}, \ldots a^l_{k+m}) \tag{42}$$

and determine the energy for all l

$$E^l_{k+m} = E(\mathcal{A}_{1,k} +\!\!+ \mathcal{A}^l_{k+1,k+m}). \tag{43}$$

(where $\mathcal{A} +\!\!+ \mathcal{B}$ stands for appending the list of moves $\mathcal{B}$ to the list $\mathcal{A}$). Next we sort these energy values and determine the best configuration

$$E^\star = E(\mathcal{A}_{1,k} +\!\!+ \mathcal{A}^\star_{k+1,k+m}) = \max_{\forall l} |E^l_{k+m} \tag{44}$$

We then generate the new configuration for the next step

$$\mathcal{A}_{1,k+1} = (a_1, a_2, \ldots a_k, a^\star_{k+1}) \tag{45}$$

if and only if $E^\star \geq 0$, i.e. the configuration is a SAW.

- Backtracking and searching for alternate solutions is done if the chain is in danger of getting trapped (which is very unlikely on a 3D-lattice).
- Finally repeat the procedure with this structure as the starting configuration.

The procedure is illustrated in fig 28. To end up at the minimum free energy conformation that way is very unlikely. Under these constraints low energy conformations are compact with a hydrophobic core and the polar residues forced to the surface, although not explicitly penalised. The algorithm has to be deterministic for the reason of avoiding random effects, destroying correlations (see Random Energy Model [20]).

In the case of no energy constraints we yield completely stretched conformations if we assume a forward step any time and more compact ones, following the geometry if we alter the hierarchy for choosing the directions to change. Random and homogeneous sequences can be treated the same in the limit of vanishing energy constraints, if all moves can be chosen at equal probability.

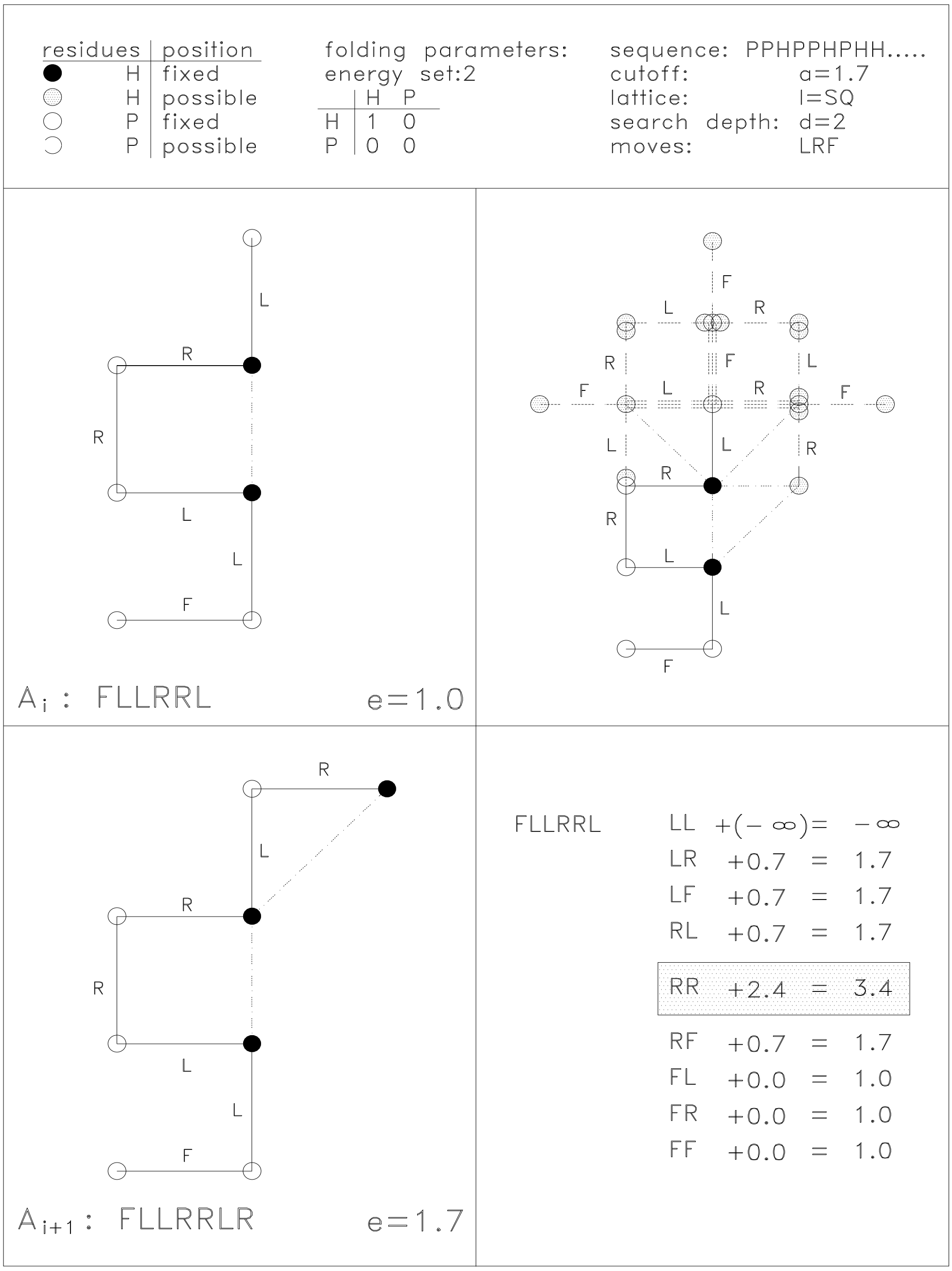


Figure 28: The greedy Chain Growth Algorithm: We assume a given configuration (top left) with one energy contributing neighbourhood between the two HH. Next all possible configurations of length 2 are appended (top right) and the overall energy is evaluated (bottom right). One configuration (LL) is forbidden and assigned an infinite energy contribution (∞). The best of all configurations, (RR) with the highest energy (3.4) is selected, and from there the first (R) move is appended. The resulting configuration (bottom left) has energy 1.7 and is used as starting configuration for the next cycle.
Full lines refer to bonds along the chain, dashed lines to possible moves, dash dotted lines to energy contributions within the cutoff distance.

6.3 Performance

As the algorithm is not designed (and not capable) to simulate the complete folding process of proteins, we do not expect to end up at the ground state. It is however important to find out, how close we come in the case of known analogies and if the algorithm is suitable for statistical investigations on the mapping to shape space.

6.3.1 CPU requirements

The performance can be estimated as follows: it is exponential in the "search - depth" m but linear in n. Time requirements therefore scales with $O((\kappa-1)^m n)$, when κ is the coordination number. The back-searching procedure eventually slows down the process, especially for low small κ when the chain gets trapped.

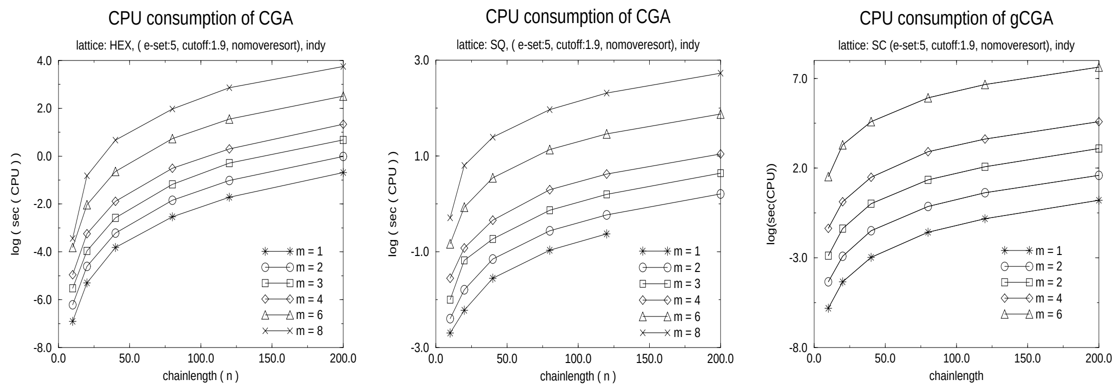


Figure 29: CPU requirements of the gCGA: the time effort for the HEX, the SQ and the SC lattice at various search depths and chain lengths. Results were obtained on an SGI-Indy with R4000 CPU and 32 Mb RAM

Data for three selected lattices are summarised in fig. 29.

6.3.2 Observations of the folding Process

Starting with a sequence we fold with different search depth m. There appears to be a limit, beyond which no clear further improvement can be observed. In the case of energy sets with repulsions it can be clearly seen that, the greater the search depth, the deeper the minimum that can be overcome. It is, nearly exclusively these regions, that contribute to the formation of better (i.e. more stable) conformations. We note the energy states, the number of traps and the number of degeneracies for random sequences of several alphabets (and energy potentials) at various search depths.

An interesting question is the length of the minimum path, i.e. $d(a) = d_{min}(a, N)$ from a given conformation a to the global (or any other local) minimum conformation N. This can be viewed as a measure for the accessibility of the native state N from any other conformation. In our implementation no conformations that are physically forbidden must be covered, but especially in the model with repulsions we can observe an eventual decrease of stability. This corresponds to regions, where the algorithm has only the choice between more or less repulsing interactions as the state, up to the current position is already frozen.

We also investigates the average behaviour of the compactness and of the energy in random ensembles. Results are shown in fig. 31. Again we do not find substantial improvement at larger search depth, which is a strong hint, that good, i.e. relatively stable and compact conformation can already be found at a very small search depth.

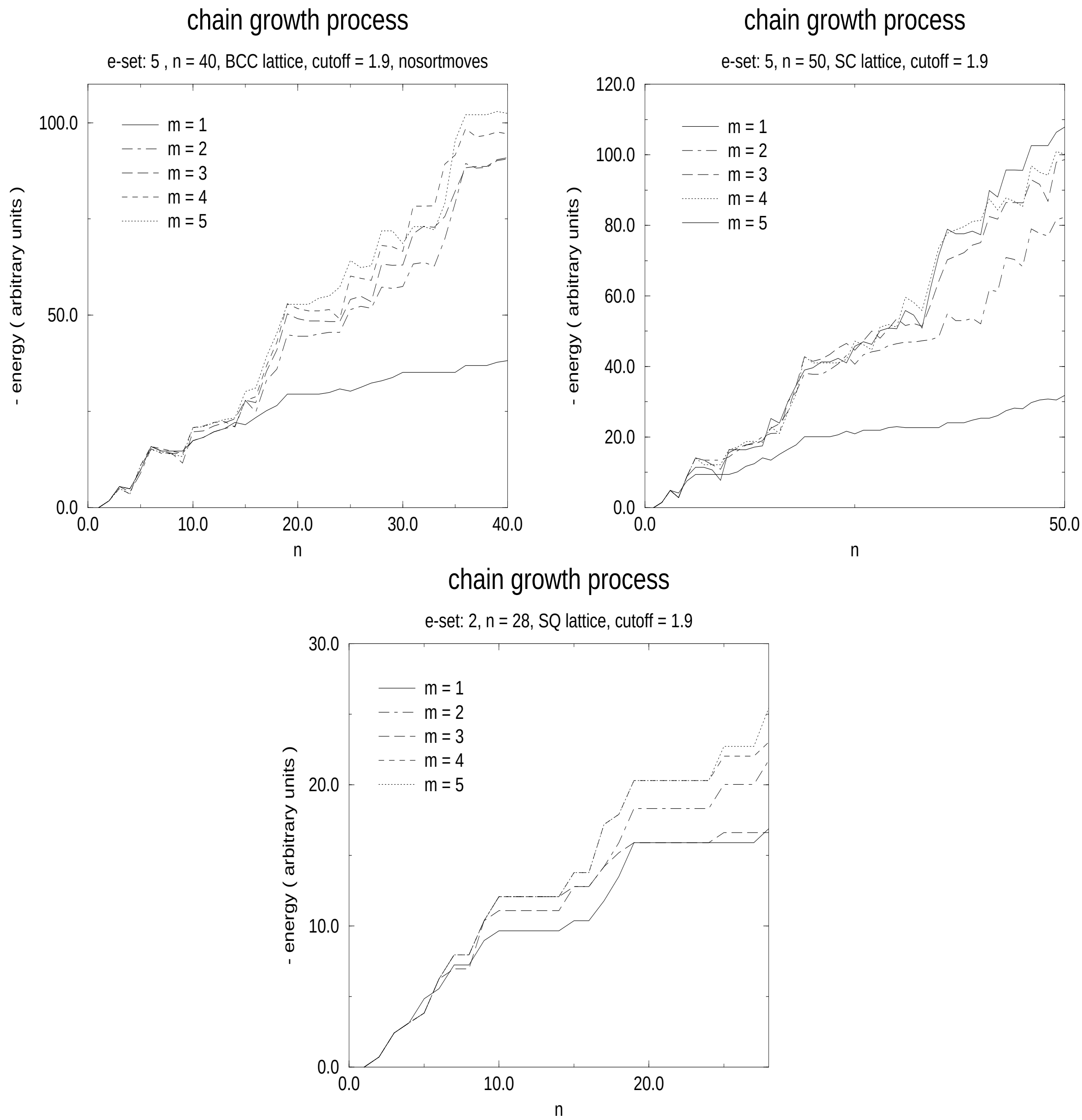


Figure 30: Observing the Folding Process at each step for the same sequence per graph and at different search depths.

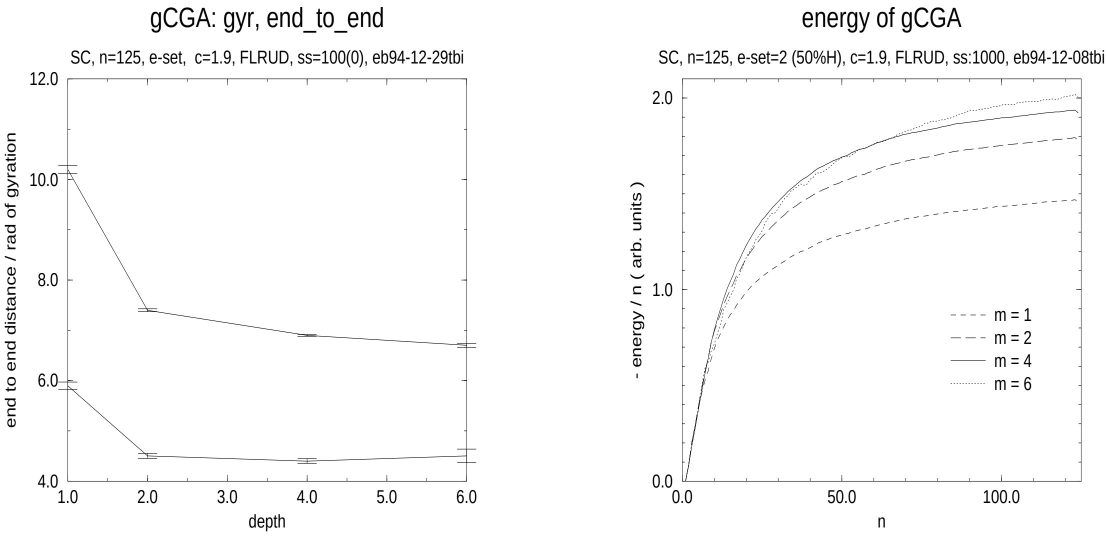


Figure 31: Observing the folding process: the energies, radius of gyration and the end-to-end distance for an ensemble of 1000 (d=6: 100) random sequences the SC lattice and chain-length n=125.

6.3.3 Secondary Structure Formation

In the following we show the statistics of the folding process and the formation of secondary structures along the chain.

We observe a number of secondary structure elements, however no significant increase with higher search depth. As the number of contacts increases as well stability and as the compactness increases, the only sufficient influence is the one on long range contacts.

The example for a sample of 1000 folded structures was analysed for the simple Lau HP alphabet and chain-length 125 on the SC lattice with cutoff a=1.9. For obvious reason we could not perform more than 100 calculations for the case m=6 so that these results are obscured by a lot of noise. Results are shown in the following fig. 32.

As it is only the energy contributing contacts that increase it is basically the possibility of long-range HH contacts that enforce a stronger compactification. If one accepts the process as the ribosomal chain growth then looking forward in a growing peptide can therefore be interpreted as looking for interactions with the already frozen part of the chain.

The start of secondary structures in dependency of the chain position is also analysed, results are shown in fig. 33. Here we find a decrease of parallel and anti-parallel sheets with the chain, but a constant factor for helixes. This can also be interpreted as the influence of long range interactions : the larger the distance between begin and end of sheets, the greater the chance is the two segments join in a parallel manner. In the case of helixes, there is an equal chance at any point as helixes are only stabilised by local interactions. This is exactly what can be observed by real proteins.

There are substantial differences with m: helixes are slightly decreased anti-parallel sheets are increased, beta-sheets remain rather uninfluenced. β sheets are the only type of secondary structural elements in this context where the range of contacts lies exactly in that range that is influenced by the increase of limited search depth.

Finally the chance of the moves was investigated, results are shown in fig. 34. As there is a strong determinism included in the procedure of the algorithm, it is interesting to see, how often decisions are made by the sequence of possible moves as they are predefined in the algorithm and how often results are outcome of the local optimisation process.

First we counted the numbers of degenerate solutions that are available at any point of decision. For obvious reasons there are more possibilities with increasing search depth. Next we looked at the statistics of the number of moves as they were chosen at any point of the folding process. There it can be clearly seen, that the determinism is subsequently battled down with increasing m and finally vanishes at m=6. After the initial folding of a core of size $\approx 25 - 30$ residues the initial conditions are overthrown.

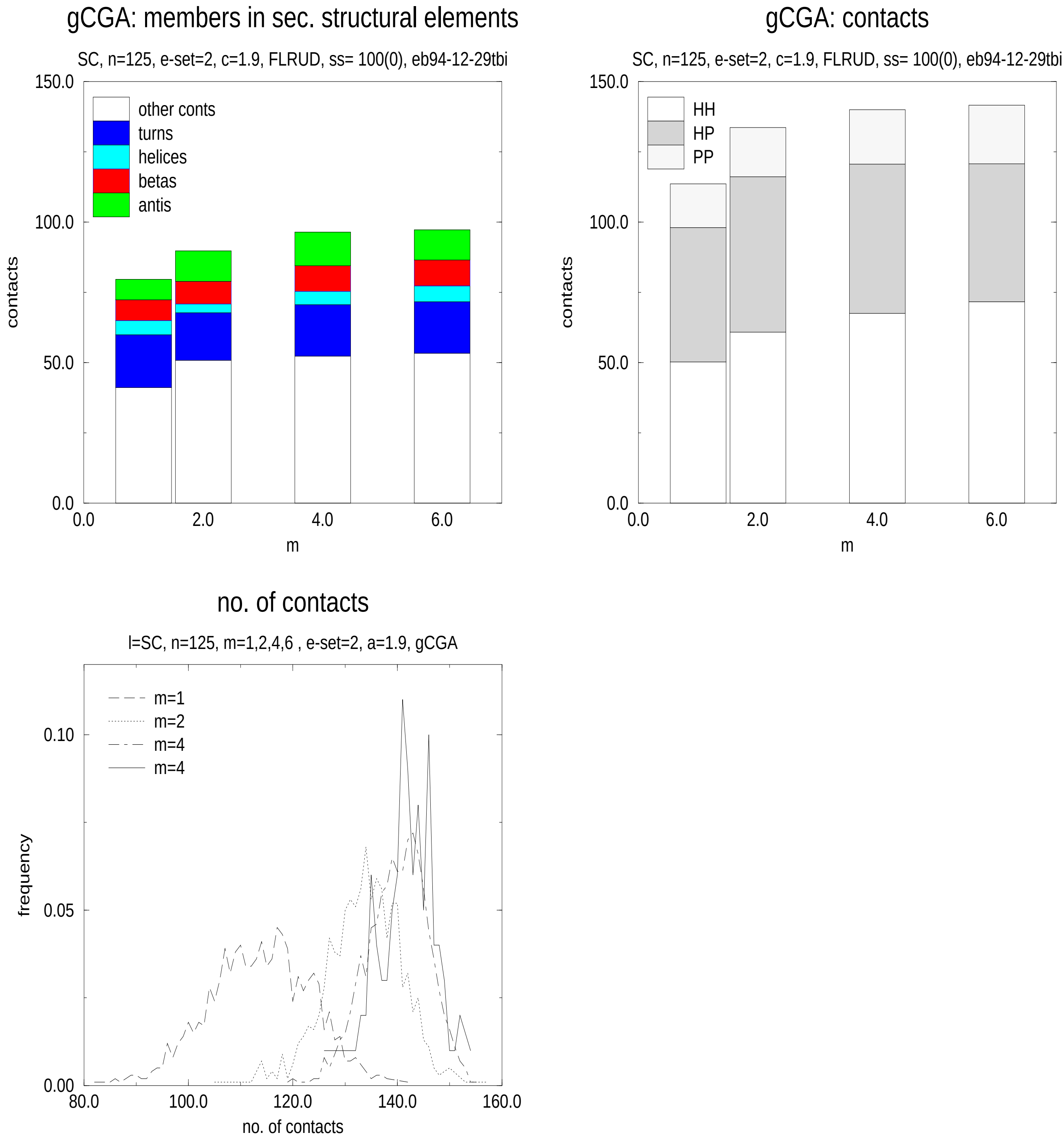


Figure 32: The number of contacts and secondary structures as a function of the search depth m. A sample of 1000 random sequences with the HP alphabet and length n=125 was folded on the SC lattice (folding parameters: a=1.9, m is varied, 50%H, e-set=2) a) Members of residues in secondary structural elements b) The number of different contacts with respect to the type of the residues as a function of the search depth m. c) The distribution of structures with a certain number of contacts at different search depths.

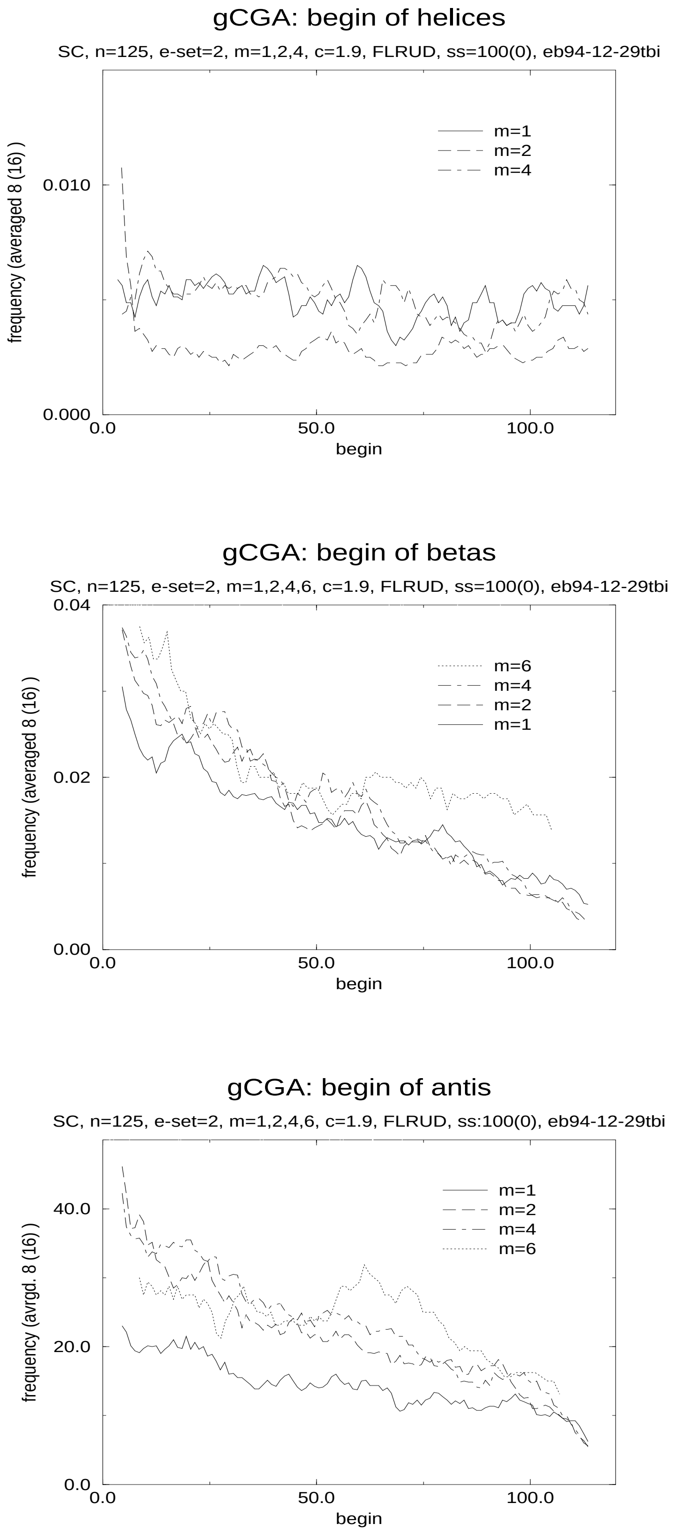


Figure 33: The position of the begin of secondary structural elements at various search depth m for an ensemble of 1000 random structures of the HP type, length n=125 and folded on the SC lattice with the same parameters as in the last figures.

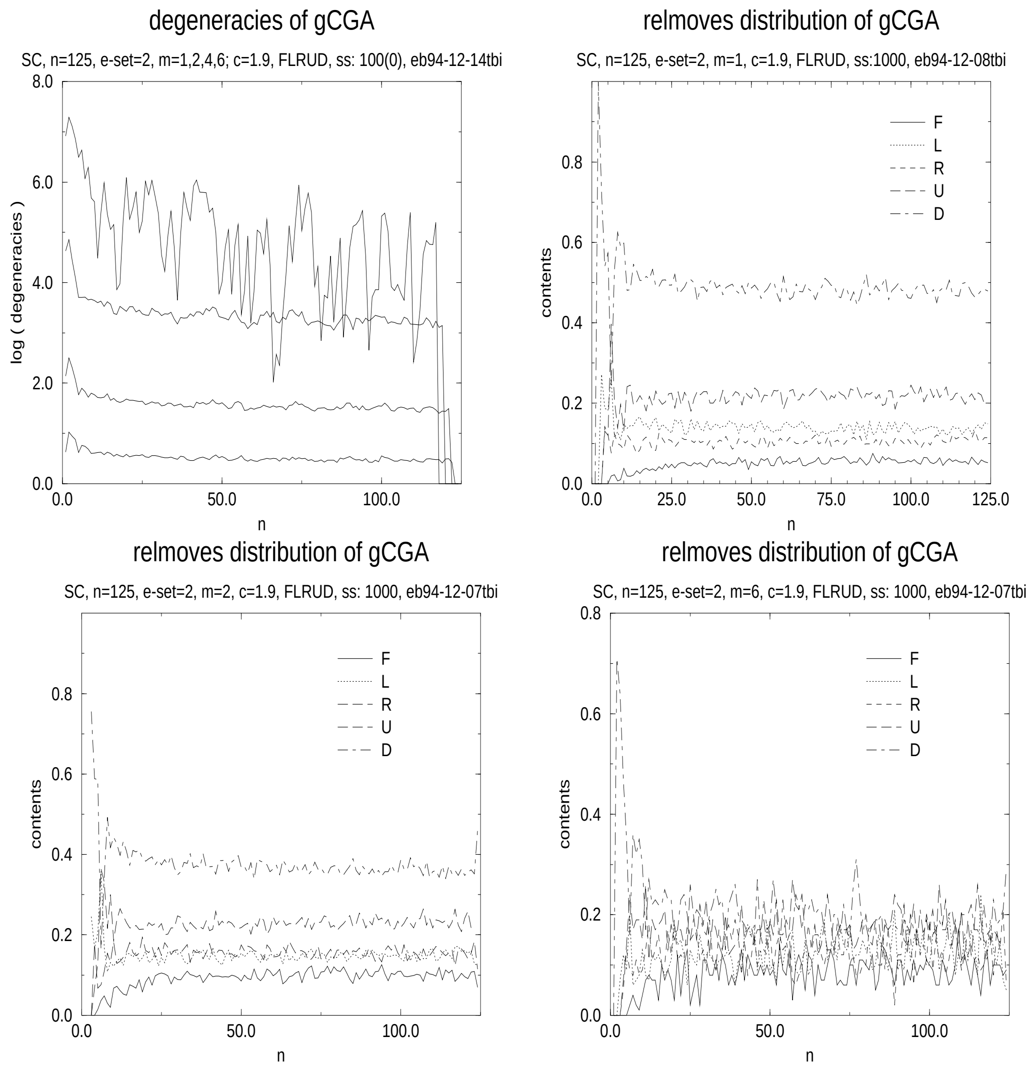


Figure 34: Moves distributions for the same ensemble as in the last figures. a)The average number of degeneracies at each step of the folding process (top left). b)The distribution of relative moves at various search depths. The default hierarchy (DRULF) is battled down at a sufficiently high search depth $m = 6$.

6.3.4 Long Range vs. Short Range Interactions

For easier visualisation we present a selected example at this point. We show the CM and DM as well as the structure for one, random sequence of n=125, folded at different search depth. It can be seen, that

- secondary structures are not really frequent
- the DM looks much like real proteins, with many local interactions and regions of long range interactions
- the regions of long range interactions are the same at all values of m but the numbers are slightly increased with m
- all structures are pretty compact
- hydrophobic residues are enriched in the center

These results are consistent with the findings from the last section on the ensemble statistics.

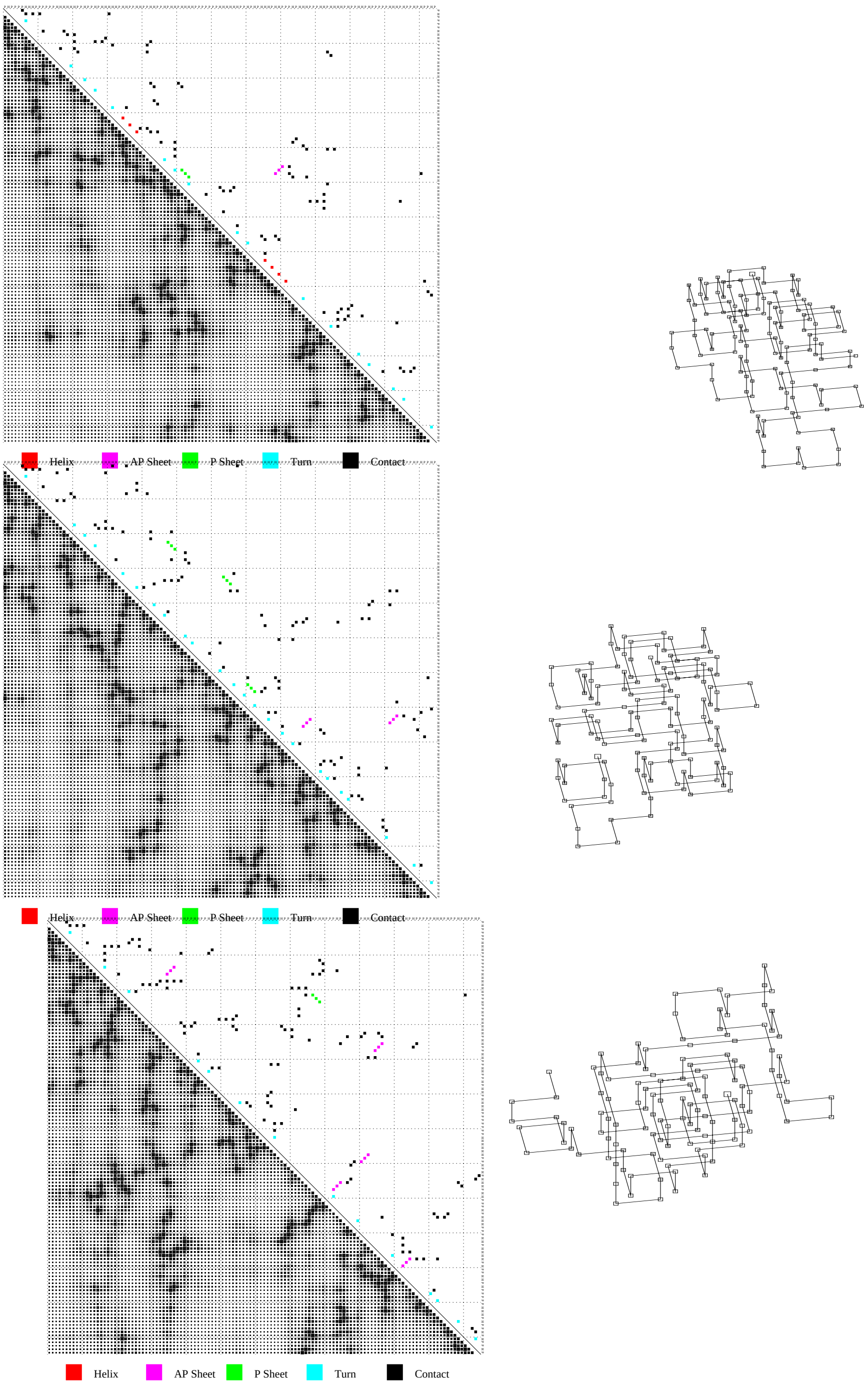


Figure 35: The same sequence, folded on the SC with the gCGA (e-set=5, a=1.9, n=125) with m=1 (top, 115 contacts), 2 (middle, 136 contacts) and 4 (bottom, 139 contacts).

6.3.5 Comparison to ground states

To quantify the outcome of the CGA algorithm we compare the results to the structures and corresponding energies, computed by exhaustive search from sequences of which a non degenerate ground state is known. We take sequences, kindly supplied by P. Stolorz [99] and fold them at various search depth.

Sequences are of length $n = 18$ which means, to find a structure by brute force one has to apply a search depth of $m = 16$ which requires too much memory and exponential CPU time. We folded all sequences up to $m = 11$. In fig. 36 it can be clearly seen, that the logarithm of the probability to find the ground state increases roughly linearly with m.

Furthermore we find in fig. 37 that it is, obviously of interest, if a sequence is folded that contains many or few residues that contribute to the structure formation. For short depth improved prediction can be observed at high contents of H, for longer depth most improvement is observed at small fractions of H.

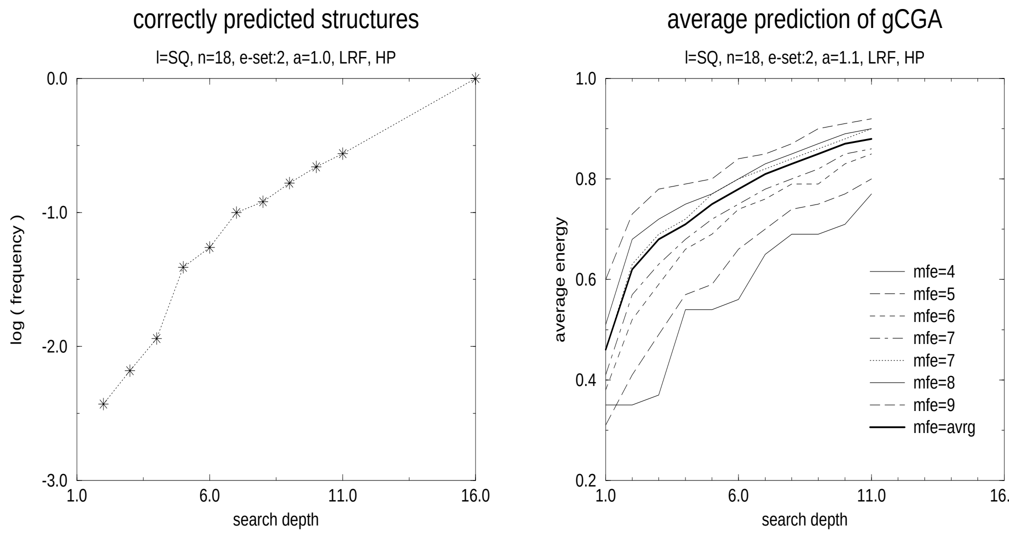


Figure 36: Performance of the Chain Growth Algorithm compared to the ground states: a) The log(fraction) of correctly predicted structures vs. the search depth. b) The average fraction of energies prediction accuracy $\langle e_{mfe}/e_{cga} \rangle$ vs. the search depth m for various values of e_{mfe} and averaged over all mfe values (thick line).

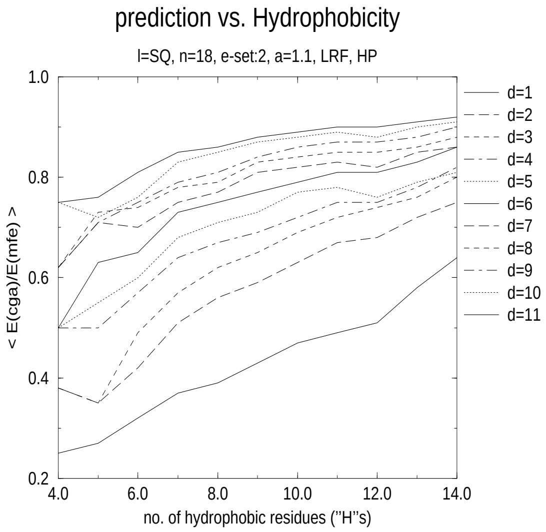


Figure 37: Performance of the Chain Growth Algorithm for structures with non-degenerate ground states: The energies at various search depths vs. the number of hydrophobic residues in the sequences.

7 A toolkit for LPs and SAWs

The toolkit has been written in ANSI C and has been tested on SUN, SGI and IBM workstations. It is based on the definition of a fundamental data structure with specific lattice constants, basisvectors etc. and can be easily extended for any regular lattice. Predefined lattices are HEX, TRI, SQ, SC, BCC, FCC, TET and TDKM. Measures for all structure distances enumerated above are provided as well as several graphic utilities for visualisation generating files in postscript language and .pdb format.
A detailed description of routines and basic data structures is given in appendix C.

7.1 Applications to Self Avoiding Walks

7.1.1 Length of SAWs

Lengths of walks and namestatistics of SAWs on all lattices were reported in [86].
Minimal lengths of traps can be easily figured and are listed in fig. 5.

TRI	6	FDDDDR
SQ	7	FRFRFRR
HEX	9	LDDDLDDDD
KM	15	FQLQLQLQLQLQLQK
TET	8	PQPPQRRP
SC	11	FLRLFDFLLRU
BCC	15	FXXYFXXZFZFYFXR
FCC	7	FRFRFRR

Table 5: The Minimal Trapping Configurations and corresponding lengths for SAWs on all lattices.

7.1.2 Collapsing SAWs

Since Flory's model [104, 14] postulatated the collapse of homopolymers to be a first order phase transition the question of how the folding is achieved is still unsolved. (The situation is even more ambigous for haeteropolymers.) The shape of the free energy landscape of (homo)polymer collapse is still not characterised exactly.
We wont't solve the problem, but we will show some results on the lattce approximation. On an ensemble of random SAWs with a given length an adaptive walk procedure of pivot moves

was applied till no improvement to the criteria smaller radius of gyration or higher number of contacts was achieved. Results imply that criteria are correlated: optimizing the number of contacts also reduces the radius of gyration significantly, in the other direction, however, there appears to be only a weak correlation if at all. Walklengths with the maximum contacts criterion were gnerally shorter. Obviously it is much more efficient to optimise the number of contacts than trying to yield a small radius of gyration. The latter would be a side product of optimising the contacts.
For obvious reasons walklengths for higher dimensional spaces were much longer. Application of the slithering snake algorithm did not result in sufficient improvement of compactness (results were reported in [86]).

7.1.3 Compact Self Avoiding Walks (cSAWs)

The scaling of the log(number) of maximal compact structures vs. the chain length is linear for all lattices and falls together with the prediction from [60] (results were reported in [86]).

7.2 Applications to LPs

7.2.1 Landscapes

We performed landscape calculations for some alphabets and the most rigid structure distance, the number of identical contacts. According to time and financial limits we could not test these results for the other structure distances and potential functions. Obviously structure landscapes exhibit nearly no significant correlation for the SC lattice. Results are shown in fig. 39 Correlations on the SC are lousy but evident which is reasonable as there are only 2 instead of 4 degenerate moves at each step. Consequently energy correlations are much longer. Furthermore we performed pilot studies for different chainlengths. Like the RNA case correlationlengths scaled approximately linearly with the chainlenth.

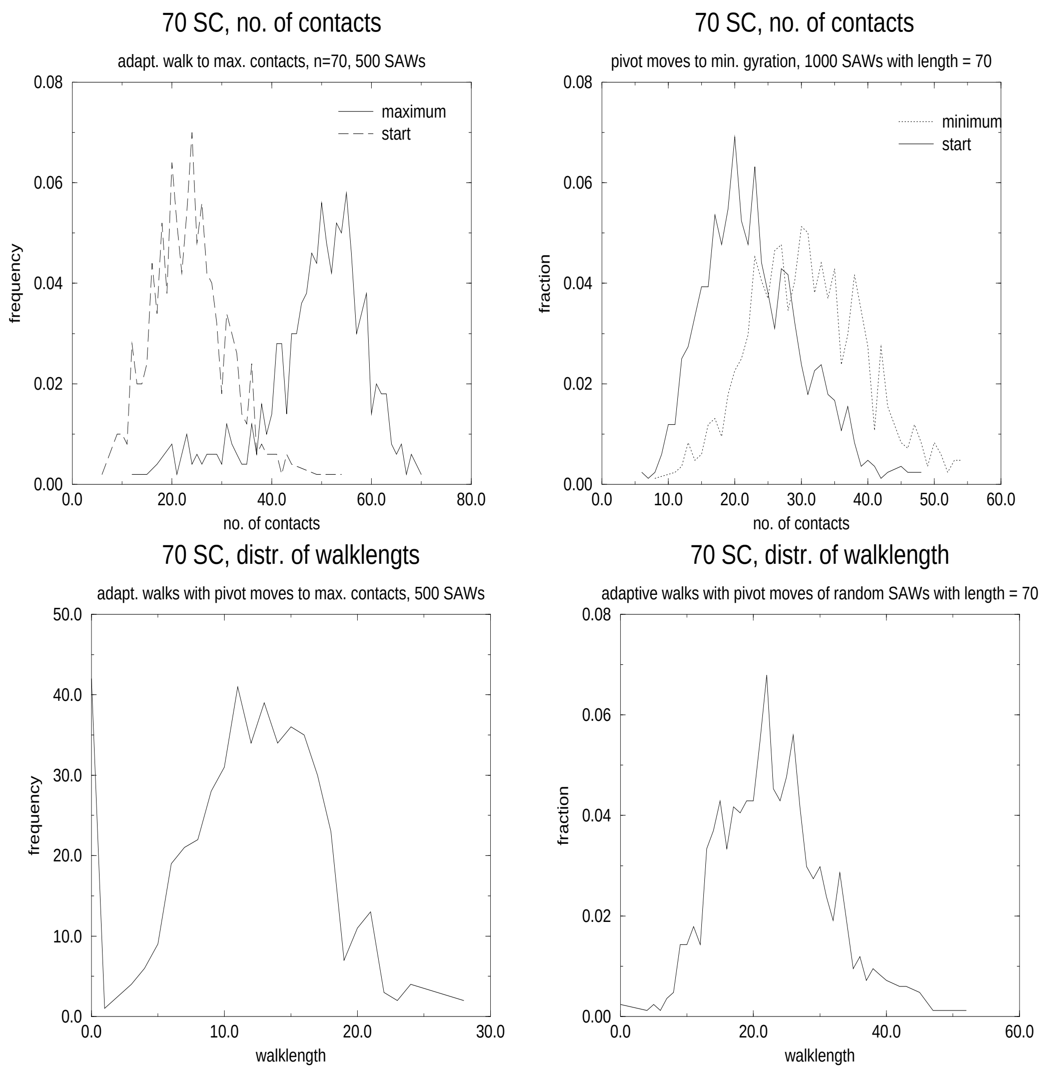


Figure 38: An example on the SC lattice, n=70 for the shifted distribution of contacts when optimizing the number of contacts or the radius of gyration respectively.

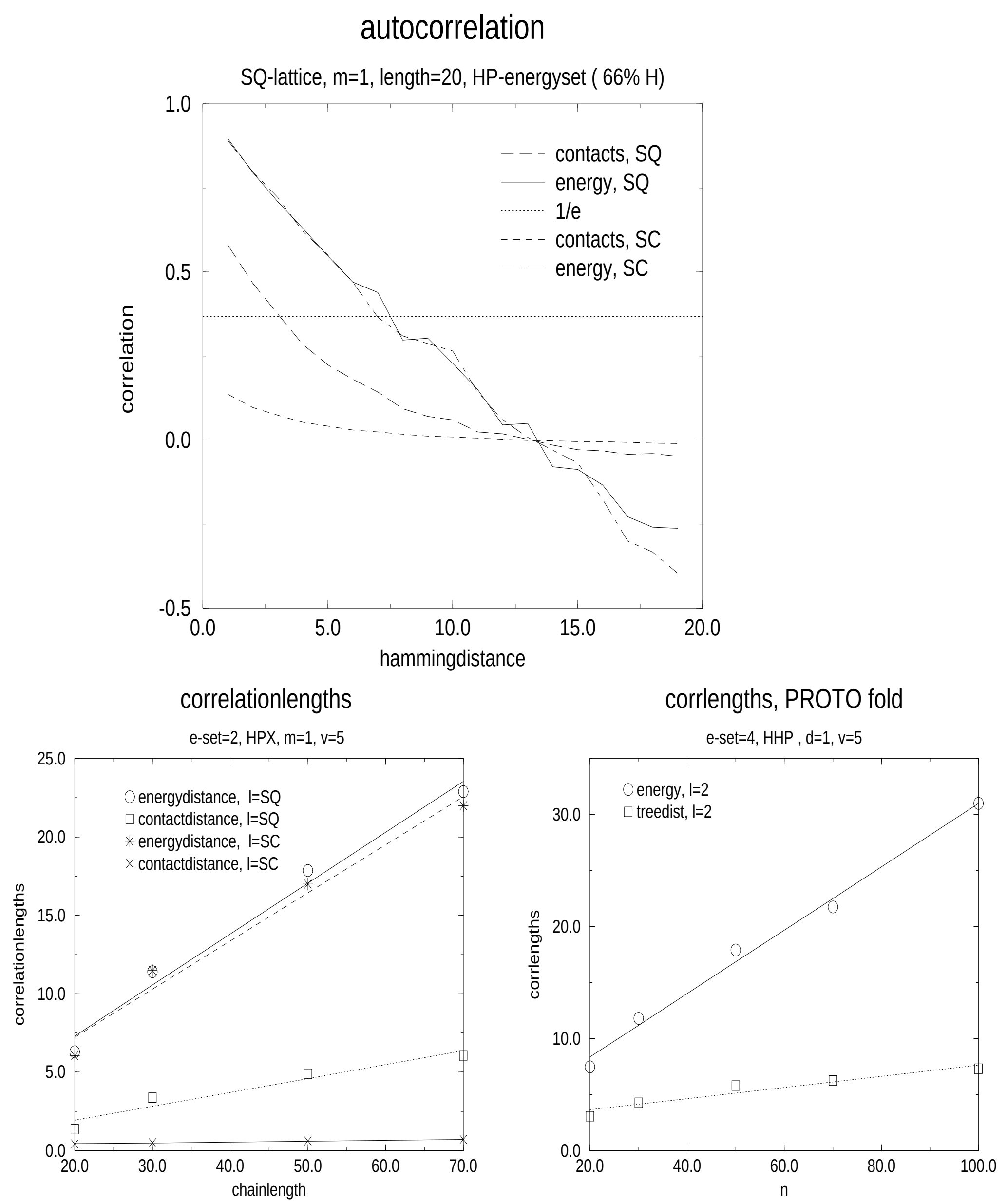


Figure 39: Landscapes of Lattice Proteins. a) top: autocorrelation functions of the energy landscape and structure combinatory map for n=20, and the simple HP alphabet with 66% H on the SQ and the SC lattice and the distance measure of identical contacts. b) correlationlengths vs. the chainlengths for the energy set 4 b) for energy set 2 and 66% H.

8 Discussion of the Protein Model

We presented a greedy chain growth algorithm and applied it to random sequences that were folded following the energy set as first proposed in 1989 by Lau and Dill [54]. The procedure has several advantages and several disadvantages: It works in $O(n)$ which gave us the possibility to use it for statistical investigations on large ensembles of random structures. It can be applied to any regular lattice, so that any lattice that seems appropriate to simulate protein structures can be used. The greatest disadvantage is the lack of a quality guarantee, such as a lower bound to the energy, compared to the ground state.

The algorithm's capabilities could not be completely investigated. Bearing in mind that there are 8 lattices, at least 4 alphabets, approximately. 4-5 different search depths, a flexible cutoff size etc. etc. there are thousands of combinations that could easily extend any theses directly to retirement. Although there were some promising pilot studies e.g. on the influence of other alphabets, I had to restrict the work to the simple HP alphabet, and investigations on the influence of a few lattices and the search depth.

Much attention was paid to the formation of secondary structures, as in analogy to the RNA model. This approach bears some potential towards the development of a coarse graining of the structure, such that enumeration of structures becomes easier. The influence of the lattices is similar to the findings from [39] and the results from SAWs [86] : some lattices favour the formation, such that secondary structures make up to 60 % of all residues. Namely the BCC lattices appears to be very suitable for that purpose. Unfortunately, however, we could not find a lattice where we could observe significant differences to the case of SAWs. This is not really a problem for the protein analogy, as several proteins are known with little or nearly no secondary structure at all. Furthermore, it is, in general a difficult task to find the right analogy of secondary structures on a lattice.

There are, however, some interesting findings from the investigations on the influence of the search depth. A significant shift towards longer range interactions can be observed in any case as well as improvement in radius of gyration, end-to-end distance and energy values. Nearly all of this improvement is due to the formation of long-range contacts with a stabilising effect, namely HH neighbourhoods.

Furthermore there are many compact states and some unoccupied beads. This could be interpreted as a state, similar to the molten globe. The differences to the global minimum can be assumed to be extremely small, such that it is likely that the global minimum is only slightly better however completely refolded due to the possibility to form some important long range interaction. Secondary structure formation remains largely uninfluenced from the value of the search depth.

Comparison to ground state proved, that it is, however only at a very limited amount possible to find the ground state with a sequential folding rule. This could indicate that it is at least in principle possible to select for structures not only with respect to their ability to fulfil certain tasks, but also if folding can be accomplished in a very simple and straightforward manner.
Another interesting idea, that appeared to be useful and makes the whole concept promising for further investigations, is to code structures with relative moves. The method is very fast and efficient for computational means.
The justification of the algorithm as a simulation of the folding of growing chains is difficult, mostly because only little is known about unguided folding from experiment.

IV: Final Remarks

9 Main Results

We have used two completely different approaches to gain some insight into the folding behaviour of biopolymers under the influence of genotypic mutations.

It is the clear base-pairing / non-pairing distinction and it's relevance for overall structure formation that makes it easy to enumerate structures in the case of RNA. The planar graph of the secondary structure can be easily converted to a coarse grained representation, which leads to a further simplification. Proteins, on the other hand, have a broad spectrum of interactions (as well as structural and catalytic tasks). This point could be partly circumvent by using th HP model. A simplified structure representation, however, is not known to date.

In detail the most interesting results from the RNA model were the following:

- A strong bias can be observed in the distribution of structure frequencies following Zipf's law.
- There are many extended neutral nets spanning sequence space.
- Both findings are independent from the chosen algorithm, structure representation or alphabet.
- A threshold value for percolation was observed. It strongly depends on the structure probability.
- Imposing strong structural requirements on the structure to be folded, reduces the accessible shape space and such the chance to find a neutral mutant.
- Due to combinatoric reasons and the ratio of sequences to structures the extension of neutral networks is smaller for small alphabet.

As there are few very common and many very rare structures, it is easy to pick a structure that belongs to a neutral net by chance. These results suggest a reasonable mechanism for optimisation: once a structure is selected for, it can explore the sequence space along a neutral network. There is a high chance that, if there is a fitter structure on some other neutral net, these nets will come close somewhere in sequence space and allow transition even when populations replicate with low error rates.

This means that optimisation of structures by evolutionary trial and error strategies is much simpler than often assumed. The combinatory map of RNA secondary structures is ideally suited for evolutionary adaptation.

In the case of lattice proteins a characterisation of the sequence to structure mapping appeared to be much more difficult. We developed a new folding algorithm that is inspired by the concept of an unguided, sequential folding process of a growing chain on the ribosome. Therefore we had to focus first on finding characteristics of this novel procedure. We had to design a comprehensive toolkit that works independently from the lattice geometry and is extendible. A large number of programming utilities was written and reported, some of them in [86]. The most interesting findings are the following:

- A structure representation based on the notion of secondary structures was not sufficient to serve as a reliable notion of coarse graining.
- The influence of increasing search depth was significant: a shift towards long range interactions was observed in any case as well as the improvement in radius of gyration, end-to-end distance and energy.
- This improvement in compactness is due to the formation of long-range contacts with a stabilising effect, namely HH neighbourhoods.
- The depth, from where no significant improvement was observed was very small, usually the greatest improvement was observed from search depth 1 to 2.
- Structures resemble the typical features from HP models: they are compact states with a hydrophobic core and few unoccupied beads.

These findings can be summarised: sequential folding of the simple HP-type lattice models leads to very good local minima resembling some crucial features of partially folded proteins, such as the molten globe state.

We also applied the concept of landscapes and combinatory maps to both systems. Correlation lengths are commonly viewed as a measure for the ruggedness of a landscape, which in turn is crucial for the performance of an evolutionary strategy. Although results from lattice proteins can only be viewed as preliminary, we find interesting similarities to the case of RNA secondary structures: correlation lengths are small compared to the diameter of the sequence space, structure correlations are smaller than energy correlations, higher for low dimensional lattices and scale linearly with the chain lengths for all parameter sets, lattices and alphabets. This leaves hope that, if the folding problem can be solved not even for the simple case of lattice approximations it should at least be possible to characterise the energy and structure mapping, such providing one with a powerful prediction strategy to estimate the influence of mutations on structure and stability.

10 Outlook

As lattice proteins and the HP model represent the simplest know representation of proteins, it is of great interest to use this models for further investigations. An interesting approach, that is currently being developed in our group, is to investigate the influence of residues with different size and shape on the number of structures that can be formed in a given volume.
Although the concept of relative moves proved to be useful, there are some caveats to that approach: this method is strictly bound to SAW models with equally sized residues and can not be easily extended to models that use a different size of side chains without giving up the advantages of easy computation and data handling. Similar problems arise, when one wants to compute surfaces, unoccupied lattice points etc.
It is interesting to compare results from sequential folding not only to energies, but also to the structures of the ground states as well as to structures on other lattices. This depends, obviously, on the availability of the data.
A threading routine, that fits structures from X-ray analysis to lattices will be written in the near future and incorporated to the package. So it will be possible to compare results to ground-states and the quality of lattices for representation.
Some distance measures have been introduced. They appear to be promising for statistical investigations, such as combinatory maps. It could thus be possible to detect, which structural features should be captured by a representation and which are most sensible to mutations.
Ceterum censeo Carthaginem esse delendam.

A Symbols and Abbreviations

n	chain length
h	hamming distance
G	Free energy
MFE	Minimum Free Energy Folding
MM	Maximum Matching
t	min. stacksize
u	min. loopsize
SAW	self avoiding walk
LP	Lattice Polymer
HEX	hexagonal lattice
TRI	trigonal lattice
KM	knight move lattice
SQ	square lattice
SC	simple cubic lattice
FCC	face centred cubic lattice
BCC	body centered lattice
TET	tetraedic lattice
TDKM, 3DKM	three dimensional knight move lattice
κ	number of nearest neighbours in a given lattice.
$s(A_i)$	residue in sequence A at pos. i
$\mathcal{A}_n$	config of seq. A with length n
a_i	rel. move of residue $s(A_i)$
$CM(\mathcal{A}) = c_{ij}$	Contact Matrix of a configuration
$DM(\mathcal{A}) = d_{ij} = d(\mathcal{A}_{ij})$	Distance matrix
$\|\|.\|\|$	euklidian norm
$gCGA$	greedy Chain Growth Algorithm
HP	binary alphabet of hydrophobic and polar residues

Table 6: Symbols and Abbreviations as used in this work.

B Data Tables

We performed landscape calculations for the AUI landscape with the RNA folding algorithm following the MFE criterion.

corrlength AUI, MFE(f)		
n	$l_e(n)$	$l_t(n)$
30	2.99	1.84
40	3.92	2.12
50	4.04	2.48
60	4.70	2.82
70	4.99	3.19
80	5.57	3.47
90	5.58	3.38
100	6.91	3.82
results from regression analysis:		
points	8	
std.dev.	1.31	.74
corrcoff.	.99	.99
slope	.05	.03

Table 7: Correlationlengths for RNA landscapes: The landscapes for the AUI alphabet and the corresponding data from regression analysis.

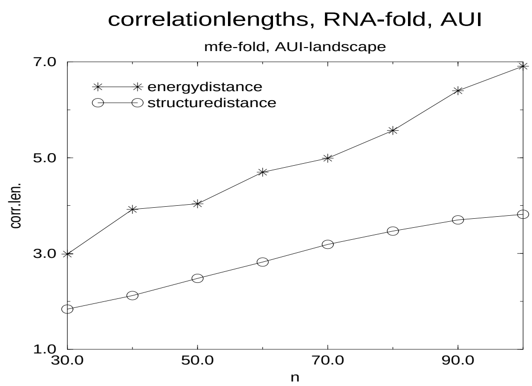


Figure 40: Correlationlengths for RNA landscapes: The landscapes for the AUI alphabet and the corresponding data from regression analysis.

Maximum Matching: corrlengths (enregy, treedist) at min. loopsize (u) vs. min. stacksize (t)																		
t :	—1—		—2—		—3—		—4—		—5—		—6—		—7—		—8—		—9—	
u	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$
n=30, AU																		
1	5.35	2.05			4.40	1.63			2.25	1.63			1.48	1.50			1.25	1.25
2																		
3	4.70	2.21			3.38	1.66			1.80	1.65			1.62	1.65			1.25	1.25
4																		
5	3.72	2.58			3.16	2.16			2.08	1.85			1.67	1.67			0.90	0.90
6																		
7	4.07	2.96			3.40	2.22			2.07	1.82			1.67	1.67			1.14	1.14
8																		
9	4.23	3.25			3.70	2.29			2.15	2.10			1.98	1.98			1.03	1.03
n=50, AU																		
1	9.26	2.62			7.56	1.93			4.32	2.07			2.71	2.50			2.40	2.32
2																		
3	6.58	2.41			6.50	1.88			4.34	2.06			2.46	2.36			2.21	2.23
4																		
5	6.11	2.83			6.14	2.65			4.34	2.34			2.39	2.45			2.15	2.19
6																		
7	6.64	3.7			5.69	2.73			3.70	2.27			2.57	2.55			2.36	2.36
8																		
9	6.67	3.88			5.97	3.17			3.48	2.33			2.70	2.59			2.26	2.26

Table 8: Correlationlengths for the structure and the energy landscapes of RNA secondary structures from binary sequences, folded with the MM algorithm.

Maximum Matching: corrlengths (enregy, treedist) at min. loopsize (u) vs. min. stacksize (t)																		
t :	—1—		—2—		—3—		—4—		—5—		—6—		—7—		—8—		—9—	
u	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$	$l_e(n)$	$l_t(n)$
n=30, AUGC																		
1	9.45	2.94			8.43	3.53			5.10	3.90			2.31	2.32			1.14	1.14
2																		
3	7.89	3.47			7.45	3.55			4.80	3.97			2.25	2.25			1.54	1.54
4																		
5	8.35	3.93			6.84	3.91			4.25	3.89			2.28	2.28			1.99	1.99
6																		
7	7.83	4.48			6.86	4.13			4.40	4.00			2.10	2.10			nep	
8																		
9	8.00	5.07			6.65	4.42			3.92	3.77			2.39	2.39			nep	
n=50, AUGC																		
1	14.57	3.14			11.81	4.09			9.60	5.74			4.76	4.54			3.84	3.81
2																		
3	12.81	3.71			8.49	4.14			8.90	6.01			4.41	4.40			3.28	3.22
4																		
5					15.18	4.44			8.55	5.75			5.02	4.70			3.03	3.00
6																		
7					14.17	4.86			8.08	5.56			4.77	4.62			4.29	4.28
8																		
9																	nep	

Table 9: Correlationlengths for the structure and the energy landscapes of RNA secondary structures from biophysical sequences, folded with the MM algorithm.

n	HPNX(3)		HPX(3)		HP(3)		HP(2)		HHP(2)		HPP(2)		HPN(7)	
	l(e)	l(c)	l(e)	l(c)	l(e)	l(c)	l(e)	l(c)	l(e)	l(c)	l(e)	l(c)	l(e)	l(c)
SQ														
20	3.40	1.52	6.30	1.35	6.20	3.86	-	-	7.46	3.06	-	-	1.82	2.48
30	4.31	2.56	11.40	3.38	6.20	3.90	8.50	2.50	11.80	4.27	10.50	1.50	2.98	2.62
50	6.39	3.87	17.86	4.90	-	-	8.30	3.00	17.90	5.80	15.00	3.40	4.34	3.88
70	10.00	4.70	22.90	6.07	-	-	-	-	21.75	6.25	10.20	4.60	4.71	4.64
100									31.00	7.30	21.60	6.03		
SC														
20	2.62	0.43	6.05	0.41	1.95	0.49	-	-	-	-	-	-	2.32	0.64
30	3.60	0.50	11.48	0.50	8.70	0.62	7.50	0.45	-	-	-	-	2.79	0.52
50	4.00	0.60	17.00	0.60	12.40	0.73	12.74	0.49	-	-	-	-	2.75	0.60
70	4.20	0.60	21.99	0.70	16.00	0.78	16.80	0.53	-	-	-	-	3.02	0.64

Table 10: Autocorrelation lengths for the energy landscape (l(e)) and the structure landcape (c(c), the number of identical contacts was used as the measure) of lattice proteins on the SQ and the SC lattice, folded with the CGA. Used alphabets and energy sets in parentheses are listed on top.

Maximum Matching: Adaptive Walks																			
t	u	n	average energy				energy of random optima				fraction	energy of local optima				walklength			
			avrg	var.	skw	kurt	avrg	var.	skw	kurt		avrg	var.	skw	kurt	avrg	var.	skw	kurt
n=30, AU																			
1	3	100	10.590	0.702	0.336	3.510	11.348	0.488	1.362	5.917	0.23	12.731	0.446	0.052	3.047	1.053	0.647	0.828	4.269
3	3	100	9.820	1.067	0.614	3.302	11.000	0.333	0.0	3.0	0.12	11.699	0.470	-0.136	2.638	1.641	1.058	0.425	2.957
5	5	100	6.130	3.733	0.941	6.497	8.000	1.556	-1.374	3.949	0.09	10.433	1.565	0.954	3.634	2.559	2.031	0.131	2.733
7	7	100	1.770	10.017	-1.254	2.637	1.555	8.469	-1336	2.786	0.09	10.586	2.887	0.153	2.069	2.164	1.418	0.187	2.805
n=50, AU																			
1	1	100	21.16	1.254	1.299	3.950	22.000	0.064	0.000	15.5	0.31	22.418	0.305	0.216	2.103	1.128	1.234	1.204	4.264
1	3	100	19.053	0.876	1.056	5.335	19.759	0.370	0.868	5.090	0.203	19.115	1.679	0.052	2.671	1.283	1.026	1.095	5.186
3	3	100	18.280	1.462	1.130	4.466	19.600	0.640	0.844	2.078	0.050	20.452	0.604	0.296	3.552	2.018	1.538	1.056	4.475
5	5	100	14.000	5.800	0.511	4.021	19.500	0.250	0.000	1.000	0.02	20.484	0.622	0.066	3.811	3.588	2.790	0.497	3.617
7	7	100	8.390	11.598	0.376	4.616	17.0	0	0	0	0.01	17.532	4.705	0.948	4.351	4.276	4.057		
n=30, AUGC																			
1	1	100	11.000	1.280	1.119	4.809	12.055	0.275	-0.067	3.585	0.18	12.424	0.366	0.269	2.738	1.357	1.099	1.083	4.766
1	3	100	9.520	1.009	0.232	3.199	10.312	0.715	-0.598	2.802	0.16	11.094	0.733	-0.039	2.964	1.449	0.939	0.378	2.721
3	3	100	7.220	3.392	0.449	2.900	10.167	0.139	-1.782	4.200	0.06	11.073	0.853	0.083	3.099	3.046	2.423	0.148	2.978
5	5	100	2.760	7.862	-0.102	1.169	3.000	9.0	0.0	0.0	0.02	9.698	3.040	0.524	2.444	3.639	2.462	0.094	2.592
7	7	100	0.14	0.96	-6.857	*	-	-	-	-	0.62	9.394	1.358	0.242	2.264	1.074	2.228	0.941	2.361
n=50, AUGC																			
1	1	100	19.370	1.313	0.597	3.481	21.0	0	0	0	0.03	21.122	0.674	0.322	2.863	1.748	0.773	0.703	3.472
1	3	100	17.050	1.167	0.337	2.933	18.111	0.765	0.217	1.314	0.09	19.084	1.025	-0.066	3.610	1.946	1.161	0.135	2.597
3	3	100	14.100	4.610	0.300	3.151	-	-	-	-	-	19.262	1.582	0.328	3.082	4.566	3.702	0.330	2.836
5	5	500	7.394	12.835	0.205	2.641	-	-	-	-	-	17.582	4.088	0.663	3.930	6.111	5.148	0.248	2.958
7	7	500	1.994	11.142	-1.117	2.357	0.416	3.343	4.215	18.993	0.12	16.12	10.003	0.583	2.756	5.659	10.301	-0125	2.422

Table 11: Adaptive Walks on the RNA landscape for the MM algorithm for n=30, n=50, the biophysical alphabet (BP) and a binary alphabet (AU).

SAWs: adaptive walks with pivot moves towards maximum number of contacts											
Dim.	Grid	length	adapt. walk		no. of contacts		radius of gyration		end to end distance		sample
			local optima	walk length	random	minimum	random	minimum	random	minimum	
2	HEX	30	0.28	1.8	4.2	6.5	8.4	5.0	9.6	6.1	500
		70	0.25	2.9	9.7	13.8	26.1	15.4	16.9	10.8	500
		100	0.27	3.5	13.7	18.9	43.9	26.4	18.9	13.7	500
	SQ	30	0.25	3.3	9.7	15.0	6.9	3.5	8.8	4.9	500
		70	0.21	5.6	24.5	34.5	21.0	11.1	15.2	8.5	500
		100	0.23	6.7	35.3	47.5	33.6	18.1	19.2	10.7	500
	TRI	30	0.22	5.1	23.5	34.3	5.8	3.0	8.0	4.3	500
		70	0.22	8.8	59.0	79.2	18.7	9.9	14.0	7.7	500
		100	0.23	11.4	84.7	112.1	30.5	14.9	17.9	9.2	500
	KM	30	0.08	7.0	10.0	26.5	18.7	6.8	12.8	5.1	500
		70	0.28	7.2	29.8	51.0	47.1	22.7	10.4	8.3	500
		100	0.37	6.7	47.1	71.0	66.9	37.4	24.2	14.7	500
3	SC	30	0.07	7.3	8.7	21.4	3.7	1.2	6.8	2.8	500
		70 +	0.08	11.8	23.9	49.1	8.1	3.0	10.2	4.2	500
		100	0.07	15.2	34.5	68.7	13.6	4.0	12.8	4.6	500
	BCC	30	0.30	5.1	16.8	28.4	3.0	1.3	6.0	3.3	500
		70 +	0.55	4.7	47.5	61.5	7.4	4.9	9.0	6.4	500
		100	0.59	5.3	70.5	86.5	11.2	7.4	11.3	8.1	500
	FCC	30	0.41	5.1	24.8	51.0	11.7	6.4	11.2	7.0	500
		70 +	0.67	4.0	98.9	116.1	30.2	21.2	18.4	13.7	500
		100	0.69	4.3	149.6	168.9	41.1	30.9	20.9	16.7	500
		500									
	TET	30	0.43	2.0	9.6	13.9	4.1	2.8	6.8	5.2	500
		70	0.59	1.9	30.1	35.8	10.4	8.4	10.6	9.0	500
		100	0.71	1.3	44.1	49.1	14.8	13.3	12.2	11.3	500
		500									

Table 12: Adaptive walks, using pivot moves towards structures with a maximum number of contacts starting from random SAWs with a given length.

SAWs: adaptive walks with pivot moves towards minimum radius of gyration											
Dim.	Grid	length	adapt. walk		no. of contacts		radius of gyration		end to end distance		sample
			local optima	walk length	random	minimum	random	minimum	random	minimum	
2	HEX	30	0.154	2.83	4.28	4.73	11.88	5.11	10.9	5.8	1000
		70	0.154	3.83	9.75	9.57	36.76	16.39	18.8	10.1	1000
		100									
	SQ	30	0.066	5.2	10.15	11.63	6.4	2.86	8.4	3.6	1000
		70 +	0.036	8.45	24.41	24.64	20.88	8.71	15.4	5.4	711
		100									
	TRI	30	0.066	6.37	23.78	27.08	8.34	2.78	9.0	3.9	1000
		70	0.051	9.22	57.98	59.44	25.98	8.51	15.8	6.2	1000
		100									
	KM	30	0.886	1.19	10.0	22.0	18.5	4.8	13.0	3.9	1000
		70	0.269	12.3	31.4	39.9	42.8	16.7	19.3	7.9	1000
		100	0.326	13.8	48.5	57.0	66.6	27.6	24.6	10.3	129
3	SC	30	0.002	13.76	8.83	16.07	3.66	0.80	6.8	2.4	1000
		70 +	0.002	22.88	22.83	30.97	9.19	*	10.4	3.2	839
		100	0.027	26.7	34.3	40.8	14.1	2.81	3.29	0.91	75
	BCC	30	0.095	10.88	16.43	21.93	3.06	0.85	6.1	2.5	1000
		70 +	0.241	13.95	47.0	46.2	7.5	2.6	9.1	4.2	1000
		100	0.28	15.29	70.6	64.8	11.4	4.2	11.2	5.2	514
	FCC	30	0.204	10.37	35.5	42.1	11.8	4.0	11.4	5.2	1000
		70 +	0.378	12.35	100.97	95.91	27.1	11.8	17.2	9.3	230
		100	0.404	14.38	152.45	140.23	41.69	17.49	20.6	10.7	156
		500									
	TET	30	0.187	5.41	9.6	10.3	4.1	1.5	6.7	3.6	1000
		70	0.36	5.84	30.0	25.9	9.7	4.9	10.2	6.4	1000
		100	0.395	6.3	45.9	37.6	14.6	7.9	12.4	8.0	1000
		500									

Table 13: Adaptive walks, using pivot moves towards structures with a maximum number of contacts starting from random SAWs with a given length.

AUGC						
1	1	40.74	0.38	142.75	0.21	2.66
	3	59.32	0.41	167.53	-0.05	2.63
	5	68.81	0.45	201.16	-0.18	2.47
3	1	70.05	0.49	157.10	0.18	2.88
	3	85.38	0.65	209.76	0.12	2.78
	5	93.46	0.73	266.21	-0.02	2.82
	7	100.84	0.54	286.62	-0.14	2.44
	9	107.99	0.55	303.30	-0.14	2.72
5	1	134.58	0.64	404.13	0.42	3.16
	3	140.76	0.66	369.98	0.29	2.95
	5	147.33	0.64	370.47	0.34	3.63
	7	153.26	0.60	360.50	0.36	3.75
	9	160.03	0.62	365.46	0.35	3.35
7	3	214.14	0.76	491.36	-0.23	2.50
	5	215.62	0.72	434.43	-0.20	2.26
	7	219.03	0.69	389.38	-0.33	2.55
	9	222.09	0.66	354.18	-0.28	2.47
9	1	239.60	0.76	147.11	-2.47	10.72
	3	241.62	0.71	120.66	-2.42	10.36
	5	242.59	0.62	93.41	-2.35	10.93
	7	241.84	0.66	102.22	-2.49	11.37
	9	243.32	0.58	67.74	-2.32	12.33
GC						
1	1	1.83	0.17	2.83	0.65	2.47
	3	5.03	0.08	6.36	0.27	2.63
	5	6.40	0.10	9.23	0.29	2.78
	7	7.96	0.32	11.23	0.07	2.62
	9	9.07	0.11	11.26	0.68	4.10
3	1	3.91	0.08	4.73	0.40	2.81
	3	7.88	0.10	10.19	0.19	2.66
	5	10.10	0.12	14.14	0.04	2.57
	7	11.96	0.14	18.15	0.05	2.56
	9	13.92	0.14	20.91	0.14	2.77
5	1	8.38	0.10	9.83	0.41	4.08
	3	12.02	0.12	14.25	0.05	3.05
	5	15.07	0.14	20.88	-0.10	2.58
	7	17.84	0.16	25.61	-0.21	2.76
	9	19.90	0.18	31.04	-0.19	2.71
7	1	20.46	0.18	32.35	0.70	3.91
	3	23.47	0.18	33.46	0.48	3.71
	5	26.23	0.19	36.55	0.39	3.62
	7	28.91	0.20	39.19	0.24	3.54
	9	31.28	0.20	41.79	0.10	3.30
9	1	44.31	0.27	69.70	0.15	3.11
	3	46.28	0.27	69.90	0.15	3.00
	5	48.26	0.26	64.46	0.15	2.85
	7	50.11	0.25	62.11	0.16	2.63
	9	51.81	0.25	61.70	0.19	2.70

Table 14: Neutral Neighbours for Maximum Matching for various minimal stack (t) and loop sizes, n=100, only one error mutants

Secondary Structure Statistics

```
======================================================================
length (n)    :  70
alphabet      :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice       : TRI               energy set   :   2
search depth  :   2               cutoff       : 1.9
---------------------------------------------------------------------
filename      : 70HP50.proTRI2280000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|            | max.   mean   +/-    var   skew kurt|
======================================================================
energies :  119.66
 p. res. :
gyration :   33.5   12.15  0.09   7.62  2.85 14.01
end to end : 16.1    6.47  0.08   6.32  0.50  3.28
----------------------------------------------------------------------
contacts :  111    102.69  0.45 202.34  0.18  3.00
type 1-1 :          38.5  37.5 %
type 1-2 :          44.5  43.3 %
type 2-2 :          19.7  19.2 %
other conts :
----------------------------------------------------------------------
turns  :     18.0   10.0   0.07   5.24  0.01  2.9
helix (no./res.):    0.17  1.38
betas (no./res.):    5.27 34.35
antis (no./res.):    0.98  4.06
======================================================================
======================================================================
length (n)    :  70
alphabet      :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice       : HEX               energy set   :   2
search depth  :   1               cutoff       : 1.9
---------------------------------------------------------------------
filename      : 70HP50.proHEX2180000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|            | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   63.24  30.5   0.25  64.17  0.28  2.95
gyration :   14.16  14.15  0.00   0.00  0.00  0.00
end to end :  5.01   5.00  0.00   0.00  0.00  0.00
----------------------------------------------------------------------
contacts :   26.00  25.00  0.00   0.00  0.00  0.00
type 1-1 :           6.3  25.1 %
type 1-2 :          12.5  49.8 %
type 2-2 :           6.3  25.1 %
other conts :26.00  19.01 0.01  0.04 32.6  998.0
----------------------------------------------------------------------
turns  :      1.00   0.00 0.00  0.00  0.00   0.00
helix (no./res.):
betas (no./res.):    2.00 7.99
antis (no./res.):
======================================================================
======================================================================
length (n)    :  70
alphabet      :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice       : HEX               energy set   :   2
search depth  :   2               cutoff       : 1.9
---------------------------------------------------------------------
filename      : 70HP50.proHEX2280000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|            | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   65.4   31.6   0.25  63.14  0.26  3.05
gyration :   19.78  15.36  0.03   1.07  0.84  3.80
end to end : 11.54   5.68  0.05   2.13  0.74  3.37
----------------------------------------------------------------------
contacts :   26.00  23.35  0.05   2.19 -0.78  3.33
type 1-1 :           6.5  27.9 %
type 1-2 :          11.5  49.1 %
type 2-2 :           5.4  23.0 %
other conts :24.0   18.41  0.06   3.89 -0.853 4.32
----------------------------------------------------------------------
turns  :     10.0    1.95  0.05   2.79  0.78  3.36
helix (no./res.):    0.06  0.67
betas (no./res.):    0.82  3.31
antis (no./res.):    0.11  0.47
======================================================================
======================================================================
length (n)    :  70
alphabet      :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice       : HEX               energy set   :   2
search depth  :   3               cutoff       : 1.9
---------------------------------------------------------------------
filename      : 70HP50.proHEX2380000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|            | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   67.0   34.2   0.25  64.28  0.27  2.91
gyration :   56.7   20.0   0.13  17.74  1.98 11.1
end to end : 26.1    8.26  0.11  12.35  0.38  3.10
----------------------------------------------------------------------
contacts :   26.0   20.24  0.06   4.05 -0.26  2.91
type 1-1 :           7.7  37.9 %
```

```
type 1-2 :          9.1  45.1 %
type 2-2 :          3.5  17.1 %
other conts :23.0   12.82  0.10  10.14 -0.02  2.86
----------------------------------------------------------------------
turns  :     11.0    5.04  0.06   3.92 -0.10  2.59
helix (no./res.):   0.57  8.05
betas (no./res.):   0.01  0.06
antis (no./res.):   0.11  0.45
======================================================================
======================================================================
length (n)   :  70
alphabet     :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice      : HEX               energy set   :   2
search depth :   4               cutoff       : 1.9
---------------------------------------------------------------------
filename     : 70HP50.proHEX2480000.zipf
sample size  : 1000
machine      : renoir
user         : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   68.85  34.3   0.25  64.59  0.12  2.62
gyration :   43.3   20.5   0.13  17.15  1.55  6.45
end to end : 21.9    9.07  0.11  12.98  0.12  2.12
----------------------------------------------------------------------
contacts :   25.0   18.34  0.07  5.60  -0.14  2.77
type 1-1 :           7.9  43.1 %
type 1-2 :           7.6  41.4 %
type 2-2 :           2.9  15.5 %
other conts :22.0   10.9   0.1   9.14   0.14  2.93
----------------------------------------------------------------------
turns  :     13.0    5.6   0.06  3.60   0.08  2.70
helix (no./res.):   0.38  4.96
betas (no./res.):   0.02  0.08
antis (no./res.):   0.17  0.71
======================================================================
======================================================================
length (n)   :  70
alphabet     :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice      : HEX               energy set   :   2
search depth :   6               cutoff       : 1.9
---------------------------------------------------------------------
filename     : 70HP50.proHEX2680000.zipf
sample size  : 1000
machine      : renoir
user         : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   63.2   37.0   0.25  63.61  0.14  2.73
gyration :   42.1   21.0   0.14  18.79  1.28  5.27
end to end :
----------------------------------------------------------------------
contacts :   26.0   18.67  0.07   5.50 -0.35  3.00
type 1-1 :          9.4    50.2 %
type 1-2 :          6.9    36.8 %
type 2-2 :          2.4    13.0 %
other conts :22.0   10.58   0.1   8.06  0.19  2.99
----------------------------------------------------------------------
turns  :     12.0    6.4    0.06  3.51 -0.10  2.81
helix (no./res.):   0.38   4.98
betas (no./res.):   0.01   0.04
antis (no./res.):   0.14   0.56
======================================================================
======================================================================
length (n)   :  70
alphabet     :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice      : HEX               energy set   :   2
search depth :   8               cutoff       : 1.9
---------------------------------------------------------------------
filename     : 70HP50.proHEX2880000.zipf
sample size  : 1000
machine      : renoir
user         : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   70.13  37.8   0.26  65.6   0.23  3.13
gyration :   56.02  20.5   0.13  16.6   1.96 11.29
end to end : 20.08   9.0   0.11  12.2   0.08  2.59
----------------------------------------------------------------------
contacts :   25.0   18.55  0.08   6.51 -0.51  3.35
type 1-1 :          9.6  51.8 %
type 1-2 :          6.5  35.3 %
type 2-2 :          2.4  12.9 %
other conts :21.0   11.06  0.08   7.11 -0.05  3.12
----------------------------------------------------------------------
turns  :     11.0    6.02  0.05   3.04 -0.31  2.86
helix (no./res.):   0.29  3.75
betas (no./res.):   0.01  0.04
antis (no./res.):   0.17  0.68
======================================================================
======================================================================
length (n)   :  64
alphabet     :  HP               bias no. 1   :  50
----------------------------------------------------------------------
lattice      :  SQ               energy set   :   2
search depth :   1               cutoff       : 1.9
---------------------------------------------------------------------
filename     : 64HP50.pro2180000.zipf
sample size  : 1000
machine      : renoir
user         : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
```

```
energies :   63.8   32.8   0.28  72.42  0.22  3.06
gyration :   33.0   13.34  0.08   6.02  2.52 13.51
end to end : 16.2    5.31  0.07   4.02  0.89  4.56
----------------------------------------------------------------------
contacts :   47.0   39.5   0.09   8.54 -0.26  2.87
type 1-1 :          13.8  34.9 %
type 1-2 :          17.5  44.3 %
type 2-2 :           8.2  20.8 %
other conts :36.0   20.4   0.14  17.51  0.18  3.25
----------------------------------------------------------------------
turns  :     22.0   11.9   0.10   9.78 -0.22  2.87
helix (no./res.):    1.87 17.12
betas (no./res.):    0.05  0.2
antis (no./res.):    0.10  0.40
======================================================================
======================================================================
length (n)    :  64
alphabet      :  HP                bias no. 1   :  50
----------------------------------------------------------------------
lattice       :  SQ                energy set   :   2
search depth  :   2                cutoff       : 1.9
---------------------------------------------------------------------
filename      : 64HP50.proSQ2280000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   68.6   36.6   0.28  77.05  0.15  2.88
gyration :   27.9   13.6   0.09   7.43  1.80  7.30
end to end : 16.6    6.3   0.08   7.16  0.56  3.20
----------------------------------------------------------------------
contacts :   47.0   39.1   0.10   9.46 -0.34  3.03
type 1-1 :          16.3  41.6 %
type 1-2 :          15.4  39.4 %
type 2-2 :           7.4  18.9 %
other conts :37.0   22.51  0.13  16.07 -0.06  3.08
----------------------------------------------------------------------
turns  :     21.0   12.46  0.09   7.42 -0.35  3.23
helix (no./res.):    1.01  8.62
betas (no./res.):    0.09  0.38
antis (no./res.):    0.18  0.74
======================================================================
======================================================================
length (n)    :  64
alphabet      :  HP                bias no. 1   :  50
----------------------------------------------------------------------
lattice       :  SQ                energy set   :   2
search depth  :   4                cutoff       : 1.9
---------------------------------------------------------------------
filename      : 64HP50.proSQ2480000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   71.6   40.6   0.27  73.92  0.143 2.92
gyration :   33.8   14.7   0.10   9.54  1.34  5.45
end to end : 17.5    7.9   0.10   9.42  0.09  2.42
----------------------------------------------------------------------
contacts :   47.0   38.2   0.11  11.30 -0.35  2.94
type 1-1 :          18.7   48.9
type 1-2 :          13.2   34.6
type 2-2 :           6.3   16.5
other conts :34.0   21.8    0.12 14.00  0.05  2.87
----------------------------------------------------------------------
turns  :     20.0   12.5    0.08  6.82 -0.32  2.81
helix (no./res.):    0.92   7.90
betas (no./res.):    0.04   0.16
antis (no./res.):    0.23   0.94
======================================================================
======================================================================
length (n)    :  64
alphabet      :  HP                bias no. 1   :  50
----------------------------------------------------------------------
lattice       :  SQ                energy set   :   2
search depth  :   6                cutoff       : 1.9
---------------------------------------------------------------------
filename      : 64HP50.proSQ2680000.zipf
sample size   : 1000
machine       : renoir
user          : eb
======================================================================
|           | max.   mean   +/-    var   skew  kurt|
======================================================================
energies :   70.0   43.1   0.28  78.35  0.06  2.82
gyration :   30.23  14.5   0.09   7.62  1.16  4.97
end to end : 16.4    7.8   0.09   8.01  0.12  2.64
----------------------------------------------------------------------
contacts :   48.0   39.0   0.11  12.20 -0.41  2.75
type 1-1 :          20.2  51.7 %
type 1-2 :          12.2  31.2 %
type 2-2 :           6.7  17.1 %
other conts :39.0   22.9   0.12  14.67 -0.04  2.90
----------------------------------------------------------------------
turns  :     20.0   12.6   0.08   5.94 -0.28  3.21
helix (no./res.):   0.75   6.28
betas (no./res.):   0.05   0.19
antis (no./res.):   0.32   1.33
======================================================================
======================================================================
length (n)    :  64
alphabet      :  HP                bias no. 1   :  50
----------------------------------------------------------------------
lattice       :  SQ                energy set   :   2
search depth  :   8                cutoff       : 1.9
---------------------------------------------------------------------
filename      : 64HP50.proSQ2880000.zipf
```

```
sample size  : 1000
machine      : renoir
user         : eb
=====================================================================
|          | max.   mean   +/-     var   skew  kurt|
=====================================================================
energies :   75.4   44.4   0.28  80.06  0.10  2.73
gyration :   34.8   14.2   0.09   7.35  1.57  8.13
end to end : 15.0    7.5   0.09   8.18  0.04  2.57
---------------------------------------------------------------------
contacts :   48.0   39.3   0.11  12.63 -0.69  3.33
type 1-1 :          20.9  53.2 %
type 1-2 :          11.6  29.6 %
type 2-2 :           6.8  17.2 %
other conts :35.0   23.1   0.12  15.11 -0.15  2.97
---------------------------------------------------------------------
turns   :     20.0   12.2   0.08   6.04 -0.31  3.43
helix (no./res.):     0.81  6.77
betas (no./res.):     0.07  0.28
antis (no./res.):     0.41  1.70
=====================================================================
=====================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       :  SC                energy set   :   2
search depth  :   1                cutoff       : 1.9
--------------------------------------------------------------------
filename      : 125HP50.proSC2180000.zipf
sample size   : 1000
machine       : renoir
user          : eb
=====================================================================
|          | max.   mean   +/-     var   skew  kurt|
=====================================================================
energies :  323.7  182.5   1.10 1217.84  0.33  3.11
gyration :   27.7   10.2   0.08    6.74  1.79  7.76
end to end : 18.1    5.9   0.07    5.53  0.59  3.60
---------------------------------------------------------------------
contacts :  143.0  113.7   0.31   95.61 -0.17  2.86
type 1-1 :          50.2  44.2 %
type 1-2 :          47.8  42.1 %
type 2-2 :          15.6  13.7 %
other conts:115.0   82.3   0.33  106.63  0.02  2.80
---------------------------------------------------------------------
turns   :     30.0   18.7   0.10   10.82 -0.17  3.88
helix (no./res.):     0.61  5.13
betas (no./res.):     1.77  7.42
antis (no./res.):     1.68  7.25
=====================================================================
=====================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       :  SC                energy set   :   2
search depth  :                    cutoff       : 1.9
--------------------------------------------------------------------
filename      : 125HP50.proSC2280000.zipf
sample size   : 1000
machine       : renoir
user          : eb
=====================================================================
|          | max.   mean   +/-     var   skew  kurt|
=====================================================================
energies :  355.3  222.7   1.14 1297.91  0.23  2.98
gyration :   14.1    7.4   0.03    0.75  2.10 11.04
end to end :  9.7    4.5   0.05    2.48  0.35  2.91
---------------------------------------------------------------------
contacts :  158.0  113.7   0.22   47.86 -0.29  3.33
type 1-1 :          60.8  45.5 %
type 1-2 :          55.3  41.4 %
type 2-2 :          17.6  13.2 %
other conts:129.0  101.7   0.25   64.23 -0.04  2.78
---------------------------------------------------------------------
turns   :     28.0   16.9   0.10    9.15 -0.07  3.46
helix (no./res.):     0.37  3.06
betas (no./res.):     1.89  8.05
antis (no./res.):     2.54 10.94
=====================================================================
=====================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       :  SC                energy set   :   2
search depth  :   4                cutoff       : 1.9
--------------------------------------------------------------------
filename      : 125HP50.proSC2480000.zipf
sample size   : 1000
machine       : renoir
user          : eb
=====================================================================
|          | max.   mean   +/-     var   skew  kurt|
=====================================================================
energies :  362.1  240.9   1.09 1196.41  0.21  3.00
gyration :    9.9    6.9   0.02    0.25  1.63  7.30
end to end :  9.7    4.4   0.05    2.03  0.27  2.88
---------------------------------------------------------------------
contacts :  156.0  140.0   0.19   34.44 -0.35  3.04
type 1-1 :          67.5  48.2 %
type 1-2 :          53.1  38.0 %
type 2-2 :          19.4  13.8 %
other conts:140.0  104.6   0.25   61.21  0.08  3.12
---------------------------------------------------------------------
turns   :     29.0   18.3   0.10    9.25 -0.17  3.62
helix (no./res.):     0.57  4.82
betas (no./res.):     2.04  8.97
antis (no./res.):     2.70 11.96
=====================================================================
=====================================================================
length (n)    : 125
```

```
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       :  SC                energy set   :   2
search depth  :   6                cutoff       : 1.9
--------------------------------------------------------------------
filename      : 125HP50.proSC2680000.zipf
sample size   : 100
machine       : renoir
user          : eb
=====================================================================
|           | max.   mean   +/-    var    skew  kurt|
=====================================================================
energies :  372.6  251.3   3.81 1448.8   0.58  2.97
gyration :    7.9    6.7   0.04    0.16  1.22  4.16
end to end :
---------------------------------------------------------------------
contacts :  155.0  141.7   0.52   27.4  -0.37  3.21
type 1-1 :          71.6  50.5 %
type 1-2 :          49.2  34.8 %
type 2-2 :          20.8  14.7 %
other conts:132.0  106.6   0.76   57.51 -0.08  3.43
---------------------------------------------------------------------
turns   :     28.0   18.4   0.33   11.12 -0.56  5.47
helix (no./res.):     0.67  5.62
betas (no./res.):     2.15  9.22
antis (no./res.):     2.44 10.76
=====================================================================
=====================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       : BCC                energy set   :   2
search depth  :   1                cutoff       : 1.9
--------------------------------------------------------------------
filename      : 125HP50.proBCC2180000.zipf
sample size   : 1000
machine       : renoir
user          : eb
=====================================================================
|           | max.   mean   +/-    var    skew  kurt|
=====================================================================
energies :  348.2  211.2   1.29 1657.03  0.29  2.94
gyration :   20.4    8.1   0.06    4.02  1.85  7.65
end to end : 12.6    5.2   0.07    4.53  0.57  3.00
---------------------------------------------------------------------
contacts :  225.0  184.3   0.45  201.4  -0.29  2.79
type 1-1 :          77.7  42.2 %
type 1-2 :          76.2  41.4 %
type 2-2 :          30.3  16.4 %
other conts:173.0  115.2   0.43  183.7  -0.17  3.22
---------------------------------------------------------------------
turns   :     46.0   30.1   0.15   21.5  -0.12  3.67
helix (no./res.):     4.43 31.57
betas (no./res.):     3.22 14.57
antis (no./res.):     3.83 16.83
=====================================================================
=====================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       : BCC                energy set   :   2
search depth  :   2                cutoff       : 1.9
--------------------------------------------------------------------
filename      : 125HP50.proBCC2280000.zipf
sample size   : 1000
machine       : renoir
user          : eb
=====================================================================
|           | max.   mean   +/-    var    skew  kurt|
=====================================================================
energies :  401.0  253.6   1.24 1543.0   0.17  3.08
gyration :   11.2    6.0   0.02    0.36  2.46 14.94
---------------------------------------------------------------------
contacts :  235.0  210.3   0.28   81.0  -0.53  3.60
type 1-1 :          93.9  44.6 %
type 1-2 :          84.8  40.3 %
type 2-2 :          31.6  15.0 %
other conts:186.0  136.4   0.31   93.4   0.05  3.47
---------------------------------------------------------------------
turns   :     39.0   27.6   0.12   15.16 -0.41  4.27
helix (no./res.):     2.40 16.05
betas (no./res.):     5.57 25.84
antis (no./res.):     5.96 27.24
=====================================================================
=====================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
---------------------------------------------------------------------
lattice       : BCC                energy set   :   2
search depth  :   4                cutoff       : 1.9
--------------------------------------------------------------------
filename      : HP50.pro280000.zipf
sample size   : 1000
machine       : renoir
user          : eb
=====================================================================
|           | max.   mean   +/-    var    skew  kurt|
=====================================================================
energies :  403.6  271.5   1.20 1437.6   0.17  2.95
gyration :    7.6    5.7   0.01    0.13  1.34  6.12
end to end :  7.5    4.0   0.04    1.50  0.23  2.89
---------------------------------------------------------------------
contacts :  237.0  216.8   0.23   54.0  -0.29  3.05
type 1-1 :         102.0  47.1 %
type 1-2 :          81.8  37.7 %
type 2-2 :          33.0  15.2 %
other conts:196.0  141.0   0.27   72.74  0.17  4.34
---------------------------------------------------------------------
turns   :     39.0   26.1   0.13   16.17 -0.22  3.70
helix (no./res.):     2.4  16.02
```

```
betas (no./res.):    6.39 29.78
antis (no./res.):    6.17 28.40
======================================================================
======================================================================
length (n)    : 125
alphabet      :  HP                bias no. 1   :  50
----------------------------------------------------------------------
lattice       : BCC                energy set   :   2
search depth  :   6                cutoff       : 1.9
---------------------------------------------------------------------
filename      : 125HP50.proBCC2680000.zipf
sample size   : 21
machine       : renoir
user          : eb
======================================================================
|           | max.   mean   +/-    var    skew  kurt|
======================================================================
energies :  339.8  275.1   7.78 1271.8   0.19  2.17
gyration :    6.7    5.8   0.08    0.13  0.56  3.01
end to end :  6.8    3.9   0.31    2.00  0.40  2.21
----------------------------------------------------------------------
contacts :  234.0  214.4   1.50   47.1   0.97  3.66
type 1-1 :         102.5  47.9 %
type 1-2 :          76.3  35.6 %
type 2-2 :          35.3  16.5 %
other conts:211.0  145.5   3.44  248.3   3.14 13.50
----------------------------------------------------------------------
turns  :     34.0   25.1   1.47   45.65 -1.71  6.97
helix (no./res.):    3.1  20.33
betas (no./res.):    5.48 25.14
antis (no./res.):    4.67 21.14
======================================================================
```

C Manual

C.1 Introduction

The toolkit is designed to compute and compare structures of Lattice Polymers (LPs). The package works independently from the chosen lattice and can be easily extended. Various methods to characterize and compare structures as well as to detect secondary strucure elements in the sense of Dill's LP analogy to natural proteins are introduced [54].
It consists of the following modules:

1. lattices : The basic data structures for working on the lattices, defining and representing structures.
2. SAWs : generating, comparing and pivoting SAWs in rel.move - notation (for a detailed description see [86]).
3. structures : Handling, comparison and interconversion of Structure representations, definition of structure distances.
4. grafic - utilities: several modules to generate files for displaying structures and matrix representations with other programs in `.pdb` or `.ps` format.
5. folding : The folding problem can not be tackeled by the use of a dynamic algorithm to find the ground state and has been proved to be NP-complete very recently [109, 33]. Efficient algorithms to fold sequences into compact structures, however not the ground state, are provided. A greedy algortihm, following a kinetic folding rule works in On where n is the size of the problem, i.e. the length of the polymer.
   Resulting structures might correspond to the molten globe state of "real proteins" in the sense of LPs. The system is designed to serve as a model for testing new methods for optimization, folding algorithms and structure prediction based on distance measures and comparison.
6. oldies but goldies: some general utilities and `.h` files.

C.1.1 Disclaimer

No responsibility for nothing. In doubt user errors are at fault.

C.2 Working on Lattices

C.2.1 Lattice Definitions and Handling of SAWs

header **lattice_types.h**

Contains the definition of the structure to be used for lattice - independent SAWs (examples for SQ in parenthesis):

```
struct Lattice {
 int     dim;      /* dimension of lattice                  (2) */
 int     nn;       /* the maximum number of neighbours      (4) */
 int     nm;       /* the maximum number of possible moves  (4) */
 char   *names;    /* move - names of absolute moves     (BFLR) */
 char   *relnames; /* move - names of relative moves      (FLR) */
 int   **Move;     /* possible moves in lattice coord.          */
 int   **Neighbor; /* possible neighbours, coordinates          */
 int   **Table;    /* transformation table rel./abs. moves      */
 float **lv;       /* Lattice Vectors                           */
 int     turns;    /* minimum steps to nearest neighbour    (3) */
 int     steps;    /* step                                  (1) */
 float   wlen;     /* average wlen, before self-traping  (71.7) */
};
```

module **lattice_types.c**

Global Definitions:

- `>> char Lattice_type[6]`

To choose one out of the predefined lattices, assign the name of desired lattice to this variable. Possible choices must coincide with one of the possibilities as listed in `struct Lattice` .

- `>> struct Lattice L`

The global structure, carrying all necessary variables during computation. the proper values must be assigned by using the procedure `define_lattices (...)` .

- `PRIVATE struct Lattice TRI, HEX, SQ, SC, BCC, TET, FCC, KM, TDKM`

The structures of type `struct Lattice` that can hold the explicit values for all variables for the available lattices TRIgonal, HEXagonal, SQare, Simple Cubic, Body Centred Cubic, TETraedic, Face Centred Cubic, Knight Moves and Three Dimensional Knight Moves respectively. This is the only data-type to be extended for incorporating new lattices into the package. Some (larger) arrays with fixed size and static definitions such as move-sets and Neighbours have to be introduced in the header of the file.

Function Calls:

- `>> void define_lattices(char *Lattice_type)`

Has to be set for initialising the proper parameter set before using any other procedures of the package. First all non-static variables for the chosen lattice are assigned, then the contents of the structure are copied to the global `struct Lattice L` which is then the only lattice structure to be used from there. This is the only procedure to be altered for incorporating new lattices into the package, valid choices are those specified at `struct Lattice` .

- `>> void free_lattices()`
Frees the space in memory after the usage of the package.

module **putils.c**:

Variables:

- `pindx`
A global array, containing an index-field to address matrices, representing structures as 5.2.

Function Calls:

- `>> void init_indx(int length)`
Initialises the index-field `int* pindx` for usage in Distance and Contact Matrix for a structure of length `len`. The exact addressing works as follows: its a 2 dimensional matrix index normally you would address a 2 dim matrix XX[1][2] with pindx: XX[pindx[1]+2].

- `>> char *Contact_Matrix (char *structure)`
Returns a char containing the Contact Matrix, using `pindx`. Each residue with a contact, according to the proper criterion, is denoted with a `1`, each non-contact residue with a `0`.

- `>> int CM_counts (char *CM)`
Returns the number of contacts from the Contact Matrix `CM`.

- `>> float *Distance_Matrix (char *structure)`
Expects the structure according to the current lattice settings in relative moves notation and returns an array containing the triangular matrix with all pair distances. Values can be accessed as usual with the `pindx` index field .

- `>> float gyr_radius(float *DM,int len)`
Returns the radius of gyration from a Distance Matrix `DM` that has been generated from a structure with length `len`.

module **SAW_utils.c**:

Function Calls:

- `>> int isSAW(char *structure)`
Expects a structure in rel.moves notation and returns 0 if structure is no SAW according to the specified lattice, a 1 otherwise.

- `>> int isSAW_last(char *structure, int **x, int i)`
Same as isSAW(), but checks correctness of the last move only. x must already contain the coordinates in positive integers.

- `>> void pivotSAW(char* structure)`
Applies a random pivot move to structure, returns only SAW - structures. At least one pivot move can always be found due to the ergodicity.

- `>> void pivot(char* structure)`

makes a random pivot move to structure, without respect if the structure is a SAW afterwards.

- `>> char *trap_SAW(int len,int maxlen, double *end_dist)`

Returns a SAW that continuous to grow until it was trapped or `maxlen` reached. `len` is required to initialise fields and should be of higher value then maxlen. This function also calculates the end to end distance and returns it as a pointer(`end_dist`).

- `>> int isSAW_last_old(char *structure)`

Same as is `SAW_last()`, requires no x values however

- `>> int isSAW_avl(char *structure)`

Same as isSAW() but with usage of AVL tree.

- `>> int isSAW_last_avl(char *structure, int **x, int pos, int last)`

Same as `isSAW_last()` but with usage of an AVL tree package.

module **avlaccess.c**

Specific functions for the use of avl-trees within our SAW package.

header **SAW_avl.h**

Definition of parameters.

C.2.2 Handling and Comparison of Structures

Structures, as generated by SAWs or folding algorithms are strings of length `strlen(sequence)-1` in the notion of relational moves. They can be converted into Contact Matrices `CM`, Bond Matrices `BM` and Distance Matrices `DM`. `BM`s and `CM`s can be converted into Contact Lists `CL`s and `BM`s into trees. 2 matrices, representing 2 (different) structures in the same notion can be compared and distances between them measured.
Further measures with respect to the nature of neighbouring residues will be provided.
For the comparison of `CL`s we need a table, showing the edit costs for each edit operation. Given a set of edit operations and edit costs, the edit distance is given by the minimum sum of costs along an edit path converting one structure into the other.

header **CL_cost.h**:
Provides the edit table for the comparison of two `CL`s.

module **SecStr.c**
Here structures can be analysed to find number and size of secondary structure elements as defined in 5.2. Values are stored in a structure `SecStr` and can be retrieved from there.

• `>> SecStr *parse_CM(char *ContactMatrix)`
Takes a Contact Matrix `CM` and converts it from a `0/1` notion for no contacts /contacts into a `H/B/P/T/C/O` (for corresponding residues being member of a Helix, antiparallelBetasheet, Parallel-betasheet, Turn, otherContact, n0contact respectively) notion.

header **SecStr.h**

```
typedef struct  blah{
  int  nt;        /* number of turns                              */
  int  npb;       /* number of parallel beta sheets               */
  int  nab;       /* number of antiparallel beta sheets           */
  int  nh;        /* number of helices                            */
  int  others;    /* number of all other contacts                 */
  int  *turns;    /* list of turns by first base                  */
  int  *pbeta_i;  /* begin1 of parallel beta sheets               */
  int  *pbeta_j;  /* begin2 of parallel beta sheets               */
  int  *pbeta_l;  /* length of parallel beta sheets               */
  int  *abeta_i;  /* begin1 of anti-parallel beta sheets          */
  int  *abeta_j;  /* begin2 of anti-parallel beta sheets          */
  int  *abeta_l;  /* length of anti-parallel beta sheets          */
  int  *helix_b;  /* begin of helix                               */
  int  *helix_e;  /* end of helix                                 */
  int  *helix_m;  /* winding height of helix                      */
}SecStr;
```

header **seqstat.h**
For each of residue, the sequence is composed of, data about the number and percentage of contacts to other types of residues are stored in this array.

```
typedef struct  pfui{
  int *count_al;        /* count of residue in total structure             */
  int *count_con_al;    /* count of contacts of residue in total structure  */
  float *weight_con_al; /* weight of contacts of residue in total structure */
  int **cont_al;        /* array of number of contacts in between alphabet   */
}surface;
```

module **seqstat.c**

• `>> surface stat_seq(char *CM,char * sequence,char *alphabet);`
Calculates statistics of neighbourhood relations and contacts of a given structure with a defined sequence where the alphabet is denoted.

module **LP_struct_dist.c**

Provides various methods to compare, convert structures and measure distances between. A simple measure can be obtained by using `hamming(char *ST1, char *ST2)` from the module `utils`, measuring the hamming distance in the directions of bonds where `ST1` and `ST2` are simply structures in the notion of relative directions.

• `>> int ident_contacts (char *CM1, char *CM2)`
Counts the number of identical contacts (= nearest neighbours) from the two Contact Matrices `CM1` and `CM2` that must have been generated from two structures of same length before.

• `>> void initialize_edit_costs(void)`
Initialises edit costs for structure alignment. Has to be set before use of any functions for structure distances.

• `>> float MV_to_SM(char *, char *)`
Generates a score matrix by exchange and gap penalty costs of relative moves as listed in `move_edit` .

• `>> float DM_dist(float *DM1, float *DM2, int para, int len)`
Compares two Distances - Matrices `DM1` and `DM2` of two structures with length `len` and returns a float, which is the RMS of all entries. The `pindx` field must be already initialised.

• `>> int **CM_to_CL(char *CM, int length)`
Transfers a Contact Matrix `CM` into a two dimensional array, the Contact List `CL` which is returned, each residue a list of contacts is assigned to. It needs the Contact Matrix `char *CM` as input, and returns a matrix, which has to be freed afterwards. The `pindx` field must be already initialised.

• `>> float **CL_to_SM(int **CLA, int **CLB, int bias)`
Generates a 2D scoring matrix, containing scores, evaluated from pairwise comparison of contact lists `CLA` and `CLB` following an editing procedure as denoted in the table `cedit` in `edit.h`.

• `>> float **DM_to_SM(float * DMA, float * DMB, int len)`
This Function compares two Distance - Matrices `DM1` and `DM2` with the length `len` of the corresponding structures and returns a score matrix `SM` . The `pindx` field must be already initialised.

• `>> char *DM_to_TREE(float *DM, int bias)`
Generates a rooted tree, encoded as a string, from the Distance Matrix `DM` following a standard clustering procedure.

• `>> float *TREE_dist(char *TR_A, char *TR_B)`
Returns the tree distance between the rooted trees `TR_A` and `TR_B` following a tree editing procedure.

the edit table `int tedit[][]` must be available in `edit.h`.

- `>> float SM_score(float **SM, int winsize, int **track)`
Returns the maximum score tracking through the Score Matrix `SM`, following a standard structure alignment procedure as described in [107]. `winsize` defines the window size and is therefore crucial for the time consumption. If `track[structure length][2]` is initialised, it can be used to retrieve the track and reconstruct the way the structures have been aligned, otherwise input `NULL`.

- `>> void align(char *A, char *B, char *AA, char *AA, int **track)`
Returns aligned sequences `A` and `B` when `track` has been filled up with matching entries already. Caution: do not allocate `AA,BB` as they might well be longer upon return.

- `>> float CM_dist(char *, char *, char *, char *, char *i, char *)`
Short form to evaluate distance between contact matrices.

C.3 Graphic - Utilities

module **graf_utils.c** and **graf_utils.h**

Provides several methods to illustrate structures in postscript or `pdb` format.

- `>> void SAW_PS(char *sequence, char *structure, char *fname, char *ALPHAbet);`

Generates a file named `fname` in postscript format, depicting the structure with balls representing each residue in the node point and a stick for each bond. Lattice and Alphabet must have been specified before usage.

- `>> void pdb_out(char *struc, char *seq, char * CM,char * pdb_fn, char *para_fn)`

Generates a file `pdb_fn` that can be used to illustrate the structure. CM has to be parsed already. A different colour is used for each different secondary structure element.

- `>> void seq_pdb_out(char *struc, char *seq, char *scont, char *pdb_fn,char *para_fn)`

Same as `pdb_out()`, but uses different colours for different residues in the sequence.

- `>> void CM_dot(char *CM, char *string, char *ps_fn)`

Needs a Contact Matrix `CM` that has been transformed into a `H/B/A/T/C/O` notion by the use of `parse_CM` and produces a file in postscript format that shows the structure in a coloured dot-plot matrix where each filled square denotes a neighbour as well as a legend, illustrating the symbols for secondary structure elements.

- `>> void CM_DM_dot(char *CM, char *DM, char *string, char *ps_fn)`

Similar to `LP_CM_dot()`, but shows both, the `CM` and `DM`, where the size of the squares is related the distance of the two corresponding residues. `CM` must be parsed already.

C.4 Folding Algorithms

We provide various algorithms to yield a structure of `LP`s. The "kinetic-ally controlled" stepwise procedure follows energy criteria as denoted in `initialise_energies()` Other algorithms are basicly used for comparison.

module **SAWs**:

Generates SAWs either with desired length or without a certain criterion, till the walk gets trapped by it's own.

module **ExHaust**:

Generates all SAWs until a certain length, and provides you with an output file, containing the (number of contacts, structure, end to end distance, radius of gyration) of specific walks fulfilling a certain criteria.

module **SFold**:

Folds a sequence, composed of a certain alphabet according to a given energy set and following the rules of a "greedy algorithm" that attaches one residue after the other after having calculated to optimal position on the lattice. In case it gets trapped, it successfully steps back till an alternative way can be found.
Energy parameters are available as described by Dill [13].
Global *Parameters* to be set before usage

```
int    depth;          /* for optimisation at current strings end   */
int    gyropt;         /* optimise according to current gyration radius      */
int    valences;       /* maximum number of considered bindings per residue   */
int    rand_choice;    /* choose randomly out of degenerate solutions         */
char  *movenames;      /* internal usage of move names                         */
```

Function Calls:

- `>> double pfold (char *q, char *ts);`

Folds a string `q` stepwise according to a greedy folding algorithm. Parameters must already be set and space for the resulting structure `ts` reserved.

- `>> void initialize_pfold ( int length )`

- `>> double free_pfold ( int length );`

- `>> double resort_moves ( void );`

- `>> double clear_nodes ( int length );`

Erases all entries from all nodes in case data structures are to used for other purpose.

module **energy_sets.c** and **energy_sets.h**

- `>> double initialize_energies ( void );`

Initialises energy settings to calculate pairwise energy contributions for a given alphabet.
Initialises energy sets and alphabets.

- `>> double DM_to_energy ( float *DM, char *sequence, int length );`

Evaluates the energy of a given sequence and a Distance Matrix DM according to predefined energy sets.

Global *Parameters* to be set before usage:

```
float  cutoff_distance; /* no contributions for contact distances byond this */
int    e_set;           /* energy set, implies alphabet and bond matrix      */
```

C.5 General Utilities

module **utilities.c**:

Some useful procedures for generating and handling strings:

- `>> void initialize_from_time(int A,int B)`

Initializes the random seeds of `urn[3]`, sleeps time in between depending on size of `A` and `B`.

- `>> char *random_string_bias(int length, char *symbols, int *biases)`

Generates a random string with a biased contents of letters denoted in `symbols` , given in percentage in the integer array `biases` where each value corresponds to the letter in the symbols - alphabet, the last may be chosen with prob. sum up to 1.

- `>> short error(char *procedure, char *message)`

Prints an error message to stdout.

- `>> int getkey()`

Scans for input.

- `>> double rd_double(FILE *fp)`

Reads a double from a file `fp`.

- `>> float rd_float(FILE *fp)`

- `>> int rd_int(FILE *fp)`

- `>> short rd_string(FILE *fp, char val[])`

- `>> short rd_short(FILE *fp)`

- `>> int digits(int status)`

Returns the number of digits from `status`.

- `>> float bico(int n, int k)`

Calculates the binomial coefficient of `n` over `k`.

- `>> void goahead()`

- `>> char *manual_input(int length, int maxlength)`

- `>> void shuffle(int *list, int n)`

Takes an array of integer (list) and it's length (n) and randomly reassigns all values to the possible positions without changing them.

- `>> void dump_int_data(int *val,FILE *fp,FILE *log_fp,long int l,int scale,int label)`

Prints the contents of the array `val` of integer values in 2 columns to a file `fp` and statistical moments to a file `log_fp`. Requires `l`, i.e. the size the array `val` has been allocated and therefore the expected maximum value. If `scale` $\neq 0$, all values are multiplied with `pow(10,scaling)` at output, so that floats can also be handled within a certain range of accuracy. The files, addressed with `fp` and `log_fp` must have been opened before usage. (`label < 0`: label is printed in first column to log file, `> 0`: same but no text is printed to log file, `= 0` no label is printed.)

• `>> void dump_subis(unsigned short mark_subi[3])`
Dumps contents of the xsubi array of "random numbers" to a file (called "fn.sub").

module **utils.c**:
Same as in the `Vienna RNA Package`.

Global *Definitions*:

• `>> double urn(void)`
Is used by `urn()` These should be set to random number seeds before the first call to `urn()`.
Global *Function Calls*:

• `>> void *space(unsigned size)`
Returns a pseudo random number in the range [0..1[, using a 48bit linear congruental method.

• `>> void nrerror(char *message)`
Writes `message` to stderr and aborts the program, exits with -1

• `>> char *time_stamp(void)`
Returns a string containing the current data in the format "Fri Nov 13 13:13:13 1994".

• `>> char *random_string(int l, char *symbols)`
Generates a"random" string of characters from `symbols` with length `l`.

• `>> int hamming(char *S1, char *S2)`
Returns the number of positions in which `S1` and `S2` differ, the so called "Hamming" distance.

• `>> char *get_line(FILE *fp)`
Reads lines of arbitrary length from `fp`.

header std.h:
Provides several, frequently used definitions for the pre-compiler.

• `PRIVATE` static calls.

• `PUBLIC` non-static calls.

• `ABS(A)` positive value of any data type.

• `FREE(S)` frees allocated space and sets pointer to NULL.

• `MAX(A,B)` returns the higher value of A or B.

C.6 Executables

(under construction)

- `LPgraf`
Expects a structure as input and produces postscript files depicting the structure, the contact matrix, the distance matrix and/or a pdb outputfile.

- `LPstruct`
Expects a structure as input and produces statistics on surface, structural elements and moments in a file.

- `fold`
Folds a single sequence according to chosen parameters with the CG-algorithm
The following files will be generated:

fn.fold contains energy status, the current move at each position

fn.log contains a summary of all parameters

fn.struct contains all data on residues in a structural element, type of contacts, etc. etc.

The options upon input are:

-n [30] length

-i input filename

-o output filename (default is: nAb.proemvxrga.m)

-m modus

1 [test] standalone

2 compare two structures

3 read a sequence file for input

4 generate a sample of random structures

5 landscape

9 evaluate the CPU requirements

-z sample size

-w [0], 1:write structures to a `fn.struct` file

-c [0], 1:continuation run

-f folding type

-l lattice type

SQ or SC, HEX, TRI, KM, SC, BCC, FCC, 3DKM

-d [2] search depth

-a [1.9] cutoff distance

-g [0], 1: optimise the radius of gyration

-v [8]: number of valences per residue, used to evaluate the energy

-x [0]: no random choice for the next move

-r [0]: don't resort the moves

-y resolution of structure representation

-e [2]: energy set

-b [1]: biases for generating the sequences

-h display a help mesg.

- `PRO-ANAL`
to analyse `.struct` files in format : `sequence length` `structure energy`
options:

-i input-filename

-l lattice type

-a cutoff

-e energy-set

-n length

- `STAT`
Takes a list of entries, each of same type and generates a file `.stat` with the statistic moments and one `.xy` with the values, sorted according to their frequency and the corresponding frequency as well as the normalised frequency.

-i input filename

-o output filename

-f scaling factor, to multiply noninterger values with

D Specification of Lattices

D.1 Definitions of SAWs

A Self-avoiding walks (SAW) [60] is a path on a lattice that that does not visit the same site more that once.

SAWs can be generated by a random walk on the lattice subject to the constraint that already occupied sites are inaccessible. Such a walk will get trapped whenever there are no neighboring unoccupied sites available. In fact, almost all SAWs constructed by this procedure terminate after a finite number N of steps. We will call these SAWs *grown self-avoiding walks* (gSAWs).

D.2 Representation of SAWs

We consider here regular [1] lattices together with a set of moves at each lattice point with the following properties: Let $x_{i+1} = x_i + m$ denote the lattice point obtained by attaching the move m to site x, then there is a move m^* such that $x_i = x_{i+1} + m^*$, i.e., $m^* = -m$ is a move as well. Furthermore we require that for all lattice points x, y and all moves m, m' there is a symmetry operation ψ of lattice such that $y = \psi(x)$ and $y + m' = \psi(x + m)$.

A walk on a lattice is completely described by its initial point, 0, and the ordered list of the N moves. Let us denote by $\mathcal{M} = \{m_1, \ldots, m_M\}$ the set of all possible moves in absolute coordinates. Not all of these M moves can be realized at each lattice point. As an example consider the hexagonal lattice, `HEX`. There are 3 possible moves at each lattice site, out of a total of 6 absolute directions. We define the set $\mathcal{R} = \{r_1, \ldots, r_m\}$ of "relative" directions as the subset of $\mathcal{M}$ consisting of all moves allowed at the reference site 0.

The representation of a walk in terms of moves in absolute coordinates is not always convenient. Instead we adopt a local point of view by rotating the coordinate frame after each move r_k in such a way that $-r_k$ is assigned the backwards direction B in the rotated coordinate system. In other words, for each relative move r there is a rotation $\tilde{r}$ of the coordinate system fulfilling $\tilde{R}(R) = -B$.

As a consequence of the symmetry requirements above the allowed moves for the next step expressed in the rotated coordinate system are again the relative moves $\mathcal{R}$. For SAWs we only need $z = m - 1$ relative directions, since the backwards direction B never occurs.

The coordinates of the point x_k obtained as the k-th step of the walk are given by $x_k = x_{k-1} + a_k$. The absolute direction a_k of the k-th step is uniquely defined by the relative direction r_k of this step, *and* the absolute direction of the previous step, a_{k-1}. Consequently, there is a mapping

$$\tau : \mathcal{M} \times \mathcal{R} \to \mathcal{M}, \ (m, r) \mapsto \tilde{r}^{-1} m$$

for each lattice determining the absolute direction of a step given its relative direction and the absolute direction of the previous step. Once we have defined τ we can represent a walk on the lattice as a string of length N over the alphabet of the m relative directions.

As a by-product, point mutation of these strings are pivot moves [61]. Mappings for all lattices are given in the next section.

[1] We call a lattice regular if for any two lattice points x and y there is a symmetry operation σ of the lattice such that $x = \sigma(y)$ and if for any two pairs of neighbors (u, v) and (x, y) there is a symmetry operation τ of the lattice such that $u = \tau(x)$ and $v = \tau(y)$.

D.3 Transformation Tables for Relative Moves

SQ	f	r	l	b
F	f	r	l	b
R	r	b	f	l
L	l	f	b	r
B	b	l	r	f

SC	b	f	l	r	u	d
B	f	b	r	l	d	u
F	b	f	l	r	u	d
L	r	l	u	d	f	b
R	l	r	d	u	b	f
U	d	u	f	b	l	r
D	u	d	b	f	r	l

HEX	f	r	l	d	u	b
F	f	r	l	d	u	b
R	r	d	f	b	l	u
L	l	f	u	r	b	d
D	d	b	r	u	f	l
U	u	l	b	f	d	r
B	b	u	d	l	r	f

BCC	b	f	p	x	q	y	r	z
B	f	b	x	p	y	q	z	r
F	b	f	p	x	q	y	r	z
P	x	p	q	y	z	r	f	b
X	p	x	y	q	r	z	b	f
Q	y	q	z	r	f	b	x	p
Y	q	y	r	z	b	f	p	x
R	z	r	f	b	x	p	y	q
Z	r	z	b	f	p	x	q	y

FCC	b	f	r	l	w	p	x	q	y	m	z	n
B	f	b	l	r	p	w	q	x	m	y	n	z
F	b	f	r	l	w	p	x	q	y	m	z	n
R	l	r	b	f	z	n	y	m	p	w	q	x
L	r	l	f	b	n	z	m	y	w	p	x	q
W	p	w	x	q	z	n	m	y	l	r	f	b
P	w	p	q	x	n	z	y	m	r	l	b	f
X	q	x	p	w	f	b	l	r	n	z	y	m
Q	x	q	w	p	b	f	r	l	z	n	m	y
Y	m	y	n	z	r	l	f	b	x	q	p	w
M	y	m	z	n	l	r	b	f	q	x	w	p
Z	n	z	y	m	q	x	w	p	f	b	r	l
N	z	n	m	y	x	q	p	w	b	f	l	r

TET	b	f	l	r	m	q	k	s
B	f	b	r	l	q	m	s	k
F	b	f	l	r	m	q	k	s
L	r	l	b	f	k	s	q	m
R	l	r	f	b	s	k	m	q
M	q	m	k	s	l	r	b	f
Q	m	q	s	k	r	l	f	b
K	s	k	m	q	f	b	l	r
S	k	s	q	m	b	f	r	l

3DKM	b	f	r	l	s	u	t	v	w	m	x	e	y	a	z	c	h	p	i	m	j	n	g	o
B	f	b	l	r	u	s	v	t	d	w	e	x	a	y	c	z	p	h	m	i	n	j	o	g
F	b	f	r	l	s	u	t	v	w	d	x	e	y	a	z	c	h	p	i	m	j	n	g	o
R	l	r	b	f	v	t	u	s	z	c	y	a	x	e	w	d	i	m	j	n	g	o	h	p
L	r	l	f	b	t	v	s	u	c	z	a	y	e	x	d	w	m	i	n	j	o	g	p	h
S	u	s	t	v	l	r	b	f	x	e	w	d	z	c	y	a	g	o	h	p	m	i	j	n
U	s	u	v	t	r	l	f	b	e	x	d	w	c	z	a	y	o	g	p	h	i	m	n	j
T	v	t	u	s	b	f	r	l	y	a	z	c	w	d	x	e	j	n	g	o	h	p	m	i
V	t	v	s	u	f	b	l	r	a	y	c	z	d	w	e	x	n	j	o	g	p	h	i	m
W	d	w	e	x	a	y	c	z	p	h	m	i	n	j	o	g	f	b	l	r	u	s	v	t
D	w	d	x	e	y	a	z	c	h	p	i	m	j	n	g	o	b	f	r	l	s	u	t	v
X	c	e	y	a	x	e	w	d	i	m	j	n	g	o	h	p	l	r	b	f	v	t	u	s
E	e	c	a	y	e	x	d	w	m	i	n	j	o	g	p	h	r	l	f	b	t	v	s	u
Y	x	e	w	d	z	c	y	a	g	o	h	p	m	i	j	n	u	s	t	v	l	r	b	f
A	e	x	d	w	c	z	a	y	o	g	p	h	i	m	n	j	s	u	v	t	r	l	f	b
Z	y	a	z	c	w	d	x	e	j	n	g	o	h	p	m	i	v	t	u	s	b	f	r	l
C	a	y	c	z	d	w	e	x	n	j	o	g	p	h	i	m	t	v	s	u	f	b	l	r
H	p	h	m	i	n	j	o	g	f	b	l	r	u	s	v	t	d	w	e	x	a	y	c	z
P	h	p	i	m	j	n	g	o	b	f	r	l	s	u	t	v	w	d	x	e	y	a	z	c
I	i	m	j	n	g	o	h	p	l	r	b	f	v	t	u	s	z	c	y	a	x	e	w	d
M	m	i	n	j	o	g	p	h	r	l	f	b	t	v	s	u	c	z	a	y	e	x	d	w
J	g	o	h	p	m	i	j	n	u	s	t	v	l	r	b	f	x	e	w	d	z	c	y	a
N	o	g	p	h	i	m	n	j	s	u	v	t	r	l	f	b	e	x	d	w	c	z	a	y
G	j	n	g	o	h	p	m	i	v	t	u	s	b	f	r	l	y	a	z	c	w	d	x	e
O	n	j	o	g	p	h	i	m	t	v	s	u	f	b	l	r	a	y	c	z	d	w	e	x

E Distance Measures and Edit Costs

E.1 Dynamic Alignment

Dynamic Alignment methods have been applied to several problems [71, 118, 111, 73].
The aim is, to search an optimal matching of two strings such that the pairings follow a given strategy, usually in matching as many as possible identical (or very similar) elements such that the sequence of the strings has not to be altered. It is based on the assumption of suboptimality and, in the case of sequence alignment, on the introduction of gap-penalties when elemnts in two strings that are not in the same position with respect to the origin are matched. This means that crossings and non-sequential overlaps are forbidden. Overall alignment for the global optimum is NP-complete [69] and extremly time consuming therefore.
The principle is explained in the following paragraph: First one generates a score matrix of size $n \times m$, where the entries S_{ij} are calculated by a well defined procedure based on defined costs from assigning all elements of one string of length n to all of the other with length m. (In fact, these data need not be produced before, but can be evaluated during the process of finding the minimum path starting at the end and recursively evaluating entries. If one has a symmetric problem, i.e. $n = m$ (f.i. distance matrices or strings of relative moves of two structures of equal length) the procedure can be reduced to the triangle matrix.) The basic recursion to generate the maximum overall score S_{ij} can be formulated as follows:

$$S_{ij} = D_{ij} + max \begin{cases} ..... & \\ \max_{k=i+2} & S_{k,j+1} + G_{k,0} \\ & S_{i+1,j+1} \\ \max_{l=j+2} & S_{i+1,l} + G_{0,l} \\ .... & \\ \max_{l=j+w} & S_{i+w,l} + G_{w,1} \\ ..... & \end{cases} \qquad (46)$$

where D_{ij} is a measure of relatedness and G a gap-penalty. The path can be recovered after having found the optimal value in a backtracking procedure, thereby one yields the corresponding alignment that resulted in the score.
Gap penalties have to be chosen carefully [111], the number of suboptimal entries, the windowsize w that are compared to find the maximum entry for the current position is crucial for both, accuracy and time consumption.

AA.			MASS	VOL	SURF	FRQ	pol	sec.str.	α	βs	βt	Ω					
GLY	G		57.06	60.1	75	7.5	hp		0	0	0.56	0.92	1.64	1.35	0.53	0.81	0.45
ALA	A		71.08	88.6	115	9.0	h	α	2.1	1.9	1.29	0.90	0.78	0.77	1.45	0.97	0.15
VAL	V	H	99.14	140.0	155	6.9	h	Π	6.3	1.7	0.91	1.49	0.47	0.64	1.14	1.65	0.08
LEU	L	H	113.17	166.7	170	7.5	h	α	7.5	0.5	1.30	1.02	0.59	0.76	1.34	1.22	0.14
ILE	I	H	113.17	166.7	175	4.6	h	π	10.5	1.0	0.97	1.45	0.51	0.68	1.00	1.60	0.15
MET	M	H	131.21	162.9	185	1.7	hp	α	5.4	16.2	1.47	0.97	0.39	0.67	1.20	1.67	0.18
PRO	P		97.12	122.7	145	4.6	hp	β	(-13.8)		0.52	0.64	1.91	1.28	0.59	0.62	0.41
PHE	F		147.18	189.9	210	3.5	h	γ	10.5	13.2	1.07	1.32	0.58	0.90	1.12	1.28	0.19
TRP	W	H	186.21	227.8	255	1.1	hp	γ	14.3	34.6	0.99	1.14	0.75	0.85	1.14	1.19	0.30
SER	S		87.08	89.0	115	7.1	p	β	-1.2	31.2	0.82	0.95	1.33	1.20	0.79	0.72	0.41
THR	T	H	101.11	116.1	140	6.0	p	γ	1.7	30.4	0.82	1.21	1.03	1.07	0.82	1.20	0.27
CYS	C		103.14	108.5	135	2.8	p	α	(-11.7)	15.2	1.11	0.74	0.80	0.94	0.77	1.30	0.31
TYR	Y		163.18	193.6	230	3.5	p	γ	9.6	35.6	0.72	1.25	1.05	1.28	0.61	1.29	0.33
ASN	N		114.11	117.7	160	4.4	p	β	-0.8	50.5	0.90	0.76	1.28	1.22	0.73	0.65	0.45
GLN	Q		128.14	143.9	180	3.9	p	α	-1.3	49.3	1.27	0.80	0.97	0.93	1.17	1.23	0.15
ASP	D		115.09	111.1	150	5.5	+	β	(-31.0)	55.7	1.04	0.72	1.41	1.22	0.98	0.80	0.33
GLU	E		129.12	138.4	190	6.2	+	α	(-41.4)	52.6	1.44	0.75	1.00	1.09	1.53	0.26	0.12
LYS	K	H	128.18	168.6	200	7.0	-	α	(-17.6)	49.8	1.22	1.08	0.69	0.83	1.07	0.74	0.27
ARG	R		156.20	173.4	225	4.7	-	α γ	(-46.9)	93.4	0.96	0.99	0.88	0.91	0.79	0.90	0.27
HIS	H		137.15	153.2	195	2.1	-	α	2.1	52.8	1.22	1.08	0.69	0.83	1.24	0.71	0.18

Table 15: Some properties of amino acids: the three letter code; the one letter code; H indicates if the acid is essentiell for humans; the mass (daltons); volume (A^3); accessible surface (A^2); the relative frequency in %; the classification as hydrophobic (h) or hydrophilic (p), with positive or negative charge; hydrophobicity and hydrophilicity compared to glycine; the preferred secondary structural element; the relative (normed) frequencies in structural elements.

E.2 Energy Functions

```
 2 |  H  P  N  X
----------------
 H | -1  0  0  0   /* selfattracting ´´solvophobic´´ residues */
 P |  0  0 -1  0   /* positive polar residues, mutually attracting with N */
 N |  0 -1  0  0   /* negative polar residues, mutually attraction with P */
 X |  0  0  0  0   /* neutral residues */
----------------

 3 |  H  P  N  X
----------------
 H | -2  1  1  0   /* stronger selfattraction */
 P |  1  0 -1  0   /* additional repulsion to nonpolar H */
 N |  1 -1  0  0   /* additional repulsion to nonpolar H */
 X |  0  0  0  0   /* neutral as can be */
----------------

 7 |  H  P  N  X
----------------
 H | -1  0  0  0   /* strange */
 P |  0 -1  0  0   /* strange */
 N |  0  0 -1  0   /* residuous */
 X |  0  0  0  0
----------------
```

E.3 Relative Moves Distances

```
TRI | F R L U D
---------------
 F  | 0 1 1 2 2
 R  | 1 0 2 3 1
 L  | 1 2 0 1 3
 U  | 2 3 1 0 2
 D  | 2 1 3 2 0
gap = 3

SQ | F L R
----------
 F | 0 1 1
 L | 1 0 2
 R | 1 2 0
gap = 2

HEX | F R L U D
---------------
 F  |
 R  |
 L  |     0   1
 U  |
 D  |     1   0
gap = 1

SC  | F R L U D
---------------
 F  | 0 1 1 1 1
 R  | 1 0 2 1 1
 L  | 1 2 0 1 1
 U  | 1 1 1 0 2
 D  | 1 1 1 2 0
gap = 2

BCC | F | P Q R | X Y Z
-----------------------
 F  | 0 | 3 3 3 | 4 4 4
----|---|-------|------
 P  | 3 | 0 2 2 | 5 3 3
 Q  | 3 | 2 0 2 | 3 5 3
 R  | 3 | 2 2 0 | 3 3 5
----|---|-------|------
 X  | 4 | 5 3 3 | 0 2 2
 Y  | 4 | 3 5 3 | 2 0 2
 Z  | 4 | 3 3 5 | 2 2 0
 ----------------------
gap = 5
```

References

[1] D. P. Bartel and J. W. Szostak. Isolation of new ribozymes from a large pool of random sequences. *Science*, 261:1411 – 1418, 1993.

[2] A. A. Beaudry and G. F. Joyce. Directed evolution of an RNA enzyme. *Science*, 257:635 – 641, 1992.

[3] J. Bluestone, S. Sharrow, S. Epstein, K. Ozato, and D. Sachs. Induction of anti-h2 antibodies in the absence of alloantigen exposure by in vivo administration of anti-idiotype. *Nature*, 219:233, 1981.

[4] S. Bonhoeffer, J. McCaskill, P. F. Stadler, and P. Schuster. RNA multi-structure landscapes, a study based on temperature dependent partition functions. *European Biophysics Journal*, 22(1):13, 1993.

[5] S. Bromberg and K. A. Dill. Side-chain entropy and packing in proteins. *Protein Science*, 3:997 – 1009, 1994.

[6] C. J. Camacho and D. Thirumalai. Kinetics and thermodynamics of folding in model proteins. *Proc. Natl. Acad. Sci., USA*, 90:6369 – 6372, 1993.

[7] G. Casari and M. J. Sippl. Structure-derived hydrophobic potential. *J. Mol. Biol.*, 224:725 – 732, 1992.

[8] T. Cech. Ribozyme engineering. *Current Opinion in Structural Biology*, 2:605, 1992.

[9] T. R. Cech. Conserved sequences and structures of groupI introns : building an active site for RNA catalysis - a review. *Gene*, 73:259–271, 1988.

[10] M. Chamberlin and T. Ryan. *The enzymes*, volume 15, pages 87 – 108. Academic Press, New York, 1982.

[11] H. S. Chan and K. A. Dill. Intrachain in polymers: effects of excluded volume. *J. Chem. Phys.*, 90:492 – 509, 1989.

[12] H. S. Chan and K. A. Dill. The effects of internal constraints on the configurations of chain molecules. *J. Chem. Phys.*, 92:3118 – 3125, 1990.

[13] H. S. Chan and K. A. Dill. Sequence space soup of proteins and copolymers. *J. Chem. Phys.*, 95:3775 – 3787, 1991.

[14] H. S. Chan and K. A. Dill. The protein folding problem. *Physics today*, 2:24 – 32, 1993.

[15] H. S. Chan and K. A. Dill. Transition states and folding dynamics of proteins and heteropolymers. *J. Chem. Phys.*, 100:9238 – 9257, 1994.

[16] G. Chapman, J. Cleese, T. Gilliam, E. Idle, T. Jones, and M. Pallin. *The complete Monty Python's Flying Circus: all the words*, volume 1. Pantheon, 1989.

[17] G. M. Crippen. Prediction of protein folding from amino acid sequence over discrete conformation spaces. *Biochemistry*, 30:4232 – 4237, 1991.

[18] C. Darwin. *The origin of species.* reprinted by the Harvard University Press, 1966.

[19] C. W. David. X-ploring extraribosomal peptide folding during synthesis. *J. Comp. Chem.*, 15:662 – 665, 1993.

[20] B. Derrida. Random-energy model: an exactly solvable model of disordered systems. *Physical Review B*, 24:2613 – 2626, 1981.

[21] K. A. Dill, K. M. Fiebig, and H. S. Chan. Cooperativity in protein-folding kinetics. *Proc. Natl. Acad. Sci., USA*, 90:1942 – 1946, 1993.

[22] G. Dueck and P. Winkler. New concepts and algorithms for portfolio choice. *Technical Report 90.06.006, IBM*, IBM Germoney, Heidelberg, Scientific Center, June 1990.

[23] M. Eigen. Selforganization of matter and the evolution of biological macromolecules. *Die Naturwissenschaften*, 10:465, 1971.

[24] M. Eigen, J. McCaskill, and P. Schuster. Molecular quasi-species. *Journal of Physical Chemistry*, 92:6881 – 6891, 1988.

[25] M. Eisenstein and R. Sharon. *Biochemie*, 73:879, 1991.

[26] A. D. Ellington and J. W. Szostak. In vitro selection of RNA molecules that bind specific ligands. *Nature*, 346:818 – 822, 1990.

[27] W. Fontana. *Ein Computermodell der Evolutionären Optimierung.* PhD thesis, University of Vienna, 1988. PhD Thesis.

[28] W. Fontana, T. Griesmacher, W. Schnabl, P. Stadler, and P. Schuster. Statistics of landscapes based on free energies, replication and degredation rate constants of RNA secondary structures. *Monatshefte der Chemie*, 122:795–819, 1991.

[29] W. Fontana, D. A. M. Konings, P. F. Stadler, and P. Schuster. Statistics of RNA secondary structures. *Biopolymers*, 33:1389–1404, 1993.

[30] W. Fontana, W. Schnabl, and P. Schuster. Physical aspects of evolutionary optimization and adaptation. *Physical Review A*, 40:3301 – 3321, 1989.

[31] W. Fontana and P. Schuster. A computer model of evolutionary optimization. *Biophysical Chemistry*, 26:123–147, 1987.

[32] W. Fontana, P. F. Stadler, E. G. Bornberg-Bauer, T. Griesmacher, I. L. Hofacker, M. Tacker, P. Tarazona, E. D. Weinberger, and P. Schuster. RNA folding and combinatory landscapes. *Phys. Rev. E*, 47(3):2083 – 2099, 1993.

[33] A. S. Fraenkel. Complexity of protein folding. *Bull. Math. Biol.*, 55:1199 – 1210, 1993.

[34] H. Frauenfelder, S. G. Sligar, and P. G. Wolynes. The energy landscape and motions of proteins. *Science*, 254:1598 – 1603, 1991.

[35] S. M. Freier, R. Kierzek, J. A. Jaeger, N. Sugimoto, M. H. Caruthers, T. Neilson, and D. H. Turner. Improved free-energy parameters for predictions of RNA duplex stability. *Proc. Natl. Acad. Sci., USA*, 83:9373–9377, 1986.

[36] R. Gesteland and J. Atkins. *The RNA World*. Cold Spring Harbour Univ. Press, 1993.

[37] M.-J. Gething and J. Sambrook. Protein folding in the cell. *Nature*, 355:33 – 45, 1992.

[38] N. Go, T. Noguti, and T. Nishikawa. Dynamics of a small globular protein in terms of low- frequency vibrational modes. *Proc. Natl. Acad. Sci., USA*, 80:3696 – 3700, 1983.

[39] A. Godzik, A. Kolinski, and J. Skolick. Lattice representations of globular proteins: How good are they? *J. Comp. Chemistry*, 14:1194 – 1202, 1993.

[40] Z. Guo, D. Thirumalai, and J. Honeycutt. Folding kinetics of proteins: a model study. *J. Chem. Phys.*, 97:525 – 535, 1992.

[41] R. Gutell and C. Woese. Higher order structural elements in ribosomal rRNAs: Pseudoknots and the use of non-canonical pairs. *Proc. Natl. Acad. Sci., USA*, 87:663, 1990.

[42] R. R. Gutell. Comparative studies of RNA: inferring higher-order structure from patterns of sequence variation. *Current Opinion in Structural Biology*, 3:313 – 322, 1993.

[43] I. L. Hofacker, W. Fontana, P. Schuster, and P. F. Stadler. Fast folding and comparison of RNA secondary structures (the Vienna RNA Package). *Monatshefte f. Chemie*, 125:167–188, 1994.

[44] J. A. Jaeger, D. H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. *Proc.Natl.Acad.Sci., USA*, 86:7706–7710, 1989.

[45] G. F. Joyce. Gelenkte evolution von biomolekülen. *Spektrum der Wissenschaft*, pages 52 – 59, 1993.

[46] W. Kabsch and C. Sander. *Biopolymers*, 22:2577, 1983.

[47] S. Kauffman. Adaptation on rugged fitness landscapes. In D. L. Stein, editor, *Lectures in the Sciences of Complexity*, pages 619–712. Addison Wesley, 1989.

[48] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *J. theor. Biol.*, 128:11 – 45, 1987.

[49] S. A. Kauffman. *The origin of order.* Oxford University Press, 1993.

[50] P. S. Kim and R. L.Baldwin. Intermediates in the folding reactions of small proteins. *Annual Reviews of Biochemestry*, 59:631–660, 1990.

[51] S. Kim, F. Suddath, G. Quigley, A. McPherson, J. Sussman, A. Wang, N. Seeman, and A. Rich. Three dimensional tertiary structure of yeast phenylalanine tRNA. *Science*, 185:435, 1974.

[52] S. Kirkpatrick, D. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671, 1983.

[53] A. I. Lamond and T. J. Gibson. Catalytic RNA and the origin of genetic systems. *Perspectives*, 6:145 – 149, 1990.

[54] K. F. Lau and K. A. Dill. Theory for protein mutability and biogenesis. *Proc. Natl. Acad. Sci., USA*, 87:638 – 642, 1990.

[55] E. Lawler, A. Lenstra, A. R. Kan, and D. Shmoys. *The traveling Salesman problem.* Wiley, New York, 1985.

[56] S. Y. Le and M. Zuker. Common structures of the 5' non coding RNA in enteroviruses and rhinoviruses, thermodynamical stability and statistical significance. *J. Mol. Biol.*, 216:729, 1990.

[57] Levinthal. Are there pathways for protein folding ? *J. Chim. Phys.*, 65:44 – 45, 1968.

[58] D. J. Lipman and W. J. Wilbur. Modelling neutral and selective evolution of protein folding. *Proc. R. Soc. Lond. B*, 245:7 – 11, 1991.

[59] C. A. Macken and A. S. Perelson. Protein evolution on rugged landscapes. *Proc. Natl. Acad. Sci. USA*, 86:6191 – 6195, 1989.

[60] N. Madras and G. Slade. The Self Avoiding Walk. Birkhäuser, 1993.

[61] N. Madras and A. D. Sokal. The pivot algorithm: A highly efficient monte carlo method for the self avoiding-walk. *J. Stat. Phys.*, 50:109–186, 1987.

[62] B. Mandelbrot. *An information theory of the statistical structure of language.* Butterworths, 1953.

[63] A. E. Mark and W. F. van Gunsteren. Simulation of the thermal denaturation of hen egg white lysozyme: Trapping the molten globule state. *Biochemistry*, 31:7745 – 7748, 1992.

[64] H. M. Martinez. An RNA folding rule. *Nucleic Acid Research*, 12:323–335, 1984.

[65] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105 – 1119, 1990.

[66] S. Miller and L. Orgel. *The origin of life on the earth.* Prentice Hall, 1974.

[67] A. A. Mironov, L. P. Dyakonova, and A. E. Kister. A kinetic approach to the prediction of RNA secondary structures. *Journal of Biomolecular Structure and Dynamics*, 2:953, 1985.

[68] A. A. Mironov and A. E. Kister. RNA secondary structure formation during transcription. *J. of Biomolecular Structure and Dynamics*, 4:1–9, 1986.

[69] A. M.Lesk. Boolean programming formulation of some pattern-matching problems in molecular biology. *Journal of Chem. Soc. Faraday Trans.*, 89:2603 – 2607, 1993.

[70] K. Nagano, H. Takagi, and M. Havel. *Biochemie*, 73:947, 1991.

[71] L. Needleman and W. Wunsch. Predicting Protein Structures by Sequence Comparison. *J. Mol. Biol.*, 48:443, 1970.

[72] A. Neumaier. The mathematics of protein folding: a feasibility study of mathematical prediction of protein structure. *preprint*, 1995.

[73] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. Kleitman. Algorithms for loop matchings. *SIAM J. Appl. Math.*, 35:68 – 82, 1978.

[74] R. Nussinov and I. Tinoco. Sequential folding of a messenger RNA molecule. *J. Mol. Biol.*, 151:519 – 533, 1981.

[75] A. Oparin. *The origin of life on earth.* Academic Press, New York, 1957.

[76] L. E. Orgel. RNA catalysis and the origins of life. *J. theor. Biol.*, 123:127–149, 1986.

[77] H. Orland, C. Itzykson, and C. de Dominicis. *J. Phys. Lett.*, 46:L353, 1985.

[78] N. Pace, G. Olson, D. Smith, and B. James. Phylogenetic comparative analysis and the secondary structure of ribonuclease P RNA - a review. *Gene*, 82:65, 1989.

[79] G. Parisi, M. Mezard, and M. Virasoro. Spin glass theory and beyond. *World Scientific Lecture Notes in Physics*, 9, 1987.

[80] A. Perelson and G. Oster. Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination. *J. theor. Biol.*, 81:645, 1979.

[81] A. E. Peritz, R. Kierzek, N. Sugimoto, and D. H. Turner. Thermodynamic study of internal loops in oligoribonucleotides: Symmetric loops are more stable than asymmetric loops. *Biochemistry*, 30:6428–6436, 1991.

[82] C. Pleij and L. Bosch. *RNA Pseudoknots: Structure, Detection and Prediction*, pages 289 – 303. Academic Press, Methods in Enzymology, methods in enzymology 180 edition, 1989.

[83] P. K. Ponnuswamy. Hydrophobic characteristics of folded proteins. *Prog. Biophys. molec. Biol.*, 59:57 – 103, 1993.

[84] I. Rechenberg. *Evolutionstrategie.* problemata, Fromman-Holzboog, 1973.

[85] C. Reidys, P. Stadler, and P. Schuster. Generic properties of combinatory maps and applications on RNA secondary structures. *preprint*, 1994.

[86] A. Renner. *Self Avoiding Walks and Lattice Polymers.* University of Vienna, 1995. Master Thesis.

[87] A. Sali, E. Shakhnovich, and M. Karplus. How does a protein fold? *Nature*, 369:248 – 251, 1994.

[88] A. Sali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding: A lattice model study of the requirements for folding to the native state. *J. Mol. Biol.*, 235:1614 – 1636, 1994.

[89] P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker. From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Roy. Soc.(London)B*, 255:279–284, 1994.

[90] P. Schuster and P. F. Stadler. Landscapes: Complex optimization problems and biopolymer structures. *Computers Chem.*, 18:295–314, 1994.

[91] P. Schuster and J. Swetina. Stationary mutant distributions and evolutionary optimization. *Bull. Math. Biol.*, 50:635, 1988.

[92] E. Shakhnovich and A. Gutin. Implications of thermodynamics of protein folding for evolution of primary sequences. *Nature*, 346:773 – 775, 1990.

[93] B. A. Shapiro. An algorithm for comparing multiple RNA secondary stuctures. *CABIOS*, 4(3):387–393, 1988.

[94] B. A. Shapiro and K. Zhang. Comparing multiple RNA secondary structures using tree comparisons. *CABIOS*, 6:3092, 1990.

[95] J. Skolnick, A. Kolinski, C. L. BrooksIII, A. Godzik, and A. Rey. A method for predicting protein structure from sequence. *Current Biology*, 3:414 – 426, 1993.

[96] Z. Slanina. Does the global energy minimum always also mean the thermodynamically most stable structure? *Journal of Molecular Structure*, 206:143 – 151, 1990.

[97] J. M. Smith. Natural selection and the concept of a protein space. *Nature*, 225:563 – 564, 1970.

[98] P. F. Stadler and W. Schnabl. The landscape of the traveling salesman problem. *Physics Letters A*, 161:337 – 344, 1992.

[99] P. Stolorz. Recursive approaches to the statistical physiscs of lattice proteins. *Proc. 27th Hawaii International Conference on System Sciences*, 1994.

[100] P. Stolorz, A. Lapedes, and Y. Xia. Predicting protein secondary structure using neural net and statistical methods. *J. Mol. Biol.*, 225:363 – 377, 1992.

[101] L. Stryer. *Biochemistry.* Freeman, 1990.

[102] N. Sugimoto, R. Kierzek, and D. H. Turner. Sequence dependence for the energetics of dangling ends and terminal base pairs in ribonucleic acid. *Biochemistry*, 26:4554–4558, 1987.

[103] N. Sugimoto, R. Kierzek, and D. H. Turner. Sequence dependence for the energetics of terminal mismatches in ribooligonuleotides. *Biochemistry*, 26:4559–4561, 1987.

[104] S. Sun. *Physical Chemistry of Macromolecules.* Wiley Interscience, 1994.

[105] M. Tacker. *Robust Properties of RNA Secondary Structure Folding Algorithms.* PhD thesis, University of Vienna, 1993.

[106] M. Tacker, W. Fontana, P. F. Stadler, and P. Schuster. Statistics of RNA melting kinetics. *Eur. J. Biophys.*, 23(1):29 – 38, 1994.

[107] W. R. Taylor and C. A. Orengo. Protein sequence alignment. *J. Mol. Biol.*, 208:1 – 22, 1989.

[108] C. Tuerk, P.Gauss, C. Thermes, D. Groebe, M. Gayle, N. Guild, G. Stormo, Y. D'Aubenton-Carafa, O. Uhleenbeck, I. J. Tinoco, E. N. Brody, and L. Gold. Cuucgg hairpins: extraordinarily stable RNA secondary structures associated with various biochemical processes. *Proc. Natl. Acad. Sci. USA*, 85:1364–1368, 1988.

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

[110] R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *J. Mol. Biol.*, 231:75 – 81, 1993.

[111] M. Vingron and M. S. Waterman. Sequence alignment and penalty choice reviews of concepts, case studies and implications. *J. Mol. Biol.*, 235:1 – 12, 1994.

[112] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the diference. *Biol. Cybern.*, 63:325 – 336, 1990.

[113] E. Weinberger and P. Stadler. Why some fitness landscapes are fractal. *J. theor. Biol.*, 163:255, 1993.

[114] D. B. Wetlaufer. Nucleation, rapid folding and globular intrachain regions in proteins. *Proc. Natl. Acad. Sci., USA*, 70:697 – 701, 1973.

[115] S. Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. volume 1, pages 356 – 366, 1932.

[116] G. Zipf. *Human behaviour and the principle of least effort.* Adison Wesley, Reading (Mass.), 1949.

[117] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[118] M. Zuker. Suboptimal sequence alignment im molecular biology alignment with error analysis. *J. Mol. Biol.*, 221:403 – 420, 1991.

[119] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46:591–621, 1984.

[120] M. Zuker and P. Stiegler. Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

[121] R. Zwanzig, A. Szabo, and B. Bagchi. Levinthal's paradox. *Proc. Natl. Acad. Sci., USA*, 89:20–22, 1992.

Curriculum vitae

Erich Bornberg - Bauer

* 1963-05-10, Vienna/Austria

1969-1973	:	Primary School, Wien 4
1973-1981	:	"Naturwissenschaftliches Realgymnasium mit Latein und Darstellender Geometrie", Waltergasse Wien 4. (Secondary + High School with scientific bias, latin and descriptive geometry) closing and graduation with first class honour
1981-1992	:	Studies in Biochemistry and Chemistry, University Vienna
8/1985-11/1985	:	Laboratory assistent at Cancer Research Institute, Medical Faculty, University of Vienna Departement Prof. U. Wintersberger: Isolation and characterisation of Ribonuclease H from Saccharomyce Cerevisiae
6/1989-1/1990	:	Employee at SAIL, A-1180 Vienna, collaboration for the design of a computer-application for geodesy and landscaping
9/1990-	:	Studies in mathematics
9/1991-5/1992	:	Diploma Thesis "The Shape Space of RNA Secondary Structures", at the "Institut für theoretische Chemie der Universität Wien" with Prof. Peter Schuster, closing with first class honour
12/1993-	:	Ph.D thesis at the Institute for Theoretical Chemistry University Vienna with Prof. Peter Schuster
3/94 - 6/94	:	Tutor for "Computerdemonstrationen zu Biochemie 4, Proteine", Institute for Theoretical Chemistry, University Vienna
7-8/1994	:	Work at Santa Fe Institute

Participation on Seminars, Courses and Congresses:

Jan. 1992, 93, 94	:	Winter Seminar on Evolution in Klosters, CH
Feb. 1994	:	FEBS Meeting: " Young Scientists Perspective of Molecular Biotechnology" in Ascona / CH. Talk and Abstract-Contribution: "Random Structures and Evolution of Biopolymers".
Apr. 1994	:	Congress: "Self Organization of Biopolymers", Jena, GER. Poster
Apr. 1994		Meeting of the " Deutsche Bunsengesellschaft für physikalische Chemie"; Talk and Poster Contribution on: " Random Structures and Folding of Biopolymers"
Jun./Jul. 1994	:	" 7th Summer School on Complex Systems " at the Santa Fe Institute Talk on "Folding of Lattice Polymers"

Publications:

[1] Erich Bauer : The Shape Space of RNA Secondary Structures, Master-thesis 1992, Inst. f. Theoret. Chemistry., University Vienna

[2] W. Fontana, P. Stadler, E. Bornberg - Bauer, et. al., P. Schuster: RNA folding and Combinatory Landscapes *Phys. Rev. E,* 47(3):2083 - 2099, 1993

[3] E. Bornberg - Bauer : Random Structures and Evolution of Biopolymers *Acta Pharamaceutica Helvetice (special issue)* in press

[4] M. Tacker, P. Stadler, E. Bornberg - Bauer, I. Hofacker, P. Schuster: Robust Properties of RNA Secondary Structure Folding Algorithms, 1994 *(preprint)*

[5] E. Bornberg - Bauer, A. Renner, P. Schuster: A computer package for folding and stucture comparison of Lattice Proteins, 1994 *(technical report)*

[6] E. Bornberg - Bauer et al.: Structure Landscapes of Lattice Polymers, 1994 *(in preparation)*

watch out! he is still alife:

wenn ist das nunstük git und slotermeyer?

ja!

...

beiherhund, das oder die

flipperwaldt gersput.