THE DEVELOPMENT OF NEARLY DETERMINISTIC METHODS FOR OPTIMISING PROTEIN GEOMETRY

Dem Department Informatik
der Fakultät für Mathematik, Informatik und Naturwissenschaften
an der Universität Hamburg
eingereichte

Dissertation

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

in der Abteilung Biomolekulare Modellierung
des Zentrums für Bioinformatik Hamburg

vorgelegt von

Dipl.-Bioinf. Gundolf Walter Hubert Schenk

geboren am 3.2.1979 in Hannover

Hamburg, 27. Oktober 2011

Tag der mündlichen Prüfung:	28. März 2012
Erstgutachter:	Herr Prof. Dr. Andrew Torda
Zweitgutachter:	Herr Prof. Dr. Jianwei Zhang

Zusammenfassung

Proteine sind langkettige Biomoleküle mit charakteristischen Funktionen, die eine Hauptrolle in allen Lebewesen einnehmen. Diese Funktion ergibt sich aus der Proteinstruktur, die wiederum durch einen komplizierten Mechanismus basierend auf der Aminosäuresequenz bestimmt wird. Der genaue Vorgang ist nicht vollständig verstanden, aber die Strukturen zu kennen ist wichtig für die pharmazeutische Industrie, sowie für die Bio- und Nanotechnologie. Leider ist es langsam und teuer sie experimentell zu bestimmen. Hohes Interesse besteht auch daran die Sequenz anzupassen um stabile industrielle Enzyme zu machen oder um Moleküle mit speziellen Formen herzustellen, z.B. für Biosensoren.

Eine Struktur am Computer anhand der Sequenz vorherzusagen ist ein klassisches Problem der theoretischen Biochemie, welches bisher nicht gelöst wurde. In dieser Arbeit liegt der Schwerpunkt auf methodologischen Verbesserungen, die verbreitete chemische Annahmen vermeiden. Eine allgemeine Methode zur Erstellung numerischer Modelle wird hier entwickelt und analysiert. Sie basiert auf einem statistischen Korrelationsmodell von Sequenz und Struktur und benutzt Ideen aus der selbst-konsistenten Mittelfeld (SCMF) Optimierung. Das Verfahren lässt sich erfolgreich auf die Strukturvorhersage- und Sequenzdesignprobleme anwenden ohne eine Boltzmann Statistik anzunehmen.

Das statistische Modell basiert auf einer Mischverteilung von bivariaten Gaußverteilungen und 20-wege Bernoulliverteilungen. Die Gaußverteilungen modellieren die kontinuierlichen Variablen der Proteinstruktur (Torsionswinkel) und die Bernoulliverteilungen erfassen die Sequenzpräferenzen. Anstelle ein Protein als statistische Einheit zu verstehen, werden hier leichter zu verarbeitende Fragmente betrachtet. Mehrere Ansätze sie wieder zusammenzusetzen werden diskutiert. Aber die Fragmente bilden lokale statistische Einheiten, die nicht notwendiger Weise miteinander übereinstimmen. Ein passendes Verfahren solche Inkonsistenzen zu behandeln, ist die SCMF Optimierung.

Mittelfeld oder SCMF Verfahren betrachten das zu optimierende System in allen Lösungszuständen gleichzeitig. In bestehenden Ansätzen wurde dazu ein Energiepotential erstellt, das gemittelte, paarweise Wechselwirkungen zwischen Untersystemen abbildet. Die Zustandsgewichte der Untersysteme wurden durch wiederholte Anwendung des Boltzmannverhältnisses alternierend in Energien und Wahrscheinlichkeiten umgerechnet bis ein selbst-konsistenter Zustand des gesamten Systems erreicht wird. Mit dem hier präsentierten Ansatz ist es möglich die Zustandswahrscheinlichkeiten direkt zu optimieren. Die Boltzmannverteilung ist keine notwendige Annahme. Daher ist die Methode auch auf Systeme mit unbekanntem Ensemble anwendbar.

Abstract

Proteins are long-chained biomolecules with distinctive functions, that take a major role in all living systems. The function is defined by the protein structure, which in turn is determined via a complicated mechanism based on the amino acid sequence. The exact procedure is not fully understood. However, knowing the structure is important for the pharmaceutical industry as well as bioengineering and nanotechnology. Unfortunately, determining it experimentally is slow and expensive. There is also much interest in being able to adapt the sequence to make stable industrial enzymes or to form molecules with specialised shapes, e.g. for biosensors.

Predicting a structure computationally from the sequence is a classic problem in theoretical biochemistry, that has not been solved yet. In this work the emphasis lies in methodological improvements, that avoid common chemical preconceptions. A general method for building numerical models is developed and analysed here. It is based on a statistical correlation scheme of sequence and structure using ideas from self-consistent mean field (SCMF) optimisation. The procedure is successfully applied to the structure prediction and sequence design problems without using a Boltzmann formalism.

The statistical model is based on a mixture distribution of bivariate Gaussian and 20-way Bernoulli distributions. The Gaussian distributions model the continuous variables of the structure (dihedral angles) and the Bernoulli distributions capture the sequence propensities. Instead of treating the protein as a statistical unit, easier to handle fragments are used. Several approaches to recombine them are discussed. But the fragments form local statistical units that do not necessarily agree with each other. A method suited to deal with such inconsistencies is SCMF optimisation.

Mean field or SCMF methods optimise a system by treating all solution states at the same time. In existing approaches, an energy potential was introduced that reflects the pairwise mean interaction between subsystems. The state weights of the subsystems were converted alternately into energies and probabilities by applying the Boltzmann relation repeatedly until a self-consistent state for the whole system is reached. With the approach presented here it is possible to optimise the state probabilities directly. The Boltzmann distribution is essentially an unnecessary assumption. Therefore, the method is also applicable to systems with an unknown ensemble.

Contents

1 Introduction **1**
- 1.1 Proteins . . . 1
- 1.2 Protein Classification . . . 2
  - 1.2.1 Protein Descriptors . . . 2
  - 1.2.2 Classifications . . . 3
  - 1.2.3 Protein Structure Prediction . . . 4
  - 1.2.4 Protein Sequence Prediction . . . 5
- 1.3 Bayesian Classification of Proteins . . . 5
  - 1.3.1 Protein Descriptors . . . 5
  - 1.3.2 Class Models . . . 7

2 Protein Reconstruction **11**
- 2.1 Probability Model . . . 11
  - 2.1.1 Dihedral Angles . . . 12
- 2.2 Methods . . . 19
  - 2.2.1 Dihedral Angles . . . 20
  - 2.2.2 Backbone Reconstruction . . . 25
  - 2.2.3 Amino Acids . . . 26
- 2.3 Results . . . 28
  - 2.3.1 Influence of Bond Lengths and Angles . . . 29
  - 2.3.2 Dihedral Angle Reconstruction . . . 29
  - 2.3.3 Sequence Reconstruction . . . 32
- 2.4 Discussion . . . 34
  - 2.4.1 Influence of Bond Lengths and Angles . . . 34
  - 2.4.2 Dihedral Angle Reconstruction . . . 35
  - 2.4.3 Sequence Reconstruction . . . 36

3 Self-consistent mean field optimisation **39**
- 3.1 Standard SCMF . . . 40
- 3.2 Purely statistical SCMF . . . 41
- 3.3 Simulated Annealing . . . 43

4 Structure Prediction **47**
4.1 Methods 49
4.1.1 Optimising Class Weights 49
4.1.2 Calculating and Sampling Structures 51
4.1.3 Refining Structures 54
4.1.4 CASP 56
4.2 Results 57
4.2.1 Most Likely Structures 57
4.2.2 Sampling 58
4.2.3 Optimising Class Weights 62
4.2.4 Scoring 64
4.2.5 CASP 66
4.3 Discussion 70
4.4 Outlook 73
4.4.1 Sequence Profiles 73
4.4.2 Loop Modelling 74

5 Sequence Prediction **77**
5.1 Methods 78
5.1.1 Optimising Residue Probabilities 79
5.1.2 Generating Sequences 80
5.2 Results 80
5.2.1 Optimising Residue Probabilities 81
5.2.2 Generating Sequences 82
5.3 Discussion 84

6 Conclusions and Outlook **87**

A Directional Statistics **93**
A.1 Means and Differences 93
A.2 Circular Distributions 94

B Analytic Derivation of the Adaptive Cooling Threshold **99**

Bibliography **103**

Curriculum Vitae **111**

Publications **112**

Acknowledgements **113**

Chapter 1

Introduction

Modelling protein molecules means trying to predict their structure given their amino acid sequence. The problem has been in the literature since the first protein structures were solved [KDS⁺60, PRC⁺60], but remains a challenging task. Finding models for protein structures experimentally is a rather costly challenge. So since the beginnings the computational modelling has been a helpful crucial tool. Only in recent years has there been much success in this area, although still there are many open questions and without experimental techniques no reliable model can be obtained [OBH⁺99, BDNBP⁺09, CKF⁺09, MFK⁺]. However, with the initiatives for structural genomics the number of experimentally solved models is increasing rapidly. The growing database of solved protein structures offers new ways to model proteins on a statistical basis. In this work, an innovative optimisation scheme based on a purely statistical protein model is developed from ideas of self-consistent mean field optimisation. With a rather simple scoring scheme we give a proof-of-principle and show how to apply our approach to structure prediction and sequence design.

1.1 Proteins

Proteins are biomolecular polymers [RR07], i.e. they are built as chains or sequences of 20 different amino acid types connected via peptide bonds found, for example, in biological cells. The chain of atoms directly involved in the peptide bond is called the backbone and the other atoms are called sidechains. Figure 1.1.1 sketches a small stretch of such a polymer. The typical lengths of protein sequences are between 50 and 800 amino acids long. In solution, e.g. in the cell, they fold to a three-dimensional structure. That means, the atoms of a protein arrange in space in a more or less compact way and this arrangement stays mostly the same throughout the protein's lifetime though sometimes flex-

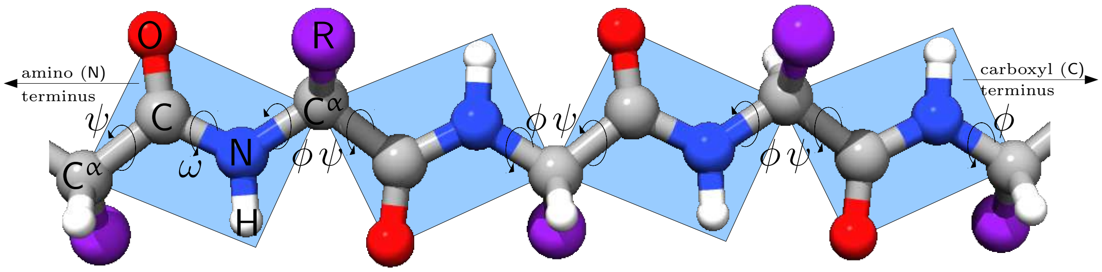


Figure 1.1.1: The backbone of a protein is connected via peptide bonds. The sidechains are only symbolised by the purple dots.

ibility has been observed. The determination of this structure is mainly driven by the polymer's amino acid composition, i.e. its sequence, and sometimes helper molecules, such as chaperones and other cofactors [Anf73, MH93, vdBWDE00].

In the biological context each protein is specialised in a distinct biochemical function, for example in a metabolic process. This function is solely determined by the proteins structure, i.e. by the positions of all protein atoms in space. Therefore, the knowledge of the protein structure is crucial in understanding its function and role in the biochemical processes that drive life.

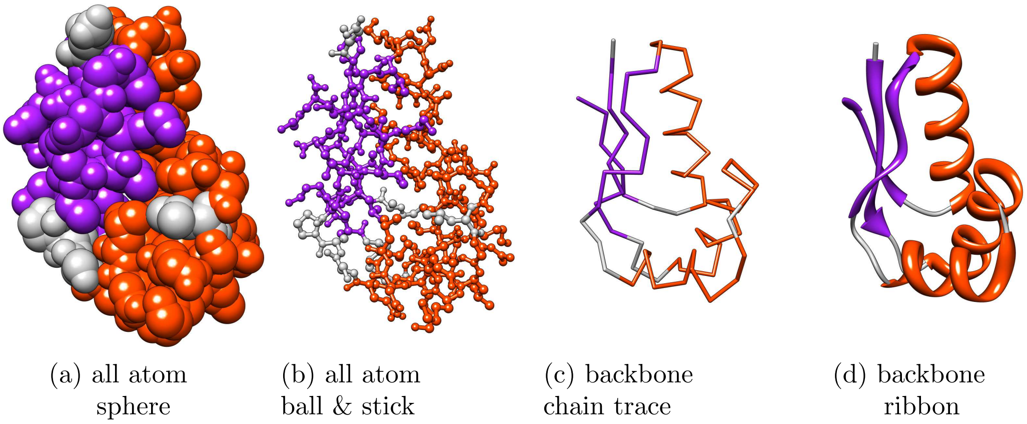
Many experimental methods are used to solve the structure of a protein. The most important ones are probably X-ray crystallography, nuclear magnetic resonance spectroscopy and electron microscopy. Structures determined via X-ray crystallography typically have the highest resolution allowing the most detailed view on the atom positions. Almost all solved structures are deposited in the Protein Data Bank [PDBa, BWF$^+$00], currently (2011-07-18) consisting of 72000 entries of which 64000 are determined by X-ray crystallography.

If a protein structure is known, one can display it via 3D rendering techniques on a computer screen. Figure 1.1.2 shows several representations and levels of abstraction that can be chosen in such programs [PGH$^+$04]. The colouring is according to secondary structure, that is, local regular structure of the backbone, e.g. α-helices in red or β-strands in purple.

1.2 Protein Classification

1.2.1 Protein Descriptors

Monomer proteins may be described by three levels, primary structure, secondary structure and tertiary structure. The primary structure is the sequence of amino



(a) all atom sphere
(b) all atom ball & stick
(c) backbone chain trace
(d) backbone ribbon

Figure 1.1.2: Representations and levels of abstraction for a small protein structure (pdb-id: 1CTF). Pictures are made by UCSF Chimera [PGH+04].

acid types, i.e. a text string. By secondary structure the sequence of local consecutive structural subunits is denoted, mostly also a character string. Tertiary structure is the three-dimensional arrangement or the spatial coordinates of the polymer, typically a table of atom positions. In this thesis, the primary structure will be called “sequence” and the tertiary structure “structure” or “fold”.

The backbone of a protein can be described solely by the sequence of dihedral angles at the α-carbons as the amide plane is known to form a rigid unit, figures 1.1.1 and 1.3.1. The sidechains are often only described by their type and are modelled separately. Figure 1.2.1 shows the histogram of the dihedral angles ϕ and ψ in the protein data bank [PDBa].

1.2.2 Classifications

Typically classifications are used to simplify the handling of proteins. Their scope is either to provide an easy, human-interpretable way for distinguishing between proteins or to enable computers to perform calculations such as comparisons and predictions in reasonable time. That means, a classification basically reduces the complexity of the task and often leads to an approximation. For a review of structural and functional protein classifications, see [OCE+03].

We have developed and successfully applied a scoring scheme to protein comparisons using sequence, structure or both [SMT08b, MST09]. It is based purely on Bayesian statistics and derived via a maximally parsimonious automatic classifier [CPT02, CS96] from overlapping protein fragments. In this work we use this classification for the sake of optimisation and prediction of unknown features, such as structure or sequence.

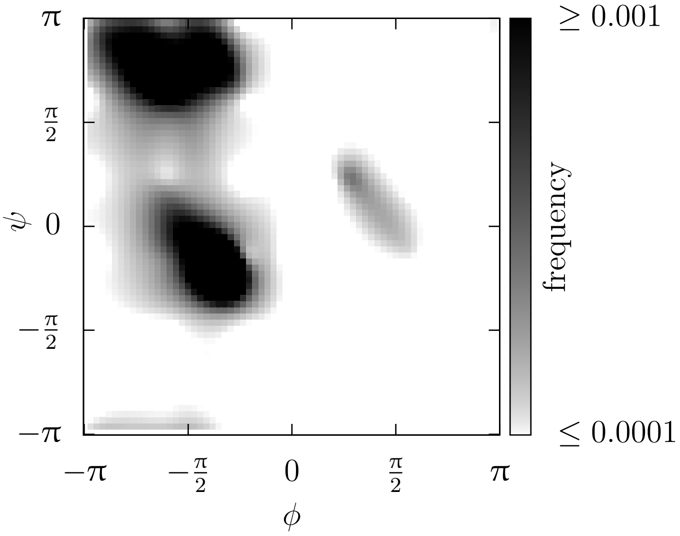


Figure 1.2.1: Distribution of the dihedral angles ϕ and ψ in the protein database.

1.2.3 Protein Structure Prediction

Structure prediction means to propose one or more structures that a given sequence would adopt in a living cell. There are two broad categories of such programs:

1. homology based and
2. *ab initio*.

In homology-based modelling one uses structural information from known templates. Programs based on this idea perform very well if homologous templates can be identified. The quality of the outcome is highly dependent on the quality and identification of a related template. Finding such a template can be a difficult task, which often gives no satisfying results. If that is the case, the *ab initio* modelling programs try to predict the structure from scratch without explicitly knowing any related structures. This approach is less reliable than homology modelling for the cases where the structure of a related protein is known, but they are the only applicable method when no template structure is available.

Given that homology-based and *ab initio* methods work best on different problems, some automatic methods attempt to select or combine the approaches. In the CASP 8 meeting the Robetta and the Zhang servers were among the top candidates [BDNBP$^+$09, CKF$^+$09]. These two programs are difficult to put into one of the described categories. They are fragment based, i.e. the amino acid sequence is split into short fragments and these fragments are used as queries on the template database. By doing that, the advantage of the homology-based approaches are also available for sequences where no complete structural template could be found. However still, if no or only bad template fragments can be found, modelling from scratch becomes necessary. In this work we propose a new approach, which is based on self-consistent mean-field optimisation (SCMF), but using a framework of descriptive statistics and simulated annealing (see section 1.3 and chapter 3).

1.2.4 Protein Sequence Prediction

Sequence prediction, or more often sequence optimisation, is the task of finding one or more amino acid sequences that fold to a given structure. It can be viewed as the inverse problem to structure prediction. This NP-hard problem [PW02] is not only fundamentally interesting but also relevant for understanding the relation between sequence and structure. It is likely to be of practical use for improving protein stability, for example thermostability in washing powder, for specialising reaction conditions in the production of biodiesel or even for nanotechnology, for example in the case of personalised medicine. Therefore, this problem is often called protein design. Despite some impressive literature results, the design steps have often been rather *ad hoc* and the method is far from routine [KB00, KAS$^+$09, FF07, SJ09, KAV05, JAC$^+$08, SDB$^+$08, Tor04]. In this work we propose an innovative general approach, which is based on the same ideas as structure prediction, i.e. self-consistent mean-field optimisation (SCMF) on a framework of descriptive statistics and simulated annealing (see section 1.3 and chapter 3).

1.3 Bayesian Classification of Proteins

The classification used in this work has been described earlier [SMT08b]. However, the main ideas are explained also here.

1.3.1 Protein Descriptors

The focus of this work is on the conformation of the protein backbone. That means, the modelling of the side-chain conformations is postponed. This is a typical procedure and simplifies the problem.

The proteins are described in terms of amino acid sequence and dihedral backbone angles. Here, the variability of bond lengths and bond angles in the backbone

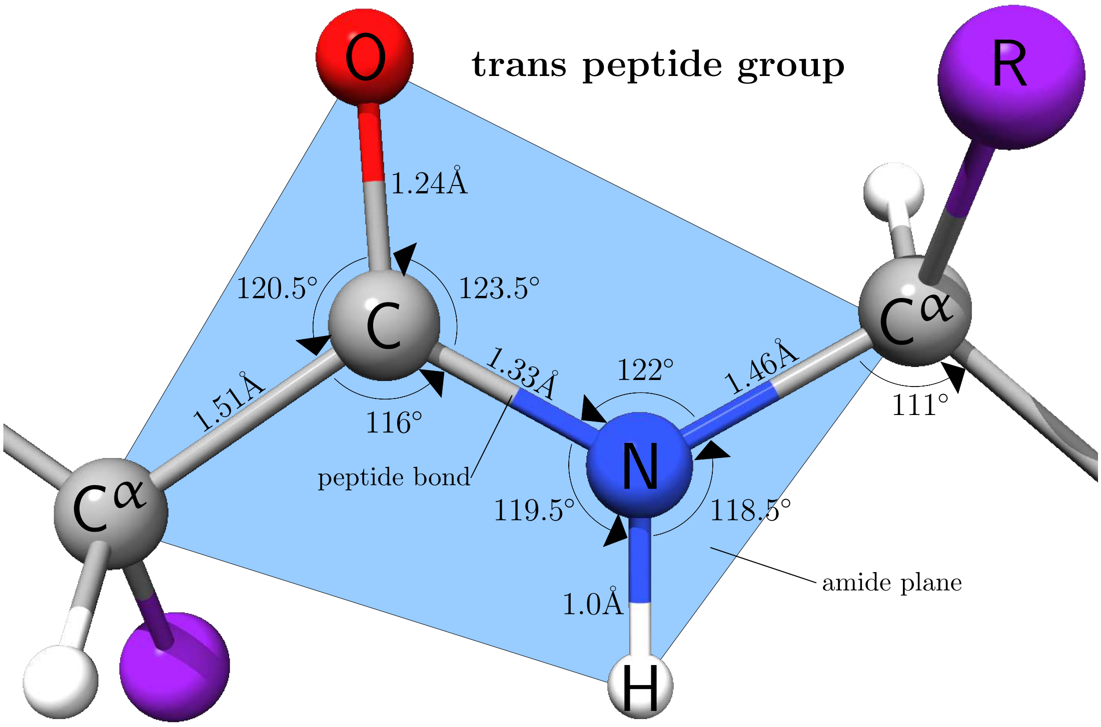


Figure 1.3.1: Standard geometry in a trans peptide group.

(the amide plane) within and across proteins is assumed very low and is therefore ignored, see figure 1.3.1. The carbonyl group and the amide group are nearly in the same plane and this unit (light blue) is very rigid [HBA$^+$07]. For its actual influence on the full backbone (re)construction see subsection 2.3.1.

For a protein consisting of l amino acids, i.e. of chain length l, $2l-2$ dihedral angles describe the conformation of the backbone, namely $l-1$ ϕ and $l-1$ ψ angles (see figure 1.1.1). In this work a Bayesian classification is used to model the statistical properties [SMT08b], see also subsection 1.3.2.

1.3.1.1 Overlapping Fragments

At the heart of the method is a classification of protein fragments. A protein is then represented as the set of all possible overlapping fragments, given by

$$\left\{ (\boldsymbol{s}_i, \boldsymbol{x}_i) \,\middle|\, \begin{array}{r} \boldsymbol{s}_i = (a_i, \ldots, a_{i+k-1})^{\mathsf{T}} \\ \wedge\ \boldsymbol{x}_i = (\phi_i, \psi_i, \ldots, \phi_{i+k-1}, \psi_{i+k-1})^{\mathsf{T}} \end{array} \forall i \in [1, l-k+1] \right\},$$

where $a_i \in [1, 20]$ is the amino acid type at residue i. The non-existent dihedral angles ϕ_1 and ψ_l are ignored and $^{\mathsf{T}}$ denotes the transpose.

1.3.1.2 Dihedral Angles, Bond Lengths and Angles

The typical distances and angles on the amide plane are shown in figure 1.3.1. As these values are fixed the backbone conformation is solely described by its dihedral angles, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ (see also figure 1.1.1). When dealing with angles, one has to be aware of the specialities characterising angles, see appendix A. The classification in use is based on Gaussian (normal) distributions and has no knowledge about the periodic nature of angles. It is therefore important to choose boundaries covering one period (no redundancy) with minimal border effects. That means, the boundaries are shifted to low populated areas. For ϕ this is $[0, 2\pi)$ and for ψ this is $\left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)$. Figure 1.3.2 shows a scatterplot of the angles pairs found in the protein database [PDBa].

1.3.2 Class Models

The class models used for the amino acid labels are multiway Bernoulli distributions $\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S} = \boldsymbol{s}_i)$ and for angles a multivariate Gauss distribution $\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} = \boldsymbol{x}_i)$. These are chosen mainly for convenience so that the AutoClass-C package [CPT02, CS96] could be used for the parameter search.

The combination of these models is a classical mixture model:

$$\mathrm{p}((\boldsymbol{S}, \boldsymbol{X}) = (\boldsymbol{s}_i, \boldsymbol{x}_i)) = \sum_{j=1}^{n} w_j \, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S} = \boldsymbol{s}_i) \, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} = \boldsymbol{x}_i), \tag{1.3.1}$$

where w_j denotes the weight for the class $\mathcal{C}_j$ and n is the number of classes. The class weights may also be interpreted as prior probabilities of the fragment $\boldsymbol{F} = (\boldsymbol{S}, \boldsymbol{X})$ being in class $\mathcal{C}_j$, which is then denoted by $\mathrm{p}(\boldsymbol{F} \sim \mathcal{C}_j) = w_j$. Formula (1.3.1) comprises two parts, the multiway Bernoulli distributions and the multivariate Gauss distribution. These are explained in the next two paragraphs.

1.3.2.1 Multiway Bernoulli Distribution

The amino acid labels are modelled by several classes of multiway Bernoulli distributions. In each class model the dependencies between residues are completely ignored. That means, a sequence fragment $\boldsymbol{s}_i \in [1, 20]^k$ is interpreted as an instance of a discrete random vector $\boldsymbol{S}$ of k independent random variables S_t with possible outcomes in $[1, 20]$ each. In each dimension each amino acid label is assigned the conditional probability of observing this label given the class model $\mathcal{C}_j$, i.e.

$$\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S} = \boldsymbol{s}_i) = \prod_{t=1}^{k} \mathrm{p}_{\mathcal{C}_j}(S_t = s_{it}). \tag{1.3.2}$$

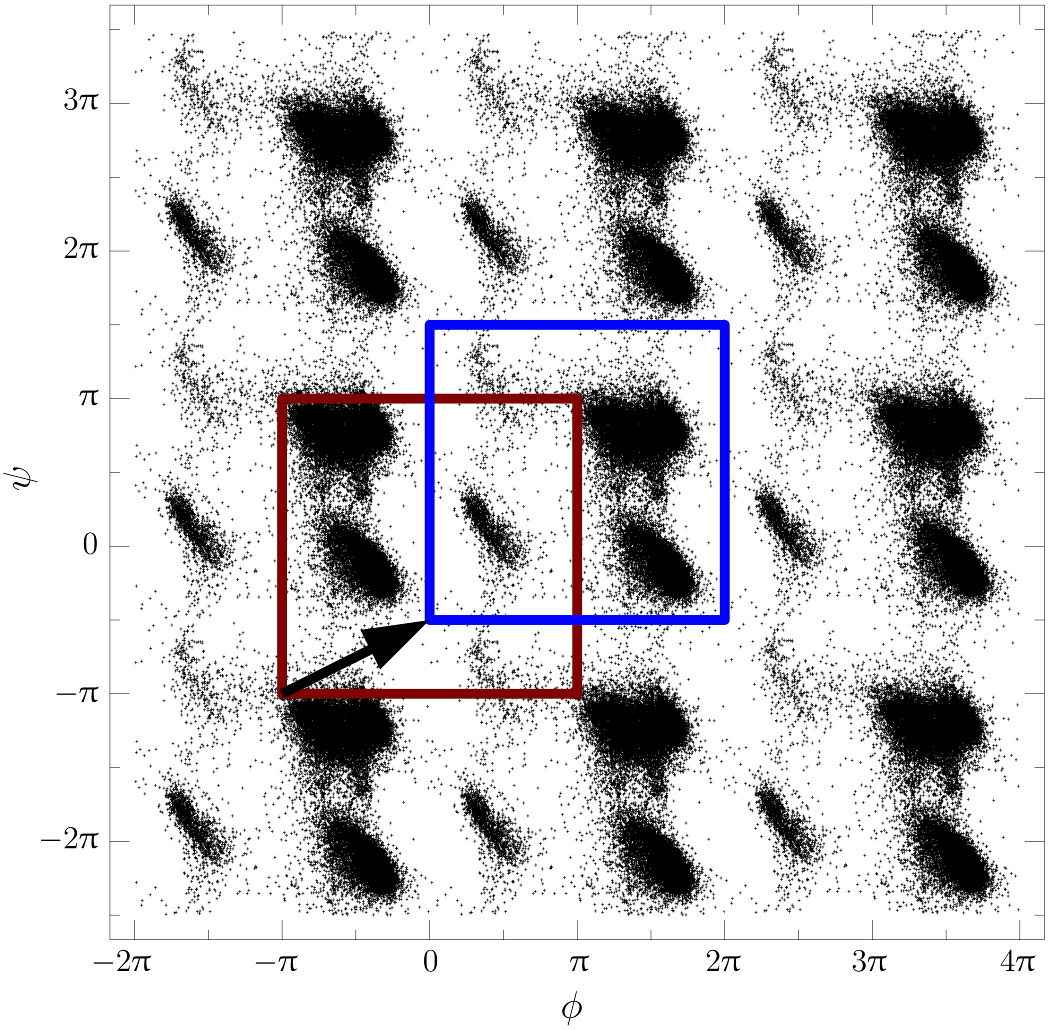


Figure 1.3.2: Periodic Ramachandran plot: red box shows the original ranges, $[-\pi, \pi) \times [-\pi, \pi)$; blue box shows the shifted borders to underpopulated areas, $[0, 2\pi) \times [-\frac{\pi}{2}, \frac{3\pi}{2})$.

However, in the weighted sum of all classes, formula (1.3.1), the residues are effectively correlated.

1.3.2.2 Multivariate Normal Distribution

The structural terms, i.e. the dihedral angles, are modelled by several classes of multivariate Gaussian or normal distributions. In each class model a fragment of dihedral angles is interpreted as an instance of a continuous random vector $\boldsymbol{X}$ of $2k$ correlated random variables. The model allows the full correlation between the dihedral angles, but, to keep computational costs low and avoid overfitting, only angle pairs $\boldsymbol{x}_{it} = \binom{\phi_t}{\psi_t}$ are allowed to correlate. However, as in the case of the Bernoulli distributions, the mixture model (formula (1.3.1)) captures the full correlation. Each fragment $\boldsymbol{x}_i$ is assigned the conditional probability density of observing the dihedral angles given the class model $\mathcal{C}_j$, i.e.

$$\begin{aligned}
\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} = \boldsymbol{x}_i) &= \frac{\exp\left[-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\boldsymbol{C}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^{\mathsf{T}}\right]}{\sqrt{(2\pi)^{2k}|\det \boldsymbol{C}_j|}} \\
&\approx \prod_{t=1}^{k} \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X}_t = \boldsymbol{x}_{it}) \\
&= \prod_{t=1}^{k} \frac{\exp\left[-\frac{1}{2}\left(\boldsymbol{x}_{it} - \boldsymbol{\mu}_{j_t}\right)\boldsymbol{C}_{j_{t,t}}^{-1}\left(\boldsymbol{x}_{it} - \boldsymbol{\mu}_{j_t}\right)^{\mathsf{T}}\right]}{\sqrt{(2\pi)^{2}|\det \boldsymbol{C}_{j_{t,t}}|}},
\end{aligned}$$

where $\boldsymbol{\mu}_j = \left(\boldsymbol{\mu}_{j_1}^{\mathsf{T}}, \ldots, \boldsymbol{\mu}_{j_t}^{\mathsf{T}}, \ldots, \boldsymbol{\mu}_{j_k}^{\mathsf{T}}\right)^{\mathsf{T}}$ with $\boldsymbol{\mu}_{j_t} = \begin{pmatrix} \mu_{j_{t_\phi}} \\ \mu_{j_{t_\psi}} \end{pmatrix}$ is the vector of means and $\boldsymbol{C}_j = \left(\boldsymbol{C}_{j_{t,t'}}\right)_{t,t' \in [1,k]}$ with $\boldsymbol{C}_{j_{t,t'}} = \begin{pmatrix} c_{j_{t_\phi,t'_\phi}} & c_{j_{t_\phi,t'_\psi}} \\ c_{j_{t_\psi,t'_\phi}} & c_{j_{t_\psi,t'_\psi}} \end{pmatrix}$ is the $2k \times 2k$ covariance matrix of class $\mathcal{C}_j$.

In the next chapter the statistical model is analysed extensively and several formal approaches for reconstructing dihedral angles and amino acids from it are developed. Chapter 3 introduces the innovative optimisation method. And in chapters 4 and 5 the methods are applied to structure prediction and sequence design, respectively. Finally, in chapter 6 conclusions and outlook are given.

Chapter 2

Protein Reconstruction

In this chapter the predictive potential of the statistical descriptor based on the Bayesian classification is analysed in order to see how well the full protein description is approximated. Therefore, the full description is reconstructed from the statistical classification and checked against the original description.

Before the reconstruction approaches are introduced in section 2.2, it is important to have a thorough understanding of the probability model. Therefore, the statistical descriptors are introduced formally along with some explanations about their theoretical properties in the next section.

2.1 Probability Model

As has been explained previously [SMT08b], the classification can be used to describe a protein by a set of class probability vectors $\{\boldsymbol{v}_i \big| i \in [1, l-k+1]\}$ constructed from overlapping fragments. First the backbone of the protein is broken into maximally overlapping fragments by sliding a window of length k. Each fragment is described by its dihedral angles (or amino acid labels), from which probability vectors are calculated. The vector elements are then associated with the Gaussian terms (Bernoulli terms) in different ways, e.g. to build mixture distributions (subsection 1.3.2) or to estimate continuous and discrete features (subsections 2.2.1 and 2.2.3). Here, the theoretical properties of the model are explained for reconstructing continuous features (i.e. dihedral angles), as their handling is more complicated than the discrete features (amino acid labels) and many insights apply for both cases.

2.1.1 Dihedral Angles

According to [SMT08b, CS96] the elements of the probability vector $\boldsymbol{v}_i = \big(v_{ij}\big)_{j\in[1,n]}$ for a structure fragment $\boldsymbol{x}_i = (\phi_i, \psi_i, \dots, \phi_{i+k-1}, \psi_{i+k-1})^{\mathsf{T}}$ are given by

$$v_{ij} = \mathrm{p}_{\boldsymbol{x}_i}(\boldsymbol{F} \sim \mathcal{C}_j) = \mathrm{p}\big(\boldsymbol{F} \sim \mathcal{C}_j \,\big|\, \boldsymbol{X} \approx \boldsymbol{x}_i\big) = \frac{w_j \, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} \approx \boldsymbol{x}_i)}{\sum\limits_{j'=1}^{n} w_{j'} \, \mathrm{p}_{\mathcal{C}_{j'}}(\boldsymbol{X} \approx \boldsymbol{x}_i)} \tag{2.1.1}$$

and can be interpreted as the n-dimensional vector of probabilities of all n classes $\mathcal{C}_j$ given the dihedral angles $\boldsymbol{x}_i$. The prior class weights w_j directly come from the classification and the structural class weights (joint probabilities) are taken to be

$$\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} \approx \boldsymbol{x}_i) = \int \cdots \int\limits_{A} \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} = \boldsymbol{x}) \, \mathrm{d}x_1 \dots \mathrm{d}x_{2k}, \tag{2.1.2}$$

where $A = [x_{i1} - \epsilon_1, x_{i1} + \epsilon_1] \times \cdots \times [x_{i2k} - \epsilon_{2k}, x_{i2k} + \epsilon_{2k}]$ is the integration domain with $\boldsymbol{\epsilon} \in \mathbb{R}^{+2k}$ being some small vector-valued error.

From first principles one might be tempted to estimate the original angles of the fragment by the expectation values of the associated mixture distribution (formula (1.3.1), page 7) [Kre98]. The vector of expectation values using the probability vector corresponding to the fragment is given by

$$\begin{aligned} \boldsymbol{x}_i^{\text{est}} &= \int \cdots \int\limits_{\mathbb{R}^{2k}} \boldsymbol{x} \sum_{j=1}^{n} v_{ij} \, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} = \boldsymbol{x}) \, \mathrm{d}x_1 \dots \mathrm{d}x_{2k} \\ &= \sum_{j=1}^{n} v_{ij} \int \cdots \int\limits_{\mathbb{R}^{2k}} \boldsymbol{x} \, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} = \boldsymbol{x}) \, \mathrm{d}x_1 \dots \mathrm{d}x_{2k} \\ &= \sum_{j=1}^{n} v_{ij} \boldsymbol{\mu}_j, \end{aligned} \tag{2.1.3}$$

where $\boldsymbol{\mu}_j$ is the vector-valued mean of the multivariate Gaussian modelling the class $\mathcal{C}_j$.

2.1.1.1 Limitations of Joint Probabilities

The formula for expectation values (2.1.3) has some principal limitations when used with random vectors. First principles say, the expectation value of a random vector is defined component-wise [Kre98]. Formula (2.1.3) can be viewed as an average of the class means, which does not lead to the right answer as the

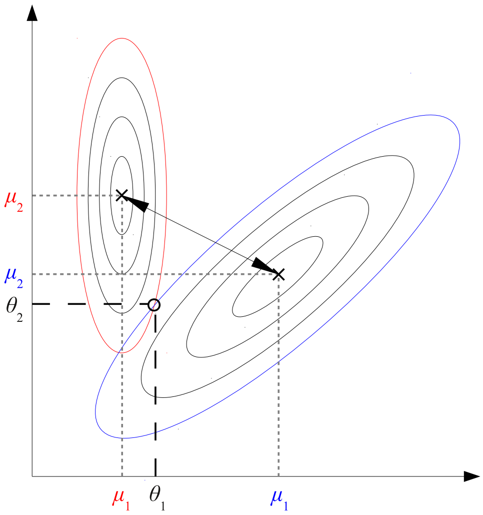


Figure 2.1.1: The ellipsoidal contours of two bivariate classes (red and blue). The angle pair to be estimated is marked by a circle.

probability vectors (formula (2.1.1)) comprise joint probabilities for multivariate classes. Let us consider the origin of these probabilities in order to understand why this a problem. For each class the probability densities of the Gaussian distribution functions define contours in the conformational space (i.e. the space of dihedral angles) in shape of ellipsoids. In the ideal case, these ellipsoids would intersect in a single point, which would be the original angles corresponding to the probability vector. However, a weighted average of two class means would lie only somewhere on a line connecting the centres of the classes. For two dimensions this problem is illustrated in figure 2.1.1 and is described formally in the following.

Let $\boldsymbol{A}_j = \boldsymbol{C}_j^{-1}$ be the inverse correlation matrix of class $\mathcal{C}_j$, then the probability density for two correlated angles θ_1 and θ_2 is given by

$$\mathrm{p}_{\mathcal{C}_j}\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} = \frac{\exp\left(-\frac{1}{2}\left(\theta_1-\mu_{j1}, \theta_2-\mu_{j2}\right)\boldsymbol{A}_j\binom{\theta_1-\mu_{j1}}{\theta_2-\mu_{j2}}\right)}{\sqrt{(2\pi)^2|\det \boldsymbol{C}_j|}}.$$

This is equivalent to

$$\begin{aligned}
&-2\log\left(\mathrm{p}_{\mathcal{C}_j}\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}\sqrt{(2\pi)^2|\det \boldsymbol{C}_j|}\right)\\
&\quad= \left(\theta_1-\mu_{j1}, \theta_2-\mu_{j2}\right)\boldsymbol{A}_j\begin{pmatrix}\theta_1-\mu_{j1}\\ \theta_2-\mu_{j2}\end{pmatrix}\\
&\quad= \sum_{t=1}^{2}\sum_{t'=1}^{2} a_{j\,t,t'}\left(\theta_t-\mu_{jt}\right)\left(\theta_{t'}-\mu_{jt'}\right).
\end{aligned} \tag{2.1.4}$$

The expression (2.1.4) is the equation for the ellipsoidal contour, also known as the Mahalanobis distance [Mah36].

Averaging class means with one single weight per class will not lead to the original angles in general. With joint probabilities of whole fragments, it is impossible to find a correct weighting, as illustrated in figure 2.1.1. Instead, the use of marginal probability densities to average the class means correctly with weights for each dimension is considered and analysed in the following

2.1.1.2 Marginal Probability Density Vectors

Marginal probability density vectors are defined in close analogy to normal probability vectors. A third index $t \in [1, k]$ reflecting the position in the fragment is added to the known indices for fragments $i \in [1, l-k+1]$ and for classes $j \in [1, n]$. For the dihedral angle pair $\boldsymbol{x}_{it} = (x_{it_\Phi}, x_{it_\Psi})^{\mathsf{T}} = (\phi_{i+t-1}, \psi_{i+t-1})^{\mathsf{T}}$ at the tth residue of a fragment $\boldsymbol{x}_i$ of length k the elements of the marginal probability density vectors $\boldsymbol{v}_i^{\text{marg}} = \left(\boldsymbol{v}_{i,1}^{\text{marg}}, \dots, \boldsymbol{v}_{i,t}^{\text{marg}}, \dots, \boldsymbol{v}_{i,k}^{\text{marg}}\right)$ are taken to be

$$\begin{aligned}
\boldsymbol{v}_{i,t}^{\text{marg}} &= \begin{pmatrix} \left(v_{i,t_\Phi\, j}^{\text{marg}}\right)_{j\in[1,n]} \\ \left(v_{i,t_\Psi\, j}^{\text{marg}}\right)_{j\in[1,n]} \end{pmatrix}^{\mathsf{T}} \\
&= \begin{pmatrix} \left(\mathrm{p}\left(\boldsymbol{F}\sim\mathcal{C}_j \,\middle|\, X_{t_\Phi} = \phi_{i+t-1}\right)\right)_{j\in[1,n]} \\ \left(\mathrm{p}\left(\boldsymbol{F}\sim\mathcal{C}_j \,\middle|\, X_{t_\Psi} = \psi_{i+t-1}\right)\right)_{j\in[1,n]} \end{pmatrix}^{\mathsf{T}} \\
&= \begin{pmatrix} \left(\dfrac{w_j\, \mathrm{p}_{\mathcal{C}_j}(X_{t_\Phi} = \phi_{i+t-1})}{\sum\limits_{j'=1}^{n} w_{j'}\, \mathrm{p}_{\mathcal{C}_{j'}}(X_{t_\Phi} = \phi_{i+t-1})}\right)_{j\in[1,n]} \\ \left(\dfrac{w_j\, \mathrm{p}_{\mathcal{C}_j}(X_{t_\Psi} = \psi_{i+t-1})}{\sum\limits_{j'=1}^{n} w_{j'}\, \mathrm{p}_{\mathcal{C}_{j'}}(X_{t_\Psi} = \psi_{i+t-1})}\right)_{j\in[1,n]} \end{pmatrix}^{\mathsf{T}} . \qquad (2.1.5)
\end{aligned}$$

Actually, each $\boldsymbol{v}_i^{\text{marg}}$ is a probability density matrix consisting of $2k$ columns and n rows. The marginal probability density functions $\mathrm{p}_{\mathcal{C}_j}(X_{t_\Phi} = \phi_{i+t-1})$ and $\mathrm{p}_{\mathcal{C}_j}(X_{t_\Psi} = \psi_{i+t-1})$ are univariate Gaussian distributions with parameters $\mu_{j_{t_\Phi}}, c_{j_{t_\Phi,t_\Phi}}$ and $\mu_{j_{t_\Psi}}, c_{j_{t_\Psi,t_\Psi}}$, respectively [Pol95].

With these marginal probability vectors the expectation values can be formulated in analogy to equation (2.1.3) for $x_{it_\Phi}^{\text{est}}$ and $x_{it_\Psi}^{\text{est}}$, respectively, by

$$\begin{aligned}
x_{it_\Phi}^{\text{est}} &= \int\limits_{\mathbb{R}} \phi \sum_{j=1}^{n} v_{i,t_\Phi\ j}^{\text{marg}}\, \mathrm{p}_{\mathcal{C}_j}(X_{t_\Phi} = \phi)\, \mathrm{d}\phi \\
&= \sum_{j=1}^{n} v_{i,t_\Phi\ j}^{\text{marg}} \int\limits_{\mathbb{R}} \phi\, \mathrm{p}_{\mathcal{C}_j}(X_{t_\Phi} = \phi)\, \mathrm{d}\phi \\
&= \sum_{j=1}^{n} v_{i,t_\Phi\ j}^{\text{marg}} \mu_{j t_\Phi} && (2.1.6_\Phi) \\
\wedge\ x_{it_\Psi}^{\text{est}} &= \sum_{j=1}^{n} v_{i,t_\Psi\ j}^{\text{marg}} \mu_{j t_\Psi}. && (2.1.6_\Psi)
\end{aligned}$$

The marginal probabilities used here, allow one to calculate expectation values in accordance with first principles by averaging class means component-wise. Still, this will not necessarily lead to the original angle values as the Gaussian probability densities do not scale linearly in the space between the class means. They rather lie on some manifold where linear Euclidean distances are not valid. The observations in paragraph 2.1.1.1 show that the manifold is defined by the Mahalanobis distance (formula (2.1.4)). However, in order to use a linear average of class means, the weights must scale linear with the angle value that led to the class weights. In the next paragraph a linearisation of this manifold is derived for the class weights.

2.1.1.3 Linearisation of Class weights

In this paragraph, weights for averaging mean values of univariate Gaussian distributions are derived step by step. Consider the simple case of two mean values μ_1 and μ_2. Given two density values p_1, p_2 from two univariate Gaussians with parameters μ_1, σ_1^2 and μ_2, σ_2^2, respectively, and assuming that p_1 and p_2 were derived from a single value α, then this value can be estimated by averaging over the class means $\mu_1 \neq \mu_2$ using suitable weights w_1 and w_2, i.e.

$$\begin{aligned}
\alpha^{\text{est}} &= w_1\mu_1 + w_2\mu_2 \\
\text{with } 1 &= w_1 + w_2.
\end{aligned}$$

As α is an angle, without loss of generality it can be assumed that $\mu_1 \leq \alpha \leq \mu_2$ holds. Otherwise one μ is substituted with its periodic image, i.e. $\mu + 2\pi$ or $\mu - 2\pi$, and exchanged with the other μ. In the following the unknown weights w_1, w_2 are analytically derived. As α^{est} is unknown, a formulation only based on the density values p_1 and p_2 is given below.

Proposition:

$$
\begin{aligned}
w_1 &= 1 + \frac{\sqrt{2}\sigma_1}{\mu_1 - \mu_2}\sqrt{-\log\,[\sqrt{2\pi}\sigma_1] - \log\,[p_1]} \\
\wedge\; w_2 &= 1 - \frac{\sqrt{2}\sigma_2}{\mu_1 - \mu_2}\sqrt{-\log\,[\sqrt{2\pi}\sigma_2] - \log\,[p_2]}
\end{aligned}
$$

Proof: Let α be the original angle. Then the densities p_1 and p_2 are by the definition of univariate Gaussian (normal) distribution functions taken to be

$$
\begin{aligned}
p_1(\alpha) &= \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left[-\frac{(\alpha-\mu_1)^2}{2\sigma_1^2}\right] \\
\wedge\; p_2(\alpha) &= \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left[-\frac{(\alpha-\mu_2)^2}{2\sigma_2^2}\right].
\end{aligned}
$$

Substituting into the proposed equations for the weights gives

$$
\begin{aligned}
w_1(\alpha) &= 1 + \frac{\sqrt{2}\sigma_1}{\mu_1 - \mu_2}\sqrt{-\log\left[\sqrt{2\pi}\sigma_1\right] - \log\,[p_1(\alpha)]} \\
&= 1 + \frac{\sqrt{2}\sigma_1}{\mu_1 - \mu_2}\sqrt{-\log\left[\sqrt{2\pi}\sigma_1\right] - \log\left[\frac{\exp\left[-\frac{(\alpha-\mu_1)^2}{2\sigma_1^2}\right]}{\sqrt{2\pi}\sigma_1}\right]} \\
&= 1 + \frac{\sqrt{2}\sigma_1}{\mu_1 - \mu_2}\sqrt{-\log\left[\sqrt{2\pi}\sigma_1\right] + \frac{(\alpha-\mu_1)^2}{2\sigma_1^2} + \log\left[\sqrt{2\pi}\sigma_1\right]} \\
&= 1 + \frac{(\alpha-\mu_1)}{\mu_1 - \mu_2} \\
\wedge\; w_2(\alpha) &= 1 - \frac{\sqrt{2}\sigma_2}{\mu_1 - \mu_2}\sqrt{-\log\left[\sqrt{2\pi}\sigma_2\right] - \log\,[p_2(\alpha)]} \\
&= 1 - \frac{(\alpha-\mu_2)}{\mu_1 - \mu_2}.
\end{aligned}
$$

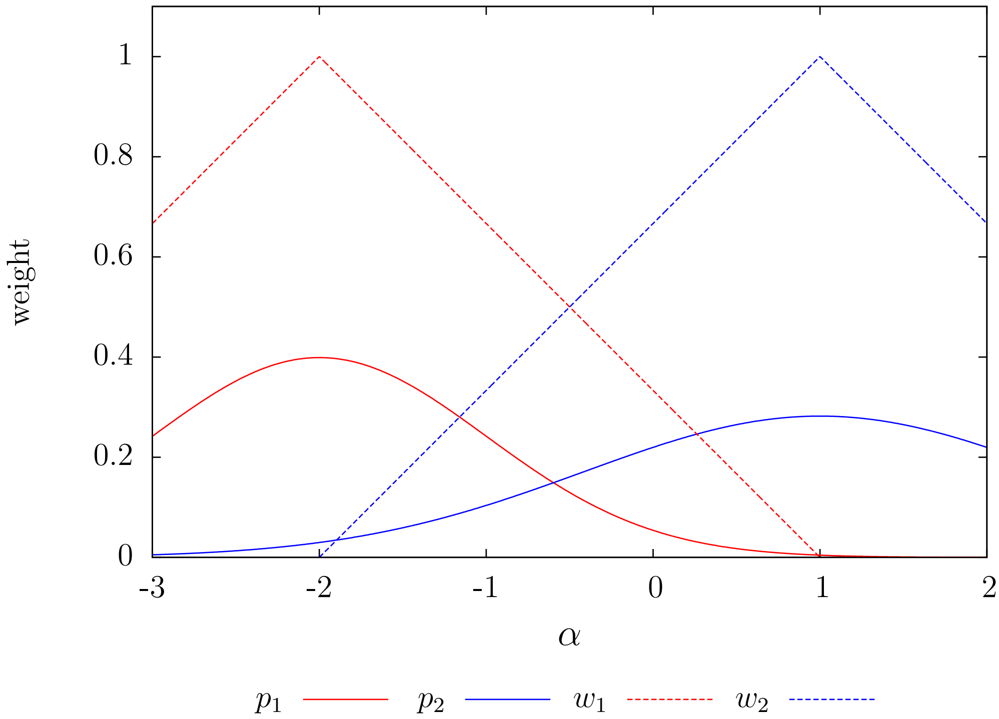


Figure 2.1.2: Two Gaussians with parameters $\mu_1 = -2, \sigma_1^2 = 1$ and $\mu_2 = 1, \sigma_2^2 = 2$ and their corresponding linearised weights.

In fact, the weights $w_1(\alpha), w_2(\alpha)$ scale linear with α. It remains to be shown that α equals α^{est}:

$$\begin{aligned}
\alpha^{\text{est}} &= w_1\mu_1 + w_2\mu_2 \\
&= \left(1 + \frac{(\alpha - \mu_1)}{\mu_1 - \mu_2}\right)\mu_1 + \left(1 - \frac{(\alpha - \mu_2)}{\mu_1 - \mu_2}\right)\mu_2 \\
&= \frac{(\mu_1 - \mu_2)\mu_1 + (\alpha - \mu_1)\mu_1 + (\mu_1 - \mu_2)\mu_2 - (\alpha - \mu_2)\mu_2}{\mu_1 - \mu_2} \\
&= \frac{\mu_1^2 - \mu_2\mu_1 + \alpha\mu_1 - \mu_1^2 + \mu_1\mu_2 - \mu_2^2 - \alpha\mu_2 + \mu_2^2}{\mu_1 - \mu_2} \\
&= \frac{\alpha\mu_1 - \alpha\mu_2}{\mu_1 - \mu_2} \\
&= \alpha
\end{aligned}$$

□

The weights and the two Gaussians are shown in figure 2.1.2.

This can be generalised to n Gaussians by sorting them according to their means and calculating the pairwise weights $w_{j,j'}\ \forall j, j' \in [1, n]$. Let $w'_{j,j'}$ be an auxiliary variable taken to be

$$w'_{j,j'} = \begin{cases} 1 + \dfrac{\sqrt{2}\sigma_j}{\mu_j - \mu_{j'}} \sqrt{-\log\left[\sqrt{2\pi}\sigma_j\right] - \log\left[p_j\right]} & \text{if } \mu_j < \mu_{j'}, \\ 1 + \dfrac{\sqrt{2}\sigma_j}{\mu_{j'} - \mu_j} \sqrt{-\log\left[\sqrt{2\pi}\sigma_j\right] - \log\left[p_j\right]} & \text{if } \mu_j > \mu_{j'}, \\ \text{undefined} & \text{else,} \end{cases}$$

then the pairwise weights are

$$w_{j,j'} = \begin{cases} 1 - \dfrac{w'_{j,j'} - 1}{1 + \frac{2\pi}{\mu_j - \mu_{j'}}} & \text{if } w'_{j,j'} + w'_{j',j} > 0 \wedge w'_{j,j'} + w'_{j',j} \neq 1 \wedge \mu_j < \mu_{j'}, \\ 1 - \dfrac{w'_{j,j'} - 1}{1 + \frac{2\pi}{\mu_{j'} - \mu_j}} & \text{if } w'_{j,j'} + w'_{j',j} > 0 \wedge w'_{j,j'} + w'_{j',j} \neq 1 \wedge \mu_j > \mu_{j'}, \\ w'_{j,j'} & \text{else.} \end{cases}$$

With this the wanted angle α^{est} can be calculated by obeying the periodicity.

$$\alpha^{\text{est}} = \begin{cases} \arctan\left(\dfrac{\bar{s}}{\bar{c}}\right) & \text{if } \bar{c} > 0 \wedge \bar{s} > 0, \\ \arctan\left(\dfrac{\bar{s}}{\bar{c}}\right) + 2\pi & \text{if } \bar{c} > 0 \wedge \bar{s} \leq 0, \\ \arctan\left(\dfrac{\bar{s}}{\bar{c}}\right) + \pi & \text{if } \bar{c} < 0, \\ \dfrac{\pi}{2} & \text{if } \bar{c} = 0 \wedge \bar{s} \geq 0, \\ \dfrac{3\pi}{2} & \text{if } \bar{c} = 0 \wedge \bar{s} < 0, \end{cases} \tag{2.1.7}$$

where $\bar{c}$ and $\bar{s}$ are the mean values of the cosines and sines of the estimated angle over all class pairs. Formally

$$\begin{aligned} \bar{c} &= \frac{1}{n} \sum_{j=1}^{n} \sum_{j'=1}^{n} w_{j,j'} \cos\mu_j + w_{j',j} \cos\mu_{j'} \\ \text{and } \bar{s} &= \frac{1}{n} \sum_{j=1}^{n} \sum_{j'=1}^{n} w_{j,j'} \sin\mu_j + w_{j',j} \sin\mu_{j'}. \end{aligned}$$

In order to apply this to estimate the dihedral angles $x_{it_\Phi}^{\text{est}}$ and $x_{it_\Psi}^{\text{est}}$, unnormalised marginal probability density vectors ${}^{\text{nonorm}}\boldsymbol{v}_i^{\text{marg}}$ must be used. For a structural fragment $\boldsymbol{x}_i$ the unnormalised marginal probability density vectors are taken to be

$$
\begin{aligned}
{}^{\text{nonorm}}\boldsymbol{v}^{\text{marg}}_{i,t} &= \begin{pmatrix} \left({}^{\text{nonorm}}v^{\text{marg}}_{i,t_\Phi\, j}\right)_{j\in[1,n]} \\ \left({}^{\text{nonorm}}v^{\text{marg}}_{i,t_\Psi\, j}\right)_{j\in[1,n]} \end{pmatrix}^{\mathsf{T}} \\
&= \begin{pmatrix} \left(w_j\, \mathrm{p}_{\mathcal{C}_j}(X_{t_\Phi} = x^{\text{est}}_{i t_\Phi})\right)_{j\in[1,n]} \\ \left(w_j\, \mathrm{p}_{\mathcal{C}_j}(X_{t_\Psi} = x^{\text{est}}_{i t_\Psi})\right)_{j\in[1,n]} \end{pmatrix}^{\mathsf{T}} . \qquad (2.1.8)
\end{aligned}
$$

Then the wanted angle α^{est} in formula (2.1.7) is substituted with $x^{\text{est}}_{i t_\Phi}$ or $x^{\text{est}}_{i t_\Psi}$ and the density value p_j is substituted with $\frac{{}^{\text{nonorm}}v^{\text{marg}}_{i,t_\Phi\, j}}{w_j}$ or $\frac{{}^{\text{nonorm}}v^{\text{marg}}_{i,t_\Psi\, j}}{w_j}$, respectively. The difference of ${}^{\text{nonorm}}\boldsymbol{v}^{\text{marg}}_{i,t}$ to the previous definition (formula (2.1.5)) is the missing normalisation. Class probabilities that are normalised can not be used for calculating exact weights, because the normalisation constant is not reconstructible. The probability vectors (formula (2.1.1)) that are commonly used in other applications [CS96, SMT08b, MT08, MST09] are based on probabilities of finding dihedral angles within some small error interval. The linearised weights that have been derived here require working with the density values directly. That means, the probabilities are not correctly defined and the approach has no sensible stochastic meaning apart from its analytical reasoning.

2.2 Methods

In this section some methods for reconstructing dihedral angles or amino acids from these probabilistic descriptors are presented and analysed for their potential to find the original angles or amino acid types. In figure 2.2.1 a test program flow is shown for reconstructing dihedral angles. It allows one to compare the constructed angles to the angles of a known structure. The steps are as follows.

1. The protein structure is broken into overlapping fragments of dihedral angles.
2. The fragments are classified, i.e. probability vectors are calculated.
3. The class weights are associated with Gaussians in order to build mixture distributions.
4. From the mixture distributions, dihedral angles are calculated and compared to the original values.
5. A three-dimensional model can be constructed from the dihedral angles and compared to the original structure.

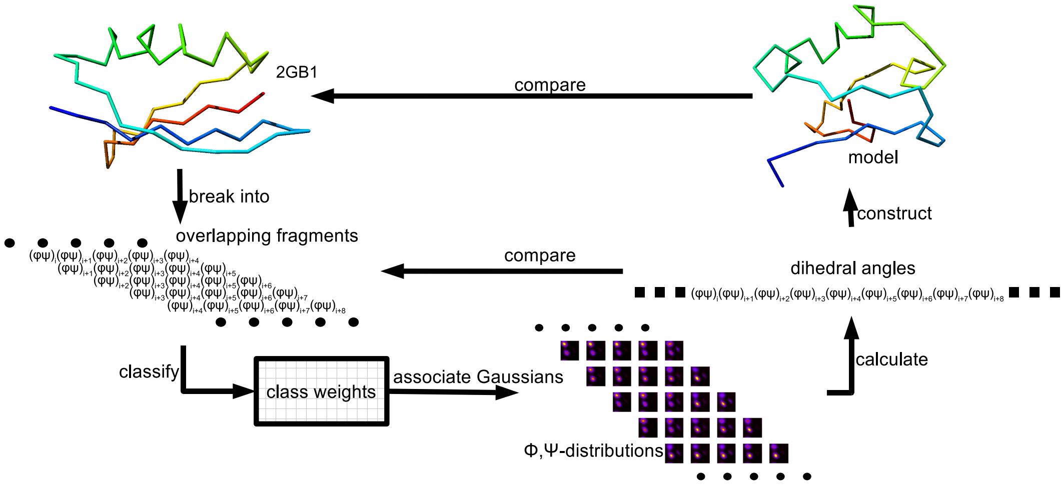


Figure 2.2.1: Method for testing the reconstruction capability of a classification.

Similarly, the amino acid label reconstruction can be tested.

2.2.1 Dihedral Angles

First some formulae for reconstructing dihedral angles from probability vectors are introduced. A protein structure of length l is represented by a set of $l-k+1$ probability vectors $\boldsymbol{v}_i = \left(v_{ij}\right)_{j\in[1,n]}$ each describing an overlapping fragment $\boldsymbol{x}_i$, where $i \in [1, l-k+1]$. This raises the problem of combining the overlapping parts. Since there is no rigorous method, several approaches were implemented and tested.

Similar to the case of sequence fragments (section 2.2.3 page 26), four principle combinations of the overlaps are proposed, which all have their advantages and disadvantages. Three combinations can be justified in terms of probability estimates. The fourth approach, could be termed an analytical-technical method. The four approaches are named after their primary underlying combination ideas. The three probabilistic approaches are

1. the “geometric mean” (formula (2.2.1)),

2. the “arithmetic mean” (formula(2.2.2)), and

3. the “maximum” approach (formula (2.2.3)),

which are introduced one after another in turn. The technical approach is called “inverse Gaussian” and is introduced in paragraph 2.2.1.4 on page 23.

2.2.1.1 Geometric Mean

Let $(\phi_i^{\mathrm{est}}, \psi_i^{\mathrm{est}})^\mathsf{T}$ denote a pair of estimated dihedral angles and $\boldsymbol{x}^{\mathrm{est}} = (\phi_1^{\mathrm{est}}, \psi_1^{\mathrm{est}}, \dots, \phi_i^{\mathrm{est}}, \psi_i^{\mathrm{est}}, \dots, \phi_l^{\mathrm{est}}, \psi_l^{\mathrm{est}})^\mathsf{T}$ the estimated dihedral angles of the whole protein, then the overlapping parts can be combined by treating them as statistically independent which means multiplying the corresponding probabilities. Normalisation then leads to a geometric mean of the probabilities of an angle pair appearing at different dimensions of the overlapping fragments. Formally given by

$$\begin{pmatrix}\phi_i^{\mathrm{est}}\\ \psi_i^{\mathrm{est}}\end{pmatrix} = \iint\limits_{\mathbb{R}^2} \begin{pmatrix}\phi\\ \psi\end{pmatrix} \sqrt[|\mathbb{I}_i|]{\prod_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v_{i'j}\, \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{X}_{i-i'+1} = \begin{pmatrix}\phi\\ \psi\end{pmatrix}\right)}\, \mathrm{d}\phi\, \mathrm{d}\psi, \tag{2.2.1}$$

where $\boldsymbol{X}_t = \begin{pmatrix}X_{t\phi}\\ X_{t\psi}\end{pmatrix}$ is the bivariate random vector modelling the values of the dihedral angle pair at the tth residue of a fragment. The index set for a residue i of all overlapping fragments $\boldsymbol{x}_{i'}$ is defined as

$$\mathbb{I}_i = [\max\{1, i-k+1\},\ \min\{l-k+1, i\}]$$

and the integrals run over the angles ϕ and ψ.

2.2.1.2 Arithmetic Mean

The second combination approach is based on the idea that the overlapping parts are actually just different stochastic models accounting for the same variable. That means the corresponding probabilities should be weighted by the respective prior model probabilities and added up. However, as the prior model probabilities are not known, they are assumed to have equal probabilities here. Therefore, the sum is normalised by the number of overlapping models $|\mathbb{I}_i|$ leading to an arithmetic mean of the overlapping probabilities. The formula is very similar to the geometric mean approach, but can be simplified to

$$\begin{aligned}\begin{pmatrix}\phi_i^{\mathrm{est}}\\ \psi_i^{\mathrm{est}}\end{pmatrix} &= \iint\limits_{\mathbb{R}^2} \begin{pmatrix}\phi\\ \psi\end{pmatrix} \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v_{i'j}\, \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{X}_{i-i'+1} = \begin{pmatrix}\phi\\ \psi\end{pmatrix}\right) \mathrm{d}\phi\, \mathrm{d}\psi \\ &= \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v_{i'j}\boldsymbol{\mu}_{j_{i-i'+1}}, \end{aligned} \tag{2.2.2}$$

where $\boldsymbol{\mu}_{j_t} = \begin{pmatrix}\mu_{j_{t\phi}}\\ \mu_{j_{t\psi}}\end{pmatrix}$ is the two-dimensional mean vector for the tth residue of class $\mathcal{C}_j$.

2.2.1.3 Maximum

The third approach is similar to the second combination approach in that it also combines the overlapping parts by an arithmetic mean. But instead of calculating the expectation values for each mixture model, the expectation values (i.e. the mean vectors) of the individual classes with highest probabilities are averaged. This is computationally less expensive and formally taken to be

$$\begin{pmatrix}\phi_i^{\mathrm{est}}\\ \psi_i^{\mathrm{est}}\end{pmatrix} = \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i} \boldsymbol{\mu}_{j_{\max}(i')_{i-i'+1}}, \tag{2.2.3}$$

where $j_{\max}(i') = \arg\max\limits_{j=1}^{n} v_{i'j}$ is the index of the class with the highest weight.

From theory it is known, that linear averaging with joint probabilities will not necessarily lead to correct estimates. In section 2.1 it has been shown, that marginal probability densities could lead to better results. With the marginal probability density vectors an arithmetic mean approach can be formulated in analogy to equation (2.2.2) for ϕ_i^{est} and ψ_i^{est}, respectively, by

$$\begin{aligned}
\phi_i^{\mathrm{est}} &= \int\limits_{\mathbb{R}} \phi \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v^{\mathrm{marg}}_{i',i-i'+1_{\upphi}j}\,\mathrm{p}_{\mathcal{C}_j}(X_{i-i'+1_{\upphi}} = \phi)\,\mathrm{d}\phi \\
&= \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v^{\mathrm{marg}}_{i',i-i'+1_{\upphi}j}\int\limits_{\mathbb{R}} \phi\,\mathrm{p}_{\mathcal{C}_j}(X_{i-i'+1_{\upphi}} = \phi)\,\mathrm{d}\phi \\
&= \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v^{\mathrm{marg}}_{i',i-i'+1_{\upphi}j}\mu_{j_{i-i'+1_{\upphi}}} && (2.2.4_{\upphi}) \\
\wedge\ \hat{\psi}_i &= \frac{1}{|\mathbb{I}_i|}\sum_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v^{\mathrm{marg}}_{i',i-i'+1_{\uppsi}j}\mu_{j_{i-i'+1_{\uppsi}}}. && (2.2.4_{\uppsi})
\end{aligned}$$

In similar ways the geometric mean and maximum approaches, i.e. formulae (2.2.1) and (2.2.3), can be reformulated using marginal probability density vectors by replacing $(\phi_i^{\mathrm{est}}, \psi_i^{\mathrm{est}})^{\mathsf{T}}$ with ϕ_i^{est} or ψ_i^{est} and $v_{i'j}$ with $v^{\mathrm{marg}}_{i',i-i'+1_{\upphi}j}$ or $v^{\mathrm{marg}}_{i',i-i'+1_{\uppsi}j}$ and $\mathrm{p}_{\mathcal{C}_j}\Big(\boldsymbol{X}_{i-i'+1} = (\phi,\psi)^{\mathsf{T}}\Big)$ with $\mathrm{p}_{\mathcal{C}_j}(\mathit{\Phi}_{i-i'+1} = \phi)$ or $\mathrm{p}_{\mathcal{C}_j}(\mathit{\Psi}_{i-i'+1} = \psi)$, respectively. This leads to the geometric mean approach for marginal probability density vectors

$$\begin{aligned}
\phi_i^{\mathrm{est}} &= \int\limits_{\mathbb{R}} \phi \sqrt[|\mathbb{I}_i|]{\prod_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v^{\mathrm{marg}}_{i',i-i'+1_{\upphi}j}\,\mathrm{p}_{\mathcal{C}_j}(\mathit{\Phi}_{i-i'+1} = \phi)\,\mathrm{d}\phi} && (2.2.5_{\upphi}) \\
\wedge\ \psi_i^{\mathrm{est}} &= \int\limits_{\mathbb{R}} \psi \sqrt[|\mathbb{I}_i|]{\prod_{i'\in\mathbb{I}_i}\sum_{j=1}^{n} v^{\mathrm{marg}}_{i',i-i'+1_{\uppsi}j}\,\mathrm{p}_{\mathcal{C}_j}(\mathit{\Psi}_{i-i'+1} = \psi)\,\mathrm{d}\psi,} && (2.2.5_{\uppsi})
\end{aligned}$$

and the maximum approach based on marginal probability density values

$$\phi_i^{\text{est}} = \frac{1}{|\mathbb{I}_i|} \sum_{i' \in \mathbb{I}_i} \mu_{j_{\max\, i-i'+1_{\phi}}} \tag{2.2.6$_\phi$}$$

$$\wedge\ \psi_i^{\text{est}} = \frac{1}{|\mathbb{I}_i|} \sum_{i' \in \mathbb{I}_i} \mu_{j_{\max\, i-i'+1_{\psi}}}. \tag{2.2.6$_\psi$}$$

These approaches are also tested with linearised weights as suggested in subsection 2.1.1.3, page 15.

2.2.1.4 Inverse Gaussian

Here the fourth rather technical approach to reconstructing dihedral angles from probability vectors is introduced and later analysed for its performance. The inverse Gaussian approach is not justified in probabilistic terms, but rather analytically derives the original angles from their probability functions.

First the inverse Gaussian approach is developed using joint probabilities. The basic idea for the inverse Gaussian approach is to construct an inverse formula by solving the following equation system for the unknown angles $\boldsymbol{x}_i$.

$$\begin{aligned}
& {}^{\text{nonorm}}v_{i1} &=&\ w_1 \frac{\exp\left[-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_1\right) \boldsymbol{C}_1^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_1\right)^{\mathsf{T}}\right]}{\sqrt{(2\pi)^{2k} |\det \boldsymbol{C}_1|}} \\
& &\vdots& \\
\wedge\ & {}^{\text{nonorm}}v_{ij} &=&\ w_j \frac{\exp\left[-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \boldsymbol{C}_j^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^{\mathsf{T}}\right]}{\sqrt{(2\pi)^{2k} |\det \boldsymbol{C}_j|}} \\
& &\vdots& \\
\wedge\ & {}^{\text{nonorm}}v_{in} &=&\ w_n \frac{\exp\left[-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_n\right) \boldsymbol{C}_n^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_n\right)^{\mathsf{T}}\right]}{\sqrt{(2\pi)^{2k} |\det \boldsymbol{C}_n|}}
\end{aligned}$$

A univariate inverse Gaussian would lead to two possible angle values, left and right from the class mean. To decide which angle is the correct one a second univariate inverse Gaussian (a second class) is needed giving another two possible angle values. Two values of these four possible values are equal or at least very similar and can be interpreted as the wanted angle. As illustrated in subsection 2.1.1.1, page 12, for two dimensions the equation system results in either 0, 1, 2, 3 or 4 solutions (an infinite number of solutions does not occur, as $\boldsymbol{\mu}_j \neq \boldsymbol{\mu}_{j'}$ holds for all $j' \neq j$). But as the equation system is over-defined (many more classes than angles to be estimated, i.e. $n \gg 2k$), a single solution could be found this way. In general, multivariate Gaussians are used to model multivariate classes of fragments of length k consisting of $2k$ dihedral angles. As shown in paragraph 2.1.1.1 for two dimensions, one $2k$-variate inverse Gaussian would lead to a $2k$-dimensional ellipsoid for a given function value (also called a contour), i.e. the solutions for a single equation lie on a hyperellipsoid of dimension $2k$. The formulae are very large and are not given here explicitly. The common points of two such ellipsoids (if they form an equation system) would describe either

- a $2k-1$-dimensional ellipsoid,
- a single point,
- two $2k-1$-dimensional ellipsoids,
- two single points,
- one $2k-1$-dimensional ellipsoid and one point,
- the original ellipsoid or
- no point.

However, the last two cases can be excluded, because the equation system is assumed to be solvable (at least approximately) since it was constructed from a single vector $\boldsymbol{x}_i$ and the ellipsoids come from different classes. In order to determine the wanted angle values, a minimum of $2k+1$ $2k$-variate Gaussians is needed. Although, the number of classes n is much higher then their dimension $2k$, it is not guaranteed to find a unique solution.

The solution becomes much simpler when the unnormalised marginal probability vectors are used for the following equation system (for ψ angles this looks analogous).

$$
\begin{array}{lrcl}
 & {}^{\text{nonorm}}v^{\text{marg}}_{i,t_\phi\;1} & = & w_1 \dfrac{\exp\left[-\frac{\left(\phi_{i+t-1}-\mu_{1t_\phi}\right)^2}{2c_{1t_\phi,t_\phi}}\right]}{\sqrt{2\pi c_{1t_\phi,t_\phi}}} \\
 & & \vdots & \\
\wedge & {}^{\text{nonorm}}v^{\text{marg}}_{i,t_\phi\;n} & = & w_n \dfrac{\exp\left[-\frac{\left(\phi_{i+t-1}-\mu_{nt_\phi}\right)^2}{2c_{nt_\phi,t_\phi}}\right]}{\sqrt{2\pi c_{nt_\phi,t_\phi}}} \\
\hline
\end{array}
$$

$$
\begin{array}{lrcl}
\Longleftrightarrow & \mu_{1t_\phi} \pm \sqrt{-2c_{1t_\phi,t_\phi}\log\left[\dfrac{{}^{\text{nonorm}}v^{\text{marg}}_{i,t_\phi\;1}}{w_1}\sqrt{2\pi c_{1t_\phi,t_\phi}}\right]} & = & \phi_{i+t-1} \\
 & & \vdots & \\
\wedge & \mu_{nt_\phi} \pm \sqrt{-2c_{nt_\phi,t_\phi}\log\left[\dfrac{{}^{\text{nonorm}}v^{\text{marg}}_{i,t_\phi\;n}}{w_n}\sqrt{2\pi c_{nt_\phi,t_\phi}}\right]} & = & \phi_{i+t-1} \\
\hline
\end{array}
\tag{2.2.7}
$$

A single equation results in two possible values and the common value of the system (if any) should be the desired angle. Theoretically this may to lead to exactly reconstructed angles.

For results on these formulae for dihedral angle reconstruction see section 2.3.2.

2.2.2 Backbone Reconstruction

Using the bond lengths and the bond angles from the literature [EH91, HBA+07] or from a known protein structure a full atom representation of the protein backbone can be build solely by knowing the dihedral angle sequence. This should work, because the amide plane is known to be a relatively rigid unit, see figure 1.3.1, page 6. Here, the same method as in [Mah09] is used to convert a sequence of dihedral angles to Cartesian coordinates of atoms and bonds. It basically takes any full atom representation of the desired length with arbitrary dihedral angles. The difference between these and the new dihedral angles is calculated and then imposed by several rotational and translational moves.

When calculating Cartesian coordinates from a given sequence of ϕ- and ψ-angles the unknown bond lengths and bond angles as well as the torsional ω-angle have to be guessed, see figures 1.1.1 on page 2 and 1.3.1 on page 6. However, compared to the variation in ϕ and ψ and the length of proteins, these values actually vary only very little and are assumed the be fixed. It is rather interesting to see how much error is introduced due to this approximation [EH91, HT04, BSDK09]. To investigate this for our implementation, the dihedral angles of known protein structures (the PDBSelect50 subset [PDBb]) are extracted and then the unknowns can be taken either from another, sufficiently long, protein structure (e.g. 2FHF), the literature (i.e. a polyalanine model constructed by USCF Chimera [PGH+04]) or from the known structures themselves, respectively. The comparison of the generated structures to the target structures reveals the introduced reconstruction error. The procedure is as follows for each structure in the test set:

1. extract the ϕ, ψ sequence from the known structure
2. set all ϕ and ψ angles to zero in a copy of the known structure, so that the structure becomes completely extended
3. do the same for 2FHF and the polyalanine and shorten the polymers to the size of the known structure
4. set the ϕ and ψ angles to the previously extracted values in the copied structure, 2FHF and the polyalanine
5. compare all three structures to the original unmodified target structure

2.2.3 Amino Acids

In this section reconstruction approaches for the discrete sequence terms are introduced. In analogy to the structural fragments the protein sequence is broken into overlapping fragments for which probability vectors are calculated (figure 2.2.1 page 20). From these, the original amino acid labels are reconstructed in order to get a feeling for the crudeness of the simplification due to the Bayesian classification. According to [SMT08b, CS96] the elements of the probability vector $\boldsymbol{v}_i = \big(v_{ij}\big)_{j\in[1,n]}$ for a sequence fragment $\boldsymbol{s}_i$ of length k are given by

$$v_{ij} = \mathrm{p}_{\boldsymbol{s}_i}(\boldsymbol{F} \sim \mathcal{C}_j) = \mathrm{p}(\boldsymbol{F} \sim \mathcal{C}_j | \boldsymbol{S} = \boldsymbol{s}_i) = \frac{w_j\,\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S} = \boldsymbol{s}_i)}{\sum\limits_{j'=1}^{n} w_{j'}\,\mathrm{p}_{\mathcal{C}_{j'}}(\boldsymbol{S} = \boldsymbol{s}_i)} \tag{2.2.8}$$

and can be interpreted as the n-dimensional vector of probabilities of all n classes $\mathcal{C}_j$ given the sequence $\boldsymbol{s}_i$. The prior class weights w_j are directly taken from the classification and $\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S} = \boldsymbol{s}_i) = \prod\limits_{t=1}^{k} \mathrm{p}_{\mathcal{C}_j}(S_t = s_{it})$ is the product of the 20-way Bernoulli probabilities. As amino acid labels are nominal features, averages or expectation values to construct the amino acid labels from the probability vectors (as done for dihedral angles) are not defined. One way to find a representative is to search for the class with the highest probability at each residue. This leads to the construction formula for the estimated amino acid label s_{it}^{est} at the tth residue of the sequence fragment $\boldsymbol{s}_i$, given by

$$s_{it}^{\mathrm{est}} = \arg\max_{a=1}^{20} \sum_{j=1}^{n} v_{ij}\,\mathrm{p}_{\mathcal{C}_j}(S_t = a), \tag{2.2.9}$$

where $\arg\max\limits_{a} \mathrm{f}(a) = a_{\max} : \mathrm{f}(a_{\max}) \geq \mathrm{f}(a)\ \forall a$ and the weighted sum forms the mixture distribution for amino acids.

However, a protein sequence of length l is represented by a set of $l-k+1$ probability vectors each describing an overlapping fragment $\boldsymbol{s}_i$ where $i \in [1, l-k+1]$. This raises the problem of combining the overlapping parts. There exists no sensible way to combine amino acid types, but the corresponding probabilities can be combined. As for the reconstruction of dihedral angles the same three probabilistic ways and one analytical-technical approach are investigated here for the same reasons as stated in section 2.2.1. They are given as extensions on formula (2.2.9) using the notation $\boldsymbol{s}^{\mathrm{est}} = (s_1^{\mathrm{est}}, \ldots, s_i^{\mathrm{est}}, \ldots, s_l^{\mathrm{est}})^{\mathsf{T}}$ for the estimated protein sequence. The linear weighted average over all classes stays for the geometric mean and the arithmetic mean approaches. However, the averaging of the overlapping parts is dropped as the absolute values are enough for finding the amino acid label with maximal probability.

2.2.3.1 Geometric Mean

Considering the overlapping parts as independent random variables leads to the geometric mean or product approach, which is taken be

$$s_i^{\text{est}} = \arg\max_{a=1}^{20} \prod_{i' \in \mathbb{I}_i} \sum_{j=1}^{n} v_{i'j}\, \mathrm{p}_{\mathcal{C}_j}(S_{i-i'+1} = a), \tag{2.2.10}$$

where $\mathbb{I}_i$ is the index set for a residue s_i of all overlapping fragments $\boldsymbol{s}_{i'}$, formally given by $\mathbb{I}_i = [\max\{1, i-k+1\}, \min\{l-k+1, i\}]$.

2.2.3.2 Arithmetic Mean

The arithmetic mean or sum approach is based on the idea that the overlapping parts are different models for the same random variable and is formulated by

$$s_i^{\text{est}} = \arg\max_{a=1}^{20} \sum_{i' \in \mathbb{I}_i} \sum_{j=1}^{n} v_{i'j}\, \mathrm{p}_{\mathcal{C}_j}(S_{i-i'+1} = a). \tag{2.2.11}$$

2.2.3.3 Maximum

The maximum approach for amino acids looks a little bit different to the maximum approach for dihedral angles, but it follows the same idea. Estimate the unknown label by taking the representative $\left(\arg\max_{a=1}^{20}\right)$ with the highest probability over all overlapping fragments $\left(\max_{i' \in \mathbb{I}_i}\right)$ and all classes $\left(\max_{j=1}^{n}\right)$. Formally written

$$s_i^{\text{est}} = \arg\max_{a=1}^{20} \max_{i' \in \mathbb{I}_i} \max_{j=1}^{n} v_{i'j}\, \mathrm{p}_{\mathcal{C}_j}(S_{i-i'+1} = a). \tag{2.2.12}$$

2.2.3.4 Inverse Calculation

The fourth rather technical approach to construct the amino acid sequence from probabilities uses marginal probability vectors, similar as in subsection 2.2.1.4 for dihedral angles. The idea is to find the amino acid that led to the probability values by inverse calculation. The unnormalised marginal probability vector for an amino acid sequence fragment $\boldsymbol{s}_i$ is defined by

$$\begin{aligned} {}^{\text{nonorm}}\boldsymbol{v}_i &= \left({}^{\text{nonorm}}v_{ijt}\right)_{j \in [1,n] \wedge t \in [1,k]} \\ &= \Big(w_j\, \mathrm{p}_{\mathcal{C}_j}(S_t = s_{it})\Big)_{j \in [1,n] \wedge t \in [1,k]}, \end{aligned} \tag{2.2.13}$$

where i is the index of the first residue in the fragment, k is the length of the fragment and n is the number of classes.

Let $j^{i,i'}_{\max} = \arg\max_{j=1}^{n} \frac{{}^{\text{nonorm}}v_{i'\,j\,i-i'+1}}{w_j}$ be the class having maximal probability for fragment i' at the mutual residue $i - i' + 1$ with fragment i. Then $i'_{\max} = \arg\max_{i' \in \mathbb{I}_i} \frac{{}^{\text{nonorm}}v_{i'\,j^{i,i'}_{\max}\,i-i'+1}}{w_j}$ is the overlapping fragment with the highest probability for that class. With this the amino acid label for the residue i can be estimated using marginal probability vectors by

$$s_i^{\text{est}} = \arg\max_{a=1}^{20} \mathrm{p}_{\mathcal{C}_{j^{i,i'_{\max}}_{\max}}} \big(S_{i-i'_{\max}+1} = a\big). \tag{2.2.14}$$

For results on these formulae for sequence reconstruction see section 2.3.3.

2.2.3.5 Substitution Matrix

The comparison of a substitution matrix calculated from a set of generated sequences to an established matrix like BLOSUM [HH92] summerises some interesting properties of our method. It shows which substitutions are under- or overrepresented and the correlation coefficient quantifies how close the sampled sequences are to being biologically relevant. In order to calculate a substitution matrix from a pool of generated sequences, the relative frequencies of aligned amino acids p_{ab} and the relative frequencies of seeing this cooccurrence by chance $p_a p_b$ have to be extracted from the sequence pool. According to [HH92] the matrix entries are then given by

$$s_{ab} = \frac{1}{\lambda} \log \left(\frac{p_{ab}}{p_a p_b} \right),$$

where λ is a scaling factor.

2.3 Results

A few classifications have been tested. Depending on the classification the results are different, but the trends stay the same. Therefore, the evaluation is restricted to a classification that did reasonably well in other projects, e.g. comparing proteins. It consists of 162 classes for fragments of length five. In this classification five Gaussian distributions cross the periodic boundary of the ψ angles (class mean ± deviation) at $-\frac{\pi}{2}$. No class crosses the ϕ boundary at 0 and no class spans more than one period.

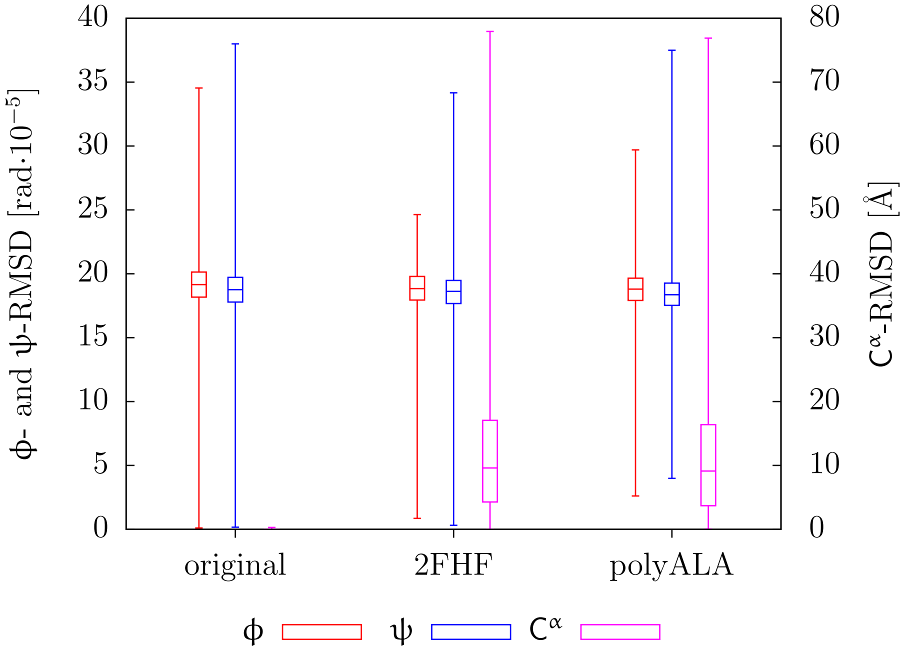


Figure 2.3.1: Reconstruction of all structures of the PDBSelect50 subset [PDBb] using their ϕ- and ψ-angles and either their original geometries, the geometry of 2FHF or a poly-Alanine model, respectively. The boxes indicate the 1st (25%), 2nd (median) and 3rd (75%) quartiles and the whiskers the minimum and maximum.

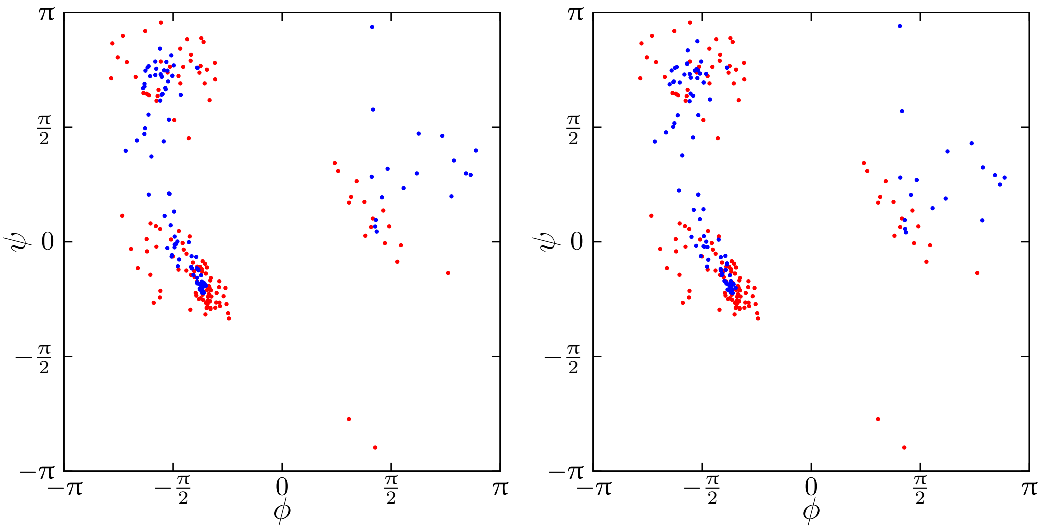
2.3.1 Influence of Bond Lengths and Angles

Figure 2.3.1 quantifies the influence of bond lengths and angles on the reconstructed coordinates of the PDBSelect50 subset using the original dihedral angles ϕ and ψ. As expected all modified structures with native geometry have no difference to the original structures, respectively, besides precision errors. The modified structures with the geometry of 2FHF have significantly large root mean squared deviations on the α-carbon positions (C^{α}-RMSD). The situation with the idealised geometry (polyalanine) does not look much different as seen in figure 2.3.1.

For a discussion on these issues see section 2.4.1 on page 34.

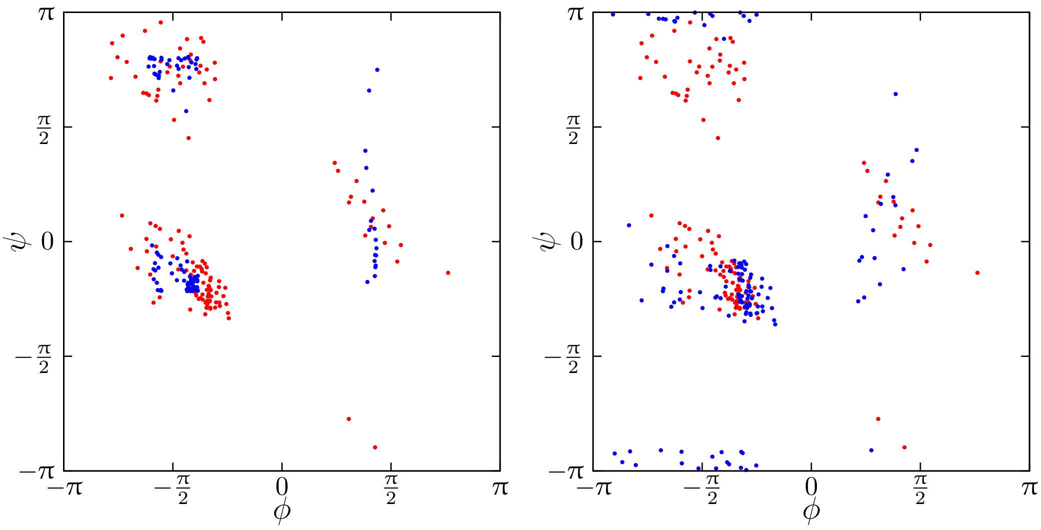
2.3.2 Dihedral Angle Reconstruction

In order to demonstrate the potential of the formulae introduced in section 2.2.1 the reconstructed dihedral angles are compared to the native values of a few example structures. Four approaches are tested altogether. Two approaches are probabilistic using joint probability vectors, namely the arithmetic mean (formula (2.2.2)) and the maximum approach (formula (2.2.3)). One probabilistic



(a) arithmetic mean on joint probabilities,
ϕ-RMSD: 0.422 (24.2°),
ψ-RMSD: 0.479 (27.5°),
C^{α}-RMSD: 17.773Å

(b) maximum on joint probabilities,
ϕ-RMSD: 0.421 (24.1°),
ψ-RMSD: 0.467 (26.8°),
C^{α}-RMSD: 27.406Å



(c) arithmetic mean on marginal probabilities with linearised weights,
ϕ-RMSD: 0.239 (13.7°),
ψ-RMSD: 0.283 (16.2°),
C^{α}-RMSD: 22.188Å

(d) inverse Gaussian on marginal probabilities with linearised weights,
ϕ-RMSD: 0.276 (15.8°),
ψ-RMSD: 0.539 (30.9°),
C^{α}-RMSD: 17.880Å

Figure 2.3.2: Four ϕψ plots of the dihedral angles of the protein 1AKI. Shown in red are the actual target values and in blue the values estimated by using an arithmetic mean (a), using a maximum of the overlapping parts (b), using marginal probabilities with an arithmetic mean of the overlapping parts and with linearised weights (c) and using marginal probabilities with the technical inverse Gaussian approach (d), respectively.

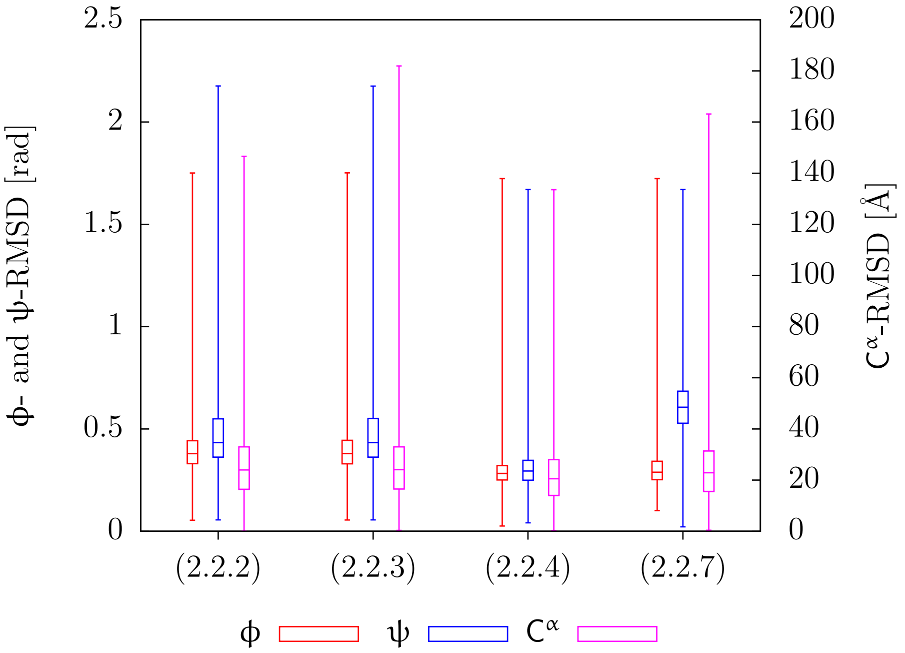


Figure 2.3.3: Performance of four reconstruction formulae tested on the PDBSelect50 subset [PDBb]. (2.2.2): arithmetic mean on joint probabilities, (2.2.3): maximum on joint probabilities, (2.2.4): arithmetic mean on marginal probabilities with linearised weights, (2.2.7): inverse Gaussian on marginal probabilities with linearised weights. The boxes indicate the 1st (25%), 2nd (median) and 3rd (75%) quartiles and the whiskers the minimum and maximum.

approach, the arithmetic mean, is also tested with marginal probabilities and linearised weights (formula (2.2.4) combined with formula (2.1.7)). And finally the fourth rather technical approach uses inverse Gaussians (formula (2.2.7)). Results for the other approaches are not shown, because either they take very long to obtain (integration in the geometric mean approaches) or they are not very different to the results presented here.

From the probability vectors for structure 1AKI the dihedral angles are reconstructed and compared to the original values. Figure 2.3.2 shows their distributions in scatter plots for each tested formula. In almost all four plots the generated angles fall into the three prominent regions, right-handed helical, strand and left-handed helical. The deviations to the original values seem to be equally high for the two probabilistic approaches using joint probability vectors. Although they differ in their C^α-RMSD, the dihedral angle difference between the arithmetic mean and maximum approach is only very little as seen in plots 2.3.2a and 2.3.2b.

The four approaches are also tested on the whole PDBSelect50 subset [PDBb]. The results are shown in figure 2.3.3. The arithmetic mean approach with marginal probabilities and linearised class weights yields the lowest dihedral angle

protein	**performance of formula** [%]			
sequence	(2.2.10)	(2.2.11)	(2.2.12)	(2.2.14)
1TU7A	38	38	33	76
1FC2C	42	42	40	79
1ENH_	33	33	37	74
4ICB_	54	55	49	82
1BDD_	40	40	35	82
1AKI_	36	35	27	72
2GB1_	41	43	36	91
1HDDC	33	32	35	72

Table 2.3.1: Reconstruction of a few example sequences. (2.2.10): geometric mean on joint probabilities, (2.2.11): arithmetic mean on joint probabilities, (2.2.12): maximum on joint probabilities, (2.2.14): inverse calculation on marginal probabilities.

RMSDs on average. Again the arithmetic mean and the maximum approach with joint probabilities perform equally well on this dataset. On average these two approaches also show no significant difference in C^{α}-RMSD.

In section 2.4.2 the dihedral angle reconstruction results are discussed.

2.3.3 Sequence Reconstruction

In order to demonstrate the potential of the statistical description of the protein sequence, the three probabilistic approaches from section 2.2.3, geometric mean (formula (2.2.10)), arithmetic mean (formula (2.2.11)) and maximum (formula (2.2.12)) with joint probabilities as well as the technical inverse calculation (formula (2.2.14)) with marginal probabilities are tested on a few example sequences and on the large PDBSelect50 subset [PDBb]. The calculated sequences are compared to the native sequences, see table 2.3.1 and figure 2.3.4. The similarity for the technical reconstruction approach (formula (2.2.14)) is highest for all sequences. The other, probabilistic approaches perform significantly worse, but give similar results compared to each other.

The average performance of reconstruction of the arithmetic mean approach (formula (2.2.11)) is 41% sequence identity (figure 2.3.4). A matrix calculated from the observed substitutions is shown in table 2.3.2 (lower triangle). The difference to the BLOSUM 40 matrix is shown in the upper part of the same table. The strong negative entries in this difference matrix originate from substitutions that were not observed for the arithmetic mean approach. The correlation coefficient is 0.15. The average reconstruction performance for the inverse calculation approach (formula (2.2.14)) is 93% sequence identity and the correlation coefficient of its substitution matrix and the BLOSUM 90 matrix is 0.5.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	
	-4	2	1	0	1	0	1	-3	2	-2	2	1	0	1	1	-1	0	2	1	-1	A
		-10	-3	0	-5	-5	1	2	-3	-5	2	-3	-7	-6	2	0	0	-6	-7	2	R
A	1		-8	-1	-6	-3	1	1	-3	-6	2	-1	-6	-5	0	-2	-2	-4	-6	2	N
R	0	-1		-8	2	0	-2	2	0	1	1	-1	1	3	1	0	1	4	2	1	D
N	0	-3	0		-23	-4	0	2	-3	-4	2	-5	-4	-6	3	0	0	-1	-3	2	C
D	-1	-1	1	1		-9	-2	1	-3	-5	2	-1	-7	-4	1	-2	-1	-7	-7	3	Q
C	-1	-8	-8	0	-7		-6	1	0	2	1	-1	1	2	-1	0	1	1	1	2	E
Q	0	-3	-2	-1	-8	-1		-7	2	1	0	1	0	1	-2	0	0	0	1	1	G
E	0	0	0	0	-2	0	1		-20	-5	2	0	-9	-6	0	0	0	4	-10	4	H
G	-2	-1	1	0	-1	-1	-2	1		-9	-2	-5	-9	-9	0	-1	-7	-5	-8	-4	I
H	0	-3	-2	0	-7	-3	0	0	-7		-6	2	-3	-2	2	3	1	1	0	-2	L
I	-3	-8	-8	-3	-8	-8	-2	-3	-8	-3		-5	-1	0	1	-1	-1	-1	-1	1	K
L	0	0	-1	-2	0	0	-1	-4	0	0	0		-14	-8	0	0	-2	-5	-9	-1	M
K	0	0	-1	-1	-8	0	0	-1	-1	-8	0	1		-17	2	-1	-7	-3	-12	0	F
M	-1	-8	-8	-2	-7	-8	-1	-2	-8	-8	0	-2	-7		-10	0	-1	3	1	1	P
F	-2	-8	-8	-1	-8	-8	-1	-2	-8	-8	0	-3	-8	-8		-4	-2	3	0	0	S
P	-1	-1	-2	-1	-2	-1	-1	-3	-2	-2	-2	0	-2	-2	1		-5	-2	-3	-5	T
S	0	-1	-1	0	-1	-1	0	0	-1	-3	0	-1	-2	-3	-1	1		-19	-5	3	W
T	0	-2	-2	0	-1	-2	0	-2	-2	-8	0	-1	-3	-8	-1	0	1		-17	1	Y
W	-1	-8	-8	-1	-7	-8	-1	-2	-1	-8	0	-3	-7	-2	-1	-2	-8	0		-4	V
Y	-1	-8	-8	-1	-7	-8	-1	-2	-8	-8	0	-2	-8	-8	-2	-2	-3	-2	-8		
V	-1	0	-1	-2	0	0	-1	-3	0	0	0	-1	0	0	-2	-1	0	0	0	1	
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	

Table 2.3.2: Substitution matrix for the arithmetic mean approach, formula (2.2.11), (lower triangle) and difference matrix (upper triangle) obtained by subtracting the BLOSUM 40 matrix [HH92] position by position.

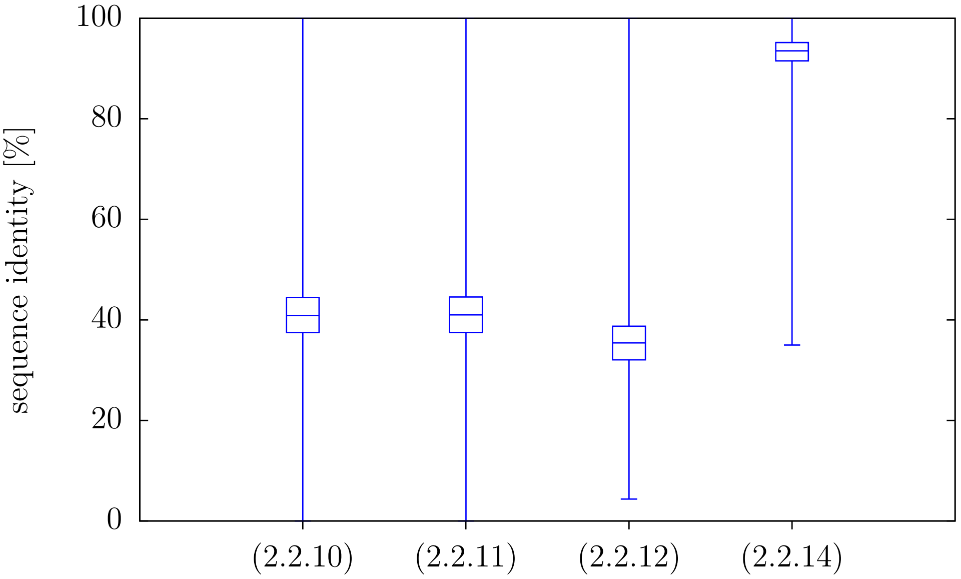


Figure 2.3.4: Sequence reconstruction of the PDBSelect50 subset [PDBb] using four different reconstruction formulae, respectively. (2.2.10): geometric mean on joint probabilities, (2.2.11): arithmetic mean on joint probabilities, (2.2.12): maximum on joint probabilities, (2.2.14): inverse calculation on marginal probabilities. The boxes indicate the 1st (25%), 2nd (median) and 3rd (75%) quartiles and the whiskers the minimum and maximum.

In section 2.4.3 the sequence reconstruction results are discussed.

2.4 Discussion

2.4.1 Influence of Bond Lengths and Angles

The influence of the bond lengths and angles on the quality of the results is quite strong. Using literature values for them leads to rather large C^α-RMSDs. There seems to be a significantly high variation [BSDK09] and one should consider using amino acid specific bond angles and lengths. This might slightly improve the results.

Despite the errors introduced by bond angles and lengths, the dihedral backbone angle ω is known to vary by a few degrees and completely points to opposite directions for Proline residues (cis-conformation of the peptide bond). In order to minimise error originating here the ω-angles should be set at least dependent on the amino acid type.

2.4.2 Dihedral Angle Reconstruction

Here, a few properties of the dihedral angle reconstruction approaches are discussed. It is important to note that, besides the approximations caused by using standard bond geometries, a few wrong dihedral angles may lead to high C^{α}-RMSDs. In figure 2.3.2 the dihedral angles RMSDs and the C^{α}-RMSDs seem not to correlate with each other. Another principle observation concerning the C^{α}-RMSD is that it may get incredibly high, up to 180Å in figure 2.3.3. With high evidence this is due to bad structures. The WURST library used can only deal with single-chained proteins without gaps [TPH04]. If these structures were filtered out, the generated structure model will tend to have lower C^{α}-RMSDs.

The difference between the performance of the arithmetic mean and the maximum approaches on joint probabilities is only very little. This means that each fragment has a high preference for one single class, as the averaging in the arithmetic mean approach (formula (2.2.2)) does not change the results significantly. This finding is consistent with other work [Hof07], where the probability vectors could be compressed using only very few classes in order to build structural alphabets. It also suggests that the classes cover the data quiet well. This does not mean there are no weird angles in the data, but they rather lie in between classes and are covered by a mixture of a few moderately populated classes.

Surprisingly, the inverse Gaussian approach does not lead to an exact reconstruction, although this is expected by construction of the formulae. Checking very carefully the numbers, it turns out, that if the class means are very close to the original value, then the angle can be reconstructed quite reliable. However, most class means are significantly far away from the original angle leading to strong precision errors, i.e. the values for one angle deviate quite a lot. Averaging over these numbers necessarily results in badly reconstructed angles. Maybe replacing the average by a weighted average, e.g. using the class weights, would improve the reconstruction.

The other approach, the arithmetic mean on marginal probabilities with linearised weights, is also exact by construction but seems to have similar problems. However, there are additional issues to be considered. The original angles often lie close to the boundaries. These angles are smaller (or larger) than any class mean. Taking a weighted average between any two class means would not lead to the desired angle. Even using periodic images would not help here, as the class weights were calculated within a single period only. Calculating class weights of two periods would help here. However, an even better solution would be to change the classification model to use periodic probability distributions, like the von-Mises models. This was not done in this work in order to use existing classifications built for other projects [SMT08b, MST09]. Extending this would require a lot of careful programming, but is strongly recommended for future projects.

The exact approaches are interesting for reconstruction, but unfortunately are unsuitable for structure prediction where the probability vectors come from sequence. There the assumption that the class weights may be interpreted to reflect some sort of distance of the wanted angle to the class means does not hold. The weights should rather be regarded as class probabilities. For example, if two classes have positive probability, then the angle does not lie only in between the class means but is likely to be found anywhere around the class means with respect to their density distributions.

Taking the arithmetic mean (formulae (2.2.2) and (2.2.4)) of the overlapping parts is theoretically and practically the most reasonable approach. The overlapping fragments are treated effectively as a random variable of different stochastic models for the dihedral angles. Each expression has equal probability. But this could be changed easily, see also final discussion chapter 6, page 87. Considering the use of marginal probability vectors results in lower RMSD values here, however the loss of correlated features leads to limited applicability for sampling approaches (section 4.1.2.1, page 52).

2.4.3 Sequence Reconstruction

Using the pairwise sequence identity to the native sequence as a quality measure for the reconstructed sequences leads to rather conservative numbers. The numbers are lower than expected from a biological point of view. This measure underestimates the biochemical similarity of the sequences. Instead of the plain identity, a similarity score based on substitution matrices like PAM or BLOSUM would better reflect the biological relevance of the generated sequences. However, choosing the correct matrix is dependent on the evolutionary distance between the sequences. But, the true (substitutional/mutational) distance between sequences folding to similar structures was not investigated here. Using the average identity of the generated sequences as an approximation of the evolutionary distance would not help, as it would not lead to a measure telling us how realistic the generated sequences are.

The best way to check the relevance of the generated sequences would be to synthesise them and determine their structure experimentally, but this is not feasible. However, on the computer sets of native sequences from the same structure cluster (fold) could be analysed in order to see if the generated sequences can be found there. For each fold a sequence profile could be built, which could serve as an estimate of the true variation. Subsequently, this profile could be compared to our substitution matrices.

The low correlation between our substitution matrices and the established matrices suggests that there is only very little similarity in the substitution patterns and the potential of the classification is limited for some affected amino acids. Comparing the substitution matrices from our reconstruction formulae to the BLOSUM 40 or BLOSUM 90 matrices is actually not perfect. The BLOSUM x matrices are generated from a pool of sequence alignments with less than $x\%$ sequence identity. As the sequence identity of our generated sequences spans a range from 0 to 100%, the observed substitutions actually should be compared to either the BLOSUM 100 matrix or, all sequences with more than $x\%$ identity should be clustered first and then compared to BLOSUM x. Additionally, there is also the problem of scaling. To avoid that, a rank correlation coefficient could be calculated. But overall the values would roughly stay the same, since these numbers are dependent on the scoring, which is intentionally kept very simple. Our scoring is solely based on structural features, whereas BLOSUM also considers effects of evolution implicitly. Nevertheless, it would be nice to have a measure for how close the pool of generated sequences is to the natural pool of sequences folding to similar structures. This would become also relevant in the light of more sophisticated scoring functions.

A median of 93% sequence identity for the set of sequences generated by the technical approach (formula (2.2.14)) can be considered the limit of what is possible to reconstruct using the classification. It is clear, that a classification is always a simplification. Here that means, that some amino acid sequences are so similar, that they form a motif in the classification and, therefore, can no longer be distinguished or reconstructed. This has also consequences for the construction and comparison of substitution matrices. Some substitutions are put together already in the classification and therefore can not be observed in the substitution matrix. Of course, this further lowers the correlation coefficients to the established matrices.

The technical approach yields the highest reconstruction rate. However, this has no meaning if the probabilities come from a protein structure. This will become relevant for sequence prediction (chapter 5). Instead, the probabilistic approaches have the advantage, that they can also be used sensibly with probability vectors created from structure. From these the arithmetic mean approach (formula (2.2.11)) seems to be the best choice for the sequence prediction task. Its calculation is simple and it can be justified most rigorously in statistical terms. The overlapping parts are effectively treated as statistical variables that can take different mixture models as values.

Chapter 3

Self-consistent mean field optimisation

In this chapter an innovative optimisation method for finding self-consistent states introduced for systems that are described by a probabilistic mean-field model [ST]. It is especially suited, but not limited to our Bayesian classification based on overlapping protein fragments. Predicting unknown features such as structure or sequence basically means to optimise the population of their states. The method described here is applied in the following chapters and allows to efficiently explore the conformational or compositional space of proteins.

Self-consistent mean field (SCMF) methods [KD96] have traditionally been used to optimise wave functions [Sto05] and have been applied to a variety of problems [HKL$^+$98, MBCS01, MM03, Edw65, Dew93, RFRO96, KD98, SSG$^+$00, DK97, CdMaT00, XJR03]. In all these applications, the system is subdivided into small subsystems which can interact through a mean field. One assumes the system to be in all states at the same time and iteratively updates the contribution of each state to the mean energy field through the Boltzmann relation. This process consists of alternating steps of calculating the mean energy of a subsystem and its interacting subsystems, updating the probabilities of each state of the subsystem, recalculating the energy and so on. These steps are iterated over all subsystems until a self-consistent state of the whole system is found. This state is reached when the probabilities of the states of the subsystems converge, i.e. they no longer change.

In order to describe the mean field previous studies invented energy functions, which are more or less directed by human preconceptions. Especially the selection and relative weighting of terms is an optimisation task in itself. In contrast, this study introduces a purely statistical version of SCMF using rather simple scoring terms. It uses the idea of overlapping subsystems and works with conditional probabilities directly without using the Boltzmann relation.

First, the standard procedure is described and some basic notation is introduced. Then, a description of the purely statistical version follows. Finally, a new cooling scheme is introduced. It adaptively lowers the entropy of the system via a temperature-like convergence parameter. Applied to our statistical SCMF version this leads to a method that smoothly narrows down the solution space.

3.1 Standard SCMF

Self-consistent mean field methods aim to find the state of lowest energy. The standard method assumes a conventionally defined energy function and is therefore extended in this work (section 3.2). It is described here for comparison. The system $\boldsymbol{\mathfrak{X}}$ is divided into small disjoint subsystems $\mathfrak{X}_i$. Each subsystem is considered to be in all possible states $\mathbb{S}_i = \{\mathfrak{x}_i\}$ at once with a certain probability $\mathrm{p}(\mathfrak{X}_i = \mathfrak{x}_i)$. This probability is adapted via an iterative procedure until the system converges to a self-consistent state, i.e. the probabilities no longer change. The energy function that one seeks to minimise is taken to be

$$\mathrm{E}_{\text{eff}}(\boldsymbol{\mathfrak{X}}, \mathrm{p}) = \sum_{i} \sum_{\mathfrak{x}_i \in \mathbb{S}_i} \mathrm{p}(\mathfrak{X}_i = \mathfrak{x}_i)\, \mathrm{E}_{\mathfrak{X}_i}(\mathfrak{x}_i).$$

Each subsystem $\mathfrak{X}_i$ in state $\mathfrak{x}_i$ feels the influence of its surrounding subsystems $\mathfrak{X}_{i'}$ in a mean field, given by the mean interaction energy

$$\mathrm{E}_{\mathfrak{X}_i}(\mathfrak{x}_i) = \sum_{i' \neq i} \sum_{\mathfrak{x}_{i'} \in \mathbb{S}_{i'}} \mathrm{p}(\mathfrak{X}_{i'} = \mathfrak{x}_{i'})\, \mathrm{E}_{\mathfrak{X}_i \mathfrak{X}_{i'}}(\mathfrak{x}_i, \mathfrak{x}_{i'}),$$

where $\mathrm{E}_{\mathfrak{X}_i \mathfrak{X}_{i'}}(\mathfrak{x}_i, \mathfrak{x}_{i'})$ is the pairwise interaction of $\mathfrak{X}_i$ being in state $\mathfrak{x}_i$ and $\mathfrak{X}_{i'}$ being in state $\mathfrak{x}_{i'}$. Ideally, this interaction should be calculated among all subsystems. However, in practice, either the energy function is not defined for long-range interactions or the calculation will lead to a combinatorial explosion. Therefore, the mean interaction energy is typically calculated on some subset $\{\mathfrak{X}_{i'} \mid i' \in \mathbb{O}_i\}$ of closely interacting subsystems, leading to

$$\mathrm{E}_{\mathfrak{X}_i}(\mathfrak{x}_i) = \sum_{\substack{i' \in \mathbb{O}_i \\ i' \neq i}} \sum_{\mathfrak{x}_{i'} \in \mathbb{S}_{i'}} \mathrm{p}(\mathfrak{X}_{i'} = \mathfrak{x}_{i'})\, \mathrm{E}_{\mathfrak{X}_i \mathfrak{X}_{i'}}(\mathfrak{x}_i, \mathfrak{x}_{i'}). \tag{3.1.1}$$

In each iteration step the mean interaction energy of each subsystem in each particular state and the mean field of its surrounding subsystems is calculated and then turned into probabilities using the Boltzmann relation, given by

$$\mathrm{p}(\mathfrak{X}_i = \mathfrak{x}_i) = \frac{\exp\left(-\beta\, \mathrm{E}_{\mathfrak{X}_i}(\mathfrak{x}_i)\right)}{\sum\limits_{\mathfrak{x}'_i \in \mathbb{S}_i} \exp\left(-\beta\, \mathrm{E}_{\mathfrak{X}_i}(\mathfrak{x}'_i)\right)}, \tag{3.1.2}$$

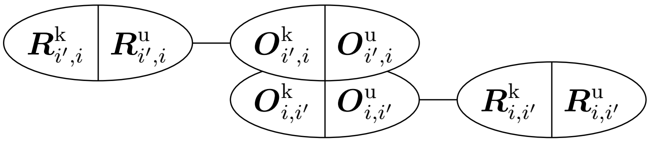

$\boldsymbol{R}^{\mathrm{k}}_{i',i}$
$\boldsymbol{R}^{\mathrm{u}}_{i',i}$
$\boldsymbol{O}^{\mathrm{k}}_{i',i}$
$\boldsymbol{O}^{\mathrm{u}}_{i',i}$
$\boldsymbol{O}^{\mathrm{k}}_{i,i'}$
$\boldsymbol{O}^{\mathrm{u}}_{i,i'}$
$\boldsymbol{R}^{\mathrm{k}}_{i,i'}$
$\boldsymbol{R}^{\mathrm{u}}_{i,i'}$

Figure 3.2.1: Notation for two overlapping systems $\boldsymbol{\mathfrak{X}}_{i'} = \left(\boldsymbol{R}^{\mathrm{k}}_{i',i}, \boldsymbol{R}^{\mathrm{u}}_{i',i}, \boldsymbol{O}^{\mathrm{k}}_{i',i}, \boldsymbol{O}^{\mathrm{u}}_{i',i}\right)$ and $\boldsymbol{\mathfrak{X}}_i = \left(\boldsymbol{O}^{\mathrm{k}}_{i,i'}, \boldsymbol{O}^{\mathrm{u}}_{i,i'}, \boldsymbol{R}^{\mathrm{k}}_{i,i'}, \boldsymbol{R}^{\mathrm{u}}_{i,i'}\right)$.

where β is the inverse, Boltzmann-weighted temperature. In order to distinguish from a single iteration step, a loop over all subsystems and all states is called a simulation step.

Ideally, the procedure finds a single self-consistent state of the system after several simulation steps. In general, the system will find itself in a number of states. In order to decrease the number of possible states, one can gradually lower the temperature of the system like in simulated annealing [KGV83], see section 3.3.

3.2 Purely statistical SCMF

Instead of using some physical or knowledge-based energy function a purely statistical score function based on conditional probabilities is applied. This allows one to quickly sample the state space without the Boltzmann formalism. In order to achieve this, the standard SCMF protocol is modified. The system is divided into overlapping subsystems, i.e. parts of a subsystem are also modelled in other subsystems. As a consequence, this overlap enables the use of conditional probabilities. That means, the effect that a subsystem has on another can be modelled statistically rather then by some (sometimes arbitrarily simplified) interaction energy.

Let us assume statistics on subsystems of systems where the states are known have been collected by treating the subsystems independently and let there be enough statistical data to sufficiently cover the population. Then, the probability of a subsystem being in a specific state can be estimated by fitting its relative frequency to some function p. Let $\boldsymbol{\mathfrak{X}}_i$ and $\boldsymbol{\mathfrak{X}}_{i'}$ be the two vectors describing the two subsystems that include sites $\mathfrak{X}_i$ and $\mathfrak{X}_{i'}$ as the first dimensions, respectively (assuming sorted dimensions), and let $\boldsymbol{\mathfrak{X}}_{i'}$ overlap with $\boldsymbol{\mathfrak{X}}_i$. Denote the overlapping parts with $\boldsymbol{O}_{i,i'}$ and $\boldsymbol{O}_{i',i}$ and the rest of the subsystems with $\boldsymbol{R}_{i,i'}$ and $\boldsymbol{R}_{i',i}$, respectively. Furthermore, assume some observation has been made on an unknown system, then the subsystems can be further split into known and unknown variables, denoted by $\boldsymbol{\mathfrak{X}}^{\mathrm{k}}$ and $\boldsymbol{\mathfrak{X}}^{\mathrm{u}}$, respectively (see figure 3.2.1). The current probability $\mathrm{p}_{\mathrm{cur}}\left(\mathfrak{X}^{\mathrm{u}}_i = \mathfrak{x}^{\mathrm{u}}_i \,\middle|\, \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\right)$ of the unknown variables of a single site $\mathfrak{X}^{\mathrm{u}}_i$ being in state $\mathfrak{x}^{\mathrm{u}}_i$ is effected by all subsystems overlapping with the subsystem $\boldsymbol{\mathfrak{X}}_i$, given by the index set $\mathbb{O}_i$. This effect can be captured by

$$\mathrm{p}_{\mathrm{cur}}\big(\mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}} \big| \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\big) = \frac{\displaystyle\prod_{\substack{i' \in \mathbb{O}_i \\ i' \neq i}} \prod_{\mathfrak{x}_{i'} \in \mathbb{S}_{i'}} \Big(p^{i,i'}_{\mathfrak{x}_i,\mathfrak{x}_{i'}}\Big)^{\frac{2}{T}\,\mathrm{p}_{\mathrm{old}}\left(\mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}} \,|\, \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\right)}}{\displaystyle\sum_{\mathfrak{x} \in \mathbb{S}_i} \prod_{\substack{i' \in \mathbb{O}_i \\ i' \neq i}} \prod_{\mathfrak{x}_{i'} \in \mathbb{S}_{i'}} \Big(p^{i,i'}_{\mathfrak{x},\mathfrak{x}_{i'}}\Big)^{\frac{2}{T}\,\mathrm{p}_{\mathrm{old}}\left(\mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}} \,|\, \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\right)}} \tag{3.2.1}$$

where the interaction term $p^{i,i'}_{\mathfrak{x}_i,\mathfrak{x}_{i'}}$ reflects the net probability contribution of the unknown pair $\mathfrak{x}_i^{\mathrm{u}}, \mathfrak{x}_{i'}^{\mathrm{u}}$ on the average or marginal probability of the known pair $\mathfrak{x}_i^{\mathrm{k}}, \mathfrak{x}_{i'}^{\mathrm{k}}$. In analogy to [Sip90] this is taken to be

$$p^{i,i'}_{\mathfrak{x}_i,\mathfrak{x}_{i'}} \;=\; \frac{\mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}},\, \mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}} \big| \mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}},\, \mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}}\big)}{\mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}},\, \mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}}\big)},$$

with the marginal probability

$$\begin{aligned}
\mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}},\, \mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}}\big) &= \mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}} \big| \mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}}\big)\, \mathrm{p}\big(\mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}}\big) \\
&= \frac{1}{|\mathbb{I}_i|} \sum_{i'' \in \mathbb{I}_i} \mathrm{p}\left(O^{\mathrm{k}}_{i'',i_1} = \mathfrak{x}_i^{\mathrm{k}} \big| O^{\mathrm{k}}_{i'',i'_1} = \mathfrak{x}_{i'}^{\mathrm{k}}\right) \\
&\quad \cdot \frac{1}{|\mathbb{I}_{i'}|} \sum_{i'' \in \mathbb{I}_{i'}} \mathrm{p}\Big(O^{\mathrm{k}}_{i'',i'_1} = \mathfrak{x}_{i'}^{\mathrm{k}}\Big),
\end{aligned}$$

and the conditional probability

$$\begin{aligned}
&\mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}},\, \mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}} \big| \mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}},\, \mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}}\big) \\
&\quad= \mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}} \big| \mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}},\, \mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}},\, \mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}}\big) \\
&\qquad \cdot \mathrm{p}\big(\mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}} \big| \mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}},\, \mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}}\big) \\
&\quad= \mathrm{p}\big(\mathfrak{X}_i^{\mathrm{k}} = \mathfrak{x}_i^{\mathrm{k}} \big| \mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}},\, \mathfrak{X}_{i'} = \mathfrak{x}_{i'}\big) \\
&\qquad \cdot \mathrm{p}\big(\mathfrak{X}_{i'}^{\mathrm{k}} = \mathfrak{x}_{i'}^{\mathrm{k}} \big| \mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}},\, \mathfrak{X}_{i'}^{\mathrm{u}} = \mathfrak{x}_{i'}^{\mathrm{u}}\big) \\
&\quad= \frac{1}{|\mathbb{I}_i|} \sum_{i'' \in \mathbb{I}_i} \mathrm{p}\big(O^{\mathrm{k}}_{i'',i_1} = \mathfrak{x}_i^{\mathrm{k}} \big| O^{\mathrm{u}}_{i'',i_1} = \mathfrak{x}_i^{\mathrm{u}},\, O_{i'',i'_1} = \mathfrak{x}_{i'}\big) \\
&\qquad \cdot \frac{1}{|\mathbb{I}_{i'}|} \sum_{i'' \in \mathbb{I}_{i'}} \mathrm{p}\big(O^{\mathrm{k}}_{i'',i'_1} = \mathfrak{x}_{i'}^{\mathrm{k}} \big| O^{\mathrm{u}}_{i'',i_1} = \mathfrak{x}_i^{\mathrm{u}},\, O^{\mathrm{u}}_{i'',i'_1} = \mathfrak{x}_{i'}^{\mathrm{u}}\big).
\end{aligned}$$

$\mathbb{I}_i$ is the index set of the subsystems modelling the site $\mathfrak{X}_i$. A derivation from formulae (3.1.1) and (3.1.2) is trivial if setting

$$\mathrm{E}_{\mathfrak{X}_i\mathfrak{X}_{i'}}(\mathfrak{x}_i, \mathfrak{x}_{i'}) = -\beta^{-1} \ln p^{i,i'}_{\mathfrak{x}_i,\mathfrak{x}_{i'}}. \tag{3.2.2}$$

In order to control the entropy of the system a convergence parameter T is introduced, which allows one to simulate cooling effects, just like temperature in simulated annealing [KGV83]. At high T, the probabilities are spread out. As T goes down, the probabilities become more concentrated in the likelier states and probabilities of less populated states will tend to zero, so that the system will end up in the most populated states (see chapter 3.3).

The response of a system is not instantaneous. Changes in one part of the system take some iterations to reach other parts of the system. This can lead to oscillations in the probabilities. To avoid this problem, the state probabilities are updated slowly using a memory factor $\lambda \in [0, 1]$. This leads to

$$\mathrm{p}_{\mathrm{new}}\big(\mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}} \big| \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\big) = \lambda\, \mathrm{p}_{\mathrm{old}}\big(\mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}} \big| \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\big) + (1-\lambda)\, \mathrm{p}_{\mathrm{cur}}\big(\mathfrak{X}_i^{\mathrm{u}} = \mathfrak{x}_i^{\mathrm{u}} \big| \boldsymbol{\mathfrak{X}}^{\mathrm{k}}\big)\,. \quad (3.2.3)$$

Iterative application of formulae (3.2.1) and (3.2.3) will populate the sites at states that are consistent with the states at the modelling subsystems, just like in standard SCMF. Interestingly, in contrast to [DK97, Sip90] the method does not need the Boltzmann relation and does not assume the underlying data to be sampled from some well-defined statistical ensemble.

3.3 Simulated Annealing

Simulated Annealing is a general optimisation method, introduced in [KGV83]. The name comes from the process of "annealing" or slowly cooling a material so that particles will have time to find their optimal positions. According to the techniques for forming single crystals of solid state bodies, simulated annealing tries to find the state of lowest energy. In order to form a unimorphous solid state body the solid is melt at a high temperature and then slowly cooled down. In the first step the crystal bonds are broken and the amorphous material is put to a state where the particles are dissolved away from their positions, i.e. they can freely move around. In the next step the mono-crystal is formed by gradually lowering the temperature thereby avoiding local energy minima (i.e. amorphous bonding). These ideas are applied to other, simulated systems to find states of lowest energy by avoiding local minima and overcoming energy barriers.

It is clear for such a task, that the optimal annealing strategy is highly dependent on the properties and behaviour of the system. However, linear and exponential cooling schemes have been reported sometimes successful [NA98]. The reason for a success of the exponential scheme is the assumption that the system first (at high temperature) finds itself in a broad valley of the energy landscape which includes the lowest energy state and where it can move around freely. Then, when the temperature is lowered, local minima play an increasing role and the temperature is lowered slower so to allow time for finding the global optimum. However, if the system undergoes phase transitions, it will behave critically for certain temperatures. In the vicinity of such critical points the relaxation times

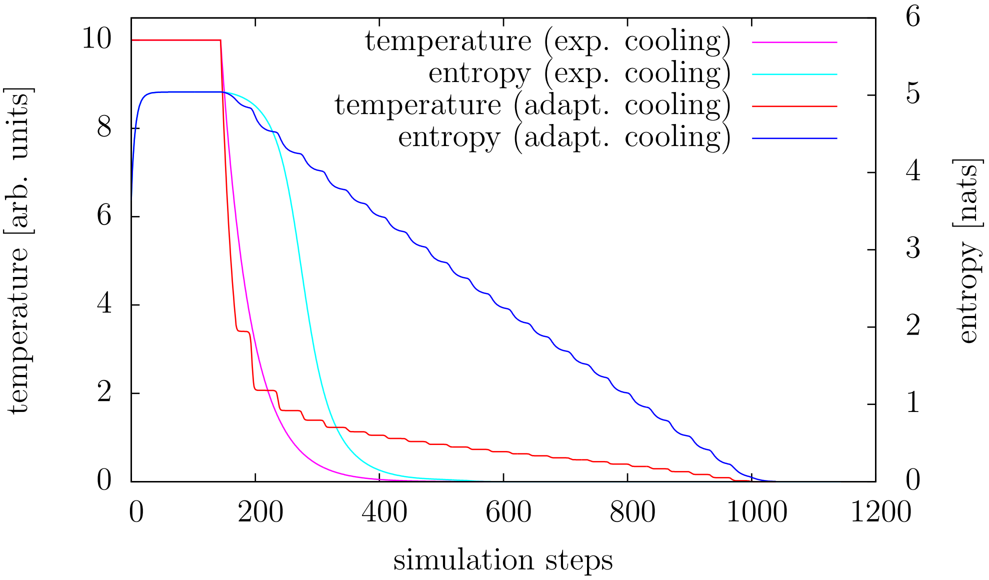


Figure 3.3.1: Exponential and adaptive cooling.

become very long [Mur03]. But at other temperatures the system might behave nicely and could be cooled faster. Therefore, the optimal strategy would be to use a cooling scheme, that adapts to the behaviour of the system.

A number of adaptive cooling strategies have been proposed [TH95, AG96, SP10]. It is necessary to have a quantity that reflects the system's behaviour. Here, an innovative adaptive cooling procedure is introduced (algorithm 1). It relies on an entropy-like measure which appropriately captures the diversification of the system. When the entropy of the system changes dramatically on a temperature change, the system could not adapt fast enough to the new temperature, probably because it is undergoing a critical transition. In order to work well, the adaptive cooling scheme has to detect the transition early enough to slow down the cooling and to allow extra time for the relaxation of the system. As the instantaneous entropy S can vary largely due to random spontaneous changes of the system, a long-time average S_{long} and a short-time average S_{short} is calculated. When the temperature is lowered, the entropy is expected to go down on average and $S_{\text{long}} > S_{\text{short}}$. For a nicely behaving system the difference $\Delta S = S_{\text{long}} - S_{\text{short}}$ should be a positive constant. If ΔS varies, then the system is no longer at equilibrium and the temperature has to be adapted. For an increasing ΔS the system might enter a critical phase and has to be cooled slower, whereas for a decreasing ΔS the system responds better to the new temperature and can be cooled faster.

The entropy-like measure can be calculated easily for the SCMF methods, given

```
$T \leftarrow T_{\text{start}}$
relax the system
calculate $S$ {instantaneous entropy}
$t_{\max} \leftarrow$ desired number of steps
$m \leftarrow \frac{S}{t_{\max}}$ {desired slope}
$\beta_{\text{long}} \leftarrow 0.9$, $\beta_{\text{short}} \leftarrow 0.5$
$\Delta S_{\text{thresh}} \leftarrow m \frac{\left(\beta_{\text{long}}\beta_{\text{short}} - \beta_{\text{long}} - \beta_{\text{short}} + 1\right)\left(\beta_{\text{long}} - \beta_{\text{short}}\right)}{\left(\beta_{\text{long}} - 1\right)^2 (\beta_{\text{short}} - 1)^2}$
$S_{\text{long}} \leftarrow S$, $S_{\text{short}} \leftarrow S$
$k \leftarrow \left(\frac{T_{\text{final}}}{T_{\text{start}}}\right)^{\frac{1}{t_{\max}}}$
$k_{\min} \leftarrow 0.8$
$\Delta S_{\text{old}} \leftarrow 0$
while $T > T_{\text{final}}$ do
    apply the system to $T$
    calculate $S$ {instantaneous entropy}
    $S_{\text{long}} \leftarrow \beta_{\text{long}} S_{\text{long}} + (1 - \beta_{\text{long}}) S$
    $S_{\text{short}} \leftarrow \beta_{\text{short}} S_{\text{short}} + (1 - \beta_{\text{short}}) S$
    $\Delta S \leftarrow S_{\text{long}} - S_{\text{short}}$
    if $\Delta S > \Delta S_{\text{thresh}} \wedge \Delta S \geq \Delta S_{\text{old}}$ then {cool slower}
        $k \leftarrow \sqrt{k}$
    else if $\Delta S < \Delta S_{\text{thresh}} \wedge \Delta S \leq \Delta S_{\text{old}}$ then {cool faster}
        if $k \geq 1.0 - \varepsilon$ then
            $k \leftarrow k - \varepsilon$
        else if $k > k_{\min}$ then
            $k \leftarrow k^2$
        end if
    end if
    $T \leftarrow kT$
    $\Delta S_{\text{old}} \leftarrow \Delta S$
end while
```

Algorithm 1: Adaptive cooling scheme based on entropy.

by

$$S = \frac{1}{N} \sum_{i=1}^{N} \sum_{\mathfrak{x}_i \in \mathbb{S}_i} \mathrm{p}(\mathfrak{X}_i = \mathfrak{x}_i) \ln \mathrm{p}(\mathfrak{X}_i = \mathfrak{x}_i) \tag{3.3.1}$$

where N is the number of (disjoint) subsystems. Here, line 2 of algorithm 1 means to iterate the update formula (3.2.3) over the entire system with a fixed starting temperature until the probabilities no longer change (i.e. after several simulation steps), whereas in line 13 only one iteration over the whole system is necessary (i.e. one simulation step). The cooling scheme assumes a linear decay to be the ideal entropy curve (see also appendix B, page 99, for a derivation of ΔS_{thresh}, line 7). The algorithm also avoids the overshooting of adaption by remembering the previous difference in entropy averages (2nd condition of lines 18 and 20). This is not essential, but it allows the algorithm to adapt faster.

Figure 3.3.1 shows two short simulations with exponential and adaptive cooling applied to protein structure prediction (see chapter 4). At the start the system is equilibrated at a high temperature. Then the two simulations show different behaviour. If the system is cooled exponentially, the entropy follows a rather steep slope. This may result in suboptimal solutions. Whereas, if the cooling adapts to the system's behaviour, the entropy can be forced to follow a linear slope on average. This allows to cool the system faster in uncritical situations while avoiding steep decays at the same time.

As an outlook, one should rigorously find out which slope would be the ideal slope. Here, a linear decay of entropy was assumed to perfectly compromise between fast and careful cooling. This means that linear cooling would be adequate if the space of simulation steps (i.e. temperature space) and the space of the entropy form an isomorphism. The adaptive cooling scheme ensures this by mapping the temperature space to a space that is isomorphous to the entropy. It remains to be shown how important an ideal slope is and what impact a different entropy slope has on the quality of the results. Currently with the simple scoring this impact can not be seen, but would become relevant for more subtle scoring functions.

Chapter 4

Structure Prediction

One of the aims of this work is to see how much can be predicted with a local, purely statistical description of proteins. As seen in chapter 2, there are certain limitations of our statistical classification even with a full structural description. Keeping those in mind, a rigorous way is shown here for structure prediction from sequence alone using our probabilistic scoring (section 1.3) and optimisation scheme (chapter 3).

Typically, protein structure prediction means to have a protein sequence where the three-dimensional structure is unknown and therefore this structure is to be predicted from the sequence. For many proteins, homologous structures have been solved and may serve as templates [EWMR$^+$06]. In recent years fragment-based approaches were the most successful predictors [BDNBP$^+$09]. Although definitely an improvement to previous approaches, their success is limited by finding reliable fragment templates. However, often enough no reliable template is known and modelling from scratch becomes necessary. In *ab initio* prediction methods, the amino acid sequence and a scoring function is the only prior knowledge used. These methods try to model the protein structure without explicitly using templates from known structures. Here, an innovative optimisation method for narrowing down the conformational space in order to enable rapid sampling of structures is introduced. Our approach is not intended to be a full blown structure predictor, but highlights elegant properties of the optimisation scheme using solely descriptive statistics with minimal preconceptions.

Similar to the more successful literature prediction approaches, our classification model is based on small protein fragments as well. With this model the question: "How probable is a structure fragment of dihedral angles given the sequence fragment of amino acid types?" can be answered. As different sequence fragments prefer different structural conformations, a sequence fragment introduces a local bias towards the allowed conformational space of the whole protein struc-

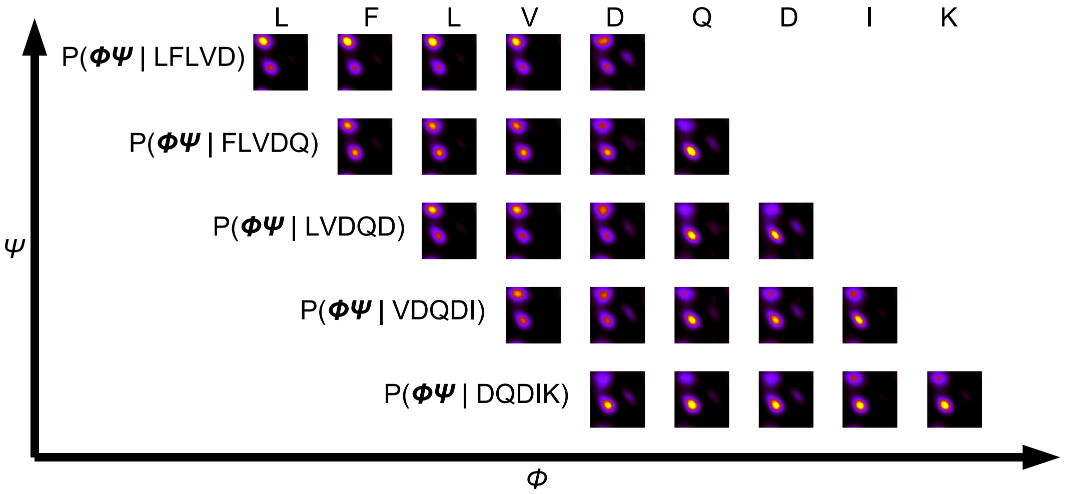


Figure 4.0.1: For angle pairs Φ, $\Psi \in [-\pi, +\pi)$ the problem of inconsistencies in the conformational space due to overlapping fragments of length 5 is illustrated. Yellow/orange indicate highly preferred regions and blue/black less preferred regions.

ture. This is condensed in the probability vectors for sequences, formula (2.2.8), page 26. Each entry in such a vector can be interpreted as a class weight and each class models a small subspace of all possible conformations by a multivariate normal distribution model. The weighted sum of all class models forms a mixture distribution and can be used to narrow down the conformational space to the subspace preferred by the sequence fragment.

When trying to predict a whole protein structure, the amino acid sequence is fragmented into overlapping fragments. The overlapping regions are then modelled in $k-1$ mixture distributions for fragments of length k. As discussed in subsection 2.2.1, the combination of these mixture models is not straight forward, especially because the mixture models are generally not consistent with each other. That means, two probability vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i'}$ with $i > i'$, that model the same angle pair (ϕ_m, ψ_m) at residue $m \in [i, i'+k-1] \neq \emptyset$, share the same number of classes, but the class weights and the class models at the common residue are different because (ϕ_m, ψ_m) appears in the corresponding fragments in different dimensions, namely $m-i+1$ and $m-i'+1$. This inconsistency is illustrated in figure 4.0.1. The five overlapping $\phi\psi$ mixture distributions for aspartate (`D`) do not agree with each other. The distributions of fragment `LFLVD` suggest that this sequence prefers an extended conformation, whereas the distributions of the overlapping fragment `DQDIK` point to a right-handed helix.

Whatever combination scheme is used for calculating a structure from the probability vectors, it would be best to minimise these inconsistencies before the combining. This leads to an iterative update scheme that has been inspired by self-consistent mean field (SCMF) methods [KD96, KD98] and is described in

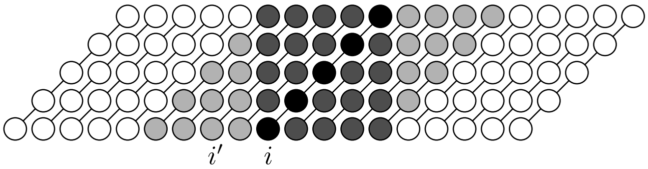


Figure 4.1.1: For a fragment of length $k = 5$, the overlapping regions are highlighted in dark gray. The gray area left of the black fragment i symbolises the index set $\mathbb{I}_i$ and the entire gray area symbolises $\mathbb{O}_i$.

detail in chapter 3. Here, our statistical SCMF version is applied to work out inconsistencies between states of dihedral backbone angles. The subsystems mentioned in chapter 3 are the overlapping fragments of dihedral angles where the probabilities are described by the probability vectors, formula (2.2.8), page 26, and the states are the classes.

4.1 Methods

4.1.1 Optimising Class Weights

Here, the purely statistical version of mean field optimisation (section 3.2) is used to derive a protein structure using solely a probabilistic framework based on fragments [SMT08b]. The system being optimised is the protein structure $\boldsymbol{\mathcal{X}}$ given its sequence $\boldsymbol{\mathcal{S}}$. That means, in the notation of the previous chapter, $\boldsymbol{\mathfrak{X}}^{\mathrm{k}} = \boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathfrak{X}}^{\mathrm{u}} = \boldsymbol{\mathcal{X}}$. For simplicity, only backbone conformations are considered here. They are described by dihedral angle pairs and amino acid labels, i.e. $\boldsymbol{\mathcal{X}} = (\phi_1, \psi_1, \dots, \phi_l, \psi_l) \in (\mathbb{R} \times \mathbb{R})^l$ and $\boldsymbol{\mathcal{S}} = (a_1, \dots, a_l) \in [1, 20]^l$. The protein is subdivided into small overlapping fragments $\{\boldsymbol{F}_i\}$ of a fixed length $k = 5$. The space of possible conformations $\boldsymbol{S}_i, \boldsymbol{X}_i$ for a fragment $\boldsymbol{F}_i$ has been discretised into $n = 162$ classes $\{\mathcal{C}_j\}$. Each class models a small subspace of conformations by k bivariate Gaussian distributions for $\boldsymbol{X}_i$ (dihedral angles) and k 20-way Bernoulli distributions for $\boldsymbol{S}_i$ (amino acid labels). See also section 1.3.

Each fragment $\boldsymbol{F}_i$ feels the influence of up to $2(k-1)$ overlapping fragments $\{\boldsymbol{F}_{i'} \mid i' \in \mathbb{O}_i\}$, where $\mathbb{O}_i = [\max\{1, i-k+1\},\ \min\{l-k+1, i+k-1\}]$ is the index set of overlapping fragments, figure 4.1.1. In order to work out the inconsistencies of overlapping parts of the fragments, the probability is considered that the current state/class $\mathcal{C}_j$ for fragment $\boldsymbol{F}_i$ is consistent with the weighted classes $\mathcal{C}_{j'}$ of the overlapping fragments $\boldsymbol{F}_{i'} : i' \in \mathbb{O}_i$. Differing from the method described in chapter 3, the implementation here deals with states of entire fragments, not just single sites. Each position $t \in [1, k]$ within the fragment $\boldsymbol{F}_i$ can be considered. Alternatively, the task may be simplified by just looking at the first position ($t = 1$). Furthermore, angles pairs (ϕ, ψ) (bivariate case) can be considered or each angle individually.

We restrict ourselves to $t = 1$ and the bivariate case. The index set $\mathbb{O}_i$ then reduces to $\mathbb{I}_i = [\max\{1, i-k+1\},\ i]$ and the update rule for state probabilities (formula (3.2.3)) is adapted to be

$$\mathrm{p}_{\mathrm{new}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big) = \lambda\, \mathrm{p}_{\mathrm{old}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big) + (1-\lambda)\, \mathrm{p}_{\mathrm{cur}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big)\,, \tag{4.1.1}$$

where $\lambda \in [0,1]$ controls the speed of adapting to the new weights. The current state probability $\mathrm{p}_{\mathrm{cur}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big)$ is reflected by the effect that the overlapping sites impose on each other. It is based on the class probability of the given sequence environment $\mathrm{p}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{F}_i \sim \mathcal{C}_j)$ and the weighted average of the level of agreement of their $\phi\psi$-distributions $o^{i',i}_{j',j}$, formally captured by

$$\mathrm{p}_{\mathrm{cur}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big) = \frac{\left(\frac{\mathrm{p}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{F}_i \sim \mathcal{C}_j)}{|\mathbb{I}_i|} \sum\limits_{i' \in \mathbb{I}_i} \sum\limits_{j'=1}^{n} \mathrm{p}_{\mathrm{old}}\big(\boldsymbol{F}_{i'} \sim \mathcal{C}_{j'} \,\big|\, \boldsymbol{\mathcal{S}}\big)\, o^{i',i}_{j',j}\right)^{\frac{1}{T}}}{\sum\limits_{j''=1}^{n} \left(\frac{\mathrm{p}_{\boldsymbol{\mathcal{S}}}\big(\boldsymbol{F}_i \sim \mathcal{C}_{j''}\big)}{|\mathbb{I}_i|} \sum\limits_{i' \in \mathbb{I}_i} \sum\limits_{j'=1}^{n} \mathrm{p}_{\mathrm{old}}\big(\boldsymbol{F}_{i'} \sim \mathcal{C}_{j'} \,\big|\, \boldsymbol{\mathcal{S}}\big)\, o^{i',i}_{j',j''}\right)^{\frac{1}{T}}}\,. \tag{4.1.2}$$

The conditional weight or probability of $\boldsymbol{F}_i$ being in class $\mathcal{C}_j$ given the sequence $\boldsymbol{\mathcal{S}}$ is taken to be the initial sequence probability vector (also defined on page 26), formally

$$\mathrm{p}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{F}_i \sim \mathcal{C}_j) = \frac{\mathrm{p}(\boldsymbol{F}_i \sim \mathcal{C}_j)\, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S}_i = (a_i, \dots, a_{i+k-1}))}{\sum\limits_{j'=1}^{n} \mathrm{p}(\boldsymbol{F}_i \sim \mathcal{C}_{j'})\, \mathrm{p}_{\mathcal{C}_{j'}}(\boldsymbol{S}_i = (a_i, \dots, a_{i+k-1}))}\,. \tag{4.1.3}$$

The prior weight of $\boldsymbol{F}_i$ being in class $\mathcal{C}_j$ as seen in the training set $\mathrm{p}(\boldsymbol{F}_i \sim \mathcal{C}_j)$ comes directly from the Bayesian classification. The conditional probability $\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S}_i = (a_i, \dots, a_{i+k-1}))$ of $\boldsymbol{S}_i$ taking the values of a k-region of the known sequence given $\boldsymbol{F}_i$ being in class $\mathcal{C}_j$ is the product of k multiway Bernoulli probabilities (formula (1.3.2), page 7). The interesting part is the interaction term $o^{i',i}_{j',j}$, which can be formulated by

$$o^{i',i}_{j',j} = \iint\limits_{\mathbb{R}^2} \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{O}^{\mathrm{u}}_{i,i'}{}_1 = \begin{pmatrix}\phi\\ \psi\end{pmatrix}\right) \mathrm{p}_{\mathcal{C}_{j'}}\left(\boldsymbol{O}^{\mathrm{u}}_{i',i}{}_1 = \begin{pmatrix}\phi\\ \psi\end{pmatrix}\right) \mathrm{d}\phi\, \mathrm{d}\psi.$$

where $\boldsymbol{O}^{\mathrm{u}}_{i,i'}$ and $\boldsymbol{O}^{\mathrm{u}}_{i',i}$ denote the unknown structural terms of the overlaps of fragments $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i'}$, respectively (defined in section 3.2). The conditional probability density $\mathrm{p}_{\mathcal{C}_j}\Big(\boldsymbol{O}^{\mathrm{u}}_{i,i'}{}_1 = \big(\begin{smallmatrix}\phi\\ \psi\end{smallmatrix}\big)\Big)$ of $\boldsymbol{O}^{\mathrm{u}}_{i,i'}{}_1$ given $\boldsymbol{F}_i$ being in class $\mathcal{C}_j$ taking the values of the dihedral angle pair $\binom{\phi}{\psi}$ is defined by the corresponding bivariate Gaussian of class $\mathcal{C}_j$. The computationally demanding evaluation of the double integral is circumvented by considering only class means instead of the complete distribution range, giving

$$\begin{aligned} o_{j',j}^{i',i} &= \mathrm{p}_{\mathcal{C}_{j'}}\left(\boldsymbol{O}^{\mathrm{u}}_{i',i_1} \approx \begin{pmatrix} \mu_{j_{1\upphi}} \\ \mu_{j_{1\uppsi}} \end{pmatrix}\right) \\ &= \iint\limits_A \mathrm{p}_{\mathcal{C}_{j'}}\left(\boldsymbol{O}^{\mathrm{u}}_{i',i_1} = \begin{pmatrix} \phi \\ \psi \end{pmatrix}\right) \mathrm{d}\phi\,\mathrm{d}\psi, \end{aligned} \tag{4.1.4}$$

where $\boldsymbol{\mu}_j$ is the mean vector of the Gaussian distributions in class $\mathcal{C}_j$ and $A = [\mu_{j_{1\upphi}} - \epsilon_{\upphi},\ \mu_{j_{1\upphi}} + \epsilon_{\upphi}] \times [\mu_{j_{1\uppsi}} - \epsilon_{\uppsi},\ \mu_{j_{1\uppsi}} + \epsilon_{\uppsi}]$ is the integration domain with $(\epsilon_{\upphi}, \epsilon_{\uppsi}) > \mathbf{0}$ being some small vector-valued error $\in \mathbb{R}^2$. The convergence parameter T in equation (4.1.2) can be interpreted as an analogon to temperature in simulated annealing and is used to gradually narrow down the system until it converges to the most probable states. The convergence is quantified by some entropy-like measure, which reflects the average number of populated classes per fragment, given by

$$S = \frac{1}{k-l-1} \sum_{i=1}^{l-k+1} \sum_{j=1}^{n} \mathrm{p}_{\mathrm{new}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big) \ln\!\big(\mathrm{p}_{\mathrm{new}}\big(\boldsymbol{F}_i \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big)\big). \tag{4.1.5}$$

The system is said to be converged, when S no longer changes. After bringing the system to a self-consistent state at a high value of T, the space of allowed dihedral angles is further narrowed by applying the cooling scheme introduced in section 3.3. Here, the temperature-like parameter T is adjusted after each iteration of formula (4.1.1).

Note that there is a fundamental difference how the states in the general method and here are defined. Here, state probabilities for entire fragments are manipulated whereas the method in chapter 3 deals with probabilities for single sites and uses the fragment probabilities only as a scoring function. That means here, the class weights change during the optimisation process.

4.1.2 Calculating and Sampling Structures

Once the system converged to a consistent state of high probability, dihedral angles can be calculated for each residue from the optimised probability vectors. Several approaches are described in chapter 2. As discussed there, only the two arithmetic mean approaches (formulae (2.2.2) and (2.2.4)) would be able to deal with probability vectors sensibly. However, the probabilities optimised here are for whole fragments, so the formula (2.2.4) based on marginal probability density values is not applicable. Since, even under perfect conditions, formula (2.2.2) does not lead to acceptable results as discussed in chapter 2, a different approach is used here to get values for the dihedral angles. These are then used to build the protein structure model.

A more natural way than the formulae in chapter 2 is to construct protein structure models from probabilities by applying a sampling approach. That means to generate many dihedral angle sequences which lead to many protein structure models. This would sample the conformational space of the protein and has the potential to discover some naturally occurring flexibility.

4.1.2.1 Mixture Sampling

One way to generate samples of dihedral angles for a protein from probability vectors is to generate them in sequential order in the following manner

1. generate angles for residues 1 to k,
2. generate the angle pair at the next residue $i = k + 1$ with the conditional probability given the previous angles until all angles are generated.

The first step can be done by first choosing the class $\mathcal{C}_j$ via the optimised class weights $\mathrm{p}_{\text{final}}\big(\boldsymbol{F}_1 \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big)$ and then drawing a sample vector from the multivariate Gaussian associated with class $\mathcal{C}_j$. For this, the ranlib.c library is used [BL]. In the second step, the class weights are recalculated for the next fragment $\boldsymbol{F}_{i'}$, $i' \in \mathbb{I}_i \setminus \{i\}$, using the sequence $\boldsymbol{\mathcal{S}}$ and the already generated angles $\boldsymbol{\mathcal{X}}_{1\ldots i-1} = (\phi_1, \psi_1, \ldots, \phi_{i-1}, \psi_{i-1})$. The class $\mathcal{C}_j$ is then chosen according to its recalculated weight given by

$$\mathrm{p}_{\text{recalc}}\big(\boldsymbol{F}_{i'} \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{X}}_{1\ldots i-1}\big) = \frac{\begin{array}{c}\mathrm{p}_{\mathcal{C}_j}\big(\boldsymbol{O}^{\mathrm{u}}_{i',i-k} \approx (\phi_{i'}, \psi_{i'}, \ldots, \phi_{i-1}, \psi_{i-1})\big) \\ \cdot\, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S}_{i'} = \boldsymbol{s}_{i'})\, \mathrm{p}_{\text{final}}\big(\boldsymbol{F}_{i'} \sim \mathcal{C}_j \,\big|\, \boldsymbol{\mathcal{S}}\big)\end{array}}{\displaystyle\sum_{j'=1}^{n} \left(\begin{array}{c}\mathrm{p}_{\mathcal{C}_{j'}}\big(\boldsymbol{O}^{\mathrm{u}}_{i',i-k} \approx (\phi_{i'}, \psi_{i'}, \ldots, \phi_{i-1}, \psi_{i-1})\big) \\ \cdot\, \mathrm{p}_{\mathcal{C}_{j'}}(\boldsymbol{S}_{i'} = \boldsymbol{s}_{i'})\, \mathrm{p}_{\text{final}}\big(\boldsymbol{F}_{i'} \sim \mathcal{C}_{j'} \,\big|\, \boldsymbol{\mathcal{S}}\big)\end{array}\right)} \tag{4.1.6}$$

Then, the next angle pair (ϕ_i, ψ_i) is generated from the corresponding bivariate Gaussian. All generated angles are wrapped to fall into one period. Finally, the resulting angle sequences can be transformed into Cartesian coordinates according to subsection 2.2.2, page 25.

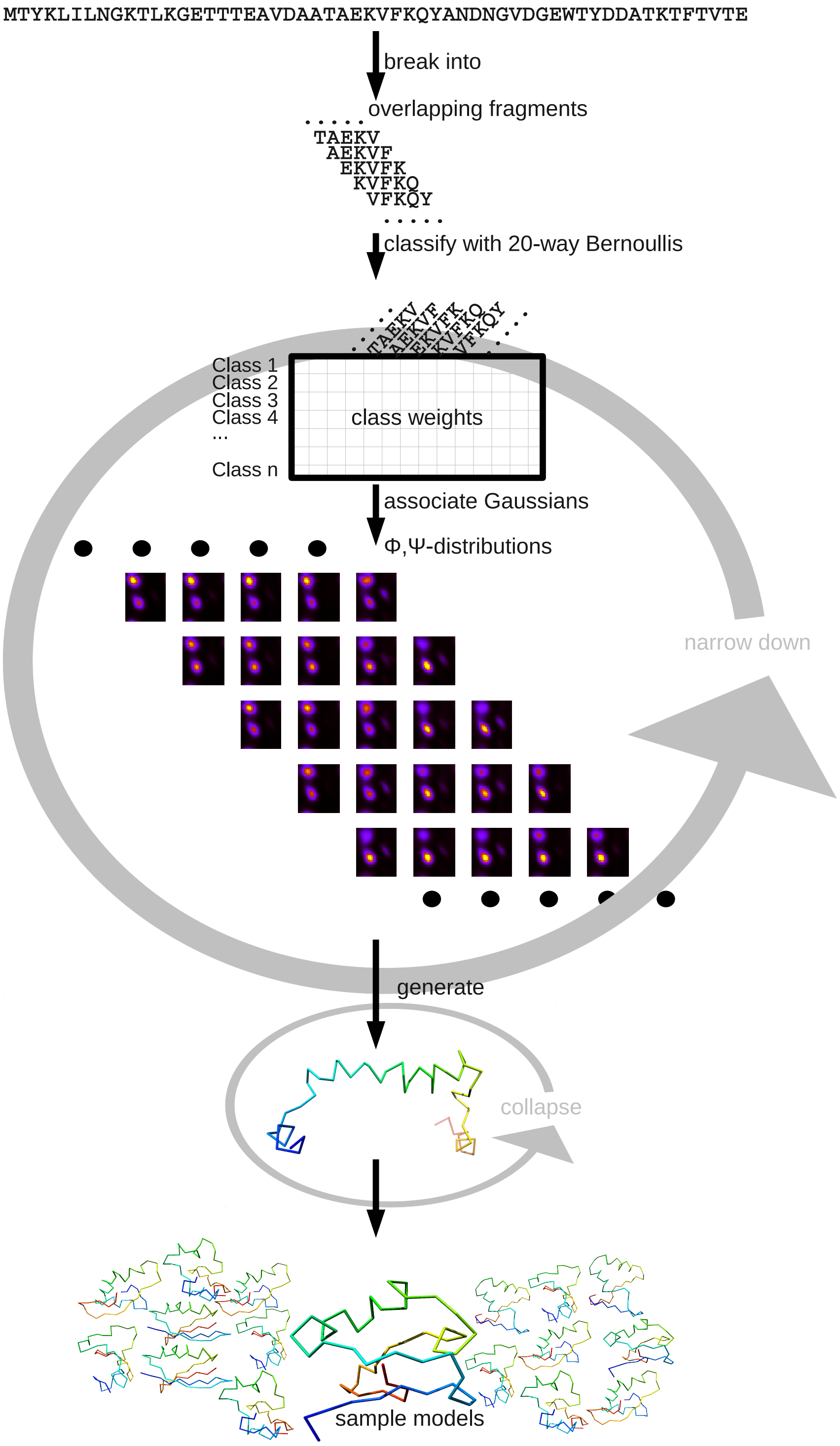
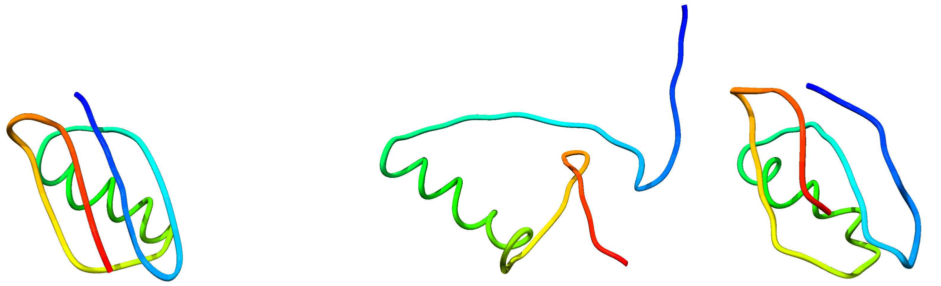


Figure 4.1.2: Steps for sampling structures from a given amino acid sequence.



(a) 2GB1 crystal structure (b) extended sample structure with clashes (c) collapsed sample structure without clashes

Figure 4.1.3: Comparison of extended (b) and collapsed (c) sample structures against the crystal structure (a) of 2GB1.

4.1.2.2 Scoring

Given an amino acid sequence a huge number of samples can be generated with the methods described previously, see also figure 4.1.2. A remaining question is, how to distinguish good models from bad ones. Ranking or scoring is a task on its own. A very popular scoring function is the Rosetta score [SKHB97]. In [Mah09] a scoring, based on the same fragment-based classification that is used here, is applied to Monte Carlo optimisation. One way accordingly would be to rank the sample structure $\big(\boldsymbol{\mathcal{X}}\,\big|\,\boldsymbol{\mathcal{S}}\big)$ given its sequence with its fragment probabilities

$$\operatorname{score}\big(\boldsymbol{\mathcal{X}}\,\big|\,\boldsymbol{\mathcal{S}}\big) = \left(\prod_{i=1}^{l-k+1}\sum_{j=1}^{n} \mathrm{p}_{\boldsymbol{\mathcal{S}}}(\boldsymbol{F}_i \sim \mathcal{C}_j)\,\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} \approx \boldsymbol{x}_i)\right)^{(l-k+1)^{-1}} . \qquad (4.1.7)$$

Thereby, the fragments are effectively scored independently. Some variations of formula (4.1.7) were used in [Mah09], but here, the basic form is preferred for simplicity.

4.1.3 Refining Structures

The generated structures constructed from sampled dihedral angles can contain steric clashes (i.e. atoms lying on top of each other). Additionally the structures might not be very compact, because a single wrong dihedral angle can move a whole domain. See figure 4.1.3, where structures are aligned on the common central helix. Two simple refinement schemes with ideas, that have been successful in [HKK06], are explained in the next two paragraphs.

4.1.3.1 Clash Removal

Like in [HKK06], a clash is defined by any two α-carbons, separated by at least two residues, being closer than 4 Å. In order to remove these clashes, the initial structures are optimised in a greedy way, see algorithm 2. Random stretches of dihedral angles are iteratively resampled until all clashes are removed.

```
 1: $\mathcal{X} \leftarrow$ initial structure of length $l$
 2: $N_{\text{clashes}} \leftarrow$ number of clashes in $\mathcal{X}$
 3: $c \leftarrow 1000$
 4: repeat
 5:   $i \in [1, l]$ randomly
 6:   $i' \in [i+1, i+15]$ randomly
 7:   generate new dihedral angles for residues $i \dots i'$ according to the procedure outlined in paragraph 4.1.2.1 by using conditional class weights (formula (4.1.6))
 8:   calculate new coordinates $\mathcal{X}^{\text{tmp}}$
 9:   $N^{\text{tmp}}_{\text{clashes}} \leftarrow$ number of clashes in $\mathcal{X}^{\text{tmp}}$
10:   if $N^{\text{tmp}}_{\text{clashes}} < N_{\text{clashes}}$ then {accept new structure}
11:     $N_{\text{clashes}} \leftarrow N^{\text{tmp}}_{\text{clashes}}$
12:     $\mathcal{X} \leftarrow \mathcal{X}^{\text{tmp}}$
13:     $c \leftarrow 1000$
14:   else
15:     $c \leftarrow c - 1$
16:   end if
17: until $N_{\text{clashes}} = 0 \lor c = 0$
```

Algorithm 2: Greedy clash removing.

4.1.3.2 Structure Collapse

In order to generate compact structures a similar scheme as for clash removal is used. In line 10 of algorithm 2 an additional criterion, the radius of gyration, is applied. It is based on the coordinates of the α-carbons and given by

$$R_{\text{g}} = \frac{1}{l\sqrt{2}} \sqrt{\sum_{i=1}^{l} \sum_{i'=i+1}^{l} |\mathsf{C}^{\alpha}_{i} - \mathsf{C}^{\alpha}_{i'}|}.$$

Only if the number of clashes does not increase and the radius of gyration is decreased, the new structure is accepted for further refinement. The refinement is finished if the radius of gyration is below some target radius of gyration, taken from a known structure or predicted by

$$R_{\mathrm{g}} = 2.0l^{0.33}.$$

In [Dew93, SKO97] other values for the constants have been given. However, our values are found by trial-and-error to work best for our test datasets.

4.1.3.3 Structure Collapse via Contact Prediction

In [Mah09] an extended structural descriptor was introduced. In addition to the dihedral angles ϕ and ψ, the number of contacts, c, within a defined sphere are used to describe and score the sample structures. This number was rigorously included in the established classification scheme [SMT08b]. It improved the results in [Mah09] in a way that the generated models looked more compact. Therefore, another criterion for collapsing structures was tested as part of a student project [Han09]. A score very similar to formula (4.1.7) is calculated for each sample. The only difference is the probability of the number of contacts as an extra factor in the summation of $\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{X} \approx \boldsymbol{x}_i)$, where $\boldsymbol{x}_i = (\phi_i, \psi_i, c_i, \ldots, \phi_{i+k-1}, \psi_{i+k-1}, c_{i+k-1})$. When this score is higher, then the new structure is accepted for further refinement. The refinement is stopped when a maximal number of unsuccessful resampling trials is reached.

4.1.4 CASP

The critical assessment of techniques for protein structure prediction (CASP) is a famous biannual community wide experiment with the goal "*to obtain an in-depth and objective assessment of our current abilities and inabilities in the area of protein structure prediction*" [MFK$^+$]. The participants try to predict as much as they can on yet unknown, but soon to be released structures. During the curse of one season the sequences of the targets are distributed among the predictors. Although our approach is not a full blown structure prediction method, two completely automatic web services were registered for fun in season eight [ST08, SMT08a]. One structure prediction protocol was customised, as there is only three days time for predictions allowed, and the scoring formula (4.1.7) was implemented for quality assessment of predictions from all groups. Both servers received a query from the CASP organisers and directed the calculation to our workstation-cluster. When the calculations were finished an email with the results was send to the organisers. Following the steps of the customised protocol for the structure prediction,

1. the probability vectors for the query sequence were calculated by formula (2.2.8),

2. 50000 sample structures were generated directly from the initial probability vectors and ranked by formula (4.1.7),
3. the 110 top-ranked structures were refined according to subsection 4.1.3 and ranked again,
4. the 5 top-ranked refined models were submitted to the organisers.

For season nine in 2010 the procedure was slightly modified. Due to improved runtimes and better hardware available, the protocol was extended to narrow down the dihedral angle space by optimising the class weights before sampling a higher number of sample structures. The steps were:

1. the probability vectors for the query sequence were calculated by formula (2.2.8),
2. the class weights were optimised and cooled in approximately 1000 steps using the adaptive cooling scheme from section 3.3 with the parameters $T_{\text{start}} = 1$, $T_{\text{final}} = 10^{-18}$ and $\lambda = 0.1$ for the update formula (4.1.1),
3. 500000 sample structures were generated from the optimised probability vectors and ranked by formula (4.1.7),
4. the 110 top-ranked structures were refined according to subsection 4.1.3 and ranked again,
5. the 5 top-ranked refined models were submitted to the organisers.

4.2 Results

In order to illustrate the properties of the methods involved in our approach, a test set of rather arbitrary known protein structures, that has been used previously [HKK06], is considered. From these examples, the performances of a suitable reconstruction formula and the sampling approach as well as the evolution of the probabilities as the system is gradually cooled are shown. Also, some results from the CASP competitions are presented.

4.2.1 Most Likely Structures

One of the formulae for reconstructing structures in chapter 2 is used here to construct models for given amino acid sequences. The other formulas are either not suited, because they assume probability vectors created from angles, or they did not perform better. The most rigorously justifiable formula is the arithmetic

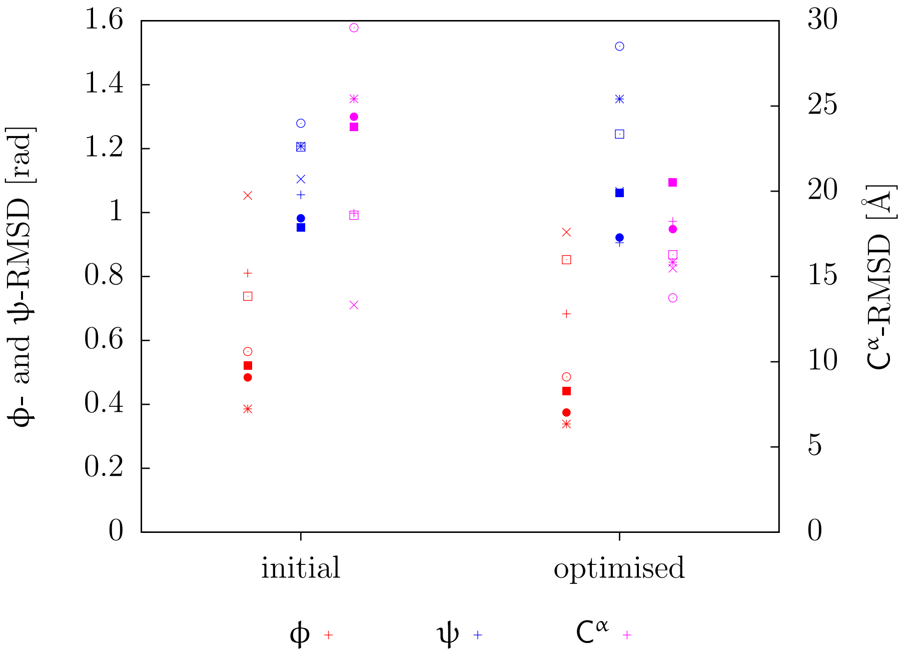


Figure 4.2.1: Performance test of the reconstruction formula (2.2.2) for prediction using initial class weights and optimised class weights of seven known protein structures. Each symbol type stands for another protein.

mean (2.2.2). It treats the overlapping sites of the fragments as different statistical models, that together form a mixture model for the angles at those sites. The angles generated this way can be seen as a prediction from the amino acid sequence and are essentially representatives of the active classes. Figure 4.2.1 shows the performance on a few known test structures that have been used previously [HKK06]. The root mean squared angular distances (RMSDs) of the dihedral angles ϕ and ψ in the calculated structures and the native structures fall in the approximate ranges $0.3 - 1.1$ ($17^\circ - 63^\circ$) and $0.9 - 1.3$ ($52^\circ - 74^\circ$), respectively. The RMSDs of the Cartesian coordinates of the C^α-atoms are all above 10Å. No big differences can be seen in figure 4.2.1 of the structures calculated from the initial and the optimised class weights. Only slight improvements for C^α-RMSDs and ϕ-RMSDs can be observed for this small test set.

4.2.2 Sampling

In table 4.2.1 a few example structures (that have been used previously [HKK06]) are reconstructed from their structure probability vectors using their original geometry with and without refinement. The positive effect of the refinement on the number of samples close to the native conformation (column 5) is easily recognised. In most cases the sample structure with the lowest C^α-RMSD also

target protein				**100000 initial samples**				
code	l	α	β	**< 6 Å**	**C^α-RMSD**	**ϕ-RMSD**	**ψ-RMSD**	**highscore**
1FC2C	43	2	0	1164	3.4 Å	0.5 (29°)	0.6 (34°)	8.5 Å
1ENH_	54	2	0	518	3.1 Å	0.3 (17°)	0.3 (17°)	12.0 Å
2GB1_	56	1	4	17	5.2 Å	0.4 (23°)	0.4 (23°)	14.2 Å
2CRO_	65	5	0	37	4.8 Å	0.3 (17°)	0.4 (23°)	10.5 Å
1CTF_	68	3	3	0	6.1 Å	0.3 (17°)	0.4 (23°)	21.4 Å
4ICB_	76	4	0	1	5.9 Å	0.4 (23°)	0.4 (23°)	11.1 Å

(a) initial samples

target protein				**100000 refined samples**				
code	l	α	β	**< 6 Å**	**C^α-RMSD**	**ϕ-RMSD**	**ψ-RMSD**	**highscore**
1FC2C	43	2	0	23659	2.8 Å	0.5 (29°)	0.5 (29°)	9.7 Å
1ENH_	54	2	0	10388	2.0 Å	0.3 (17°)	0.3 (17°)	5.9 Å
2GB1_	56	1	4	6619	3.4 Å	0.5 (29°)	0.4 (23°)	7.3 Å
2CRO_	65	5	0	2024	3.6 Å	0.3 (17°)	0.4 (23°)	11.4 Å
1CTF_	68	3	3	156	4.6 Å	0.3 (17°)	0.4 (23°)	11.9 Å
4ICB_	76	4	0	218	4.5 Å	0.4 (23°)	0.4 (23°)	9.2 Å

(b) clash free, collapsed samples

Table 4.2.1: Effect of the clash removal and collapsing of 100000 sample structures directly generated from non-optimised structural probability vectors (formulae (2.1.1)). Columns 1-4: PDB code, length, number of α-helices and number of β-strands of the target protein. Columns 5-9: number of samples with C^α-RMSD below 6Å, lowest C^α-RMSD, lowest ϕ-RMSD, lowest ψ-RMSD and C^α-RMSD of the best scoring sample.

becomes closer to the native conformation. Clearly, the longer proteins are harder to model. Interestingly as can be seen in figure 4.2.1, the lowest ψ-RMSD is often slightly higher than the lowest ϕ-RMSD. However, the clash removal and collapsing seem to have no significant effects on these numbers and therefore are not listed in table 4.2.3.

In table 4.2.2 the effects of the two collapsing refinement criteria on the radii of gyration are shown for a few target proteins (numbers taken from [Han09]). For them, the radii of gyration are closer to the native values after the refinement using no contact information.

In table 4.2.3 the results of five different sampling runs for the known test cases are shown. In 4.2.3a the samples are generated from initial probability vectors of the respective sequences before any optimisation of the class weights. The numbers of unrefined samples being close to the native conformations are very low and the best samples are far away from the native structures. Often not even a single sample can be considered sufficiently close. As expected the situation improved in all test cases for the refined samples. Here near-native samples

target protein					**refined samples**	
code	l	α	β	R_g [Å]	$\phi\psi$ R_g [Å]	$\phi\psi c$ R_g [Å]
1ENH_	54	2	0	7.1	7.4 ±0.1	8.4 ±1.1
2GB1_	56	1	4	7.2	7.7 ±0.3	8.8 ±1.0
2CRO_	65	5	0	7.1	7.9 ±0.2	9.0 ±0.9
1CTF_	68	3	3	7.5	8.1 ±0.3	9.2 ±1.1

Table 4.2.2: Effect of two collapsing criteria on the refinement of sample structures generated from non-optimised structural probability vectors. Columns 1-5: PDB code, length, number of α-helices, number of β-strands and radius of gyration of the target protein. Columns 6-7: radius of gyration after refinement using dihedral angles only and radius of gyration after refinement using dihedral angles with number of contacts simultaneously.

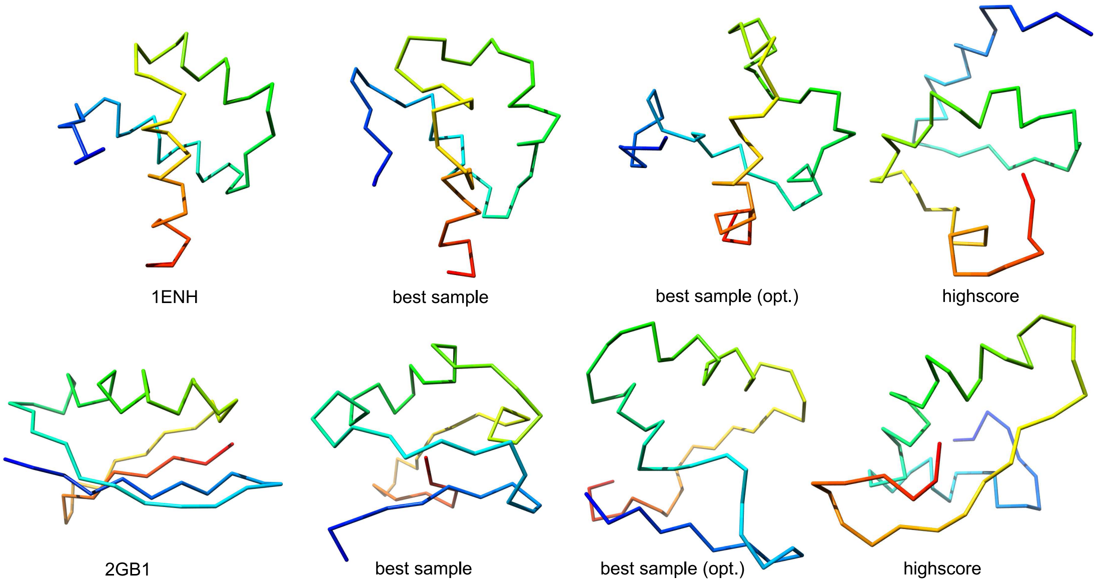


Figure 4.2.2: Two Examples from the evaluation. “best sample” is the sample with the lowest C^{α}-RMSD. “highscore” is the sample with the highest score.

are generated for all test cases. A similar trend can be seen in 4.2.3b for the samples generated from optimised probability vectors. In most cases the numbers are much better here. However, two cases, proteins 2GB1 and 1CTF, which comprise β-strands, perform significantly worse than the other cases no matter if using initial or optimised probability vectors. For proteins 1ENH and 2GB1 three sample structures are depicted along with the native conformation in figure 4.2.2, respectively. In contrast to the number of near-native structures and as expected from the C^{α}-RMSD values in table 4.2.3, the quality of the best sample is not necessarily increased for optimised class weights. These findings are consistent with the trends reported in [HKK06]. Some results are shown in table 4.2.3c, where the numbers are roughly of the same order of magnitude.

protein	**initial samples**			**refined samples**		
code	**< 6 Å**	**C^{α}-RMSD**	**highscore**	**< 6 Å**	**C^{α}-RMSD**	**highscore**
1FC2C	296	4.0Å	14.9Å	8877	3.4Å	9.2Å
1ENH_	43	4.8Å	14.6Å	2495	3.4Å	5.4Å
2GB1_	0	6.7Å	17.0Å	19	5.3Å	9.4Å
2CRO_	0	6.3Å	18.6Å	141	4.9Å	7.4Å
1CTF_	0	6.3Å	18.3Å	3	5.3Å	11.2Å
4ICB_	0	6.2Å	11.2Å	29	5.0Å	9.7Å

(a) initial and refined samples from initial sequence probability vectors

protein	**initial samples**			**refined samples**		
code	**< 6 Å**	**C^{α}-RMSD**	**highscore**	**< 6 Å**	**C^{α}-RMSD**	**highscore**
1FC2C	847	4.1Å	5.8Å	19043	3.0Å	8.4Å
1ENH_	44	5.1Å	10.2Å	3168	3.8Å	6.1Å
2GB1_	0	6.4Å	13.9Å	4	5.9Å	9.4Å
2CRO_	1	5.4Å	14.3Å	967	4.2Å	9.8Å
1CTF_	0	6.6Å	17.5Å	3	5.6Å	10.2Å
4ICB_	2	5.3Å	13.9Å	113	4.8Å	11.5Å

(b) initial and refined samples from optimised sequence probability vectors

target protein				**100000 samples**	
code	**l**	**α**	**β**	**< 6 Å**	**C^{α}-RMSD**
1FC2C	43	2	0	9593	2.7Å
1ENH_	54	2	0	6595	2.5Å
2GB1_	56	1	4	37	4.9Å
2CRO_	65	5	0	464	3.9Å
1CTF_	68	3	3	9	5.4Å
4ICB_	76	4	0	89	4.3Å

(c) FB5-HMM approach [HKK06]

Table 4.2.3: 100000 sample structures generated from initial (formula (2.2.8)) and from optimised sequence probability vectors after simulated adaptive cooling (section 3.3) with and without refinement compared to the results from a similar approach [HKK06]. For column description see table 4.2.1.

4.2.3 Optimising Class Weights

The conditional distributions of the dihedral backbone angles, ϕ and ψ, given a sequence of amino acids are analysed. A comparison of the conditional distributions before and after the optimisation with histograms of the training data is done. The latter can be interpreted as prior propensities of a certain amino acid in adapting different values for ϕ and ψ with its neighbours. Since Ramachandran [RRS63], three broad regions can be recognised: right-handed helical, extended and left-handed helical. It is also undoubtedly known that, depending on the side chain, the regions are populated differently. This can be seen in the left column of figure 4.2.3 for glycine, aspartate and asparagine. These amino acids are able to populate all three regions and therefore are chosen as examples. The other columns in figure 4.2.3 show pairs of $\phi\psi$-distributions in a few rather arbitrarily chosen sequence environments. They are calculated as a sum of mixtures of bivariate Gaussians. For each amino acid, the parameters are taken from the classification either using class weights derived from the sequence of a certain protein (as in formula (4.1.3)) for the left part or using optimised class weights (formula (4.1.1)) for the right part.

The residues selected for figure 4.2.3 are known to show prominent preferences for the ϕ and ψ angles. Looking at glycine, the histogram of the training data suggests that it can adopt any observed value of ϕ and ψ, with extended regions less populated. Taking the local sequence from 4ICB around glycine59 into account does not change much in the predicted distributions. However, after optimisation the only predicted region accessible is the left-handed helix, where also the native angles fall into. In another context (4ICB, pos. 8), glycine goes from a non-specific distribution towards the right-handed helical region after optimisation. Again the native angles fall into the same region. Aspartate58 in 4ICB is another amino acid starting in smeared out regions and travelling towards narrower parts of the $\phi\psi$-plots at the end of the optimisation. Like glycine, asparagine adopts to changing contexts. In most cases shown here, the sequence-specific $\phi\psi$-distributions are narrowed down after optimisation. An exception to this is 2GB1 aspartate40, which lies in a loop region. This is generally not observed for amino acids directly succeeded by a proline. Even the initial distributions are very much distorted and the optimised distributions show only slightly preferred regions. Another interesting case is 2KIC glycine40. Here the smeared regions seem to correspond with the flexibility of this loop region in the NMR structure ensemble.

Figure 4.2.4 illustrates the evolution of the $\phi\psi$-distributions of the target protein 2GB1 as the system is gradually cooled. At the start, i.e. before any equilibration or optimisation has taken place, the system is found in a number of preferred $\phi\psi$-regions for most residues. Only a few trends can be observed at this stage.

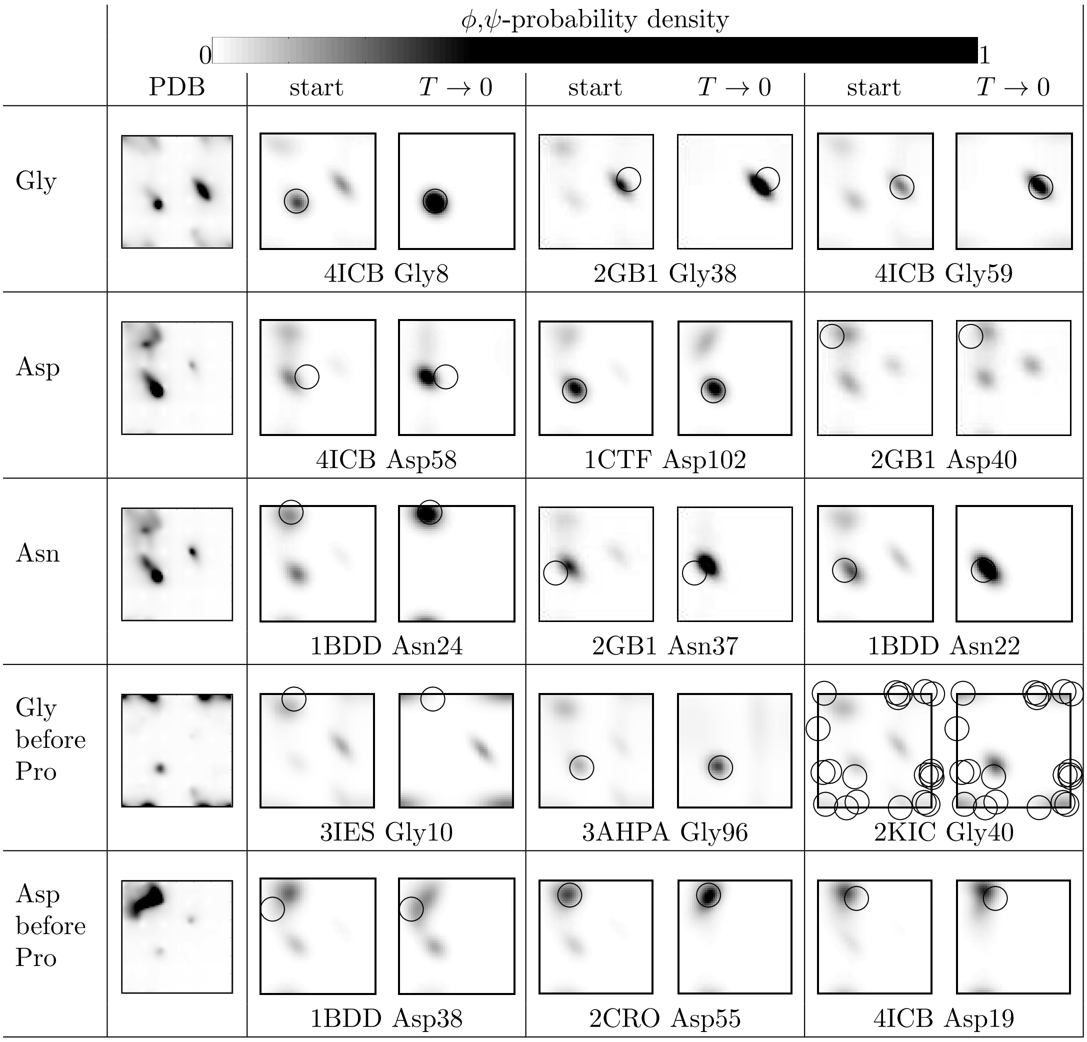


Figure 4.2.3: Histograms of training data (PDB) and ϕ, ψ-distributions of certain sequences at the beginning (start) and at the end ($T \to 0$) of the optimisation. The native values are indicated by circles. Created with kin2Dcont [WR02] and GNUPlot [WKM^{+}10].

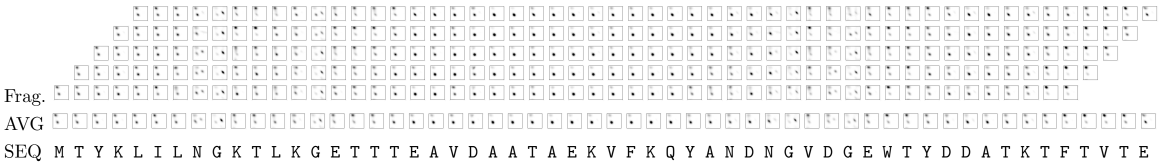
For example, a helical conformation seems to be preferred in the middle of the sequence from `E19` to `A34`. Also the glycines (`G`) at positions 9 and 38 seem to form right-handed turns. After the optimisation, i.e. when the system has been cooled down, these trends become much clearer. Other preferences show up as well. Considering just the average $\phi\psi$-distributions per residue clearly reveals a α-helix from `A23` to `Q32` and glycine-turns at `G9` and `G38`. The rest of the preferences is not so easily recognised. However, taking into account positions of the glycine-turns, the start of the helix and the strong preference for a left-handed turn of aspartate `D57`, the $\phi\psi$-distributions allow for four β-strands at the `N`- and `C`-termini and in between.

Although the optimised $\phi\psi$-distributions point in rather preferable directions, no significant improvement for the arithmetic mean prediction formula (2.2.2) could be achieved by optimising class weights for the small test set as observed in section 4.2.1. However, if the structures are generated by sampling from probability vectors, the number of near-conformations in the refined sample sets increases for most cases. The other numbers in tables 4.2.3a and 4.2.3b roughly stay at the same order of magnitude.

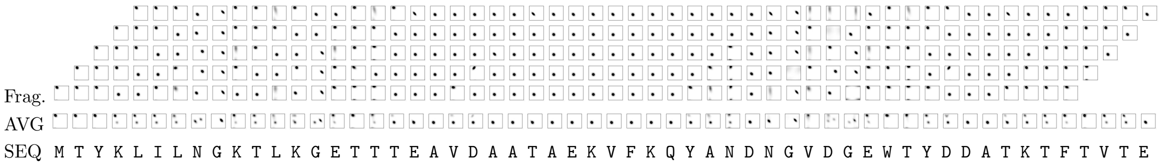
4.2.4 Scoring

The C^α-RMSD values for the high scoring samples in tables 4.2.1 and 4.2.3 are all quite high. Therefore, the generated structures can not be considered near-native. In the case of structure probability vectors (table 4.2.1) the numbers seem not to improve generally for the refined samples. Whereas in the case of initial sequence probability vectors (table 4.2.3a) the C^α-RMSD values improve for all test proteins. And for optimised sequence probability vectors (table 4.2.3b) the high scoring sample structures also improve in almost all test proteins, but not as strongly as in 4.2.3a.

In order to check if the score shows a good behaviour, it is compared to another established score and the C^α-RMSD. A very common score is the Rosetta score [SKHB97, LBXL08, BDNBP+09], which actually scales like energy, i.e. lower numbers are better. Figure 4.2.5 shows the Rosetta and the [Mah09]-score (formula (4.1.7)) versus the C^α-RMSD of some generated sample structures from different optimisation runs of our test proteins. A perfect energy would correlate well with C^α-RMSD and a perfect score would show anti-correlation. For the sample structures depicted here, the correlation of Rosetta with C^α-RMSD seems rather low. Most structures scatter close to zero for Rosetta, even structures with very high C^α-RMSD. A few outliers can also be observed for structures with relatively low C^α-RMSD. But no structures scored by Rosetta are found in the upper right corner of the plot. The [Mah09]-score (formula (4.1.7)) seems not to show any significant correlation or anti-correlation. The points scatter all over the



(a) before optimisation



(b) after optimisation

Figure 4.2.4: Cooling effects on the ϕψ-distributions for target 2GB1 (white: low probability, black: high probability). The length of the overlapping fragments is five residues. SEQ: amino acid sequence, AVG: average distribution per residue, Frag.: fragment-wise distributions (diagonals).

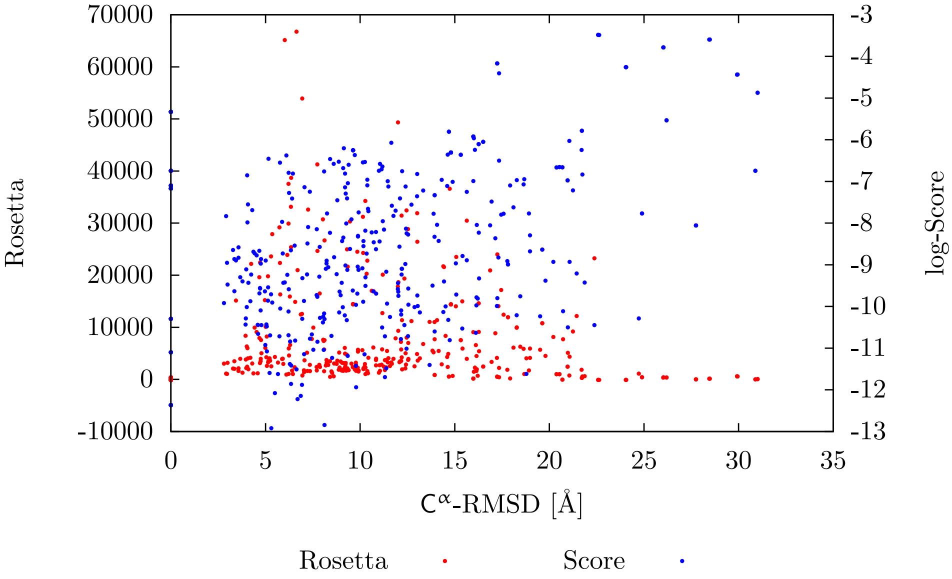


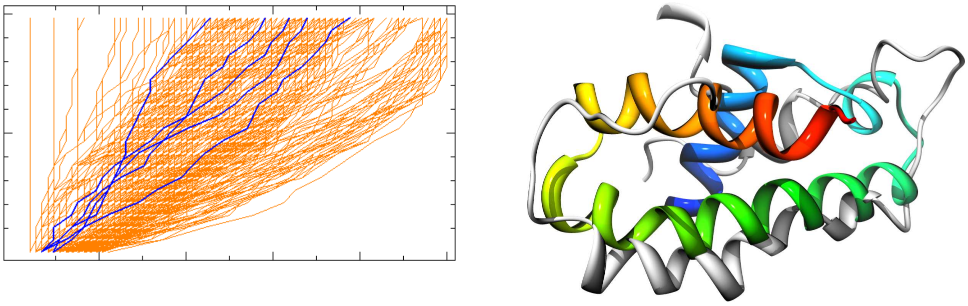
Figure 4.2.5: Scatterplot of some generated sample structures from different runs showing the distribution of two scores versus the C^α-RMSD. See [SKHB97] and formula (4.1.7).

plot. Structures with very high C^α-RMSD may get a good score and near-native structures may also get a bad score. However, there is a less populated area in the very lower left corner of the plot. This suggests at least a little anti-correlation for small C^α-RMSD values. The points with zero C^α-RMSD correspond to the native conformations of the proteins in our test set. Rosetta consistently places them close to zero, which means a high rank. The [Mah09]-score, which is also used for the CASP protocols, ranks the native conformation sometimes worse than some generated sample structure.

4.2.5 CASP

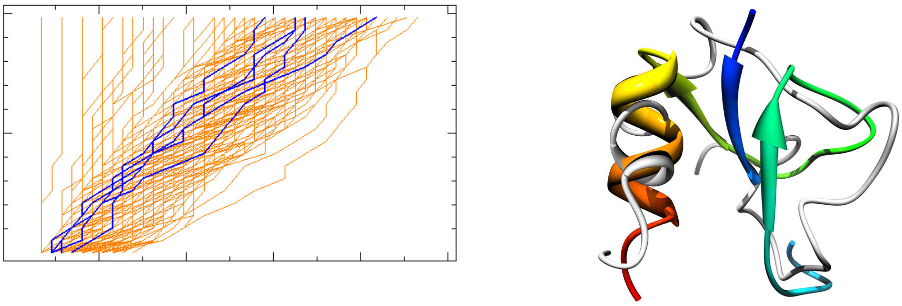
In CASP8, the automatic prediction server successfully submitted 550 structure models for 110 target proteins out of 121 total in the given time frame of 72 hours [MFK$^+$]. The protocol described in section 4.1.4 was optimised to meet this time constraint with the available hardware. The most time-consuming step is the refinement, especially the collapsing of the sample structures. In the final evaluation our server was ranked within the last three out of 72 automatic servers.

Nevertheless, some of the submitted models are interesting enough to have a closer look (figures 4.2.6 and 4.2.7). The left subfigures show the results of the global distance test (GDT) [MFK$^+$]. Each line in these plots corresponds to a model submitted by some prediction group. The more a line lies to the lower right corner of the plot, the better the corresponding model fits to the native



(a) GDT plot of target T0405-D1

(b) Superposition of model 3 (white) with target T0405-D1 (colour).



(c) GDT plot of target T0510-D3

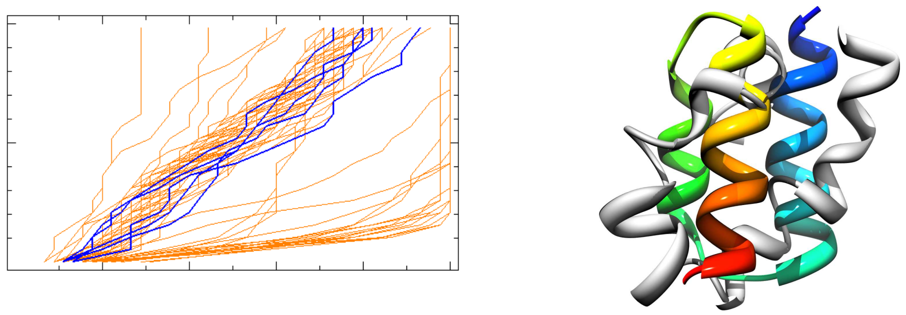
(d) Superposition of model 2 (white) with target T0510-D3 (colour).

Figure 4.2.6: Left: GDT analysis: largest set of C^{α} atoms that can fit under C^{α}-RMSD cutoff. Blue lines: the five models submitted by our server, orange lines: models from other servers. Right: alignment of predictions for targets T0405-D1 and T0510-D3.

conformation. The right subfigures depict superpositions of models submitted by our server with the native target structure.

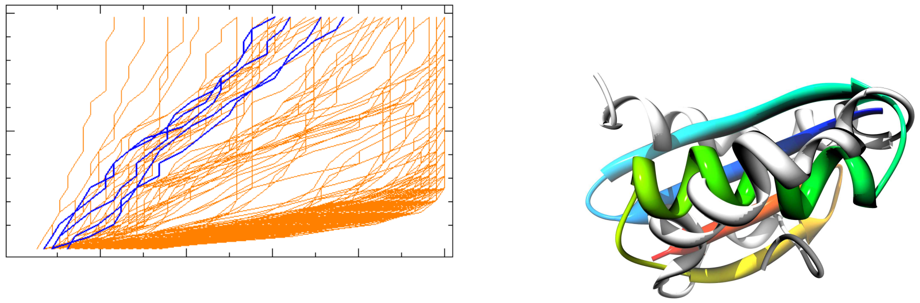
In figure 4.2.6 two difficult targets from CASP8 are shown. None of the prediction groups were able to submit a model that shows a good fit. Our submitted models are within the bulk of predictions. The superposition in 4.2.6b shows that the server recognised the central helix well, but the other two helices of the native conformation are not modelled. For domain 3 of target 510 the submitted models of all groups failed the GDT even worse (figure 4.2.6c). The fitting fractions of the predicted models are less on average. Again our predictions are within the bulk of the models. As can be seen in the superposition (figure 4.2.6d) the starting helix and some loops and strands are modelled, but not arranged in the same way as in the target structure. However, even the experimentally derived target structure does not contain reliable information about the loop conformations (see the missing links).

The targets 498 and 499 are also two difficult proteins, but with an interesting story behind them (figure 4.2.7). Their sequence identity is 95%, but their



(a) GDT plot of target T0498-D1

(b) Superposition of model 1 (white) with target T0498-D1 (colour).



(c) GDT plot of target T0499-D1

(d) Superposition of model 2 (white) with target T0499-D1 (colour).

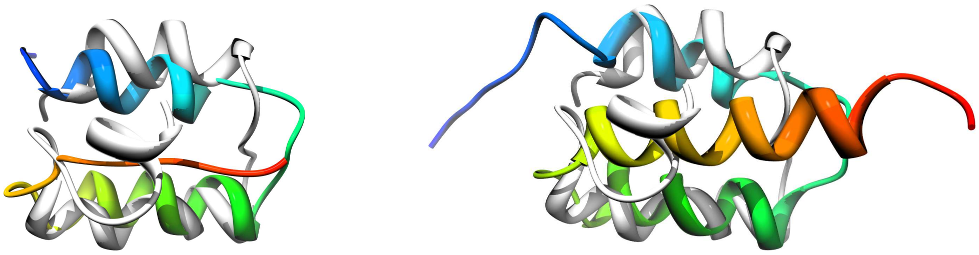
Figure 4.2.7: Left: GDT analysis: largest set of C^{α} atoms that can fit under C^{α}-RMSD cutoff. Blue lines: the five models submitted by our server, orange lines: models from other servers. Right: alignment of predictions for targets T0498-D1 and T0499-D1. These two targets are 95% sequence identical.

structures are completely different. T0498-D1 consists of three helixes, whereas T0499-D1 comprises one helix and four strands. This is very surprising considering the high sequence identity. In fact, these proteins are design studies, where the goal was to find a sequence that could be changed, with as little mutations as possible, from a helical structure to a structure comprising strands [AHC+09]. For CASP8 both proteins were classified as template-free modelling targets. That means, no templates are detectable by standard sequence search methods. Normally, this would be considered an interesting situation for the *ab initio* prediction methods. However, during the CASP8 experiment it turned out that some groups were able to find a template structure for target 499. So this was basically a misclassification and therefore the competition here was slightly unfair. For target 499 our method was not able to find a near-native conformation, except maybe the central helix. The GDT of the other target 498 (figure 4.2.7a) shows only very few submissions and most of them seem not to model the target structure exceptionally well. There is a small group of submissions which somehow managed to find the correct conformation. Looking at the superposition in subfigure 4.2.7b, our server seems to recognise the three helices. However, one of them is misplaced by 90 degrees. Interestingly, according to the CASP organisers none of the servers were able to distinguish between the two targets and submitted wrong predictions for at least one of the two. The models shown here, especially the model for target 498, are good examples of the big effect that slightly wrong angles can have on the overall C^α-RMSD.

For checking reasons, the six test proteins used previously are modelled also by the CASP protocols. Thereby, the models for target protein 1FC2C consistently show a beginning helix at the C-terminus. Wondering if this is real, a BLAST search was performed on the PDB in order to find other homologous structures [AGM+90]. The result is the entry 1BDD, which is the same protein but solved in solution by nuclear magnetic resonance. It turns out that 1BDD comprises three helices at the same positions that are modelled in our predictions (figure 4.2.8). Analysing 1FC2C reveals that it is part of a heterodimer having been solved by X-ray crystallography. Compared to 1BDD, the sequence of 1FC2C is shorter. The C-terminus of 1FC2C is in contact with a neighbouring unit cell, so maybe the sequence has been shortened to facilitate crystallisation. Therefore, the missing third helix either could result from the dimerisation or could be an artifact from the crystallisation.

The quality assessment server ranked models for 116 targets in CASP8 with the [Mah09]-score [MFK+]. The best ranking has a Spearman rank correlation coefficient of 0.532. This is relatively low compared to the other servers having coefficients up to 0.994.

In CASP round 9, our prediction server successfully submitted 160 models for 32 target proteins using the improved protocol. The server's performance is ranked



(a) Superposition of model 2 (white) with 1FC2C (colour).

(b) Superposition of model 2 (white) with 1BDD (colour).

Figure 4.2.8: Superpositions of prediction model 2 for target 1FC2C with 1FC2C and 1BDD.

among the last six out of 79 servers based on model 1 of five only. The evaluation of the CASP9 results shows the same trends as for CASP8.

4.3 Discussion

The method presented allows a sequence-specific sampling of the dihedral angles, ϕ, ψ, of the protein backbone, as can be seen from the different distributions in figure 4.2.3. Most of the time the preferred regions are narrowed down and this indicates a success of converging the class probabilities of the system. The local sequence context shifts the $\phi\psi$-distributions to regions roughly agreeing with the native values. A reason, why the exact values are not always found, could be that the fragments are too short. The influence on the $\phi\psi$-distributions remains local and the propagation of this influence during the iterative optimisation along the sequence is not strong. For example, tertiary contacts stabilise the formation of sheets. Such interactions are not modelled in our simple classification.

Therefore, the resampling of structures, like the removal of steric clashes and collapsing, is an important but simple post-processing method. It allows to consistently push the distribution of samples to obey additional constraints without worsening the dihedral angle distributions as seen in table 4.2.1. In order to collapse an initial sample structure to a given radius of gyration a rather large number of refinement steps is necessary. This leads to long runtimes that become especially critical for tough competitions like CASP, where only a limited computational time frame is allowed. As also seen in [Han09] the acceptance criterion could lead to local optima, so that a high rejection rate occurs and only rather large moves lead to better structures. This greedy strategy should be replaced eventually by Monte Carlo optimisation as in [Mah09]. Nevertheless, the refinement is an essential step towards near-native sample structures. It also improves the ranking quality for sample structures predicted from sequence.

The optimisation successfully narrows down the angular distributions of 2GB1 towards a 1-helix-4-strands fold, which can be recognised by visual inspection. In contrast, the generated sample structures are not very close to the native conformation, which in fact consists of one α-helix and four β-strands forming an anti-parallel β-sheet. The main reasons are probably a few angles pointing not in perfect directions and no constraint in the refinement brings together β-strands in order to form β-sheets.

The regular secondary structure (helix and extended strands) is often recognised by the method. But two problems can be observed here. Already slightly wrong angles and the use of standard geometry can lead to disturbed structures, see also chapter 2. The classification can be viewed as a discretisation which might be too broad for some classes. The smallest variance seen in the Gaussian terms is 0.4 (23°). This originates from the process of finding the classification in a training set of dihedral angles having 0.4 radians uncertainly [SMT08b]. So the generated models can not be expected to match known structures very well. The angular RMSDs therefore can not get below that threshold as has been found for the ideal reconstruction situation in chapter 2. The second problem is that right-handed helical regions might be preferred. Helices are the most common motif in the database leading to a high prior weight. Considering this, the C^{α}-RMSDs are fairly good, if the native bond lengths and angles are used. A C^{α}-RMSD below 6Å can be considered good for these test cases [HKK06]. However, the structures calculated from our test set are often still different from accurate conformations. That also means that the slight differences in RMSD using initial or optimised class weights should not be over-interpreted and probably are not significant. It is debatable if the trends become obvious if more test cases would be included.

Sometimes, the $\phi\psi$-space is narrowed down too much. For some residues the remaining classes are too narrow, i.e. the native angles are just on the border of the final classes, see figure 4.2.3. In these cases generating or sampling the correct angles becomes hard. For sequence optimisation (chapter 5) better results are obtained, if the samples are generated from states with higher entropy. For the hard cases it might be worth to additionally sample structures from intermediate states.

Considering the locality of the scoring, one might be tempted to compare the results to the native secondary structure or to other secondary structure prediction programs like GOR [KTJG02]. While this can be done manually, doing it automatically requires a definition of secondary structure based entirely on the dihedral angles. The assignments computed by common programs like DSSP or STRIDE [KS83, FA95] require an intact tertiary structure. Among other terms their definitions are based on H-bonds and contacts stabilising helices or sheets and would not work here. Therefore, this comparison would be a task in its own and mostly interesting from a scientific point of view, as the aim here clearly was the optimisation of tertiary structure.

Remembering the numbers from table 4.2.3, the poor performance at the CASP competitions is not surprising. Both the optimisation and the refinement improved the quality of the generated structures, but are not used at all or at least not exhaustively here. This leads to the ranking problem, where the protocols are especially fragile. Only a few samples may be selected for refinement in order to keep computational costs low. Unfortunately, the scoring is not very reliable and often samples reaching high scores show rather large C^{α}-RMSD values. An improvement, that most other prediction servers have, would be a parallel refinement. This would provide a bigger pool of sample structures to choose from while keeping within the time frame.

Despite the CASP results, considering the numbers of protein-like sample structures, the generation of tertiary structures is a promising approach. A remaining question is how to find the near-native conformations inside this sample set. This is a task for scoring and ranking the models. Using the Rosetta score for ranking the generated sample structures does not work as well as expected. The ranking does not correlate very much with C^{α}-RMSD. The samples are probably too far away from the native conformation in order to be guided decently to a near-native structure. The higher the C^{α}-RMSD is, the less reliable becomes the score. A rather simple scoring based on our classification is certainly not sophisticated enough. The score reflects the deviation of the structure from a typical conformation given the classification and the sequence. This is the reason why sometimes even the native conformation might get a lower rank than some sample structure. Therefore, the next steps should be to include chemically more sophisticated terms.

An extension to the current approach, where the discretisation of the $\phi\psi$-space is done by the classification, is actually not a trivial task. In order to include new scoring terms while staying consistent with the old scoring, one would have to redo the classification with the old and new terms together. In [Mah09] solvation terms in the form of contact counts could be introduced to the classification rather straight forward. Special challenges would be the inclusion of locally stabilising features, such as H-bonds or long-range interactions. Only recently a statistical description of H-bonds was published [PPMH10]. If and how this description could be included to our classification remains to be investigated.

The resampling idea is a very nice approach to rigorously impose additional constraints on the initial sample structures. Another way to include chemically more sophisticated terms would be possible, if a similar approach to sequence design (chapter 5), where the scores are not changed during the optimisation, would be implemented for structure prediction. This would mean to use a grid on the $\phi\psi$-space. Each bin would get a probability by which it would get selected in order to draw $\phi\psi$-samples uniformly from it. This would have the advantage that the scoring function would be easier to extend or replace, for example to include non-local interaction terms.

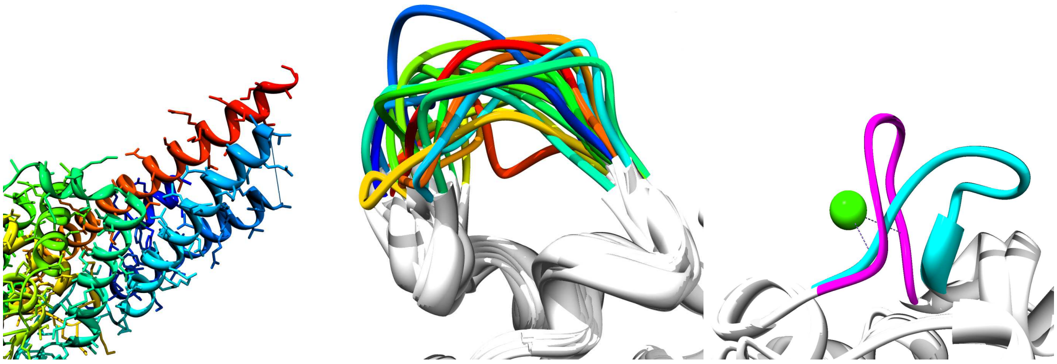
4.4 Outlook

4.4.1 Sequence Profiles

In the FB5-HMM approach [HKK06] secondary structure, in terms of helix, strand and coil, was used as an additional input to sequence and improved the prediction quality. This would not be very surprising, if only the real secondary structure would have been used. But it becomes quiet remarkable when predicted secondary structure also led to better results, because the accuracy of secondary structure prediction programs is less than 75% [KTJG02]. No direct way to apply this idea in our approach exists. Most advanced secondary structure prediction programs are based on multiple sequence alignments. These can be used to construct sequence profiles, which may substitute the single sequence input here. This technique has been successful for homology-based tertiary structure prediction where it is used to find and include distantly related templates [AMS$^+$97]. A profile is defined by using probabilities reflecting the amino acid propensities for each residue. Such a profile is then used as search query in template databases. Typically, it is build from multiple sequence alignments of homologous proteins. The hope is that profiles capture the important features of a protein family. Therefore, they are more suited to broaden the search space, but at the same time restrict it to relevant hits only. This idea is also tried here.

From the probability vectors used in this work, profiles could be build easily. However, so far, these probabilities are calculated from a single protein sequence. In a preliminary investigation, the program Ψ-BLAST is used to build a profile from multiple sequences aligned to the given sequence [AMS$^+$97]. The sequences are taken from a database search result. This is not a trivial task and requires some parameter tweaking. If the BLAST search parameters are set too conservative, the profile will not differ much from using a single sequence. Whereas, if the parameters are set too loose, then unrelated sequences might be included in the profile, leading to a weak query due to averaging effects. Using the automatically optimised parameters from the protein threading server WURST [TPH04] should be a reliable compromise. These so build profiles, $\boldsymbol{\mathcal{P}} = (p_{i,a})_{i\in[1,l]\wedge a\in[1,20]}$, are then turned into probability vectors, given by

$$\boldsymbol{v}_i = \left(\frac{w_j \, \mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S} | \boldsymbol{\mathcal{P}})}{\sum\limits_{j'=1}^{n} w_{j'} \, \mathrm{p}_{\mathcal{C}_{j'}}(\boldsymbol{S} | \boldsymbol{\mathcal{P}})} \right)_{j\in[1,n]} ,$$



(a) a missing loop (dashed line) between two helices (b) a flexible loop in an NMR-ensemble (c) two conformations of a Calcium-binding loop

Figure 4.4.1: Reasons for loop modelling.

where

$$\mathrm{p}_{\mathcal{C}_j}(\boldsymbol{S}|\boldsymbol{\mathcal{P}}) = \prod_{t=1}^{k}\sum_{a=1}^{20} p_{i+t-1,a}\,\mathrm{p}_{\mathcal{C}_j}(S_t = a).$$

These can be used just like the vectors from a single sequence (formula (2.2.8)). Preliminary results suggest that the structure prediction does not improve, but sometimes even worsens. This might originate from weak smeared profiles that contain information from unrelated sequences. Or it could be that bad predictions get stronger, i.e. classes pointing away from the native conformation get more pronounced because of the simplicity of the terms used in the classification. Probably, the results are a mixture of both. In order to further investigate this, one could build profiles from clusters in the PDB50 or PDB90 subsets [PDBb] and see how close the predicted structure samples are to the structures in the clusters, respectively.

4.4.2 Loop Modelling

Sometimes parts of a protein structure are more flexible than others, see figure 4.4.1. These often correspond to loop regions connecting regular secondary structure elements. The loops might adopt different binding modes (figure 4.4.1c) or just do not have a fixed conformation (figure 4.4.1b). This can lead to poor electron density for X-ray models and missing xyz coordinates for the loop atoms (figure 4.4.1a). Another reason for missing loops are gaps in the alignment for building homology models from templates. However, these loops are often involved in protein function. Therefore, the exploration of their conformational space is of high interest. Here, we show a way to predict possible loop conform-

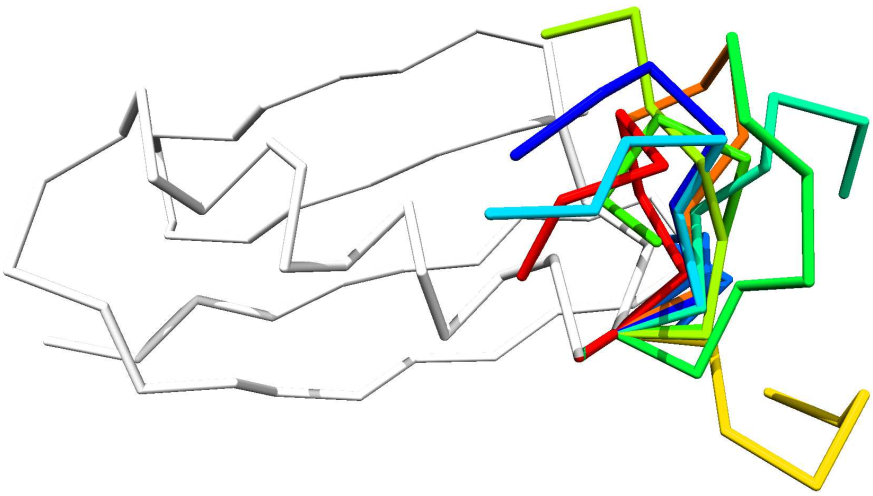


Figure 4.4.2: A few open loop conformations in structure 2GB1 with gaps at the C-terminus.

ations while accounting for their flexibility. This leads to an ensemble of both geometrically and biologically relevant conformations.

The problem of loop modelling is more constrained than the prediction of a whole protein structure. The loop is much shorter than the entire structure and the structural environment is already known. At a first glance this seems to lead to a simpler task. However, the problem actually becomes harder as the loop has to connect to the rest of the structure without introducing clashes. For example, a single, slightly wrong angle can cause an open loop.

The approach described here has been implemented to some extent in a student project [Küh10]. Let us define a loop as an element $\mathcal{L}$ of a set $\mathbb{L} \subset \mathcal{X}$ of unknown structural regions in $\mathcal{X}$ given the sequence $\mathcal{S}$. That way, a loop is any stretch of the protein, that has unknown structure. Then the class weights can be calculated from the sequence alone or from the sequence plus the known parts of the structure. In the loop modelling process the known parts are fixed and residues directly preceeding or succeeding a loop act as anchors. The generation of a loop conformation follows the resampling procedure outlined in subsection 4.1.2.1. The loops generated this way from N- to C-terminus are open conformations. In order to close the gap at the C-terminus, either many generated conformations are ranked and filtered by the distance and orientation to the C-terminal anchor, or by iteratively resampling little parts of a single conformation until the gap is closed sufficiently. The closeness criteria are

1. the distance of the C^{α} atoms of the C-terminal anchors in the fixed part and in the loop and
2. the RMSD of the four backbone atoms of the C-terminal anchors capturing the relative orientation.

Placing side-chain atoms is omitted, but could be added with a program like SCWRL [CSD03].

Currently, the program of [Küh10] is able to generate open loop samples based on sequence alone. However, the use of overlapping fragments allows for conditional probabilities given the fixed part of the structure. This leads to loop samples dependent on both sequence and structure. As can be seen in figure 4.4.2, the loops are not filtered yet. Therefore, the next steps would be to include the known pre- and succeeding structure in the class weights calculation and to implement a post processing step. This should filter out loops that introduce steric clashes or are not sufficiently closed in terms of orientation and distance of the anchors. Another post processing step would follow the refinement ideas by resampling the loops until the constraints are fulfilled.

Chapter 5

Sequence Prediction

Proteins acting as biocatalysts are called enzymes and can perform their task amazingly fast. They can facilitate biochemical reactions, that would otherwise not take place or only at a very low rate. The speedup is estimated to be up to 10^{17}-fold yielding a rate of several million reactions per second [RW95]. Industry is highly concerned, since there are many applications in biotechnology and medicinal chemistry with potentially high impact. Although harvesting naturally occurring enzymes has been done for decades, most applications require some modifications of the molecule. For example, proteases have been found that digest the dirt on clothes in washing machines. The naturally occurring forms are bound to work at the biological temperature, say 37 °C, but sometimes it is necessary to wash the clothes at higher or lower temperatures. So the washing powder industry tries to modify the proteases to be thermostable. Another example is the production of biofuel on a large scale. Here very special chemical conditions have to be matched. A last prominent example is the design of antibodies for the therapeutic treatment of patients.

Despite some impressive literature results, the design steps have often been rather *ad hoc* and the method is far from routine [KB00, KAS$^+$09, FF07, SJ09, KAV05, JAC$^+$08, SDB$^+$08, Tor04]. This is partly due to the fact that most of what is known about proteins is at native physiochemical conditions. In order to design or optimise a protein sequence, the correlation between the sequence and its structure has to be understood. Here protein design means to exchange side chains without changing the overall and essential structure, i.e. the backbone of the protein. Changing all side chains can be regarded as the inverse problem to structure prediction. For a given structure $\boldsymbol{\mathcal{X}}$ a suitable sequence $\boldsymbol{\mathcal{S}}$ folding to that structure has to be found. That means, the new sequence is optimised in terms of some energy or scoring function. An innovative approach is proposed here, which is based on self-consistent mean field (SCMF) methods, but using a framework of descriptive statistics. The approach is very similar to the structure

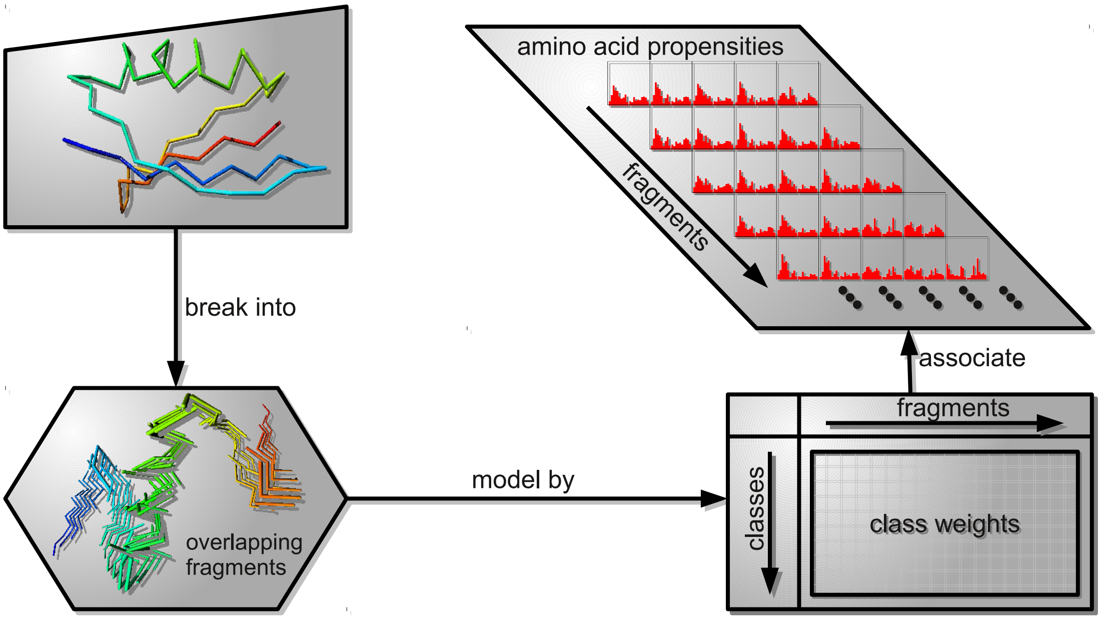


Figure 5.1.1: Preparation steps for sequence optimisation.

prediction methods in chapter 4. The same probabilistic classification, that has been described earlier in [SMT08b], in section 1.3 and in chapter 2, is combined with a purely statistical version of SCMF optimisation and simulated annealing, described in chapter 3.

5.1 Methods

Figure 5.1.1 illustrates the preparation steps for the given structure. First, the protein structure is subdivided into overlapping fragments of length $k = 5$. For each fragment and class a weight can be calculated, leading to a total of $n(l - k + 1)$ class weights with $n = 162$ the number of classes and l the length of the protein, see also section 2.1.1. These class weights are then used to build mixture distributions for the amino acid labels of each fragment using the associated 20-way Bernoulli probabilities. Each fragment feels the influence of up to $2(k - 1)$ overlapping fragments. Therefore, each residue is modelled by up to k mixture distributions, which may not entirely agree with each other. A way to work out these inconsistencies is the statistical SCMF method. Following the notation in section 3.2, the known terms are the dihedral angles of the backbone structure $\boldsymbol{\mathcal{X}} = (\phi_1, \psi_1, \ldots, \phi_l, \psi_l)^{\mathsf{T}}$ and the unknowns are the amino acid labels of the sequence $\boldsymbol{\mathcal{S}} = (a_1, \ldots, a_l)^{\mathsf{T}}$, i.e. $\boldsymbol{\mathfrak{X}}^{\mathrm{k}} = \boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathfrak{X}}^{\mathrm{u}} = \boldsymbol{\mathcal{S}}$. A solution to the problem could be approached analogously to the structure prediction problem (chapter 4), where the class weights are optimised per fragment. However, this turns out to be less flexible than using mutation probabilities for each residue like in other SCMF algorithms [DK97].

5.1.1 Optimising Residue Probabilities

Working with residue-wise probabilities leads to a description, which is closer to the general one introduced in chapter 3. Here, the states to be found are the amino acid labels $\mathbb{S}_i = \{a_i\} \subseteq [1, 20]$ for each residue $i \in [1, l]$ and so the update formula (3.2.3) can be reformulated to be

$$\mathrm{p}_{\mathrm{new}}\big(\mathcal{S}_i = a_i \big| \boldsymbol{\mathcal{X}}\big) = \lambda\, \mathrm{p}_{\mathrm{old}}\big(\mathcal{S}_i = a_i \big| \boldsymbol{\mathcal{X}}\big) + (1-\lambda)\, \mathrm{p}_{\mathrm{cur}}\big(\mathcal{S}_i = a_i \big| \boldsymbol{\mathcal{X}}\big)\,,$$

where $\lambda \in [0, 1]$ is a memory factor. The current probability of residue $\mathcal{S}_i$ labelled by amino acid a_i given the structure $\boldsymbol{\mathcal{X}}$ is formulated by

$$\mathrm{p}_{\mathrm{cur}}\big(\mathcal{S}_i = a_i \big| \boldsymbol{\mathcal{X}}\big) = \frac{\displaystyle\prod_{\substack{i' \in \mathbb{O}_i \\ i' \neq i}} \prod_{a_{i'} \in \mathbb{S}_{i'}} \Big(p^{i,i'}_{a_i, a_{i'}}\Big)^{\frac{2}{T}\, \mathrm{p}_{\mathrm{old}}\left(\mathcal{S}_{i'} = a_{i'} \,\middle|\, \boldsymbol{\mathcal{X}}\right)}}{\displaystyle\sum_{a \in \mathbb{S}_i} \prod_{\substack{i' \in \mathbb{O}_i \\ i' \neq i}} \prod_{a_{i'} \in \mathbb{S}_{i'}} \Big(p^{i,i'}_{a, a_{i'}}\Big)^{\frac{2}{T}\, \mathrm{p}_{\mathrm{old}}\left(\mathcal{S}_{i'} = a_{i'} \,\middle|\, \boldsymbol{\mathcal{X}}\right)}}, \tag{5.1.1}$$

where the interaction terms are weighted geometric means (inner product with weights in the exponent) over the pairwise interaction of all residues in the set of neighbours $\mathbb{O}_i$ (outer product). In analogy to [Sip90], the pair interaction $p^{i,i'}_{a_i,a_{i'}}$ of two residues i and i' is taken to be the net probability of the amino acid pair $(a_i, a_{i'})$, which yields the probability contribution of the pair $(a_i, a_{i'})$ to the average or marginal probability of the dihedral angles $\binom{\phi_i}{\psi_i}$ and $\binom{\phi_{i'}}{\psi_{i'}}$. The probabilities are given by the sum over the set $\mathbb{I}_i$ of all overlapping fragments. This leads to

$$\begin{aligned}
p^{i,i'}_{a_i,a_{i'}} &= \frac{\displaystyle\sum_{i'' \in \mathbb{I}_i} \mathrm{p}\left(\boldsymbol{O}^{\mathrm{k}}_{i'',i_1} \approx \begin{pmatrix}\phi_i \\ \psi_i\end{pmatrix},\ \boldsymbol{O}^{\mathrm{k}}_{i'',{i'}_1} \approx \begin{pmatrix}\phi_{i'} \\ \psi_{i'}\end{pmatrix} \middle| O^{\mathrm{u}}_{i'',i_1} = a_i,\ O^{\mathrm{u}}_{i'',{i'}_1} = a_{i'}\right)}{\displaystyle\sum_{i'' \in \mathbb{I}_i} \mathrm{p}\left(\boldsymbol{O}^{\mathrm{k}}_{i'',i_1} \approx \begin{pmatrix}\phi_i \\ \psi_i\end{pmatrix},\ \boldsymbol{O}^{\mathrm{k}}_{i'',{i'}_1} \approx \begin{pmatrix}\phi_{i'} \\ \psi_{i'}\end{pmatrix}\right)} \\
&= \frac{\displaystyle\sum_{i'' \in \mathbb{I}_i} \sum_{j=1}^{n} \left(\begin{array}{l} \mathrm{p}\big(\boldsymbol{F}_{i''} \sim \mathcal{C}_j \big| O^{\mathrm{u}}_{i'',i_1} = a_i,\ O^{\mathrm{u}}_{i'',{i'}_1} = a_{i'}\big) \\ \cdot\, \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{O}^{\mathrm{k}}_{i'',i_1} \approx \begin{pmatrix}\phi_i \\ \psi_i\end{pmatrix}\right) \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{O}^{\mathrm{k}}_{i'',{i'}_1} \approx \begin{pmatrix}\phi_{i'} \\ \psi_{i'}\end{pmatrix}\right) \end{array}\right)}{\displaystyle\sum_{i'' \in \mathbb{I}_i} \sum_{j=1}^{n} \left(\begin{array}{l} \mathrm{p}\big(\boldsymbol{F}_{i''} \sim \mathcal{C}_j\big) \\ \cdot\, \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{O}^{\mathrm{k}}_{i'',i_1} \approx \begin{pmatrix}\phi_i \\ \psi_i\end{pmatrix}\right) \mathrm{p}_{\mathcal{C}_j}\left(\boldsymbol{O}^{\mathrm{k}}_{i'',{i'}_1} \approx \begin{pmatrix}\phi_{i'} \\ \psi_{i'}\end{pmatrix}\right) \end{array}\right)}
\end{aligned}$$

where $n = 162$ is the number of classes and the conditional class weights are given by

$$\begin{aligned}
&\mathrm{p}\big(\boldsymbol{F}_{i''} \sim \mathcal{C}_j \,\big|\, O^{\mathrm{u}}_{i'',i_1} = a_i,\ O^{\mathrm{u}}_{i'',{i'}_1} = a_{i'}\big) \\
&= \frac{\mathrm{p}(\boldsymbol{F}_{i''} \sim \mathcal{C}_j)\,\mathrm{p}_{\mathcal{C}_j}\Big(O^{\mathrm{u}}_{i'',i_1} = a_i\Big)\,\mathrm{p}_{\mathcal{C}_j}\Big(O^{\mathrm{u}}_{i'',{i'}_1} = a_{i'}\Big)}{\displaystyle\sum_{j'=1}^{n}\mathrm{p}(\boldsymbol{F}_{i''} \sim \mathcal{C}_{j'})\,\mathrm{p}_{\mathcal{C}_{j'}}\Big(O^{\mathrm{u}}_{i'',i_1} = a_i\Big)\,\mathrm{p}_{\mathcal{C}_{j'}}\Big(O^{\mathrm{u}}_{i'',{i'}_1} = a_{i'}\Big)}.
\end{aligned}$$

The probability $\mathrm{p}_{\mathcal{C}_j}\Big(\boldsymbol{O}^{\mathrm{k}}_{i'',i_1} \approx \binom{\phi_i}{\psi_i}\Big)$ of the angle pair $\binom{\phi_i}{\psi_i}$ at the first residue of the overlap of fragments $\boldsymbol{F}_{i''}$ and $\boldsymbol{F}_i$ being in class $\mathcal{C}_j$ is defined as in formula (2.1.2) on page 12 and $\mathrm{p}_{\mathcal{C}_j}\Big(O^{\mathrm{u}}_{i'',i_1} = a_i\Big)$ is accordingly defined by the corresponding multiway Bernoulli distribution of class $\mathcal{C}_j$. The convergence parameter T in formula (5.1.1) allows to smoothen the probability landscape, analogous to temperature in simulated annealing, and to force the system to a single answer. As for structure optimisation, the convergence can be measured by an entropy-like measure, given by

$$S = \frac{-1}{l}\sum_{i=1}^{l}\sum_{a\in\mathbb{S}_i}\mathrm{p}_{\mathrm{new}}\big(\mathcal{S}_i = a\big|\boldsymbol{\mathcal{X}}\big)\ln\big(\mathrm{p}_{\mathrm{new}}\big(\mathcal{S}_i = a\big|\boldsymbol{\mathcal{X}}\big)\big). \tag{5.1.2}$$

It is worth noting, that the structural terms in the equations are not necessarily only ϕ- and ψ-angles but may be any (local) features of the structure.

5.1.2 Generating Sequences

Several approaches to (re)construct a sequence from probability vectors or class weights are introduced and tested in chapter 2. The method here works with residue-wise probabilities and therefore a sampling approach is more suited for generating sequences. Sampling sequences is actually very simple. At each residue $\mathcal{S}_i$ the amino acid label a is chosen according to its final optimised probability $\mathrm{p}_{\mathrm{final}}\big(\mathcal{S}_i = a\big|\boldsymbol{\mathcal{X}}\big)$.

5.2 Results

In order to illustrate the properties of the method some rather arbitrary proteins are considered. These are also used for structure prediction in chapter 4. From our examples, the evolution of the probabilities is shown as the system is gradually cooled. Then sample sequences are generated and analysed how protein-like their compositions are.

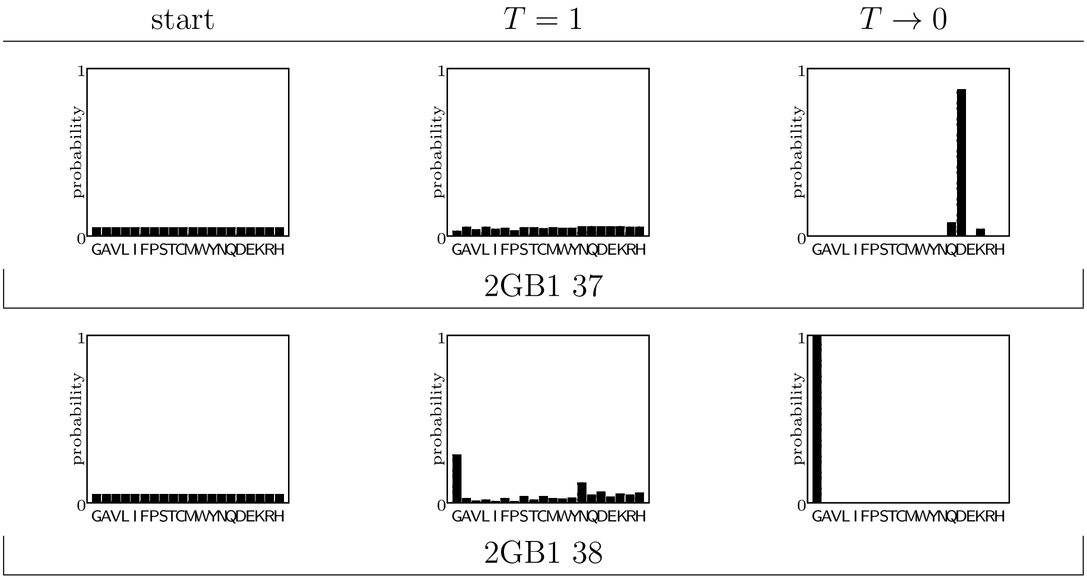
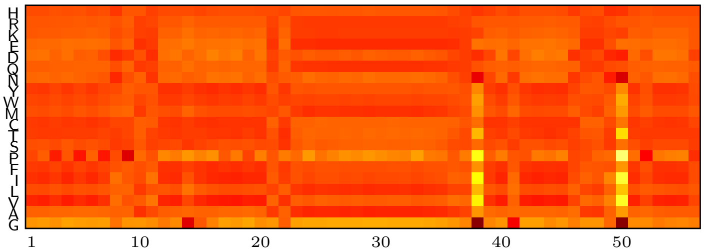


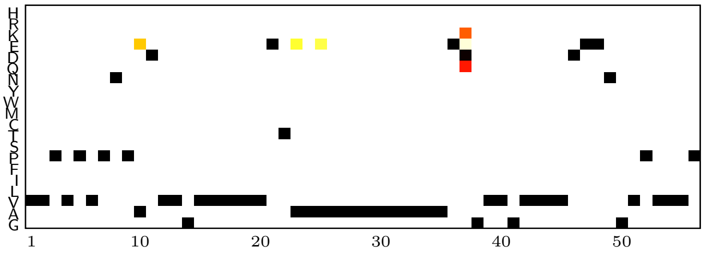
Figure 5.2.1: Amino acid distributions at certain positions in an arbitrary chosen structure before and after cooling.

5.2.1 Optimising Residue Probabilities

We have compared the conditional propensities of the 20 amino acids given a sequence of dihedral backbone angles, ϕ and ψ, before and after the optimisation. In figure 5.2.1 two example residues of protein 2GB1 are shown. At the first site (pos. 37) the probabilities are smeared out at high temperature and get more pronounced as the system cools down. In the second example the structural environment shows a clear preference for certain kinds of amino acid. Here and in all other test cases, the amino acid propensities are successfully narrowed down to one or a few remaining. Staying with the example structure 2GB1, this is quantified by the entropy-like measure (formula (5.1.2)) and can be easily visualised in the converged mutation matrix, figure 5.2.2. The entries in this matrix are the amino acid probabilities per residue. The probabilities in the hot matrix are smeared out, i.e. the system can almost freely move from one state to another. But finally the system is cooled so much, that only the most probable states remain accessible. Then, the new sequence almost can be read off directly from the cooled matrix. Blocks of repetitive composition are clearly visible. These trends correlate with the secondary structure as shown in figure 5.2.3a.



(a) $T = 1$



(b) $T \to 0$

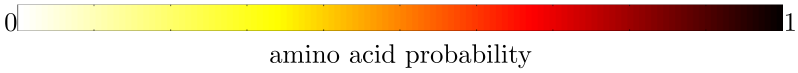
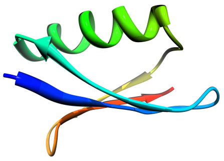


Figure 5.2.2: Mutation matrix of 2GB1 before and after cooling. The probabilities are colour coded in a heat map.

5.2.2 Generating Sequences

Figure 5.2.3 shows a few sample sequences for three target structures generated at intermediate optimisation steps. That means that each sequence is chosen from a set of 100000 sample sequences generated at different temperatures T. Each depicted sequence shows the highest similarity to the native sequence in terms of pairwise amino acid identities. In order to emphasise the properties of the sample sequences, the native sequence along with its secondary structure assignment is given. Comparing the sample sequences to the native sequence reveals only little similarity. A comparison with the assigned secondary structure gives more insight here. Particularly noticeable is the repetitive composition of sample sequences generated at low temperatures along regular helices and strands. Valine is found for strands and alanine or leucine for helices. Glycine is placed between regular secondary structure if the turn is left-handed. The native sequences agree here on a glycine as well. Proline, asparagine and aspartate are also often placed onto turns, but are sometimes shifted by one or two positions in the native sequence. However, if the sequences are generated at earlier stages of the simulation, i.e. at higher values for T, the repetitiveness becomes less obvious and the system gives a more heterogeneous amino acid composition.

There are some obvious and some rather unusual trends concerning the influence of the temperature-like convergence factor T on the sequence identity. For the

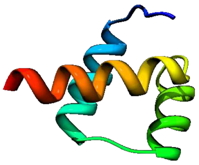


```
MTYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDDATKTFTVTE
 EEEEEE      EEEEEE   HHHHHHHHHHHHH      EEEEE    EEEEE
```

T

```
MVPICICNDCEFCGVYIVHVLTAAAQEWAAKAQAIDAGTIGIWIYDAEHGVPTYYP
VTVIVVVNPPHCTGFWCIVFLTRAAAEKEQEQMALKNGYVGVFCYCDKTGFPVVVF
VIPIPIPNPQDIVGIVVVVVESAAAAEEAAAQLAAASGVVGIVVVDDANGVPVVVP
VVVVVVPTDPDAVGVVVVVVTSPEELLLLLLLLLLALGLPGVVVVDPETGVPVVVP
VVPVPVPNPADVVGVVVVVVETEAEEAAAAAAAAAAEDGVVGVVVVDEENGVPVVVP
```

(a) 2GB1

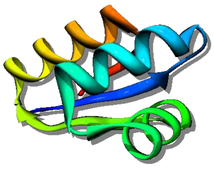


```
RPRTAFSSEQLARLKREFNENRYLTERRRQQLSSELGLNEAQIKIWFQNKRAKI
       HHHHHHHHHHHHH     HHHHHHHHHH    HHHHHHHHHHHHHHH
```

T

```
WQRTPVHMEQLSEYECEMGIPVHTTWCDREDLTTLKGCEFQRSLNVFVNQRAAS
IPPTIPPLLQIAALAAAKKEIRVSPELFRQMMMRDEGCPQQARKLWEMAVLAEI
PPPPIPPEEAAAEAAREEAEVEIVPEAAAQQLAAMLGIPLAMAAEAAALAAAEA
PPPPVPPAEAAAAAAAAAAAAVEVVPAAAAAAAAAEEGVPAEEAAAAAAAAAAAA
```

(b) 1ENH



```
EFDVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAPAALKEGVSKDDAEALKKALEEAGAEVEVK
 EEEEEEE   HHHHHHHHHHHHHH HHHHHHHHHH   EEEEEEEHHHHHHHHHHHHHHH  EEEEE
```

T

```
VMCMFSITQGADILAAAAEFKIHAGLFWCENKMLFESPYVHWCHGVLRFQLEFWFFRLNEQGPYVIVC
FFVVITDTKGDNARAAESRWMLLKGSTLEEAWEWADDWPAPQFTGPPARQAEALEKAEEQFGWRLCTC
FIVVVCWWLGANMAAAKKAALEQSGIPEKEKIELAQKPPVPEVVGIPQPKVEAAEKALEYAGLVFIII
VTVVIVHCNGAHELAQAEAQRQEYGPPLMEAAQENENPPVVWVVGCPAWEARALLKREEQEGITVVVV
TVVIILKWDGENEEEQAAAEAAMEGPPEKEAKLLEEDPPVPLVVGVPAAARMEMQQALAEAGPVVVVI
VVVVIIAVDGENAAEARAAEAAAAGPPLAEAQLAEENPPVPLVIGVPAMAAEALAAAAEEAGPVVVVV
VVVVVVEVDGENAAAAAAAAAAAEQGPPAAAAAAAAAENPPVPEVVGVPAAAAAAAAAAAAEEGPVVVVV
```

(c) 1CTF

Figure 5.2.3: Designed sequences for the structures of 2GB1, 1ENH and 1CTF. The native sequence and DSSP's secondary structure assignment [KS83] is given in bold letters (`H`: right-handed helix, `E`: extended/strand, ␣: turn/coil). Each sample sequence corresponds to a simulation stopping at different values for the convergence parameter T.

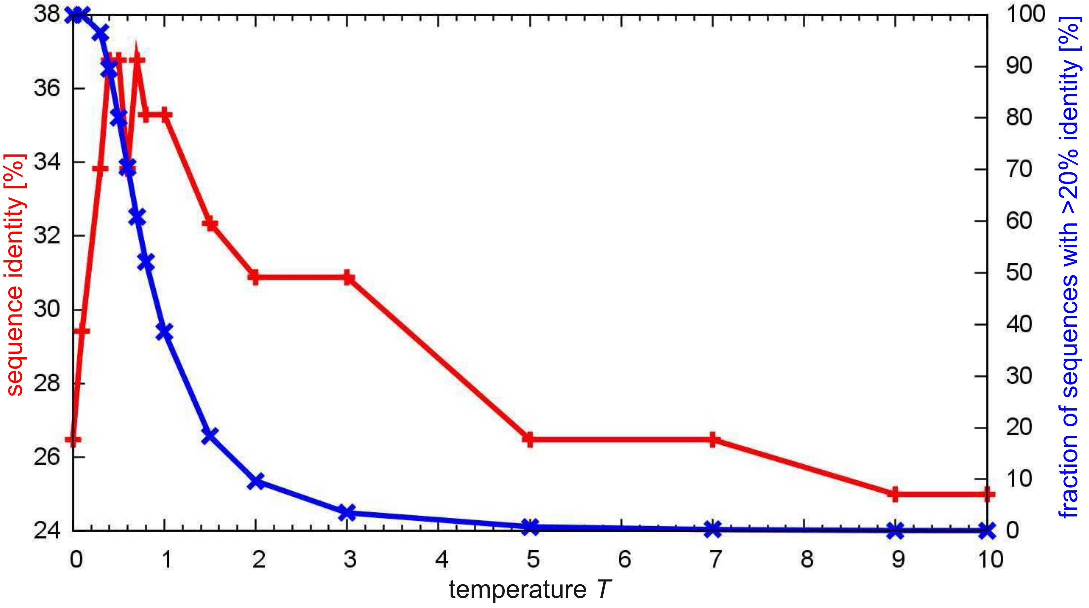
targets 1CTF and 1ENH figure 5.2.4 illustrates the dependence of the highest sequence similarity found in a set of 100000 sample sequences and the temperature-like parameter T. At high temperature the sample sequences have only random similarity of less than 25% in analogy to a hot physical system. When the temperature is decreased the sequences start showing some similarity to the native composition. There is a peak around $T = 1$. At lower temperatures the similarity goes down together with the diversity of the system.

5.3 Discussion

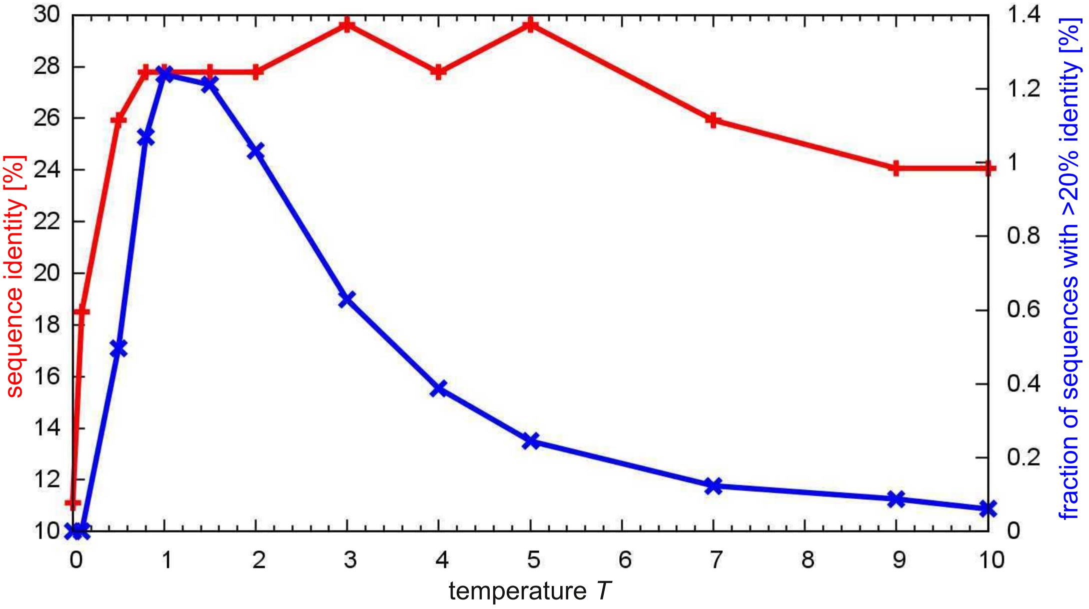
The approach introduced in this chapter allows one to generate sequences for a given structure. The fold of this structure is modelled in terms of a probabilistic description of dihedral backbone angles. Therefore the sequences are optimal in the context of these terms. The SCMF-like optimisation scheme successfully narrows down the state space. It can be used to control the heterogeneity of the amino acid composition.

A direct comparison between the generated sequences and the native sequence is not very sensible as shown in the results. A low similarity is is not necessarily bad as many sequences may fold to the same structure. In order to maintain its function the protein fold must be stable against mutation. The native sequence is just one, that has been shaped under evolutionary pressure. That means for example, it is not only designed to fold to the given backbone but also to perform a certain biochemical function. The native sequence can be viewed as a compromise between these two goals, whereas our sequences are only fitted against the fold in terms of dihedral angles. Some interesting features that are otherwise not obvious can be revealed by this comparison. For example, figure 5.2.1 shows that the native asparagine at position 37 in 2GB1 may mutate to an aspartate. To a biochemist this might not be a big surprise, however, it indicates that the method is able to find chemically sensible mutations. This exchange is also geometrically sensible as the dihedral angle distributions of these two amino acids are very similar, compare figure 4.2.3.

The method generates sequences that correlate well with secondary structure elements and common motives like glycine-turns or prolines at the ends of helices are successfully recognised. The method is able to sample context-specific sequences. However, a comparison to secondary structure assignments reveals strong correlations of repetitive stretches with regular helices and strands. This is not very surprising, as the scoring is based entirely on dihedral angles. At regular secondary structure the dihedral angles are also repetitive. This is not necessarily bad, as those sequences may act as reasonable starting points [KAS⁺09]. Furthermore, the heterogeneity of the generated sequences is adjustable by sampling at higher



(a) 1CTF



(b) 1ENH

Figure 5.2.4: Red: highest sequence identities out of 100000 samples versus temperature of 1CTF and 1ENH. Blue: fraction of sequences closer than 20% identity versus temperature.

temperatures. This trick successfully leads to more protein-like compositions.

At very low temperatures, only those sequences remain that optimally fit the dihedral angles. The native sequence is not necessarily optimal for dihedral angles alone, but was also formed to fulfil a biochemical function and by evolutionary pressure. In figure 5.2.4 the curve of the fraction of similar sequences to the native sequence is very sensitive to the arbitrarily chosen threshold of 20% identity. For some targets this is a good number, but for others it does not work. Maybe trying a number of different values would reveal more properties of the method. Using other similarity measures, like substitution scores based on BLOSUM matrices that reflect chemical and evolutionary similarity, would allow more sensible comparisons.

The method is built on top of a rather simple scoring scheme, based only on local statistics. But this does not mean that the optimisation scheme is restricted to it. It is possible to extend or replace the scoring with chemically more sophisticated models. The inclusion of solvent accessibility and tertiary packing is clearly one of the next tasks.

Another useful extension would be to have a ranking of the generated sequences. This should reflect the likelihood of a sequence folding to the target structure and allows to create a candidate list for further experimental testing.

One application that is possible already with the simple scoring would be mutation studies. This problem is much more constrained, as only small changes in a given sequence environment are considered.

As the sequences generated by our method are optimised for the given structure without any evolutionary constraints (as are the native sequences), it would be interesting to compare our sequences to other designed proteins. For example the two target structures T0498 and T0499 from the CASP8 competition, which share 95% of their sequence, or the four helix bundle used in [RD88], which has a rather repetitive sequence, would be exciting candidates.

The method tends to pick only the most probable sequence for the given structure. This may be too restrictive and can be relaxed to some extent by sampling at higher temperatures. Another way to broaden the solution space to other relevant sequences would be the use of profiles. Similar to the sequence profiles in homology searches for structure prediction, multiple structure alignments may be used to generate structure profiles for sequence prediction. The program HANSWURST creates multiple structure alignments based on the same probabilistic classification scheme used in this work [MT08]. This may act as a starting point for profiles and can lead to more distantly related sequences.

Chapter 6

Conclusions and Outlook

A single scoring and optimisation scheme for both structure prediction and sequence design is introduced with this work. It avoids arbitrary simplifications and comes with very little preconceptions. It makes almost no assumptions about the training set. It is rather common with so called knowledge-based forcefields to use the Boltzmann formalism. We could show that this essentially not necessary. This is quite advantageous, as the protein database is no defined statistical mechanics ensemble. The structures have been in all sorts of chemical conditions, e.g. varying pH, different buffers, temperatures etc.. We have, however, our own approximations. Modelling angles with Gaussian distributions is not ideal. But in the case of the dihedral angles ϕ and ψ, the use of non-periodic distribution functions is not critical as the angle boundaries can be shifted to sparsely populated areas. An extension that uses circular von Mises distributions to model directional features would nevertheless improve the approximation. Especially if other directional features shall be included, which do not show strongly underpopulated regions, then the availability of a periodic probability distribution model becomes extremely useful.

Considering the simplicity of the scoring the generated structures and sequences show rather interesting features. However, the low fraction of relevant structure samples and the repetitive sequence composition call for extensions of the scoring by including chemically more sophisticated terms, like tertiary packing and solvation. The addition of new terms to the existing scoring requires careful parametrisation. First of all, any new terms should be orthogonal to the others. Second, if extra terms are not rigorously included in our classification scheme, then the weighting becomes a major task in its own. However, some terms, like solvation, are easier to include than others, e.g. H-bonds. This has been shown in [Mah09]. Part of the difficulty is the consistent modelling of local and long-range terms, as the first perfectly fits our fragment-based approach but the latter might involve features that can be calculated only across fragments.

Another difficulty is to statistically capture the directionality of, for example, H-bonds. Only recently, a promising approach was published [PPMH10], where a statistical model of hydrogen bonding patterns is build. The next question is, how to use the new terms for our two favourite applications, sequence optimisation and structure prediction. For finding sequences, it would simply mean to know more about the given structure. This would give more accurate conditional probabilities and would further narrow the solution space already at the start of the simulation. For predicting structures via SCMF-like methods, this mainly would lead to better post-filtering options. However, it also offers to predict other structural properties than just the conformation, i.e. dihedral angles or atom coordinates.

The cooling scheme successfully narrows down the solution space for both sequence optimisation and structure prediction. As this might be a good situation for problems where a single solution is desired, like structure prediction, it limits the solution space too much in cases where many alternative solutions are desired, e.g. to propose sequences as candidates for further design studies. However, this is not the only reason to look for broader spaces. As our scoring is built of chemically rather simple terms, we do not expect to find a near-native solution necessarily within the top ranks. If the solution space is kept broad, e.g. by sampling at temperatures greater zero, solutions closer to the native can be obtained. This could be shown for sequence optimisation, where sequences could be sampled that show more protein-like compositions. The high number of unrelated structure samples generated at low temperature is not the result of insufficient sampling or bad optimisation, but rather comes from the very simple scoring. As is the case for sequence samples, generating structures at higher temperatures, and consequently states of higher entropy, would broaden the solution space and might increase the fraction of promising candidates.

The sampling approach works much better than our trials to predict a single answer of maximal probability. The reason is clearly the simplicity of the current scoring. Therefore, the generated samples are good starting points for further refinement. This is demonstrated successfully for the structure models with two rather simple constraints, by removing steric clashes and enforcing compactness. More sophisticated refinement should be possible, for example, via Monte-Carlo optimisation using other existing scoring schemes. This would avoid the creation of chemically sophisticated scores, but from a scientific point of view would gain only little compared to other existing approaches. Though, what is an innovation in that context is our (re)sampling approach entirely based on Boltzmann-free statistics. When using this approach as a move strategy for Metropolis Monte Carlo, then the proposed structures would be closer to a relevant answer than with random moves. However, the acceptance criterion has to account for the bias introduced. Otherwise detailed balance would be violated.

In principle, the concept of overlapping subsystems for propagating local biases over the entire system seems a promising idea. Application to the scoring of a state probability matrix of non-overlapping subsystems offers the most flexibility for extention of the score function. It would not harm so much to add extra terms that are derived from modelling schemes different to our classification. Parametrisation would be an issue, here. However, if the state probabilities of the overlapping subsystems themselves are updated (as done for structure optimisation), then heavy parametrisation problems would arise. Any new scoring terms would need to be included consistently in our classification, which can be difficult [Mah09].

One hope of our optimisation procedure was to find consistent, highly probable states. With very little limitations due to the suboptimal entropy measure the final states are consistent with each other and also the most probable with respect to the scoring. However, the sequential propagation of local biases does not lead to solutions showing real tertiary motifs, like β-sheets, hydrophobic packing or alternating hydrophobic/hydrophilic sequence patterns. The next step clearly is the use of chemically more sophisticated scores.

Another question is, how the cooling influences the results. Clearly, the solution space is narrowed down towards the most probable states. However, these states can already be seen at high temperatures. The states are just less pronounced. The system does not show any obvious quick phase transitions that would change the relative population of states. In that sense it seems unnecessary to cool the system. However, this is only true with respect to our scoring. Using different quality measures like RMSD or sequence identity (though they have their own pitfalls), a peak can be seen at lower temperatures but significantly above zero still. This suggests that there is an optimal temperature for sampling. For sequence sampling preliminary results point to a number slightly larger than 1.0. Finding it also for structure sampling would certainly increase efficiency.

The entropy-like measure used in the simulated annealing of our structure prediction approach reflects the average number of populated classes per fragment. As this approximation is sufficient in most cases, it sometimes leads to overlapping fragments being in different classes. Of course no two overlapping fragments can ever be literally in the same class, but at the end of the optimisation the final class means and shapes should not differ much. In order to account for that, we would need a different entropy-like measure that reflects not only the number of classes per fragment but rather the width of the class population per residue. Numerical integration would be necessary to compute the actual entropy per site. However, a summation of the entropies at each site of each class ($S_{j_m} = \frac{1}{2}\ln((2\pi\mathrm{e})^2|\det \mathrm{C}_{j_{m,m}}|)$, since bivariate Gaussians are used) would be an illegal approximation but might be a working. Another possibility could be based on the Mahalanobis distances between class means. This would not lead directly to a measure which scales like entropy and therefore needs more thinking and maybe an adaption of the annealing algorithm.

So far, the update formula for state probabilities is based on a product of the probabilities of interacting sites. But other measures are possible for calculating consistency or the level of agreement between overlapping fragments. For example, the Kullback-Leibler divergence between the state probability distributions of overlapping fragments could be used.

A thorough testing of different classifications should be done, in order to check the effect on the prediction quality. However, in the case of protein comparisons different choices for classifications were tested and found to be uncritical there. Preliminary results (not shown) for prediction suggest only slight changes, but the trends stay the same.

One application that would be possible already with our simple scoring is mutation studies. There the problem is much more constrained as only small changes in a fixed sequence environment are considered. This would be a similar problem as the loop modelling task. Class weights of the fixed parts can be calculated using both sequence and structure features simultaneously.

The secondary structure content and distribution in the generated structures seems to match the native secondary structure. Although it was not in the focus of this project, it would be interesting to see how our structure prediction results simply based on dihedral angles compares to secondary structure prediction programs like GOR [KTJG02]. To do so it would be necessary to assign secondary structure solely based on dihedral angles. Secondary structure assignment via DSSP or STRIDE [KS83, FA95] would not recognise β-strands in our models as these programs look for tertiary motifs, like H-bond patterns.

Sophistication of the optimisation scheme is always possible. Currently, for example, for calculating the level of agreement between overlapping fragments the overlapping parts are treated equally important. However, weighting these parts according to their influence values would be possible. These weights can be taken from the AutoClass-C classification reports and reflect how important each feature was for the training of the classes. Looking at them reveals that the midportion of fragments is most reliable, which is what is typically expected from a sliding window approach.

The preliminary results of the loop sampling do not show any principle limitations for exploring conformations of missing loops. In order to predict entire protein structures, loop modelling can help to bring initial homology models to full scale if parts are missing in the template. The overlapping nature of our approach is especially suitable to find reasonable conformations for parts of a model, for which the uncertainty is very high. This is the case in fragment-based approaches, if two neighbouring fragments are derived from different template structures and the correct structure of the bridge in between is not known. Then our approach allows for finding conformations consistent with the rest of the structure by considering the bias from the existing, reliable parts and resampling the unreliable parts. For modelling missing loops, the generated models would need to be connected to the anchors of the known parts. The number of open loop conformations should be checked. Then open loop models could be closed via inverse kinematics. Another extension would be to place side chains, which might also require to slightly remodel parts of the structure.

Clearly for the two main applications, structure prediction and sequence optimisation, the most urgent extension is the use of chemically more sophisticated scoring. The innovative statistical optimisation scheme introduced in this work is independent of the exact scoring function to a wide extent and can be considered a general search method. It is especially applicable to a range of optimisation tasks for systems with hard to define energy functions or even non-physical systems.

Appendix A

Directional Statistics

The field of directional statistics deals with the statistics of directions and angles. As the proteins in this work are also described by dihedral angles, a short introduction to the field is given and some thoughts are discussed. Directional features, such as angles, have special properties compared to ordinary numbers. Most important of all properties is that angles are periodic. That is, if $x \in \mathbb{R}$ is a real number then an angle can be defined as $\alpha = x \mod 2\pi + o$ where o is an offset or phase shift. Another way to represent an angle α is as a point on the unit circle, i.e. $\boldsymbol{p}_\alpha = (\cos\alpha, \sin\alpha)$.

A.1 Means and Differences

In order to get an impression of the specialities of angles the calculation of differences and means is demonstrated here.

Let $x_1 = 15$, $x_2 = 5$ and $x_2 = 355$ be three numbers. The differences are $|x_1 - x_2| = |15 - 5| = 10$, $|x_1 - x_3| = |15 - 355| = 340$ and $|x_2 - x_3| = |5 - 355| = 350$. Now, let $\alpha_1 = 15°$, $\alpha_2 = 5°$ and $\alpha_2 = 355°$ be three angles. Here the differences are no longer valid. The real differences are 10°, 20° and 10°, respectively. It can be shown that the formula for numbers is only valid for angles that are less than 180° apart from each other. For the other angles the periodic image, i.e. $\alpha + 360°$ or $\alpha - 360°$, of one angle has to be calculated so that one can use the difference formula for numbers again.

When calculating the mean of the numbers $x_1 = 15$, $x_2 = 5$ and $x_2 = 355$ one gets $\frac{x_1+x_2+x_3}{3} = \frac{15+5+355}{3} = 125$. The actual mean for angles $\alpha_1 = 15°$, $\alpha_2 = 5°$ and $\alpha_2 = 355°$, however, is $\frac{15°+5°+(-5°)}{3} = 5°$. Again, one has to calculate the periodic image of an angle to be able to treat them just like numbers.

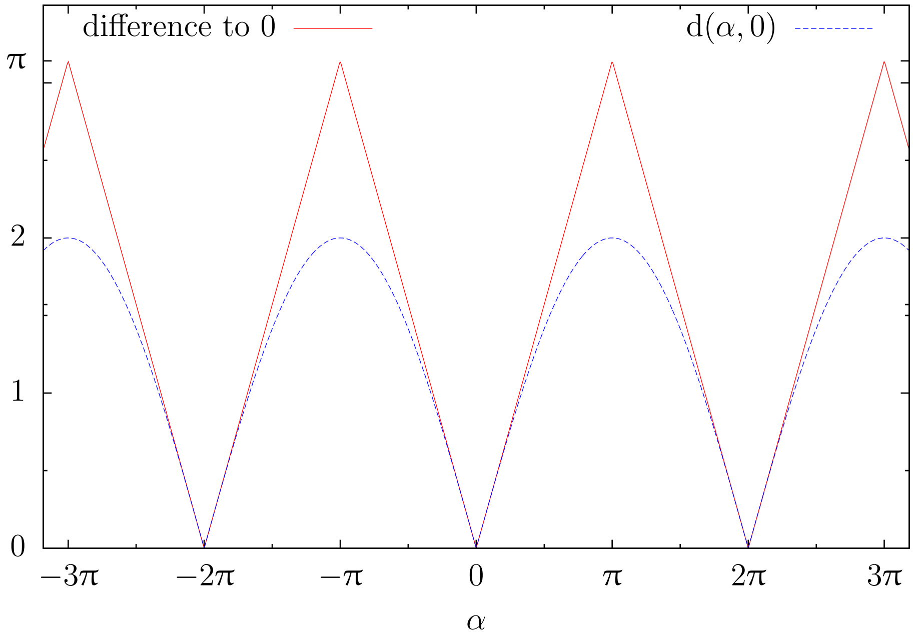


Figure A.1.1: The difference of the angle α and 0 is shown in red and the corresponding Euclidean distance on the unit disk is shown in blue.

If the angles are represented as points on the unit circle one can define an Euclidean distance measure by

$$\mathrm{d}(\alpha_1, \alpha_2) = \left|\boldsymbol{p}_{\alpha_1}, \boldsymbol{p}_{\alpha_2}\right| = \sqrt{(p_{\alpha_1\mathrm{X}} - p_{\alpha_2\mathrm{X}})^2 + (p_{\alpha_1\mathrm{Y}} - p_{\alpha_2\mathrm{Y}})^2},$$

where $p_{\alpha\mathrm{Y}} = \sin\alpha$ and $p_{\alpha\mathrm{X}} = \cos\alpha$. Unfortunately, this measure does not scale linearly with the actual difference of the angles. Figure A.1.1 gives an idea of the scaling. With this distance measure there are also attempts to perform principal component analysis on dihedral angle information obtained from molecular dynamics simulations [MNS04, ANHS07]. However, this so called dihedral angle principal component analysis (dPCA) works in the forbidden space off the unit circle or the surface of a unit sphere, if more than one angle is analysed. A better way to do PCA on spaces like surfaces of unit spheres is to perform the analysis on the manifold directly, leading to a special kind of geodesic analysis [LLV04, LV04].

A.2 Circular Distributions

When it comes to modelling angles by statistical distributions the specialities of directional features become quite important. However, specialised distributions have become available only recently. We will shortly introduce some ways to model angles.

Gaussian

The Gaussian normal distribution is quite efficient in modelling numbers. It needs only two parameters, the mean and the variance. But it does not account for the periodicity of angles. Therefore, when modelling angles directly with Gaussians one has to ensure the periodic boundary conditions. For example, if one wants to know the probability of an angle the distance to the mean has to be less than π. And, if this not the case, this can be achieved by translating the angle to its periodic image closest to the mean. Real problems occur if the variance gets close to or even over the size of a period.

Another way would be to model the point representation of the angles. Here, one has to deal with number pairs in the range $[-1,1]^2$. At the first glance, this seems to be feasible to model with bivariate Gaussians on the logarithmic values, i.e.

$$\left(\log\left(\log\left(\frac{2}{\cos\alpha+1}\right)\right),\log\left(\log\left(\frac{2}{\sin\alpha+1}\right)\right)\right)\in\mathbb{R}^2.$$

However being in the perfect range, $[-\infty,\infty]^2$, for a Gaussian model, the angle differences would overweigh close to the four singularities, i.e. at $\alpha=0,\frac{\pi}{2},\pi,\frac{3\pi}{2}$. The other problem of the use of point representations is, that the points follow a graph line and are not spread like real Gaussians, see figure A.2.1.

Multivariate Gaussian

If one has to deal with more than one angle, these angles are described by vectors $\boldsymbol{\alpha}$ of dimension k which can be modelled by the k-variate Gaussian distribution similar to the univariate case. The density is given by

$$\mathrm{N}_\mathrm{k}\big(\boldsymbol{\alpha}\,\big|\,\boldsymbol{\mu},\mathbf{C}\big)=\frac{\exp\left[-\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\mu})\mathbf{C}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})^\mathsf{T}\right]}{\sqrt{(2\pi)^\mathrm{k}|\det\mathbf{C}|}},$$

where $\boldsymbol{\mu}$ is the mean angle vector and $\mathbf{C}$ is the covariance matrix.

Wrapped Gaussian

The wrapped Gaussian is a distribution for angles [Bah06]. It seems to have the least modifications compared to the original Gaussian normal distribution. Whereas the original Gaussian is defined on numbers from $-\infty$ to $+\infty$, the wrapped Gaussian models angles $\alpha\in[0,2\pi)$. It can be defined by a sum of the

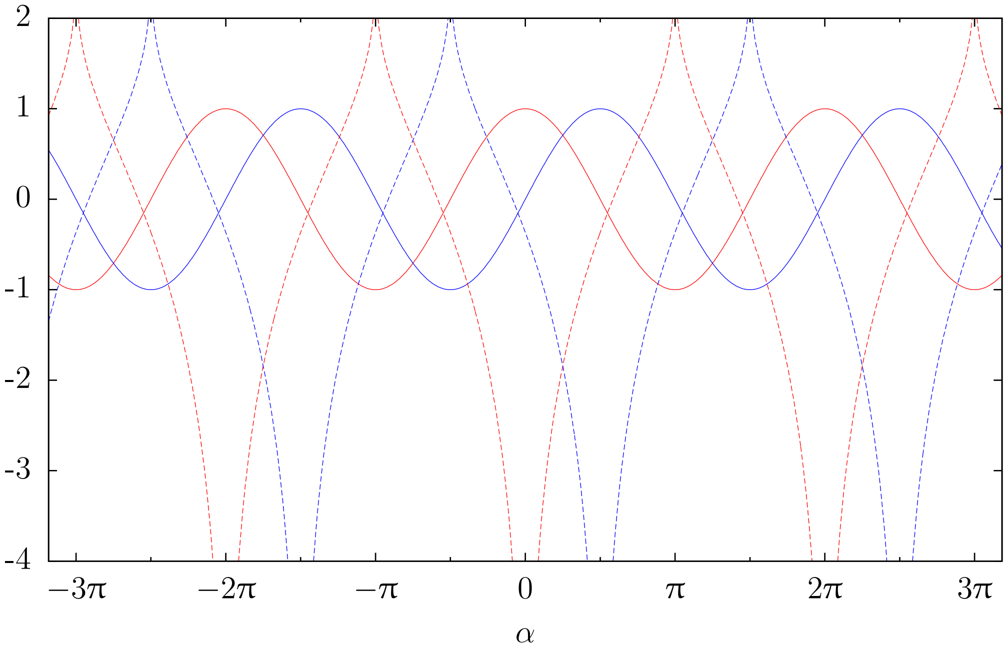


Figure A.2.1: The sine (blue) and cosine (red) of the angle α and the corresponding logarithmic representation (same colour, dashed).

normal density of all periodic images of α, given by

$$\mathrm{N}^{\mathrm{wrap}}\big(\alpha \,\big|\, \mu, \sigma\big) = \sum_{t=-\infty}^{+\infty} \mathrm{N}_1\big(\alpha + 2\pi t \,\big|\, \mu, \sigma\big).$$

This formulation, however, is known to have some drawbacks concerning the parameter estimation [MHTS07].

Multivariate wrapped Gaussian

The multivariate case looks similar [Bah06]. The density is given by

$$\mathrm{N}_{\mathrm{k}}^{\mathrm{wrap}}\big(\boldsymbol{\alpha} \,\big|\, \boldsymbol{\mu}, \mathbf{C}\big) = \sum_{t_1=-\infty}^{+\infty} \cdots \sum_{t_{\mathrm{k}}=-\infty}^{+\infty} \mathrm{N}_{\mathrm{k}}\big(\boldsymbol{\alpha} + 2\pi t_1 \mathbf{e}_1 + \cdots + 2\pi t_{\mathrm{k}} \mathbf{e}_{\mathrm{k}} \,\big|\, \boldsymbol{\mu}, \mathbf{C}\big),$$

where $\mathbf{e}_i$ is the ith Euclidean basis vector (with an entry of 1 at the ith element and 0 elsewhere).

Von Mises

The von Mises distribution is the most prominent among the univariate circular distributions and is a natural analogue to the univariate Gaussian normal distribution. For angles α its density function is given by

$$\mathrm{M}\left(\alpha \middle| \kappa, \mu\right) = \frac{\exp[\kappa \cos(\alpha - \mu)]}{2\pi \, \mathrm{I}_0(\kappa)},$$

where μ is the mean angle, $\kappa \geq 0$ is the concentration parameter and $\mathrm{I}_0(\kappa)$ is the modified Bessel function of the first kind and order 0.

Von Mises-Fisher

The von Mises-Fisher distribution is a generalisation of the von Mises distribution to the k-dimensional sphere. If $\mathrm{k} = 1$ it reduces to the von Mises distribution. For $(\mathrm{k}+1)$-dimensional point vectors of unit length, $\boldsymbol{p}_{\boldsymbol{\alpha}}$ (build by k angle variables), it is given by

$$\mathrm{MF}\left(\boldsymbol{p}_{\boldsymbol{\alpha}} \middle| \kappa, \boldsymbol{\mu}\right) = \frac{\kappa^{\frac{k-1}{2}} \exp\left[\kappa \boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{p}_{\boldsymbol{\alpha}}\right]}{(2\pi)^{\frac{k+1}{2}} \, \mathrm{I}_{\frac{k-1}{2}}(\kappa)},$$

where $\boldsymbol{\mu}$ is the mean angle vector, $\kappa \geq 0$ is the concentration parameter and $\mathrm{I}_{\frac{k-1}{2}}(\kappa)$ is the modified Bessel function of the first kind and order $\frac{k-1}{2}$. This distribution does not allow for non identical variance in the different dimensions nor for covariances.

Fisher-Bingham or Kent

The 5-parameter Fisher-Bingham or Kent distribution is an analogue to the bivariate normal distribution on the unit sphere with an unconstrained covariance matrix [Ken82]. It uses a point representation of the angle pair $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ given by

$$\boldsymbol{p}_{\boldsymbol{\alpha}} = \begin{pmatrix} \cos\alpha_1 \\ \sin\alpha_1 \cos\alpha_2 \\ \sin\alpha_1 \sin\alpha_2 \end{pmatrix}.$$

The density function is then given by

$$\mathrm{FB}_5\left(\boldsymbol{p}_{\boldsymbol{\alpha}} \middle| \kappa, \beta, \boldsymbol{\Gamma}\right) = \frac{\exp\left[\kappa \boldsymbol{\gamma}_1^{\mathsf{T}} \boldsymbol{p}_{\boldsymbol{\alpha}} + \beta \left((\boldsymbol{\gamma}_2^{\mathsf{T}} \boldsymbol{p}_{\boldsymbol{\alpha}})^2 - (\boldsymbol{\gamma}_3^{\mathsf{T}} \boldsymbol{p}_{\boldsymbol{\alpha}})^2\right)\right]}{\mathrm{c}(\kappa, \beta)},$$

where $\kappa \geq 0$ is the concentration, $\beta \geq 0$ is the ovalness and the matrix $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3)$ describes the mean direction ($\boldsymbol{\gamma}_1$), the major axis ($\boldsymbol{\gamma}_2$) and the minor axis ($\boldsymbol{\gamma}_3$). $\mathrm{c}(\kappa, \beta)$ is a normalising constant [Ken82].

Multivariate von Mises

A multivariate von Mises distribution was recently proposed [MHTS07]. Its probability density function for angle vectors $\boldsymbol{\alpha}$ of dimension k is given by

$$\mathrm{M}_{\mathrm{k}}\left(\boldsymbol{\alpha}\,\middle|\,\boldsymbol{\mu},\boldsymbol{\kappa},\boldsymbol{\Lambda}\right) = \frac{\exp\left[\boldsymbol{\kappa}^{\mathsf{T}}\boldsymbol{c}(\boldsymbol{\alpha},\boldsymbol{\mu}) + \frac{1}{2}\boldsymbol{s}(\boldsymbol{\alpha},\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{s}(\boldsymbol{\alpha},\boldsymbol{\mu})\right]}{\mathrm{T}(\boldsymbol{\kappa},\boldsymbol{\Lambda})},$$

where $\boldsymbol{c}(\boldsymbol{\alpha},\boldsymbol{\mu}) = \begin{pmatrix} \cos(\alpha_1 - \mu_1) \\ \vdots \\ \cos(\alpha_{\mathrm{k}} - \mu_{\mathrm{k}}) \end{pmatrix}$, $\boldsymbol{s}(\boldsymbol{\alpha},\boldsymbol{\mu}) = \begin{pmatrix} \sin(\alpha_1 - \mu_1) \\ \vdots \\ \sin(\alpha_{\mathrm{k}} - \mu_{\mathrm{k}}) \end{pmatrix}$, the matrix $\boldsymbol{\Lambda}$ is symmetric with only zeros on the diagonal and $\mathrm{T}(\boldsymbol{\kappa},\boldsymbol{\Lambda})$ is a normalising constant. All conditional distributions are again multivariate von Mises and the marginals are symmetric around their means and either uni- or bimodal [MHTS07, SHD02].

Although the multivariate von Mises distribution seems to be the most elegant model for the dihedral angles of protein fragments, there is no parameter estimation program available, which can deal both with discrete and continuous descriptors for fragments. Therefore and to reduce unnecessary programming, in this work multivariate Normal distributions were used with the AutoClass-C program [CS96, CPT02].

Appendix B

Analytic Derivation of the Adaptive Cooling Threshold

In this appendix the threshold used for the cooling criterion in algorithm 1 on page 45 is derived analytically. The condition for slower cooling is given if the entropy difference of the short term average and the long term average is below some threshold ΔS_{thresh}, formally $S_t^{\text{long}} - S_t^{\text{short}} < \Delta S_{\text{thresh}}$. The averages are calculated on the fly by $S_t^{\text{short}} = \beta_{\text{short}} S_{t-1}^{\text{short}} + (1-\beta_{\text{short}}) S_t$ and $S_t^{\text{long}} = \beta_{\text{long}} S_{t-1}^{\text{long}} + (1-\beta_{\text{long}}) S_t$. A derivation of ΔS_{thresh} is shown in terms of the instantaneous entropy S_t at time t and the parameters β_{long} and β_{short}. The ideal slope of the entropy is assumed to be a linear decay, that is $S_t = -mt + S_0$, where S_0 is the initial entropy and $0 < m = \frac{S_0}{t_{\max}}$ with $t_{\max}$ as the number of desired simulation steps. The actual number will be close to $t_{\max}$ only if the cooling rate is not adjusted or if the adjustments average out. In advance, it is hard to say what the actual number of steps will be. The properties of geometric sums are used to derive at a closed formula for $S_{t_n}^{\text{short}}$ or $S_{t_n}^{\text{long}}$, respectively, for some time point $t_n > 0$.

$$
\begin{array}{rcl}
S_0^{\text{short}} & = & S_0 \\
\wedge\ S_{t_n}^{\text{short}} & = & \beta_{\text{short}} S_{t_n-1}^{\text{short}} + (1-\beta_{\text{short}}) S_{t_n} \\
\hline
\Longleftrightarrow S_{t_n}^{\text{short}} & = & (1-\beta_{\text{short}}) S_{t_n} + [(1-\beta_{\text{short}}) S_{t_n-1} + [(1-\beta_{\text{short}}) S_{t_n-2} + \dots \\
 & & \quad \dots + [(1-\beta_{\text{short}}) S_1 + \beta_{\text{short}} S_0]\, \beta_{\text{short}} \dots]\, \beta_{\text{short}}]\, \beta_{\text{short}} \\
 & = & \beta_{\text{short}}^{t_n} S_0 + \beta_{\text{short}}^{t_n-1} (1-\beta_{\text{short}}) S_1 + \dots + \beta_{\text{short}}^{t_n-t_n} (1-\beta_{\text{short}}) S_{t_n} \\
 & = & \beta_{\text{short}}^{t_n} S_0 + \displaystyle\sum_{t=1}^{t_n} \beta_{\text{short}}^{t_n-t} (1-\beta_{\text{short}}) S_t
\end{array}
$$

Assuming $S_t = -mt + S_0$, then

$$\begin{aligned}
S_{t_n}^{\text{short}} &= \beta_{\text{short}}^{t_n} S_0 + (1 - \beta_{\text{short}}) \sum_{t=1}^{t_n} \beta_{\text{short}}^{t_n - t} S_t \\
&= \beta_{\text{short}}^{t_n} S_0 + (1 - \beta_{\text{short}}) \sum_{t=1}^{t_n} \beta_{\text{short}}^{t_n - t} (-mt + S_0) \\
&= \beta_{\text{short}}^{t_n} S_0 + (1 - \beta_{\text{short}}) \left[-m\beta_{\text{short}}^{t_n} \sum_{t=0}^{t_n} \beta_{\text{short}}^{-t} t + S_0 \beta_{\text{short}}^{t_n} \left(\sum_{t=0}^{t_n} \beta_{\text{short}}^{-t} - 1 \right) \right]
\end{aligned}$$

Applying the geometric sum formula leads to

$$\begin{aligned}
S_{t_n}^{\text{short}} &= \beta_{\text{short}}^{t_n} S_0 + (1 - \beta_{\text{short}}) \left[S_0 \beta_{\text{short}}^{t_n} \left(\sum_{t=0}^{t_n} \beta_{\text{short}}^{-t} - 1 \right) - m\beta_{\text{short}}^{t_n} \sum_{t=0}^{t_n} \beta_{\text{short}}^{-t} t \right] \\
&= \beta_{\text{short}}^{t_n} S_0 + (1 - \beta_{\text{short}}) \left[\begin{array}{c} S_0 \beta_{\text{short}}^{t_n} \left(\frac{\beta_{\text{short}}^{-t_n - 1} - 1}{\beta_{\text{short}}^{-1} - 1} - 1 \right) \\ -m\beta_{\text{short}}^{t_n} \frac{t_n \beta_{\text{short}}^{-t_n - 2} - (t_n + 1)\beta_{\text{short}}^{-t_n - 1} + \beta_{\text{short}}^{-1}}{(\beta_{\text{short}}^{-1} - 1)^2} \end{array} \right] \\
&= S_0 + m \frac{\beta_{\text{short}}^{t_n + 1} - \beta_{\text{short}}^{t_n} - (t_n + 1)\beta_{\text{short}} - t_n \beta_{\text{short}}^{-1} + 2t_n + 1}{\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2}
\end{aligned}$$

Using $m = \frac{S_0}{t_{\max}}$ leads to

$$\begin{aligned}
S_{t_n}^{\text{short}} &= \left[S_0 + S_0 \frac{\beta_{\text{short}}^{t_n + 1} - \beta_{\text{short}}^{t_n} - (t_n + 1)\beta_{\text{short}} - t_n \beta_{\text{short}}^{-1} + 2t_n + 1}{t_{\max} \left(\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2 \right)} \right] \\
&= S_0 \left[\begin{array}{l} \frac{t_{\max}\left(\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2\right)}{t_{\max}\left(\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2\right)} \\ \quad + \frac{\beta_{\text{short}}^{t_n + 1} - \beta_{\text{short}}^{t_n} - (t_n + 1)\beta_{\text{short}} - t_n \beta_{\text{short}}^{-1} + 2t_n + 1}{t_{\max}\left(\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2\right)} \end{array} \right] \\
&= S_0 \left[\begin{array}{l} \frac{\beta_{\text{short}}^{t_n + 1} - \beta_{\text{short}}^{t_n} + (t_{\max} - t_n - 1)\beta_{\text{short}} + (t_{\max} - t_n)\beta_{\text{short}}^{-1}}{t_{\max}\left(\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2\right)} \\ \quad + \frac{-2t_{\max} + 2t_n + 1}{t_{\max}\left(\beta_{\text{short}}^{-1} + \beta_{\text{short}} - 2\right)} \end{array} \right]
\end{aligned}$$

The optimal difference $\Delta S_{\text{thresh}} = S_t^{\text{long}} - S_t^{\text{short}}$ could now be calculated at each time t. This an undesired situation as first the assumption that the entropy follows a linear decay is not realistic, and second this calculation would be too expensive to be performed at each step of the simulation. Therefore, the optimal difference of the two averages should stay constant.

$$
\begin{aligned}
\Delta S &= S_{t_n}^{\text{long}} - S_{t_n}^{\text{short}} \\
&= \left[\begin{array}{l}
S_0 + m\frac{\beta_{\text{long}}^{tn+1}-\beta_{\text{long}}^{tn}-(t_n+1)\beta_{\text{long}}-t_n\beta_{\text{long}}^{-1}+2t_n+1}{\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2} \\
\quad -\left(S_0 + m\frac{\beta_{\text{short}}^{tn+1}-\beta_{\text{short}}^{tn}-(t_n+1)\beta_{\text{short}}-t_n\beta_{\text{short}}^{-1}+2t_n+1}{\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2}\right)
\end{array}\right] \\
&= m\left[\begin{array}{l}
\frac{\beta_{\text{long}}^{tn+1}-\beta_{\text{long}}^{tn}-(t_n+1)\beta_{\text{long}}-t_n\beta_{\text{long}}^{-1}+2t_n+1}{\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2} \\
\quad -\frac{\beta_{\text{short}}^{tn+1}-\beta_{\text{short}}^{tn}-(t_n+1)\beta_{\text{short}}-t_n\beta_{\text{short}}^{-1}+2t_n+1}{\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2}
\end{array}\right] \\
&= m\left[\begin{array}{l}
\frac{\beta_{\text{long}}^{tn+1}\beta_{\text{short}}^{-1}-\beta_{\text{long}}^{tn}\beta_{\text{short}}^{-1}-\beta_{\text{long}}\beta_{\text{short}}^{-1}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{\beta_{\text{long}}^{tn+1}\beta_{\text{short}}-\beta_{\text{long}}^{tn}\beta_{\text{short}}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{-2\beta_{\text{long}}^{tn+1}+2\beta_{\text{long}}^{tn}+\beta_{\text{long}}-\beta_{\text{long}}^{-1}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{-\beta_{\text{long}}^{-1}\beta_{\text{short}}^{tn+1}+\beta_{\text{long}}^{-1}\beta_{\text{short}}^{tn}+\beta_{\text{long}}^{-1}\beta_{\text{short}}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{-\beta_{\text{long}}\beta_{\text{short}}^{tn+1}+\beta_{\text{long}}\beta_{\text{short}}^{tn}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{2\beta_{\text{short}}^{tn+1}-2\beta_{\text{short}}^{tn}-\beta_{\text{short}}+\beta_{\text{short}}^{-1}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)}
\end{array}\right]
\end{aligned}
$$

For big enough t_n this simplifies to

$$
\begin{aligned}
\lim_{t_n\to\infty}\Delta S &= m\left[\begin{array}{l}
\frac{-\beta_{\text{long}}\beta_{\text{short}}^{-1}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{\beta_{\text{long}}-\beta_{\text{long}}^{-1}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{\beta_{\text{long}}^{-1}\beta_{\text{short}}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)} \\
\quad +\frac{-\beta_{\text{short}}+\beta_{\text{short}}^{-1}}{\left(\beta_{\text{long}}^{-1}+\beta_{\text{long}}-2\right)\left(\beta_{\text{short}}^{-1}+\beta_{\text{short}}-2\right)}
\end{array}\right] \\
&= m\frac{\left(\beta_{\text{long}}\beta_{\text{short}}-\beta_{\text{long}}-\beta_{\text{short}}+1\right)\left(\beta_{\text{long}}-\beta_{\text{short}}\right)}{\left(\beta_{\text{long}}-1\right)^2\left(\beta_{\text{short}}-1\right)^2}
\end{aligned}
$$

Bibliography

[AG96] B. Andresen and J. M. Gordon: *Constant Thermodynamic Speed Simulated Annealing*, Inverse Methods (B. Jacobsen, K. Mosegaard and P. Sibani, eds.), Lecture Notes in Earth Sciences **63**, 1996, pp. 303–311.

[AGM^{+}90] S. Altschul, W. Gish, W. Miller, E. W. Meyers and D. J. Lipman: *Basic Local Alignment Search Tool*, J. Mol. Biol. **215**, 1990, pp. 403–410.

[AHC^{+}09] P. A. Alexander, Y. He, Y. Chen, J. Orban and P. N. Bryan: *A minimal sequence code for switching protein structure and function*, Proc. Natl. Acad. Sci. U.S.A. **106**, 2009, pp. 21149–21154.

[AMS^{+}97] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman: *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*, Nucleic Acids Res. **25**, 1997, pp. 3389–3402.

[Anf73] C. B. Anfinsen: *Principles that Govern the Folding of Protein Chains*, Science **181**, 1973, pp. 223–230.

[ANHS07] A. Altis, P. H. Nguyen, R. Hegger and G. Stock: *Dihedral angle principal component analysis of molecular dynamics simulations*, Chem. Phys. **126**, 2007, pp. 244111–1–244111–10.

[Bah06] C. Bahlmann: *Directional features in online handwriting recognition*, Pattern Recogn. **39**, 2006, pp. 115–125.

[BDNBP^{+}09] M. Ben-David, O. Noivirt-Brik, A. Paz, J. Prilusky, J. L. Sussman and Y. Levy: *Assessment of CASP8 structure predictions for template free targets*, Proteins **77**, 2009, pp. 50–65.

[BL] B. W. Brown and J. Lovato: *RANLIB.C – Library of C Routines for Random Number Generation*, University of Texas. `http://orion.math.iastate.edu/burkardt/c_src/ranlib/ranlib.html`, [accessed 21. August 2011].

[BSDK09] D. S. Berkholz, M. V. Shapovalov, R. L. J. Dunbrack and P. A. Karplus: *Conformation Dependence of Backbone Geometry in Proteins*, Cell Structure **17**, 2009, pp. 1316–1325.

[BWF$^+$00] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne: *The Protein Data Bank*, Nucleic Acids Res. **28**, 2000, pp. 235–242.

[CdMaT00] S. A. Cannas, A. C. N. de Magalhães and F. A. Tamarit: *Evidence of exactness of the mean-field theory in the nonextensive regime of long-range classical spin models*, Phys. Rev. B **61**, 2000, pp. 11521–11528.

[CKF$^+$09] D. Cozzetto, A. Kryshtafovych, K. Fidelis, J. Moult, B. Rost and A. Tramontano: *Evaluation of template-based models in CASP8 with standard measures*, Proteins **77**, 2009, pp. 18–28.

[CPT02] D. Cook, J. Potts and W. Taylor: *AutoClass-C 3.3.4*, University of Texas at Arlington and NASA Ames Research Center, Jan 2002. `http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass/autoclass-c-program.html`, [accessed 22. January 2008].

[CS96] P. Cheeseman and J. Stutz: *Bayesian Classification (AutoClass): Theory and Results*, Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds.), 1996, pp. 153–180.

[CSD03] A. A. Canutescu, A. A. Shelenkov and R. L. Dunbrack: *A graph-theory algorithm for rapid protein side-chain prediction*, Prot. Sci. **12**, 2003, pp. 2001–2014.

[Dew93] T. G. Dewey: *Protein structure and polymer collapse*, Chem. Phys. **98**, 1993, pp. 2250–2257.

[DK97] M. Delarue and P. Koehl: *The inverse protein folding problem: self consistent mean field optimisation of a structure specific mutation matrix*, Pacific Symposium on Biocomputing (R. B. Altman, K. Dunker, L. Hunter, K. Lauderdale and T. E. Klein, eds.), 1997, pp. 109–121.

[Edw65] S. Edwards: *The statistical mechanics of polymers with excluded volume*, Proc. Phys. Soc. **85**, 1965, pp. 613–624.

[EH91] R. A. Engh and R. Huber: *Accurate Bond and Angle Parameters for X-ray Protein Structure Refinement*, Acta Chryst. **A47**, 1991, pp. 392–400.

[EWMR+06] N. Eswar, B. Webb, M. A. Marti-Renom, M. Madhusudhan, D. Eramian, M.-y. Shen, U. Pieper and A. Sali, *Comparative Protein Structure Modeling Using Modeller*, Current Protocols in Bioinformatics, 2006, ch. 5.6, pp. 5.6.1–5.6.30.

[FA95] D. Frishman and P. Argos: *Knowledge-based protein secondary structure assignment*, Proteins **23**, 1995, pp. 566–579.

[FF07] C. A. Floudas and H. K. Fung: *Mathematical Modeling and Optimization Methods for De Novo Protein Design*, Systems Biology I, 2007, pp. 42–66.

[Han09] B. Hansen: *Evaluation of Protein Structure Prediction Methods*, Software project report, ZBH - Centre for Bioinformatics, Universität Hamburg, Hamburg, 2009.

[HBA+07] H. Hansson, G. Berglund, E. Andersson, M. Sandgren and M. Selmer: *Introduction to protein structures: The oxygen binding proteins of muscle and blood*, Uppsala Universitet, Feb 2007. `http://xray.bmc.uu.se/kurs/BiostrukfunkX2/Practical_1/practical_1.html`, [accessed 21. January 2008].

[HH92] S. Henikoff and J. G. Henikoff: *Amino acid substitution matrices from protein blocks*, Proc. Natl. Acad. Sci. U.S.A. **89**, 1992, pp. 10915–10919.

[HKK06] T. Hamelryck, J. T. Kent and A. Krogh: *Sampling Realistic Protein Conformations Using Local Structural Bias*, PLoS Comput. Biol. **2**, 2006, pp. 1121–1133.

[HKL+98] E. S. Huang, P. Koehl, M. Levitt, R. V. Pappu and J. W. Ponder: *Accuracy of Side-Chain Prediction Upon Near-Native Protein Backbones Generated by Ab Initio Folding Methods*, Proteins **33**, 1998, pp. 204–217.

[Hof07] S. Hoffmann, *Using index based techniques in protein structure comparison*, Master's thesis, ZBH - Centre for Bioinformatics, Universität Hamburg, Hamburg, 2007.

[HT04] J. B. Holmes and J. Tsai: *Some fundamental aspects of building protein structures from fragment libraries*, Prot. Sci. **13**, 2004, pp. 1636–1650.

[JAC+08] L. Jiang, E. A. Althoff, F. R. Clemente, L. Doyle, D. Röthlisberger, A. Zanghellini, J. L. Gallaher, J. L. Betker, F. Tanaka, C. F. Barbas, D. Hilvert, K. N. Houk, B. L. Stoddard and D. Baker: *De Novo Computational Design of Retro-Aldol Enzymes*, Science **319**, 2008, pp. 1387–1391.

[KAS$^+$09] R. L. Koder, J. L. R. Anderson, L. A. Solomon, K. S. Reddy, C. C. Moser and P. L. Dutton: *Design and engineering of an O_2 transport protein*, Nature **458**, 2009, pp. 305–310.

[KAV05] S. K. Koh, G. K. Ananthasuresh and S. Vishveshwara: *A Deterministic Optimization Approach to Protein Sequence Design Using Continuous Models*, Int. J. Robot. Res. **24**, 2005, pp. 109–130.

[KB00] B. Kuhlman and D. Baker: *Native protein sequences are close to optimal for their structures*, Proc. Natl. Acad. Sci. U.S.A. **97**, 2000, pp. 10383–10388.

[KD96] P. Koehl and M. Delarue: *Mean-field minimization methods for biological macromolecules*, Curr. Opin. Struct. Biol. **6**, 1996, pp. 222–226.

[KD98] P. Koehl and M. Delarue: *Building protein lattice models using self-consistent mean field theory*, Chem. Phys. **108**, 1998, pp. 9540–9549.

[KDS$^+$60] J. C. Kendrew, R. E. Dickerson, B. E. Strandberg, R. G. Hart, D. R. Davies, D. C. Phillips and V. C. Shore: *Structure of Myoglobin: A Three-Dimensional Fourier Synthesis at 2Å. Resolution*, Nature **185**, 1960, pp. 422–427.

[Ken82] J. T. Kent: *The Fisher-Bingham Distribution on the Sphere*, J. Roy. Statist. Soc. Ser. B **44**, 1982, pp. 71–80.

[KGV83] S. Kirkpatrick, C. D. J. Gelatt and M. P. Vecchi: *Optimization by Simulated Annealing*, Science **220**, 1983, pp. 671–680.

[Kre98] U. Krengel, *Einführung in die Wahrscheinlichkeitstheorie und Statistik*, ed. 4, 1998.

[KS83] W. Kabsch and C. Sander: *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features*, Biopolymers **22**, 1983, pp. 2577–2637.

[KTJG02] A. Kloczkowski, K.-L. Ting, R. L. Jernigan and J. Garnier: *Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence*, Proteins **49**, 2002, pp. 154–166.

[Küh10] J. Kühl, *Probabilistic Prediction and Reconstruction of Protein Loops*, Master's thesis, ZBH - Centre for Bioinformatics, Universität Hamburg, Hamburg, 2010.

[LBXL08] S. C. Li, D. Bu, J. Xu and M. Li: *Fragment-HMM: A new approach to protein structure prediction*, Prot. Sci. **17**, 2008, pp. 1925–1934.

[LLV04] J. A. Lee, A. Lendasse and M. Verleysen: *Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis*, Neurocomputing **57**, 2004, pp. 49–76.

[LV04] J. A. Lee and M. Verleysen: *How to project 'circular' manifolds using geodesic distances?*, ESANN 2004 - European Symposium on Artificial Neural Networks, 2004, pp. 223–230.

[Mah36] P. C. Mahalanobis: *On the Generalized Distance in Statistics*, Proc. Indian Inst. Sci. **2**, 1936, pp. 49–55.

[Mah09] N. Mahmood, *Protein Structure Prediction using Coarse Grain Force Fields*, Ph.D. thesis, ZBH - Centre for Bioinformatics, Universität Hamburg, Hamburg, 2009.

[MBCS01] J. Mendes, A. M. Baptista, M. A. Carrondo and C. M. Soares: *Implicit solvation in the self-consistent mean field theory method: sidechain modelling and prediction of folding free energies of protein mutants*, J. Comput.-Aided Mol. Des. **15**, 2001, pp. 721–740.

[MFK$^+$] J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost and A. Tramontano: *Protein Structure Prediction Center*, University of California, Davis. `http://www.predictioncenter.org`, [accessed 8. July 2008].

[MH93] J. Martin and F. U. Hartl: *Protein folding in the cell: molecular chaperones pave the way*, Structure **1**, 1993, pp. 161–164.

[MHTS07] K. V. Mardia, G. Hughes, C. C. Taylor and H. Singh: *A Multivariate von Mises Distribution with Applications to Bioinformatics*, Research Report STAT07-03, University of Leeds, Department of Statistics, School of Mathematics, Leeds, 2007.

[MM03] G. L. Moore and C. D. Maranas: *Identifying residue-residue clashes in protein hybrids by using a second-order mean-field approach*, Proc. Natl. Acad. Sci. U.S.A. **100**, 2003, pp. 5091–5096.

[MNS04] Y. Mu, P. H. Nguyen and G. Stock: *Energy Lanscape of a Small Peptide Revealed by Dihedral Angle Principal Component Analysis*, Proteins **58**, 2004, pp. 45–52.

[MST09] T. Margraf, G. Schenk and A. E. Torda: *The SALAMI Protein Structure Search Server*, Nucleic Acids Res., 2009.

[MT08] T. Margraf and A. E. Torda: *HANSWURST: Fast Efficient Multiple Protein Structure Alignments*, From Computational Biophysics to Systems Biology (U. H. E. Hansmann, J. H. Meinke, S. Mohanty, W. Nadler and O. Zimmermann, eds.), NIC Series **40**, 2008, pp. 313–316.

[Mur03] K. P. N. Murthy, *An Introduction to Monte Carlo Simulation of Statistical Physics Problems*, 2003.

[NA98] Y. Nourani and B. Andresen: *A comparison of simulated annealing cooling strategies*, J. Phys. A: Math. Gen. **31**, 1998, pp. 8373–8385.

[OBH+99] C. Orengo, J. Bray, T. Hubbard, L. LoConte and I. Sillitoe: *Analysis and assessment of ab initio three-dimensional prediction, secondary structure, and contacts prediction*, Proteins **37**, 1999, pp. 149–170.

[OCE+03] C. A. Ouzounis, R. M. R. Coulson, A. J. Enright, V. Kunin and J. B. Pereira-Leal: *Classification Schemes for Protein Structure and Function*, Nature Rev. Genet. **4**, 2003, pp. 508–519.

[PDBa] *The Protein Data Bank website.* `http://www.pdb.org`, [accessed 8. April 2009].

[PDBb] *The PDBSelect50 cluster service.* `http://www.pdb.org/pdb/rest/representatives?cluster=50`, [accessed 13. July 2010].

[PGH+04] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng and T. E. Ferrin: *UCSF Chimera - A visualization system for exploratory research and analysis*, J. Comput. Chem. **25**, 2004, pp. 1605–1612.

[Pol95] D. S. G. Pollock: *Lectures in Mathematical Statistics*, University of Leicester, 1995. `http://www.le.ac.uk/users/dsgp1/COURSES/MATHSTAT/PROSTAST.HTM`, [accessed 18. August 2011].

[PPMH10] J. Paulsen, M. Paluszewski, K. V. Mardia and T. Hamelryck: *A probabilistic model of hydrogen bond geometry in proteins*, 29th Leeds Annual Statistical Research Workshop (A. Gusnanto, K. Mardia, C. Fallaize and J. Voss, eds.), 2010, pp. 61–64.

[PRC+60] M. F. Perutz, M. G. Rossmann, A. F. Cullis, H. Muirhead, G. Will and A. C. T. North: *Structure of Haemoglobin: A Three-Dimensional Fourier Synthesis at 5.5-Å. Resolution, Obtained by X-Ray Analysis*, Nature **185**, 1960, pp. 416–422.

[PW02] N. A. Pierce and E. Winfree: *Protein Design is NP-hard*, Prot. Eng. **15**, 2002, pp. 779–782.

[RD88] L. Regan and W. F. DeGrado: *Characterization of a helical protein designed from first principles*, Science **241**, 1988, pp. 976–978.

[RFRO96] B. A. Reva, A. V. Finkelstein, D. S. Rykunov and A. J. Olson: *Building self-avoiding lattice models of proteins using a self-consistent field optimization*, Proteins **26**, 1996, pp. 1–8.

[RR07] D. C. Richardson and J. S. Richardson: *The Anatomy and Taxonomy of Protein Structure*, Adv. Protein Chem., **34**, 1981–2007.

[RRS63] G. N. Ramachandran, C. Ramakrishnan and V. Sasisekharan: *Stereochemistry of polypeptide chain configurations*, J. Mol. Biol. **7**, 1963, pp. 95–99.

[RW95] A. Radzicka and R. Wolfenden: *A proficient enzyme*, Science **267**, 1995, pp. 90–93.

[SDB$^+$08] C. Stordeur, R. Dallüge, O. Birkenmeier, H. Wienk, R. Rudolph, C. Lange and C. Lücke: *The NMR solution structure of the artificial protein M7 matches the computationally designed model*, Proteins **72**, 2008, pp. 1104–1107.

[SHD02] H. Singh, V. Hnizdo and E. Demchuk: *Probabilistic model for two dependent circular variables*, Biometrika **89**, 2002, pp. 719–723.

[Sip90] M. J. Sippl: *Calculation of Conformational Ensembles from Potential of Mean Force*, J. Mol. Biol. **213**, 1990, pp. 859–883.

[SJ09] M. Suárez and A. Jaramillo: *Challenges in the computational design of proteins*, J. R. Soc. Interface **6**, 2009, pp. S477–S491.

[SKHB97] K. T. Simons, C. Kooperberg, E. Huang and D. Baker: *Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions*, J. Mol. Biol. **268**, 1997, pp. 209–225.

[SKO97] J. Skolnick, A. Kolinski and A. R. Ortiz: *MONSSTER: a method for folding globular proteins with a small number of distance restraints*, J. Mol. Biol. **265**, 1997, pp. 217 – 241.

[SMT08a] G. Schenk, N. Mahmood and A. E. Torda: *The GN-Score Webservice*, University of Hamburg and ZBH Centre for Bioinformatics, May 2008. `http://cardigan.zbh.uni-hamburg.de/~mahsch/qa-ms/`.

[SMT08b] G. Schenk, T. Margraf and A. E. Torda: *Protein sequence and structure alignments within one framework*, Algorithms Mol. Biol. **3**, 2008.

[SP10] J. J. Schneider and M. Puchta: *Investigation of acceptance simulated annealing – A simplified approach to adaptive cooling schedules*, Physica A **389**, 2010, pp. 5822–5831.

[SSG+00] C. T. Shih, Z. Y. Su, J. F. Gwan, B. L. Hao, C. H. Hsieh and H. C. Lee: *Mean-Field HP Model, Designability and Alpha-Helices in Protein Structures*, Phys. Rev. Lett. **84**, 2000, pp. 386–389.

[ST] G. Schenk and A. E. Torda: *Self-consistent mean field optimization of proteins with statistical scoring*, in preperation.

[ST08] G. Schenk and A. E. Torda: *The G-Opt Webservice*, University of Hamburg and ZBH Centre for Bioinformatics, Apr 2008. `http://cardigan.zbh.uni-hamburg.de/~mahsch/schenk/`.

[Sto05] J. R. Stone: *Self-consistent Hartree-Fock mass formulae: a review*, J. Phys. G: Nucl. Part. Phys. **31**, 2005, pp. R211–R230.

[TH95] R. Tafelmayer and K. H. Hoffmann: *Scaling features in complex optimization problems*, Comput. Phys. Commun. **86**, 1995, pp. 81–90.

[Tor04] A. E. Torda: *Protein Sequence Optimization–Theory, Practice, and Fundamental Impossibility*, Soft Materials **2**, 2004, pp. 1–10.

[TPH04] A. E. Torda, J. B. Procter and T. Huber: *Wurst: A protein threading server with a structural scoring function, sequence profiles and optimized substitution matrices*, Nucleic Acids Res. **32**, 2004, pp. W532–W535.

[vdBWDE00] B. van den Berg, R. Wain, C. M. Dobson and R. J. Ellis: *Macromolecular crowding perturbs protein refolding kinetics: implications for folding inside the cell*, EMBO **19**, 2000, pp. 3870–3875.

[WKM+10] T. Williams, C. Kelley, P. Mikulik et al.: *GNUPlot 4.4*, Mar 2010. `http://www.gnuplot.info`, [accessed 27. May 2010].

[WR02] J. M. Word and D. C. Richardson: *kin2Dcont 1.8*, Duke University, Jul 2002. `http://pibs.duke.edu/software/kincon.php`, [accessed 27. May 2010].

[XJR03] E. P. Xing, M. I. Jordan and S. Russell: *A generalized mean field algorithm for variational inference in exponential families*, 19th Conference on Uncertainty in Artificial Intelligence, 2003.

Curriculum Vitae

Research:

09/11 – dato	Post-doctoral researcher Biological small-angle X-ray scattering Dr Dmitri Svergun European Molecular Biology Laboratory Hamburg Outstation (Germany)
08/06 – 06/11	Dr rer. nat. in structural bioinformatics University of Hamburg (Germany) Thesis: "The Development of Nearly-Deterministic Methods for Optimising Protein Geometry" Prof. Dr Andrew E. Torda Centre for Bioinformatics Hamburg

Study:

10/03 – 06/06	German Diplom in bioinformatics Main emphasis: Data analysis and prediction University of Hamburg (Germany) Thesis: "Image Alignment for Time-Series Analysis of Protein Crystallisation Trials" Dr Victor Lamzin European Molecular Biology Laboratory Hamburg Outstation
10/00 – 09/03	Bachelor of science in applied computer science Applying subject: Molecular biology and genetics University of Göttingen (Germany) Thesis: "Separating DNA Sequences with Support Vector Machines" Prof. Dr Stephan Waack Institute for Numerical and Applied Mathematics
09/99 – 09/00	Undergraduate courses in physics and mathematics Faculty of Physics and Faculty of Mathematics University of Göttingen (Germany)

Education:

07/85 – 06/98	German Abitur (university entrance qualification) 10 years in Germany, 2 in Bourgas (Bulgaria), 1 in Athens (Greece)

Publications

Journal Articles:

"Self-consistent mean field optimization of proteins with statistical scoring"
G. Schenk and A. E. Torda, in preparation

"The SALAMI Protein Structure Search Server"
Th. Margraf, G. Schenk and A. E. Torda, Nucleic Acids Res. 2009

"Protein Sequence and Structure Alignments within one Framework"
G. Schenk, Th. Margraf and A. E. Torda, Algorithms Mol. Biol. 2008

Conference contributions:

"Sequence Optimization in Probabilistic Fields"
G. Schenk and A. E. Torda (Talk)
German Conference on Bioinformatics, Brunswick (Germany) 2010

"Narrowing Down Probabilistic Protein Space"
G. Schenk and A. E. Torda (Talk)
Methods of Molecular Simulation, Heidelberg (Germany) 2009

"Protein Sequence and Structure Optimisation in one Probabilistic Framework"
G. Schenk and A. E. Torda (Talk)
Intelligent Systems for Molecular Biology and European Conference on Computational Biology, Stockholm (Sweden) 2009

"Fragment Assembly in Probabilistic Fields"
G. Schenk and A. E. Torda
Critical Assessment of Techniques for Protein Structure Prediction, Cagliari (Italy) 2008

"Nearly Deterministic Methods for Optimising Protein Geometry"
G. Schenk and A. E. Torda
Proceedings: From Computational Biophysics to Systems Biology, Jülich (Germany) 2008
European BioPerspectives and BioTechnica, Hanover (Germany) 2008
Computer Simulation and Theory of Macromolecules, Hünfeld (Germany) 2008
Methods of Molecular Simulation, Heidelberg (Germany) 2007

"Image Alignment for Time Series Analysis of Protein Crystallisation Trials"
G. Schenk and A. E. Torda
German Conference on Bioinformatics, Tübingen (Germany) 2006

"Bayesian Fragmented Protein Comparisons"
G. Schenk and A. E. Torda
German Conference on Bioinformatics, Hamburg (Germany) 2005

Acknowledgements

The research project was funded by the city of Hamburg by means of a three-year half-time wiss. Mitarbeiter position and a one-year Abschlussstipendium according to HmbNFG at the University of Hamburg.

I would like to express my deepest gratitude to Prof. Dr. Andrew Torda for the opportunity to work on my doctoral research in his group. He has always been available for idea-sparking discussions with his profound knowledge and continuously motivated me by showing a serious interest in my work.

Prof. Dr. Jianwei Zhang has been so kind to examine my work from the point of view of a computer scientist. Although very busy, he was happily willing to let me explain my research to him. I appreciated this very much.

The secretary, Frau Annette Schade, helped me with all bureaucratic issues of the university. I perceive her as the soul of the group and would like to warmly thank her for the welcome atmosphere.

I very much appreciated the friendly and inspiring working atmosphere with my colleagues, Stefan Bienert, Björn Hansen, Jörn Lenz, Nasir Mahmood, Thomas Margraf, Marco Matthies, Martin Mosisch and Paul Reuter. Together with Jens Kleesiek and Tim Wiegels we had dinner at various occasions. Moreover, I enjoyed the daily recreative lunch breaks with the ZBH staff members.

It would be only fair to mention my project and Master students Birutè Frercks, Michał Lorenc, Björn Hansen, Jacques Kühl, Dalia Klundt, Selim Keskin and Patrick Löffler, who assisted in my research.

Also, I would like to thank my former fellow student, Akira Hattesohl. With him I had several entertaining chats and discussions about all sorts of topics concerning scientific writing.

Without my lovely girl friend and knowing partner, Svenja Riekeberg, I would not know how I came this far. She was always there with her continuous support and ability to motivate me, when it was needed.