LU TP 01-20
May 31, 2001

Quantifying Local Reliability of Sequence Alignments Using Mean Field Annealing

Master of Science Thesis by Maximilian Schlosshauer

Thesis Advisors:
Carsten Peterson, Mattias Ohlsson

LUND UNIVERSITY

DEPARTMENT OF THEORETICAL PHYSICS
LUND UNIVERSITY

Abstract

The subject of this thesis is the introduction of a novel method for quantifying the local reliability of algorithmically obtained sequence alignments.

Comparing protein sequences by means of an alignment algorithm has become an ubiquitous task of modern molecular biology. From such an alignment, evolutionary, structural and functional relationships between proteins can be delineated. However, as the alignment is obtained from a mathematical optimization algorithm that in turn is based on a set of parameters, it is inevitable to raise the question whether similiarities deduced from such an alignment actually mirror biological truth. Hence, a procedure is desired that assesses the reliability of sequence alignments produced by an algorithm.

In this work, we will present a novel method that allows the assignment of a reliability index to every single aligned pair in the alignment. The technique is based on a recast of the classical Needleman–Wunsch algorithm in terms of mean field annealing that allows an implicit quantification of the extent of local alternatives to the alignment problem in question. The study of suboptimal solutions to deduce a measure of reliability has commonly been used also in earlier works; our method, however, exhibits several important advantages. The reliability measure arises naturally from the dynamics of the optimization problem of sequence alignment and does therefore not require the introduction of a separate algorithm distinct from the alignment problem itself. The obtained reliability measure can directly be translated into an absolute value of likelihood for a correct alignment, without the need for additional calibration with external reference data. Finally, it can equally be assigned to gapped regions.

An extensive validation of our reliability measure with a large set of reference alignments has shown superiority of the method over a typical representative [16] of previous approaches.

Contents

1 Introduction **1**

2 Sequence Alignment **4**
- 2.1 The Definition of the Correct Alignment 4
- 2.2 Scoring Alignments 6
  - 2.2.1 The similarity matrix 7
  - 2.2.2 Gap penalties 7
  - 2.2.3 The alignment score 8
- 2.3 The Alignment Matrix Representation 9
- 2.4 Algorithms for Sequence Alignment 12
  - 2.4.1 Sequence alignment and network routing 12
  - 2.4.2 The Needleman–Wunsch algorithm 13

3 Reliability of Sequence Alignments **17**
- 3.1 The Nature of the Problem 17
  - 3.1.1 The significance of an alignment 17
  - 3.1.2 The local reliability of an alignment 18
- 3.2 Current Methods 19
  - 3.2.1 Delineating reliably aligned regions 20
  - 3.2.2 Quantifying reliability of individual pairs 22

4 Methods **27**
- 4.1 Introduction to Mean Field Annealing 28
  - 4.1.1 Optimization problems and spin systems 28
  - 4.1.2 Simulated annealing 29
  - 4.1.3 The mean field equations 30
- 4.2 Implementing Mean Field Annealing 31
  - 4.2.1 Introducing spin variables 32
  - 4.2.2 Introducing an energy function 33
  - 4.2.3 Applying the mean field approximation 35
- 4.3 Obtaining a Reliability Measure 37
  - 4.3.1 Monitoring mean field dynamics 37
  - 4.3.2 Simulated smelting 40

4.3.3 Choice of integration limits 41
4.3.4 Summary of the algorithm 42
4.3.5 Implementation of the algorithm 43
4.4 Validation Data 46

5 Results and Discussion 49
5.1 Reliability Index and Correct Alignment 49
5.1.1 Similarity class 40–50% 50
5.1.2 Classes of lower similarity 52
5.1.3 Influences on the performance 53
5.2 Computational Effort 55
5.2.1 Scaling of the computational effort 55
5.2.2 Absolute computational effort 57

6 Summary and Outlook 58

Chapter 1

Introduction

DNA and proteins are the two essential macromolecules for life on earth. DNA, the carrier of all genetic information, and proteins, omnipresent in virtually all biological processes, can be viewed as chains whose links are drawn from a limited number of fundamental building blocks. In the case of DNA, four building blocks (nucleotides) code the entire genome which is expressed in form of 20 different amino acids (residues). The amino acids, in turn, play the role of being the building blocks of proteins. The particular composition of the DNA or protein chain in terms of nucleotides or amino acids can be represented by a string of letters drawn from a four and 20-letter alphabet, respectively, where each letter uniquely denotes one of the building blocks, nucleotides or amino acids. Such a string is said to represent the *sequence* of the DNA or the protein. In what follows, we shall consider the case of proteins only, but our method presented in this work can in general be expected to be similarily applicable to DNA sequences.

Sequence comparison. During the last decades, the comparison of protein sequences has become an ubiquitous task of molecular biology. As highly efficient sequencing techniques rapidly increase the number of known sequences, each new sequence is routinely compared to previously determined sequences to infer evolutionary, structural and functional homologies. For instance, the functional properties of proteins are to a large extent dominated by the local and global three-dimensional structure. However, direct resolution of this structure using laboratory techniques such as NMR and X-ray crystallography is frequently a cumbersome and costly procedure, and structural predictions derived from the sequence using sophisticated computer models are not yet feasible with respect to reliability and computational effort. On the other hand, given the same environment, the structure of two proteins is entirely dictated by their amino acid sequence alone which can be determined fairly easily. This allows one to deduce structural and thus functional properties if a comparison of the protein's sequence with the sequence of a protein of known structure yields appropriate homologies. Similarily, the evolutionary distance between two protein sequences can be determined from which an evolutionary tree can be inferred that depicts the relations between allegedly very different proteins.

Sequence alignment. The comparison of the sequences of two proteins can be performed by means of an algorithm that mutually *aligns* the two sequences. Each possible pairing of amino acids is assigned a defined degree of similarity, which, for a pair of two different residues, can biologically be viewed as a measure for the probability that the two amino acids change into each other by a mutation in the DNA region that codes the amino acid, a process denoted as *substitution*. The alignment task would be a trivial translational shift of the two sequences with respect to each other if we did not allow for the insertion of *gaps*. A gap can indistinguishably correspond to either a deletion or insertion of a letter in one of the sequences, or from a biological point of view, to a deletion or insertion of nucleotides in the coding DNA region of one of the sequences. Insertion of gaps by the alignment algorithm corresponds to a mutational process of certain probability to which an appropriate *gap penalty* is assigned. Adding up similiarity scores and subtracting gap penalties along the alignment defines the *alignment score*. The purpose of the alignment algorithm becomes then to determine the *optimal alignment* that maximizes the mutual similarity between the two sequences, quantified by the alignment score. The resulting alignment can be depicted by writing one sequence on top of the other in such way that an aligned pair of letters or a letter aligned to a gap share the same column. Sequence alignment and its algorithmical realization shall be described in more detail in Chapter 2 of this work.

Reliability. Apparently, the alignment produced by an algorithm is essentially just the result of a mathematical optimization procedure which in turn depends upon the choice of the similarity and gap penalty parameters. Relationships between proteins, however, might be imposed by a multiplicity of other factors that dominate over the requirement of a pure maximization of similarity. Therefore, it is of utmost importance to raise the question of the local and global *reliability* of alignments produced by the algorithm, if the evolutionary, structural and functional properties of proteins deduced from these alignments should serve as a meaningful mirror of biological truth.

Various methods have been proposed in the past to define and to quantify reliability of algorithmically obtained sequence alignments; a review will be given in Chapter 3, together with a more detailed discussion of the issues of alignment reliability. The availability of a suitable reliability measure would allow the identification of regions in the two sequences that are sufficiently reliably aligned to be used for the desired deduction of common properties.

Current approaches to alignment reliability are mainly based on a study of suboptimal solutions to the alignment problem in question; regions in the alignment that remain unaltered among a large set of suboptimal alignments are considered to be more reliably aligned than areas that are found to be represented only in a few suboptimal alignments. Previous realizations of this idea, however, often exhibit certain drawbacks, such as complicated algorithmical structures, dependency on external databases of reference alignments, and the inability to assign reliability to gapped regions in the alignment.

Our method. In this work, a novel method is presented that elegantly resolves these problems. The sequence alignment algorithm is recasted in terms of a mean field annealing approach, a technique that has been successfully applied to various optimization problems in the past. This allows a natural deduction of a reliability measure for each aligned pair from the dynamics of the optimization task of sequence alignment itself. This reliability measure has a direct and intuitive interpretation as the likelihood for a correct alignment, does not require any calibration with an external database and can be assigned to gapped regions in the same way as to aligned residues. Our method will be described in detail in Chapter 4. The algorithm has been implemented into a computer program for extensive validation; results of these studies will be presented in Chapter 5. In particular, we shall compare the performance of our technique to the method of [16].

Chapter 2

Sequence Alignment

This chapter provides an introduction to sequence alignment and its algorithmical realization. First of all, in order to define the goal of sequence alignment and to validate algorithms for sequence alignment, we need to establish a meaningful definition of the *correct* alignment, often referred to as the *standard of truth*; Sec. 2.1 will be concerned with this issue. Then, in Sec. 2.2, we shall discuss the scoring parameters that serve as the basis for the determination of the optimal alignment by the sequence alignment algorithm. Sec. 2.3 will introduce an *alignment matrix* that allows a direct visualization of the space of possible alignments. Finally, we shall present algorithms for sequence alignment in Sec. 2.4.

2.1 The Definition of the Correct Alignment

Motivation. In general, the purpose of a sequence alignment is to delineate regions of similarity between proteins, and we may view an alignment as 'correct' if it properly pinpoints these relations. However, the similarities can be manifold; we may have evolutionary, structural and functional relations, that may but in general do not need to overlap. In itself, any alignment is essentially an artificial construct that can not be verified through a comparison with some real existing counterpart in nature. As a consequence, the correct alignment becomes a matter of definition. At least two possible definitions arise immediately; 1) the *evolutionarily correct* alignment that delineates evolutionary relationships, and 2) the *structurally correct* alignment that mirrors similarities in the local and global 3-D structures of the proteins. Functional relationships, in contrary, are usually *derived* from evolutionary and structural similarities and are therefore not suitable for a direct definition of the standard of truth.

The evolutionarily correct alignment. Suppose we are given two protein sequences $\mathbf{A} = (A_1 A_2 \ldots A_M)$ and $\mathbf{B} = (B_1 B_2 \ldots B_N)$ composed of M

1-letter code	3-letter code	amino acid name
A	Ala	alanine
C	Cys	cysteine
D	Asp	aspartate
E	Glu	glutamate
F	Phe	phenylalanine
G	Gly	glycine
H	His	histidine
I	Ile	isoleucine
K	Lys	lysine
L	Leu	leucine
M	Met	methionine
N	Asn	asparagine
P	Pro	proline
Q	Gln	glutamine
R	Arg	arginine
S	Ser	serine
T	Thr	threonine
V	Val	valine
W	Trp	tryptophan
Y	Tyr	tyrosine

Table 2.1: The letter coding of the standard 20 amino acids that are directly expressed by a translation of the genetic code.

and N amino acids, respectively, where each A_i, B_j denotes a letter in the standard 20-letter alphabet of amino acids (Tab. 2.1). By convention, the sequence string is always written from the 5' end (N terminus) to the 3' end (C terminus). From an evolutionary point of view, we might regard these two sequences as having evolved from a common ancestor sequence $\mathbf{O} = (o_1 o_2 \ldots o_L)$ of length L by distinct mutations in the corresponding coding DNA regions. The evolutionarily correct alignment could then be defined as the particular alignment that optimally reflects the true series of evolutionary events which transformed the ancestor sequence into the two distinct sequences **A** and **B**.

As an example, let us consider the two sequences $\mathbf{A}$ = ATTEV and $\mathbf{B}$ = GSTEV. We could think of these two sequences as having evolved from the common ancestor sequence $\mathbf{O}$ = GSTTEV through substitutions and deletions in the following way (residues to be changed in the next step are marked in bold face):

$$\begin{aligned} &\mathbf{G}\mathrm{STTEV} \xrightarrow{\mathrm{del}} \text{–}\mathbf{S}\mathrm{TTEV} \xrightarrow{\mathrm{subst}} \text{–}\mathrm{ATTEV},\\ &\mathrm{GST}\mathbf{T}\mathrm{EV} \xrightarrow{\mathrm{del}} \mathrm{GST}\text{–}\mathrm{EV}, \end{aligned}$$

where the symbol '–' stands for a gap. Hence, the evolutionarily correct alignment would be

$$\begin{array}{cccccc} - & \mathrm{A} & \mathrm{T} & \mathrm{T} & \mathrm{E} & \mathrm{V} \\ \mathrm{G} & \mathrm{S} & \mathrm{T} & - & \mathrm{E} & \mathrm{V}. \end{array} \tag{2.1}$$

We shall see that algorithms for sequence alignment mimick, in essence, this series of evolutionary events in order to determine the evolutionarily correct alignment, given the two final sequences. However, since the true series of mutational events that have occured in the evolution of the two sequences is usually unknown, the evolutionarily correct alignment is more an idealized than a practicable standard of truth. For the evaluation of sequence algorithms, we therefore need another reference that is readily available.

The structurally correct alignment. Such a practical standard of truth is commonly deduced from a superposition of the three-dimensional structures of the proteins, which defines the structurally correct alignment. Here, the chains of protein pairs with resolved 3-D structure are optimally matched to each other such that the mutual geometrical distance is minimized. Each two residues, one from each chain, that comes to lie spatially closest to each other, define an aligned residue pair in the two sequences. A residue in one chain that is geometrically not matched to a nearby residue of the other chain implies a gap in the sequence alignment. In this sense, a reference sequence alignment can be deduced from the structural alignment.

It turns out that in a vast range of cases, structural relationship between proteins also implies evolutionary and functional relationship, and vice versa. That is, a region in the structurally correct alignment that indicates similarity between the proteins will presumably also delineate a connection from an evolutionary point of view. Therefore, forced by the general absence of explicit evolutionary information, alignments obtained from structural superpositions have been widely used as the standard of truth in benchmarking algorithms for sequence alignments. In this work, we shall follow this common approach; whenever we refer to the term “correct alignment” or “standard of truth”, we mean the alignment obtained from a structural superposition.

2.2 Scoring Alignments

Referring to our outline in the introduction, we can formulate the goal of a sequence alignment algorithm concisely as the maximization of similarity between the two sequences with a minimum number of inserted gaps. To this extent, we will now introduce measures that score the alignment of a particular residue–residue pair or the insertion of gaps, and on which the score that quantifies the overall similarity between the two aligned sequences will be based.

2.2.1 The similarity matrix

Simple approaches. The *similarity matrix* is a symmetrical 20×20 matrix of numbers that defines a similarity score for each possible amino acid pairing. Commonly, the matrix values are deduced from the frequencies of mutations that have been observed in closely related proteins. A low probability for a particular substitution of a residue X by a different residue Y implies that the alignment of the two residues is unfavorable from an evolutionary point of view and is hence associated with a low similarity, and vice versa.

The knowledge of the substitution probabilities allows a sequence alignment algorithm to determine the alignment that most likely models the true series of evolutionary events. For example, a change in the position of the first gap with respect to the alignment (2.1) above yields another possible alignment,

$$\begin{array}{|cc|cccc} \hline \mathrm{A} & - & \mathrm{T} & \mathrm{T} & \mathrm{E} & \mathrm{V.} \\ \mathrm{G} & \mathrm{S} & \mathrm{T} & - & \mathrm{E} & \mathrm{V} \\ \hline \end{array} \tag{2.2}$$

If now, say, the probability for the substitution A $\leftrightarrow$ S (and thus the similarity between A and S) is higher than for a mutation A $\leftrightarrow$ G, the algorithm will be able to identify alignment (2.1) as the evolutionarily more likely one and thus suggest it as the 'correct' one. Commonly [27], one counts the matching of two identical residues (the best possible match since it does not involve any mutational events) as strictly positive and allows negative similarity scores for the pairing of residues that would correspond to a highly unlikely mutation.

The PAM matrices. The PAM series [5, 6] of similarity matrices, based on observations of mutational frequencies, has found widespread application in the field of sequence alignment. A basic matrix was derived by first aligning a number of families of protein sequences of very small evolutionary distance (at most one mutation per 100 amino acids in average, *i.e.* 1% or less divergence between the two sequences) by eye. Then, the observed amino acid substitutions within the families were counted. Taking logarithms of the normalized frequencies defined the PAM-1 matrix which was extrapolated to higher divergences to allow for the comparison of evolutionarily more distant sequences. Most commonly, the PAM-250 matrix is used, ccorresponding to 250% divergence and to about 20% sequence identity. PAM matrices have been found to perform very well [28] and were subject only to minor refinements [8, 12] in the course of the years.

2.2.2 Gap penalties

Motivation. The algorithm for sequence alignment must allow for the insertion of gaps that represent the insertion or deletion of one or several amino acids, on the genomic level corresponding to an insertion or deletion of a segment in the coding DNA region. The insertion of gaps must be *penalized* to maintain an appropriate balance to the pairing of residues.

Linear gap penalty functions. In the evolution of a sequence, the insertion or deletion of a segment of adjacent amino acids occurs as a single event. That implies that the significant event is the opening of a gap. Therefore, it is biologically reasonable to assign a penalty score $g(l)$ to a gap of length l that is smaller than the sum of l independent gaps of length 1. Commonly, a linear gap penalty function

$$g(l) = g_{\mathrm{open}} + (l-1)g_{\mathrm{ext}} \tag{2.3}$$

is used, where the opening of the gap receives a fixed penalty score g_{open} and the extension to a segment of length l a penalty proportional to the length of the extension, with $g_{\mathrm{ext}} < g_{\mathrm{open}}$. Usually, one chooses both parameter to be positive and subtracts them from similarity scores. Linear gap penalties have been shown to have efficiency advantages over more elaborate penalty functions [9]. Gap penalties can be viewed on the same footing as the similarity scores of Sec. 2.2.1 as a measure for probabilities of evolutionary events, namely the insertion or deletion of amino acids.

Terminal gaps. A special case occurs for gaps at the beginning or the end of aligned sequences (*terminal gaps*). Such gaps may, but do not necessarily need to stem from actual insertions or deletions related to evolutionary activity. For example, a frequent task is to align sequences whose similarity is restricted to a particular motif or domain that we would like the algorithm to identify by an alignment, whereas the remaining regions are more or less unrelated and thus of no interest. Often, we also deal with the alignment of two sequences that are fragments of similar parent sequences, sharing a certain region of overlap which corresponds to a particular section in the alignment of the parent sequences and which we wish the algorithm to identify by an alignment. To facilitate the translational shift of the two sequences as a whole with respect to each other, algorithms for sequence alignment commonly use terminal gaps that are less penalized than gaps within the alignment; often, terminal gaps are not penalized at all (free terminal gaps).

2.2.3 The alignment score

Definition. With the scoring schemes for residue pairs and gapped regions introduced in the previous sections, the task of aligning two sequences such that the alignment most likely mirrors the true evolutionary events has been reduced to a simple optimization procedure. Summing up the individual similarity scores and subtracting (positive) gap penalty scores along a particular alignment of the two sequences $\mathbf{A} = (A_1 A_2 \ldots A_M)$ and $\mathbf{B} = (B_1 B_2 \ldots B_N)$ yields an *alignment score* $\mathcal{A}$. Denoting the similarity score for aligning A_i with B_j by $s(A_i, B_j)$ and the number of gaps of length l in the alignment by $N_{\text{gap}}(l)$, we define the alignment score $\mathcal{A}$ accordingly as

$$\mathcal{A} = \underbrace{\sum_{i=1}^{M} \sum_{j=1}^{N} N_{ij}\, s(A_i, B_j)}_{\text{sum up similarities}} - \underbrace{\sum_{l} N_{\text{gap}}(l)\, g(l)}_{\text{subtract gap penalties}} , \tag{2.4}$$

where $N_{ij} = 1$ if the alignment contains the pair (A_i, B_j), and $N_{ij} = 0$ otherwise; $g(l)$ represents, as usual, the (positive) gap penalty function. Since similarity and gap penalty scores reflect the likelihood of evolutionary events, the optimal alignment (*i.e.* the particular alignment that receives the largest alignment score) will delineate the series of evolutionary events that most likely have taken place.

Global and local alignments. Generally, one distinguishes between a *global* and *local* alignment procedure. In a global alignment, the algorithm determines the alignment that maximizes the alignment score $\mathcal{A}$ over the full length of both sequences. Global alignments are a reasonable approach for sequences that are related over their entire lengths.

However, in the context of terminal gaps (Sec. 2.2.2), we have already mentioned the possibility that two sequences share significant similarity only in limited regions (motifs, domains), whereas the remaining regions are mainly unrelated. Thus, the attempt of the algorithm to align the entire sequences would essentially be a biologically futile task. In such cases, an algorithm can be used that computes the optimal *local* alignment, determined by finding the pair of substrings of the full sequences whose alignment yields the highest alignment score among the set of all substrings and their possible alignments.

2.3 The Alignment Matrix Representation

To aid the visualization of the vast space of possible alignments, we shall introduce an *alignment matrix* that allows for an intuitive and readily representation of sequence alignments and will additionally serve as a basis for the algorithms described later.

Construction of the alignment matrix. We let the two sequence strings $\mathbf{A} = (A_1 A_2 \ldots A_M)$ and $\mathbf{B} = (B_1 B_2 \ldots B_N)$ label the rows and columns, respectively, of a $(M + 1) \times (N + 1)$ array, the alignment matrix, as shown in Fig. 2.1. Every possible alignment of the two sequences can now be represented by a directed path, represented by the arrows in the figure, that leads from

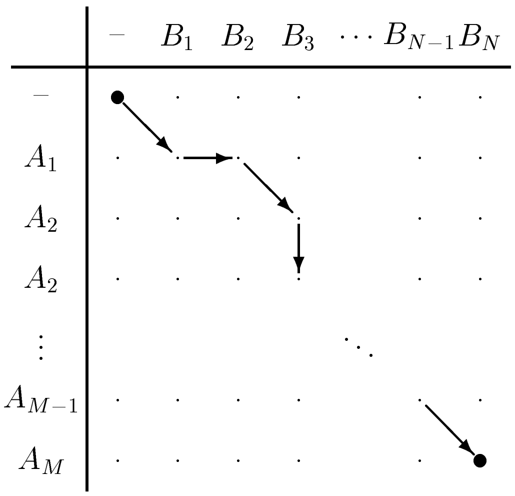

Figure 2.1: The alignment matrix representation for an alignment of two sequences $\mathbf{A} = (A_1 A_2 \ldots A_M)$ and $\mathbf{B} = (B_1 B_2 \ldots B_N)$.

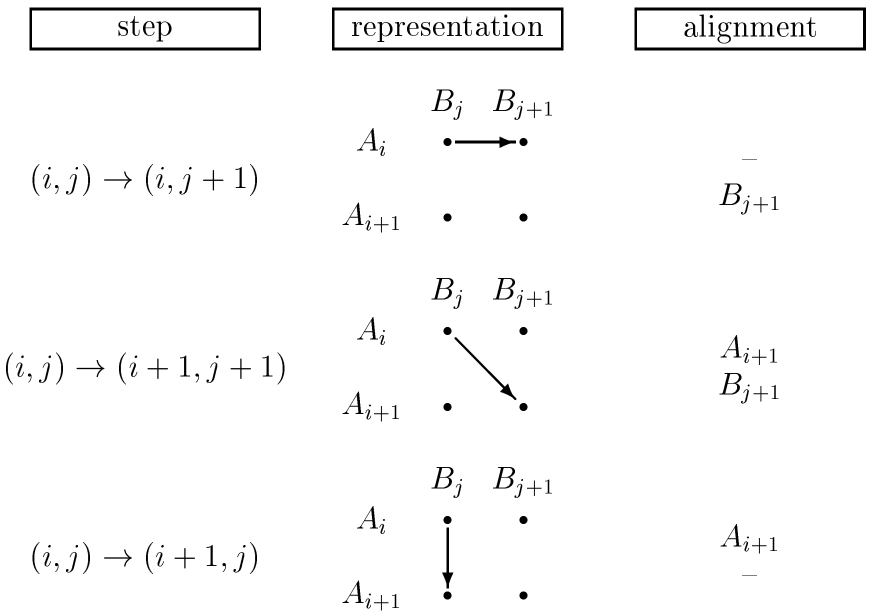

Figure 2.2: The three possible next steps from the element (i, j), their representation in the alignment matrix and the corresponding alignment: $(i, j) \to (i, j + 1)$, corresponding to the insertion of a gap in the sequence string $\mathbf{A}$, that is, the letter B_{j+1} of sequence $\mathbf{B}$ is not aligned to any letter in sequence $\mathbf{A}$; $(i, j) \to (i + 1, j + 1)$, representing the alignment of the letter A_{i+1} of sequence $\mathbf{A}$ with the letter B_{j+1} of sequence $\mathbf{B}$; $(i, j) \to (i + 1, j)$, corresponding to the insertion of a gap in the sequence string $\mathbf{B}$, that is, the letter A_{i+1} of sequence $\mathbf{A}$ is not aligned to any letter in sequence $\mathbf{B}$.

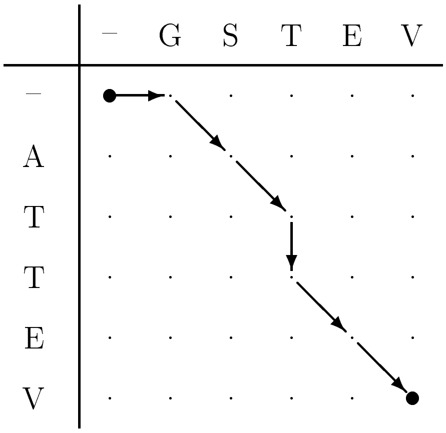

Figure 2.3: The representation of the alignment of $\mathbf{A}$ = ATTEV and $\mathbf{B}$ = GSTEV, cf. (2.1), as a directed path in the alignment matrix.

the upper left corner down to the lower right corner, connecting the dots '·' of the alignment matrix.

We start out from the matrix element $(-, -)$ whose coordinates we define to be $(0, 0)$ such that for $1 \leq i \leq M$, $1 \leq j \leq N$ the matrix element (i, j) corresponds to (A_i, B_j). From every matrix element (i, j) on the path, $0 \leq i < M$, $0 \leq j < N$, the next possible continuation steps are $(i, j+1)$, $(i+1, j+1)$ or $(i+1, j)$, corresponding to the insertion of a gap in the sequence string $\mathbf{A}$, to the alignment of the letter A_{i+1} of sequence $\mathbf{A}$ with the letter B_{j+1} of sequence $\mathbf{B}$, and to the insertion of a gap in the sequence string $\mathbf{B}$, respectively. These three different options are illustrated in Fig. 2.2.

In the special case of $i = M$ or $j = N$, we have reached the bottom or right margin of the matrix, so the only possible remaining continuation steps are $(i, j) \to (i, j+1)$ and $(i, j) \to (i+1, j)$, respectively. We continue step by step until we reach the final element (M, N) in the bottom right corner of the alignment. Every such path describes uniquely a possible alignment, and every alignment can be represented by such a path. An illustration of this technique is shown in Fig. 2.3 for the alignment (2.1) of the two sequences $\mathbf{A}$ = ATTEV and $\mathbf{B}$ = GSTEV considered in Sec. 2.1.

Relation to the alignment score. Since each step along a path in the alignment matrix corresponds to either the alignment of a pair of residues or to the insertion of a gap in one of the sequences, we can associated it with a certain similarity or gap penalty score. Thus, adding up and subtracting the individual similarity scores and gap penalties, respectively, of each individual step along a directed path that starts at (i, j) and ends at (i', j') in the alignment matrix, yields the alignment score $\mathcal{A}$, Eq. (2.4), of the alignment of $(A_i A_{i+1} \dots A_{i'})$ with $(B_j B_{j+1} \dots B_{j'})$. The optimal (global) alignment corresponds then to the *longest path length* (in terms of the resulting alignment score) connecting $(0,0)$ with (M,N).

2.4 Algorithms for Sequence Alignment

Motivation. The number of possible alignments grows exponentially with the length of the sequences which makes the determination of the optimal alignment by an explicit evaluation of all possible alignments an essentially hopeless task. For example, there are $\mathcal{O}(100^{600})$ possible global alignments of two sequences of length 1,000 [35]. An elegant solution to this intractable search problem is given by a class of iterative computer algorithms known as *dynamic programming*. Dynamic programming algorithms store the values computed in the preceding iteration cycle in arrays such that they can easily be accessed in the next cycle without the need for recomputation. Using dynamic programming algorithms for sequence alignment greatly reduces the alignment space that needs to be investigated and requires a computational effort that grows only as $\mathcal{O}(MN)$ when a linear gap penalty is used.

Outline. We shall first briefly point out the connection of sequence alignment to the general shortest path problem in network routing (Sec. 2.4.1). In Sec. 2.4.2, we will then present the *Needleman–Wunsch algorithm*, the most widely used dynamic programming algorithm for global sequence alignment.

2.4.1 Sequence alignment and network routing

Determining the *optimal path* in the alignment matrix, *i.e.* the path that maximizes the total score along the way, can be reduced to the general problem of network routing that has, already in the 1950's, been shown to be exactly solveable by means of a dynamic programming algorithm in polynomial time (Bellman–Ford algorithm, [2, 7]). There, the object is to find the shortest (or longest) path from a source node to any other node in a network of interconnected nodes, with a step between any two connected nodes associated with a certain cost (or gain). We can readily apply this network structure to our problem of sequence alignment by identifying the elements in the alignment matrix with the nodes of a network, where every node (i,j) is directly connected to $(i-1,j)$, $(i-1,j-1)$ and $(i,j-1)$ by a single step (with obvious restrictions at the boundaries, $i=0$ and $j=0$), and to all nodes (i',j') with $0 \le i' \le i$, $0 \le j' \le j$ in multiple steps. Associating the steps with the corresponding similarity and gap penalty scores recovers our goal of finding the optimal alignment as determining the longest path length between the source node $(0, 0)$ and the 'sink' (M, N).

2.4.2 The Needleman–Wunsch algorithm

Overview. The Needleman–Wunsch algorithm [19] is often attributed as the first application of dynamic programming in molecular biology. It can readily be derived from the general Bellman–Ford algorithm, but instead we shall follow the historical development and motivate the algorithm directly from principles of sequence alignment. Slightly different, but essentially equivalent [25] formulations of the Needleman–Wunsch algorithm were described by Sellers [24] and Waterman *et al.* [33]. The implementation of the algorithm commonly follows Gotoh [9].

Derivation. Suppose we wish to find the optimal alignment of the sequence prefix $(A_1 A_2 \dots A_i)$ of the whole sequence $\mathbf{A} = (A_1 A_2 \dots A_M)$ with the sequence prefix $(B_1 B_2 \dots B_j)$ of the whole sequence $\mathbf{B} = (B_1 B_2 \dots B_N)$. This alignment will receive an optimal (*i.e.* maximum) alignment score $\mathcal{A}_{\text{opt}}(i, j)$. The key observation is now that this score can be easily calculated if $\mathcal{A}_{\text{opt}}$ is already known for smaller alignment problems, that is, for alignments of $(A_1 A_2 \dots A_{i'})$ with $(B_1 B_2 \dots B_{j'})$ where $i' < i$ and/or $j' < j$. To this extent, we note that optimal alignment of $(A_1 A_2 \dots A_i)$ with $(B_1 B_2 \dots B_j)$ can end in only one of the following three ways:

1. A_i is aligned to B_j. Referring to Fig. 2.2, this corresponds to a diagonal step in the alignment matrix from node $(i-1, j-1)$ to (i, j). Then the alignment score $\mathcal{A}(i, j)$ for aligning the prefices $(A_1 A_2 \dots A_i)$ and $(B_1 B_2 \dots B_j)$ will be the similarity score $s(A_i, B_j)$ for aligning A_i to B_j, plus the optimal alignment score $\mathcal{A}_{\text{opt}}(i-1, j-1)$ for the remaining optimal alignment not including A_i and B_j.

2. A_i is aligned to a gap. This is represented by a vertical step from node $(i-1, j)$ to (i, j). Assuming an arbitrary (positive) gap penalty function $g(l)$ where l is the length of the gap, the algorithm needs to know where that gap started, so it can count how many consecutive residues at the end of $\mathbf{A}$ are aligned to gaps in order to use the correct penalty $g(l)$. This just means finding the best scoring alternative for aligning B_j to some residue A_a, where $a < i$. The length of the gap is thus $i - a$. Then $\mathcal{A}(i, j)$ will be $\mathcal{A}_{\text{opt}}(\hat{a}, j) - g(i - \hat{a})$ for the best choice of $\hat{a}$, which is found by maximizing $\mathcal{A}_{\text{opt}}(a, j) - g(i - a)$ over all $0 \le a < i$.

3. B_j is aligned to a gap. This corresponds to a horizontal step from node $(i, j-1)$ to (i, j) and is analoguous to case 2, except now the last residues of $\mathbf{B}$ instead of $\mathbf{A}$ are aligned to gaps. Accordingly $\mathcal{A}(i, j) = \mathcal{A}_{\text{opt}}(i, \hat{b}) -$

$g(j-\hat{b})$, where $\hat{b}$ is now chosen such that $\mathcal{A}_{\text{opt}}(i,b)-g(j-b)$ is maximized over all $0 \le b < j$.

Recursion relation. The optimal alignment score $\mathcal{A}_{\text{opt}}(i,j)$ is then the maximum of these three alternatives $\mathcal{A}(i,j)$. Hence, $\mathcal{A}_{\text{opt}}(i,j)$ is defined recursively in terms of solutions of the smaller subproblems $\mathcal{A}_{\text{opt}}(i-1,j-1)$, $\mathcal{A}_{\text{opt}}(a,j)$ and $\mathcal{A}_{\text{opt}}(i,b)$ according to

$$\mathcal{A}_{\text{opt}}(i,j) = \max \begin{cases} \max_{0\le b<j}\{\mathcal{A}_{\text{opt}}(i,b) - g(j-b)\}, \\ \mathcal{A}_{\text{opt}}(i-1,j-1) + s(A_i,B_j), \\ \max_{0\le a<i}\{\mathcal{A}_{\text{opt}}(a,j) - g(i-a)\}. \end{cases} \tag{2.5}$$

Initialization. Since the nodes $(0,j)$, $j=1,\ldots,N$, on the top and the nodes $(i,0)$, $i=1,\ldots,M$, on the left margin of the alignment matrix can only be reached by a strictly horizontal and vertical path, respectively, corresponding to the insertion of initial gaps in sequence **A** and **B**, respectively, their optimal alignment score $\mathcal{A}_{\text{opt}}$ is a priori known from the gap penalty function $g_{\text{term}}(l)$ used for terminal gaps. We can therefore initialize

$$\begin{aligned} \mathcal{A}_{\text{opt}}(0,0) &= 0, \\ \mathcal{A}_{\text{opt}}(i,0) &= -g_{\text{term}}(i), \qquad i=1,\ldots,M, \\ \mathcal{A}_{\text{opt}}(0,j) &= -g_{\text{term}}(j), \qquad j=1,\ldots,N. \end{aligned} \tag{2.6}$$

In the special case of free terminal gaps, $g_{\text{term}} \equiv 0$, the initialization reduces to $\mathcal{A}_{\text{opt}}(i,0) = \mathcal{A}_{\text{opt}}(0,j)$ for all $i=0,\ldots,M$ and $j=0,\ldots,N$.

Recursive calculation. Recursive application of Eq. (2.5) together with the initial values Eqs. (2.6) allows the calculation of $\mathcal{A}_{\text{opt}}(i,j)$ for all nodes (i,j). The alignment matrix formulation provides an intuitive and elegant way of performing this calculation. With the initial values Eqs. (2.6), we are in position to immediately calculate $\mathcal{A}_{\text{opt}}(1,1)$, $\mathcal{A}_{\text{opt}}(1,2)$, $\ldots$, $\mathcal{A}_{\text{opt}}(1,N)$, in that order, since at each node $(1,j)$, $\mathcal{A}_{\text{opt}}(0,j-1)$, $\mathcal{A}_{\text{opt}}(0,j)$ and $\mathcal{A}_{\text{opt}}(1,j-1)$ are already known, and Eq. (2.5) states that these three values are sufficient to compute $\mathcal{A}_{\text{opt}}(1,j)$.

In this manner, we can proceed row by row in the alignment matrix and calculate for each node (i,j) the optimal alignment score $\mathcal{A}_{\text{opt}}(i,j)$ that corresponds to the optimal alignment of $(A_1A_2\ldots A_i)$ with $(B_1B_2\ldots B_j)$. We store each calculated score value as an entry in a $(M+1)\times(N+1)$ array and also record from which node the calculation was propagated, *i.e.* which of the three options in Eq. (2.5) has yielded the optimal score. When the final node (M,N) is reached, the node (M,N) will then be associated with the alignment score of the optimal alignment of the full sequences **A** and **B**.

Retracing the optimal alignment. The optimal alignment itself can be obtained by a backtrace through the alignment matrix, starting from the node (M, N) and following the recorded directions of propagation to retrace the optimal path. The optimal path can also be retrieved by explicit recomputations starting from node (M, N) which makes the necessity to store informations about the direction of propagation in principle obsolete. However, for practical realizations of the algorithm, saving directions is usually the preferred method since it requires no additional computations which reduces both the needed execution time and overall complexity of the algorithm.

Multiple optimal alignments. It might occur that more than one of the three options in the recursion relation Eq. (2.5) represent an identical score. This means that the optimal alignment of $(A_1 A_2 \dots A_i)$ with $(B_1 B_2 \dots B_j)$ can end in more than one possible way. Globally, it implies the presence of multiple possible optimal alignments of the sequences $\mathbf{A}$ and $\mathbf{B}$. We could in principle account for such degeneracies in the trace-back procedure by placing yet unexplored directions associated with ambiguous nodes into a last in–first out stack to which the algorithm returns after each output of a possible optimal alignment, until the entire set of optimal alignments has been determined [35].

Simplification for linear gap penalties. For an arbitrary gap penalty function $g(l)$, the Needleman–Wunsch algorithm requires a computational effort of $\mathcal{O}(M^2N + MN^2)$, or $\mathcal{O}(N^3)$ if $M = N$ [35]. With a linear gap penalty, this reduces to $\mathcal{O}(MN)$ [9], for the algorithm only needs to know whether the insertion of a gap corresponds to an opening or an extension of the gap to assign the correct penalty, instead of checking the whole row or column. For instance, to evaluate

$$\max_{0 \leq a < i} \{\mathcal{A}_{\text{opt}}(a, j) - g(i - a)\},$$

in the recursion relation Eq. (2.5), it only has to be checked whether the calculation of $\mathcal{A}_{\text{opt}}(i-1, j)$ was propagated from node $(i-2, j)$, which implies the extension of a gap, or not, which requires the opening of a gap. Analogously, for the calculation of $\max_{0 \leq b < j}\{\mathcal{A}_{\text{opt}}(i, b) - g(j - b)\}$, only the knowledge of the direction of propagation for node $(i, j-2)$ is required.

Modification for local alignments. A small modification of the original Needleman–Wunsch algorithm that allows the determination of the optimal *local* alignment has been introduced by Smith and Waterman [26] (*Smith–Waterman algorithm*). The key difference is based on the introduction of a zero as a fourth option in the recursion relation Eq. (2.5) which has the effect of terminating any path in the alignment matrix whose score drops below zero. After optimal alignment scores have been calculated for all nodes in the usual recursive way, the optimal local alignment is found by locating the node with the largest alignment score and performing a trace-back starting from this node, until a node is encountered whose alignment score is equal to zero.

Chapter 3

Reliability of Sequence Alignments

In this chapter, we will raise the question of the significance and reliability of alignments produced by a mathematical optimization algorithm, such as the Needleman–Wunsch algorithm presented in the preceding chapter. The introduction of the problem in Sec. 3.1 will be followed by a review of current approaches to the quantification of local reliability in Sec. 3.2. In this context, we shall also discuss the problems intrinsically related to these current methods in order to motivate the novel approach that has been pursued in this thesis work.

3.1 The Nature of the Problem

The goal of sequence alignment is to pinpoint regions of high similarity between the two sequences. From such *conserved regions*, evolutionary and structural relationships between the two proteins can be inferred. An algorithm for sequence alignment will in general pursue this identification by maximizing a certain score that quantifies the similarity between the sequences in the particular alignment. Hence, the algorithm for sequence alignment is essentially just a mathematical optimization procedure based on a predefined scoring scheme, and it is therefore legitimately to ask to what extent the produced alignment will also reflect optimality in the sense of *biological* truth. This poses the two-fold problem of 1) judging the statistical *significance* of an alignment as a whole, and 2) quantifying the local *reliability* of its parts.

3.1.1 The significance of an alignment

The problem. It is clear that an algorithm for sequence alignment will align *any* two sequences. In the extreme case, the two sequences may be entirely random and therefore completely unrelated. Accordingly, we would be misled by inferring any biological relations between these two sequences from their alignment. Hence, a criterion is needed that indicates whether the alignment contains biological significance, and in that way allows one to judge whether the two sequences share a sufficient degree of homology.

Quantifying significance. Since, besides the sequences themselves and the scoring scheme, no additional external input of detailed empirical data is available, biological significance is commonly deduced from a statistical one [31, 32]. In essence, one compares the score of an optimal alignment of the two original sequences with the average score obtained from optimal alignments of sequences generated from random permutations of the letters in the original pair of sequences. If the score of the original alignment exceeds the average score by a significant amount, safely outside of the range set by the standard deviation of the mean score, one can conclude that the optimal alignment exhibits significance which implies a relationship between the two proteins.

Restriction to established similarity. In the following, we shall not concern us furtherly with the problem of determining the statistical significance of an alignment. Instead, we will assume that the sequences under consideration are in fact related, and that therefore their alignment is a biologically reasonable and meaningful task.

3.1.2 The local reliability of an alignment

The significance of an alignment just provides a global measure to judge whether two sequences are overally related to each other or not. It does not discern parts that are correctly aligned (with respect to the standard of truth) from others that are incorrectly aligned.

Sources of deviations from the correct alignment. As a matter of fact, complex evolutionary, structural and functional relationships between proteins can not always be retraced by a simple, however in most cases surprisingly effective algorithmical optimization of some score. There might exist alternative, *suboptimal* alignments that score slightly lower than the optimal alignment and are therefore disregarded by the algorithm, but actually mimick the correct alignment better. For example, higher-dimensional structural affinities might be present that represent the actual crucial relationships between two proteins, but are not contained in the one-dimensional sequence information and can therefore not be delineated by an alignment that is solely based on a maximization of sequence similarity.

Furthermore, as the algorithm is based on a scoring scheme of similarities and gap penalties, changing these parameters results often, in particular in the case of sequences of low similarity, in drastic alterations of the resulting alignment [1, 30]. A lack of robustness with respect to such changes in the parameters then raises the problem of selecting the proper optimal alignment, since there is no general biological argument for a specific choice of parameters; in turn, this implies little security in whatever alignment is chosen.

Importance of quantifying reliability. From this discussion, it has become apparent that even for sequences whose overall homology, *i.e.* statistical and presumably therefore biological significance has been established, the optimal alignment resulting from the algorithmical procedure will generally not be a mirror of biological truth in all its parts. It will therefore allow a biologically valid evaluation only when the local reliability of its parts is quantified. Thereby, regions of low reliability, where the alignment produced and hence the relationship implied is doubtful, can be discerned from parts for which we may safely assume that the relationship delineated by the alignment pinpoints in fact biologically conserved regions whose analysis can yield the desired implications about evolutionary, structural and functional homologies. Information on local reliability will be highly valuable for anyone using programs for sequence alignment to ensure that only biologically reliable and valid conclusions are drawn from the produced alignment. Therefore, it is hard to underestimate the importance of such a reliability measure for a proper interpretation of any algorithmically obtained sequence alignment.

Definition of local reliability. There exist numerous possibilities for a definition and quantification of local reliability that are suitable for an algorithmical realization. The basic idea of most approaches relies on an implicit or explicit study of the presence of locally different suboptimal alignments competing with the optimal solution, which in turn is assumed to reflect the local robustness of the alignment. Local reliability of an aligned region relates therefore to the degree of the presence of alternatives that would, with respect to the objective optimization function and the used parameters, also serve as favorable and thus biologically meaningful solutions to the alignment problem.

3.2 Current Methods

We shall now present concrete realizations for the assessment of the local reliability of sequence alignments that have been suggested in various earlier works. We shall not only describe these methods and thereby illustrate the rather abstract procedure outlined in the above definition of local reliability, but also pinpoint their limitations and drawbacks in order to motivate the novel approach that has been developed in the course of this thesis work. Results of a representative method [16] will be presented and discussed in more detail to allow later comparison with our technique.

3.2.1 Delineating reliably aligned regions

Commonly, the first attempt to formulate a measure for the local reliability of pairwise sequence alignments is attributed to Vingron and Argos [29]; their approach has also served as the foundation of most later methods. The idea is based on the study of a set of so called ϵ-suboptimal alignments, that is, alignments whose alignment score is within a range ϵ below the optimal score. Regions that are shared by *all* ϵ-suboptimal alignments, called *stable regions*, are then regarded as reliably aligned.

Algorithm. The explicit computation of all suboptimal alignments is in principle a possible [3, 34], for practical applications however not feasible approach due to the enormous number of resulting, often widely similar alignments. Instead, Vingron and Argos introduce an alternative method based on a modification of the original Needleman–Wunsch algorithm. Besides the normal alignment score $\mathcal{A}_{\text{opt}}$ of the optimal alignment, they compute in time $\mathcal{O}(MN)$ for each possible residue pairing (A_i, B_j) the optimal score $\mathcal{A}_{\text{opt}}^{ij}$ of an alignment that is enforced to contain this particular pair. This alignment belongs to the set of ϵ-suboptimal alignments if and only if

$$\mathcal{A}_{\text{opt}} - \mathcal{A}_{\text{opt}}^{ij} \leq \epsilon, \tag{3.1}$$

which in turn defines the collection $P(\epsilon)$ of pairs (A_i, B_j) that can be part of a ϵ–suboptimal alignment ("ϵ–suboptimal residue pairs"). A match of the residue pair A_i with B_j in the optimal alignment is then considered to be reliably aligned if the pair (A_i, B_j) is contained in $P(\epsilon)$ but no other pair (A_i, B_k) or (A_l, B_j) with $k \neq j$ and $l \neq i$, since in this case (A_i, B_j) must be part of all ϵ–suboptimal alignments.

Suboptimal dot plot. By plotting a dot for each pair (A_i, B_j) contained in $P(\epsilon)$ into the i–th row and j–th column of the alignment matrix, the possible ϵ–suboptimal alignment paths can be visualized, and stable regions can be directly identified by inspection. Additional weighting of the paths can be attributed by using different colors or shades for the dots (i, j) depending on their corresponding alignment score $\mathcal{A}_{\text{opt}}^{ij}$. An example for such a *suboptimal dot plot* is shown in Fig. 3.1. A single alignment path, such as in the upper left corner of the matrix, corresponds to a stable region and is therefore predicted to be reliably aligned. In contrary, the 'fuzzier' regions in the middle indicate a multiplicity of possible alternative matches and thus lower thrustworthiness into this part of the optimal alignment.

Interpretation. Clearly, the larger the tolerance parameter ϵ is chosen, the more pairs (A_i, B_j) will be part of $P(\epsilon)$ and thus the more unlikely it is that a given match (A_i, B_j) in the optimal alignment is shared by all ϵ–suboptimal

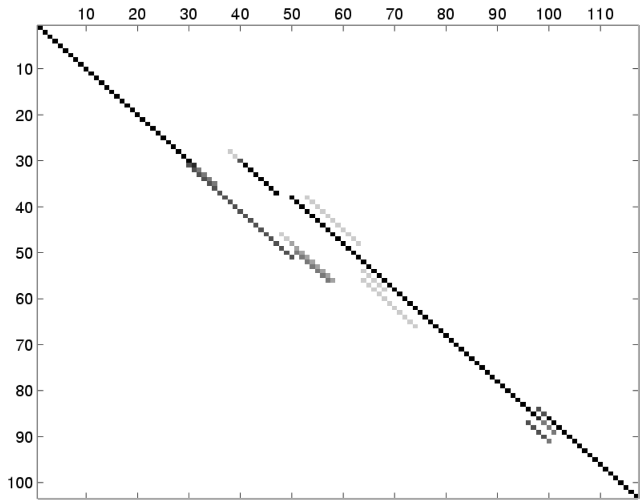

Figure 3.1: A ϵ–suboptimal dot plot for a pair of immunoglobulins. More lightly shaded dots indicate ϵ–suboptimal residues that correspond to lower alignment scores $\mathcal{A}_{\text{opt}}^{ij}$ of the ϵ–suboptimal alignment. The plot was adapted from [32].

alignments; in turn, the number of ϵ–suboptimal residues in the optimal alignment is expected to decrease. Given the prediction that the ϵ–suboptimal residues in fact delineate reliably residues, this implies that residues that are still part of stable regions at larger values of ϵ are presumably more reliably aligned than those that lost their affilation as ϵ was increased.

Evaluation and results. In order to confirm this expectation and thereby validate their reliability measure, the authors studied several alignments for which the standard of truth was known from structural superpositions. In the investigated cases, the authors demonstrated that regions which remained stable at larger values of ϵ exhibited also stronger agreement with the reference alignment. Increasing ϵ therefore also increases the probability that residues predicted to be reliably aligned according to their affilation to the set of ϵ–suboptimal residues are in fact correctly matched. On the other hand, the larger ϵ is chosen, the fewer regions are identified as reliable; excluding the regions not marked as reliably aligned from the optimal alignment will therefore decrease the percentage of the reference alignment to be found by the optimal alignment.

Discussion. The method introduced by Vingron and Argos is the first and already remarkably successful attempt to assess the local reliability of a sequence alignment. It establishes and confirms the idea that local reliability is related to the behavior of suboptimal alignments. The problem of an explicit evaluation of an often astronomically large number of suboptimal alignments is bypassed by a $\mathcal{O}(MN)$ calculation of the limited set $P(\epsilon)$ of ϵ–suboptimal residue pairs without evaluating the alignments themselves.

Disadvantageously, the algorithm can only predict whether a certain residue pair is reliably aligned or not, but it does not distinguish furtherly between higher and lower reliability. Hence, for a trustworthy further analysis, one will usually be confined to entirely disregard all regions in the optimal alignment that are predicted to be unreliably aligned. As a consequence, in many cases only a few parts of the correct alignment will be obtained in that way. Since the algorithm does not produce any suboptimal alignments explicitely that could be used to elucidate the remaining regions, the algorithm is not too suitable for a practical application. Furthermore, the method is obviously not capable of assessing reliability to gapped regions.

Modifications and enhancements. The method described in the preceding section was extended to local Smith–Waterman alignments by Zuker [36], who studied a wider range of example alignments that essentially confirmed the basic observations made by Vingron and Argos. Saqi and Sternberg [23] explicitely calculated a limited set of suboptimal alignments that differ nontrivially from each other and can be used to identify larger parts of the correct alignment.

Naor and Brutlag [18] performed a detailed analysis of global ϵ–suboptimal alignment paths composed of the ϵ–suboptimal residues obtained from the algorithm introduced by Vingron and Argos. By means of an algorithmical procedure that is able to determine the number of suboptimal alignments for each value of ϵ, they suggested an alternative method for the quantification of the statistical significance of an alignment by showing that the number of suboptimal alignments for real biological sequences sharing similarity is significantly lower than for randomly generated and therefore completely unrelated sequences. This may open a new perspective on a unified treatment of statistical significance and local reliability.

3.2.2 Quantifying reliability of individual pairs

The methods presented so far aim mainly at a identification of reliably aligned *regions* in the optimal alignment. The attempt to quantify the reliability of each individual aligned pair has been approached first in a work by Chao *et al.* [4]. A modified and thoroughly verified application of their concepts particularily to protein sequence alignment has been described by Mevissen and Vingron [16].

The idea. The approach introduced by Chao *et al.* is closely related to the idea of Vingron and Argos described above, but establishes a different point of view. While the latter method deduces reliability from an overlap of local regions among the set of suboptimal alignments at a particular *fixed* value of ϵ, the idea of Chao *et al.* is to monitor when a particular pair in the alignment loses it stability (*i.e.* its affiliation to all ϵ–suboptimal alignments) as a function of increasing ϵ. The corresponding threshold value ϵ_0 is used to define an *algorithmic robustness index* $r(i,j)$ of the pair (A_i, B_j).

Algorithmic realization. The threshold ϵ_0 and therefore the algorithmic robustness index can be determined by computing the alignment score $\mathcal{A}_{\text{opt}}^{\overline{ij}}$ of the optimal alignment that *avoids* a match of A_i to B_j. This is identical to the (best, if more than one) suboptimal alignment that arises when the threshold $\epsilon = \epsilon_0$ is just reached such that the pair (A_i, B_j) loses its stability. Then, at this point, the inequality in (3.1) becomes an equality, and hence

$$r(i,j) \equiv \epsilon_0 = \mathcal{A}_{\text{opt}} - \mathcal{A}_{\text{opt}}^{\overline{ij}}. \tag{3.2}$$

Chao *et al.* describe a sophisticated algorithm that allows the calculation of $\mathcal{A}_{\text{opt}}^{\overline{ij}}$ and hence $r(i,j)$ for every residue pair in the alignment in time $\mathcal{O}(MN)$. This algorithm is also able to assign a robustness index to gapped regions. Mevissen and Vingron [16] adopt the idea, but simplify the algorithm at the price of not being able to assign robustness to gapped regions; the computational time remains $\mathcal{O}(MN)$. The minimum and maximum value of the robustness index is used to normalize $r(i,j)$ to the interval $[0,1]$.

Verification. The desired connection between the robustness of a particular pair and its alignment reliability can be established by verifying that residue pairs that receive a high robustness index are in fact more frequently correctly aligned with respect to the standard of truth than pairs with a low robustness index.

This has been extensively done in the work by Mevissen and Vingron. Here, a large number of sequence pairs whose structurally correct alignment was known from the `3D_ali` database [20] were aligned using the global Needleman–Wunsch algorithm with a PAM-250 similarity matrix and a linear gap penalty with free terminal gaps. For each residue pair in the optimal alignment, the normalized robustness $r(i,j)$ was computed; the interval $[0,1]$ of possible values was subdivided into 10 bins such that the robustness index could be reported as integer numbers between 1 and 10. The optimal alignment was then compared with the `3D_ali` reference alignment to obtain the fractions of correctly aligned residues for each of the 10 possible assigned robustness indices.

It was found that with increasing robustness index the average frequency of a correct alignment increased as well (Fig. 3.2). For more similar and therefore more easily alignable sequences, residue pairs with robustness indices larger than 4–5 were in virtually all cases correctly aligned. Having thereby verified the interpretation of the robustness index as a reliability measure, the authors rename $r(i,j)$ as *reliability index* of the pair (A_i, B_j).

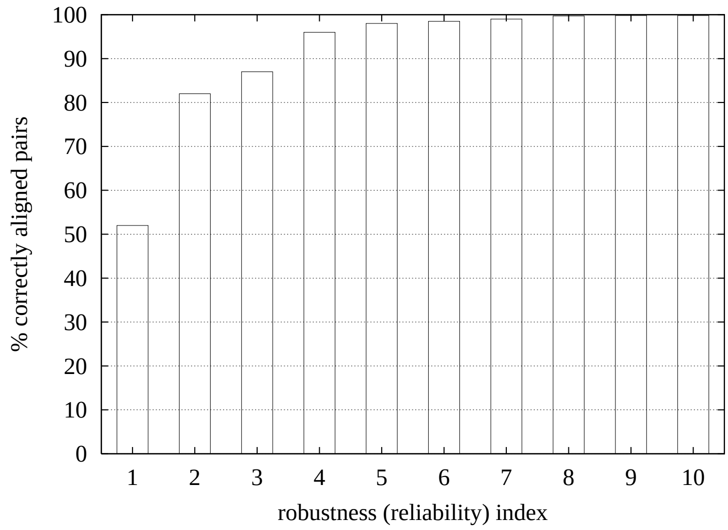

Figure 3.2: Validation of the robustness index as a measure of reliability in the work by Mevissen and Vingron [16]. The plot shows the percentage of correctly aligned pairs for each robustness index between 1 and 10. The analyzed reference sequences shared high similarity (40–50% sequence identity in the optimal alignment). As the robustness index increases, the fraction of correctly aligned pairs increases accordingly, thus establishing the desired connection between robustness and reliability. Data derived from a plot given in [16].

Deducing an absolute reliability measure. As one can easily see from Fig. 3.2, the particular value of the reliability index $r(i, j)$ does not directly correspond to a distinct and absolute measure of reliability. For example, the plot shows that essentially all residue pairs with reliability indices between 5 and 10 are correctly aligned with probabilities close to 100%. Therefore, a pair that receives a reliability index of 5 is in average roughly as reliably aligned as a pair with an index of 10, making the numerical differences between these indices essentially meaningless.

This requires the reliability index calculated for a new alignment to be calibrated in the following way. Firstly, the optimal alignment and the corresponding reliability indices $r(i, j)$ are computed with the given scoring scheme. Then, the optimal alignments of a set of sequence pairs, extracted from the `3D_ali` database and having similar homology as the alignment under consideration, are calculated, and the average frequencies of correct alignment are deduced from a comparison with the corresponding structurally correct alignments. These frequencies are finally used to translate the reliability indices $r(i, j)$ into absolute probabilities for the pairs (A_i, B_j) to be correctly aligned. For example, from Fig. 3.2 we see that a $r = 2$ residue pair in the 40–50% homology class would receive an $\approx 80\%$ probability to be correctly aligned.

Discussion of the method by Mevissen and Vingron. The procedure described by Mevissen and Vingron yields a meaningful and thoroughly verified measure for the local reliability of each individual pair in an optimal alignment. However, the method suffers from certain disadvantages. Most severely, the algorithmically obtained reliability index requires a calibration with an external data source, namely with the reference alignments deduced from the `3D_ali` database. This calibration has to be repeated for each alignment that is computed for a set of parameters or a similarity class for which no analysis of reference alignments has been performed previously. Therefore, this method is not capable of assigning a distinct and absolute reliability measure solely obtained from an algorithm. Additionally, the reliability of gapped regions can not be assessed.

Moreover, the method results in significant, not particularily elegant algorithmical and computational complications. For example, a set of submatrices has to be computed and evaluated; reference sequence pairs have to be extracted from the `3D_ali` database and their optimal alignment must be computed; the resulting alignments have to be compared with the reference to calculate the frequencies of correct alignment, and the like. Remedying these disadvantages has to a large extent motivated the technique developed in the course of this work.

Other approaches. Alternative methods for quantifying local reliability of individual pairs have been proposed in the context of a probabilistic interpretation of the alignment score [14, 17]. Miyazawa [17] attributes a statistical weight proportional to $\exp(\mathcal{A}/T)$ to an alignment receiving an alignment score $\mathcal{A}$. T denotes a parameter that depends on the particular scoring scheme chosen. The mathematical form of the weight ensures that the probability of an alignment constructed from disjoint segments can be expressed as the product of the individual propabilities of the segments, as the alignment score is additive.

Recognizing the similarity of this weight expression to the Boltzmann distribution when interpreting $\mathcal{A}$ as a negative *energy* and T as a fictitious *temperature*, Miyazawa introduces a *partition function*

$$Z = \sum_{\ell} \exp\left(\frac{\mathcal{A}(\ell)}{T}\right), \tag{3.3}$$

where the sum runs over all possible alignments; an algorithm is presented that allows the computation of Z in time $\mathcal{O}(MN)$. The *probability* $\mathcal{P}(\ell)$ for a particular alignment ℓ is then given by

$$\mathcal{P}(\ell) = \frac{1}{Z} \exp\left(\frac{\mathcal{A}(\ell)}{T}\right), \tag{3.4}$$

from which an expression for the individual probabilities $p(A_i, B_j)$ of a certain residue–residue pairing (A_i, B_j) is readily derived. In the zero-temperature limit, Eq. (3.4) recovers the maximum-similarity alignment.

Using several proteins whose structurally correct alignment is known, the author demonstrates that pairs with a high probability value $p(A_i, B_j)$ are more likely to be correctly aligned than pairs with low probability. This observation suggests that the quantity $p(A_i, B_j)$ can serve as a measure of reliability. The probabilities can be represented by a dot plot similar to Fig. 3.1, where now the amount of shading of the dots corresponds to the probability value. The traces outlined in such a plot can be used to delineate suboptimal alignments. Hence, this probability method is, in essence, an alternative approach to retrieve information about local suboptimality.

Chapter 4

Methods

This chapter will present the novel method for assigning local reliability to sequence alignments that has been developed and implemented in the course of this thesis work.

Technique. In Chapter 3, it has been demonstrated that the reliability of a sequence alignment can be defined by quantifying the conservation of aligned regions, down to individual pairs of residues, among a set of suboptimal alignments. We shall now introduce a different approach towards the implicit evaluation of suboptimality that is based on a recast of the Needleman–Wunsch algorithm in a *mean field approximation*. Thereby, the path of the optimal alignment will loose its rigidity and will become 'fuzzy'; monitoring the evolution of the local fuzziness along a process of *simulated annealing* will serve as the basis for our reliability measure. The combination of a mean field approximation with simulated annealing shall be referred to as *mean field annealing* in the following.

Outline. We will begin with an introduction to the technique of mean field annealing as developed in the context of optimization problems (Sec. 4.1). In Sec. 4.2, we shall reformulate the Needleman–Wunsch algorithm in terms of a spin system with an energy function to allow for a straightforward application of the mean field annealing technique. The deduction of our reliability measure from the mean field dynamics of the system will be presented in Sec. 4.3. The algorithm has been implemented into a computer program to allow for an extensive validation of our reliability measure with reference data extracted from the `3d_ali` data base; the selection and classification of this data shall be described in Sec. 4.4.

Restriction to global alignments. In the following treatment, we will restrict ourselves, for the sake of simplicity and clarity, to a global algorithm of the Needleman–Wunsch type. Nevertheless, this restriction implies no loss of generality since our technique can equally be applied to local alignment algorithms of the Smith–Waterman type.

4.1 Introduction to Mean Field Annealing

Motivation. The technique of mean field annealing has originally been developed in the context of combinatorial optimization problems. This is a class of problems that are particularily hard to solve, as the number of possible solutions often grows exponentially or worse with the size of the system under consideration, which makes the determination of the exact solution in many cases an essentially hopeless task. Therefore, heuristic methods are frequently used to obtain reasonably good solutions. Mean field annealing is a member of this class of heuristic approaches that has successfully been applied to various classical combinatorial optimization problems [10, 11, 21].

Application to sequence alignment. Sequence alignment does obviously not belong to the class of hard combinatorial optimization problems since we have already seen that a dynamic programming algorithm of the Needleman–Wunsch type allows the determination of the optimal alignment which maximizes the alignment score in polynomial time $\mathcal{O}(MN)$. We shall see, however, that a recast of the sequence alignment algorithm in the language of mean field annealing can be used to deduce a measure of local reliability.

4.1.1 Optimization problems and spin systems

Problem mapping. In physics, a frequent object is to find the particular configuration $\mathcal{S} = (s_1, s_2, \ldots, s_N)$ of a system of N interacting spins s_i (actually, magnetic moments) that minimizes a specific energy function $E(\mathcal{S})$ based on spin interaction and constraints terms; for instance, the well-known *Ising model* is a simplified representative of such a system. It turns out that also a large class of optimiziation problems can be mapped onto such a spin system with a problem-specific energy function such that the optimiziation task is expressed in finding the spin configuration of the system that minimizes this energy.

Ising vs. Potts spin variables. For certain problems, it is sufficient to use one–dimensional binary *Ising* spin variables $s_i \in \{0, 1\}$ only to encode the configuration of the system. In many other cases, it might be advantageous to have multi-dimensional *Potts* spin variables $\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{iK})$ with binary components $s_{ik} \in \{0, 1\}$ subject to the constraint

$$\sum_j s_{ij} = 1. \tag{4.1}$$

Customarily, one chooses K equal to the number of desired possible spin states; the j-th possible state is then defined by $s_{ij} = 1$ and $s_{ik} = 0$ for all $k \neq j$. Hence, the j-th possible state can be represented by the K dimensional Euclidean unit vector $\mathbf{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the '1' as the j-th component. These vectors point to the corners of a regular K simplex, are both normalized and mutually orthogonal and fulfill automatically Eq. (4.1). Using Potts spins when encoding an optimization problem often leads to a reduction in the number of terms needed in the energy function which simplifies the energy landscape and hence the determination of the global energetic minimum.

4.1.2 Simulated annealing

Motivation. For combinatorial optimization problems, the energy landscape in the space of possible spin states $\mathcal{S}$, defined by the energy function $E(\mathcal{S})$, is usually of such complex structure that simple optimization algorithms such as the gradient descent method will almost certainly get trapped in local minima close to the starting point that correspond to non-optimal solutions, without reaching the global minimum.

Method. A stochastic method that allows for "uphill moves" in the energy landscape is therefore a more suitable strategy. Using stochastic neighborhood search methods such as the Metropolis algorithm [15], a series of configurations $\mathcal{S}$ is generated that emulate a Boltzmann distribution

$$P(\mathcal{S}) = \frac{1}{Z} e^{-E(\mathcal{S})/T}, \tag{4.2}$$

where T is a fictitious (positive) *temperature* of the system that represents its noise level, and Z is the *partition function*

$$Z = \sum_{\mathcal{S}} e^{-E(\mathcal{S})/T}. \tag{4.3}$$

In the zero–temperature limit $T \to 0$, the Boltzmann distribution will become concentrated to the particular configuration $\mathcal{S}'$ that minimizes the energy, that is, $P(\mathcal{S}') \to 1$ and $P(\mathcal{S}) \to 0$ for $\mathcal{S} \neq \mathcal{S}'$. The system is now subject to a process of *simulated annealing* [13]. Initially, T is chosen sufficiently high such that the mobility within the energy landscape is high, and large regions of the state space can therefore be explored as new configurations are generated. Then, T is successively lowered which reduces the mobility more and more (annealing), until in the limit $T = 0$ the system will end up in a energetic minimum that can be hoped to be the global minimum. The configurations generated in this annealing process are less likely to get trapped in a local minima than if $T = 0$ had been chosen from the start.

4.1.3 The mean field equations

Motivation. The disadvantage of using simulated annealing with a stochastic method such as the Metropolis algorithm is the required computational time. The idea of mean field annealing is therefore to introduce a mean field approximation that replaces stochastic with deterministic equations and thereby allows for a much faster computation. We shall carry out the derivation for the case of Potts spin variables, since this is the type of spins that will be used for our application to sequence alignment.

Derivation. To this extent, we note that if we knew the *thermal averages* $\langle \mathbf{s}_i \rangle_T$ of the Potts spins $\mathbf{s}_i$ at temperature T, given by (cf. Eq. (4.2))

$$\langle \mathbf{s}_i \rangle_T = \sum_{\mathcal{S}} \mathbf{s}_i P(\mathcal{S}) = \frac{1}{Z} \sum_{\mathcal{S}} \mathbf{s}_i e^{-E(\mathcal{S})/T}, \tag{4.4}$$

we could just let $T \to 0$ to obtain the ground state configuration $\langle \mathbf{s}_i \rangle_{T=0}$ which represents the solution to the combinatorial problem. To obtain an expression that allows a time-efficient approximative calculation of $\langle \mathbf{s}_i \rangle_T$, we first rewrite Eq. (4.4) as

$$\langle s_{ij} \rangle_T = \left\langle \frac{e^{-E|_{\mathbf{s}_i = \mathbf{e}_j}/T}}{\sum_k e^{-E|_{\mathbf{s}_i = \mathbf{e}_k}/T}} \right\rangle_T \tag{4.5}$$

for the j-th component of $\langle \mathbf{s}_i \rangle_T$. $E|_{\mathbf{s}_i = \mathbf{e}_j}$ is the energy of the system in some configuration $\mathcal{S}$ where the Potts spin $\mathbf{s}_i$ is fixed to be in the j-th state. The mean field approximation consists now of approximating the thermal average on the right-hand side of Eq. (4.5) by taking away $\langle \cdot \rangle_T$ and replacing the binary variables s_{ij} by continuous *mean field* variables v_{ij} that serve as approximations to the thermal averages $\langle s_{ij} \rangle_T$. That is,

$$\langle s_{ij} \rangle_T \approx v_{ij} = \frac{e^{-E|_{\mathbf{v}_i = \mathbf{e}_j}/T}}{\sum_k e^{-E|_{\mathbf{v}_i = \mathbf{e}_k}/T}} = \frac{e^{-E_{ij}/T}}{\sum_k e^{-E_{ik}/T}}, \tag{4.6}$$

where

$$E_{ij} = E|_{\mathbf{v}_i = \mathbf{e}_j} - E|_{\mathbf{v}_i = \mathbf{0}} \tag{4.7}$$

is the *local field* of the Potts spin component v_{ij}.

Mean field annealing algorithm. Eqs. (4.6) and (4.7) form the set of mean field equations that is solved iteratively in the following steps:

1. Start at a high temperature T and initialize the spin components v_{ij} close to the infinite-temperature limit value $v_{ij} = 1/K$.

2. Lower the temperature according to $T \to \alpha T$ ($\alpha < 1$). Compute iteratively the local energy from Eq. (4.7) and update the spin variables $\mathbf{v}_i$ according to Eq. (4.6) until the v_{ij} do not change considerably anymore.

3. Repeat step 2 until the spin variables have all converged sufficiently closely to the limiting binary values 0 or 1. Then, the solution of the optimization problem can be decoded from the values of the spin variables.

We hence see that the stochastic method of randomly drawing new configurations according to the Boltzmann distribution Eq. (4.2) has been replaced by an iteration of the deterministic mean field equations, Eqs. (4.6) and (4.7).

Probabilistic interpretation. One readily sees from Eq. (4.6) that

$$0 < v_{ij} < 1, \qquad \sum_j v_{ij} = 1, \tag{4.8}$$

which suggest an interpretation of the mean field Potts spin component v_{ij} as the *probability* for the i-th Potts spin to be in the state j.

Mean field dynamics. Typically, at large temperatures, all spin components have values close to the limiting infinite-temperature value $1/K$ which can be viewed as a trivial fixpoint of the system. When the temperature is lowered and a certain critical temperature is passed, a phase transition (bifurcation) occurs where the spin variables become unstable and start to acquire particular values close to one or zero, determining the new, now non-trivial fixpoint of the system that represents the solution to the optimization problem in question.

One frequently encounters spin variables that exhibit a late onset of decision and/or show more or less strong oscillatory fluctuations before finally a fixed state emerges, which implies that it is particularily hard to find the optimal local configuration of the system represented by the specific spin variable. Remembering the probablistic interpretation of the values of the spin variables, this in turn suggest the existence of alternative, mutually competing solutions that vary among each other in the region corresponding to the 'undecisive' spin variable. This behavior shall be exploited for the application of mean field annealing to sequence alignment which will be described in the next section.

4.2 Implementing Mean Field Annealing

We are now in position to recast the Needleman–Wunsch algorithm in a mean field annealing approach. As mentioned earlier, it is first necessary to map the problem of finding the optimal alignment onto a spin system with a suitably chosen energy function such that minimizing this energy is, in essence, equivalent to iterating the Needleman–Wunsch recursion relation Eq. (2.5); this will be done in Secs. 4.2.1 and 4.2.2. In Sec. 4.2.3, we shall then apply the mean field approximation to the spin variables such that the task of sequence alignment can be performed using the mean field annealing algorithm.

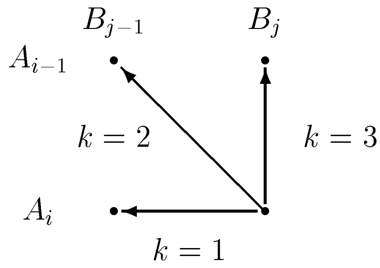

Figure 4.1: The integer variable k specifies the possible nodes adjacent to (i, j) through which an allowed alignment path that ends in (i, j) can pass. Compare also with Fig. 2.2.

4.2.1 Introducing spin variables

Encoding alignment paths. Since every possible alignment can be represented as a directed path that connects adjacent nodes in the alignment matrix (cf. Sec. 2.3), the configuration of the system can be encoded by assigning a spin variable to each node (i, j) whose state specifies the continuation node to be chosen for the optimal alignment path if the node (i, j) is part of the path. This is synonymous to recording for each node (i, j) from which direction k, see Fig. 4.1, the calculation of the optimal alignment score $\mathcal{A}_{\text{opt}}(i, j)$ has been propagated from, *i.e.* which of the three options in the recursion relation Eq. (2.5) has been chosen as to yield the maximum score $\mathcal{A}_{\text{opt}}(i, j)$.

Introducing Potts spins. Since we have three different possible directions k, we assign a $K = 3$ Potts spin variable $\mathbf{s}_{ij}$ to each node (i, j) with the three components $s_{ij,k}$ denoting the direction of propagation. Following the conventions introduced earlier, the state k is then represented by the three-dimensional unit vector $\mathbf{e}_k$, that is, $\mathbf{s}_{ij} = \mathbf{e}_k$ if the calculation has been propagated from the direction k with respect to the node (i, j). We note that, as we now deal with a two-dimensional array of nodes, two indices i and j are needed to specify a particular spin variable.

Retracing alignments. Hence, the $\mathbf{s}_{ij}$ encode the direction of optimal alignment paths. Starting from a node (i, j) and following the directions represented by the states of the Potts spins $\mathbf{s}_{ij}$ up to the initial node $(0, 0)$ will recover the optimal alignment path of the sequence prefix $(A_1 A_2 \ldots A_i)$ of the whole sequence $\mathbf{A} = (A_1 A_2 \ldots A_M)$ with the sequence prefix $(B_1 B_2 \ldots B_j)$ of the whole sequence $\mathbf{B} = (B_1 B_2 \ldots B_N)$. In particular, taking $(i, j) = (M, N)$ as starting node will retrace the optimal alignment of $\mathbf{A}$ with $\mathbf{B}$.

Initializing fixed spins. Since for nodes in the first column or row of the alignment matrix only at most one directly precessing node exists, the values of the corresponding variables $s_{ij,k}$ are just constants, independent of the alignment, that can accordingly be set as initial values, namely

$$\begin{aligned} s_{00,k} &= 0, & k &= 1,2,3, \\ s_{i0,k=1} = s_{i0,k=2} = 0,\ s_{i0,k=3} &= 1, & i &= 1,\ldots,M, \\ s_{0j,k=1} = 1,\ s_{0j,k=2} = s_{0j,k=3} &= 0, & j &= 1,\ldots,N. \end{aligned} \tag{4.9}$$

4.2.2 Introducing an energy function

The second step in the procedure of problem mapping is now the construction of a suitable energy function of the system that is a function of the spin states $\mathbf{s}_{ij}$ and whose optimization yields the optimal alignment.

Optimization goal. To this extent, we remember that the Needleman–Wunsch algorithm finds the optimal alignment by computing $\mathcal{A}_{\rm opt}(i,j)$ for *all* nodes (i,j) in the alignment matrix, that is, by maximizing the path lengths (in terms of the score contributions from residue matches and gap penalties along the path) for valid alignment paths connecting the source node $(0,0)$ with every node (i,j). Therefore, the optimization goal is to find the spin configuration $\mathcal{S} = \{\mathbf{s}_{ij} : 0 \le i \le M, 0 \le j \le N\}$ for which the local alignment score $\mathcal{A}(i,j)$ is maximized for *each* node. As we have seen earlier, the Needleman–Wunsch algorithm determines this optimal configuration $\mathcal{S}$ by iteratively executing the recursion relation Eq. (2.5).

Reformulating the recursion relation. Which of the three options in the recursion relation has been chosen as to yield the maximum score $\mathcal{A}_{\rm opt}(i,j)$ is encoded in the Potts spin $\mathbf{s}_{ij}$, so we can replace the 'max' function in Eq. (2.5) by the sum

$$\mathcal{A}_{\rm opt}(i,j) = \sum_{k=1}^{3} s_{ij,k}\mathcal{A}_{\rm opt}(i,j;k), \tag{4.10}$$

where the $\mathcal{A}_{\rm opt}(i,j;k)$ denote the resulting alignment scores for the three possible options of propagation,

$$\begin{aligned} \mathcal{A}_{\rm opt}(i,j;k=1) &\equiv \max_{0\le b<j}\{\mathcal{A}_{\rm opt}(i,b) - g(j-b)\}, \\ \mathcal{A}_{\rm opt}(i,j;k=2) &\equiv \mathcal{A}_{\rm opt}(i-1,j-1) + s(A_j,B_i), \\ \mathcal{A}_{\rm opt}(i,j;k=3) &\equiv \max_{0\le a<i}\{\mathcal{A}_{\rm opt}(a,j) - g(i-a)\}, \end{aligned} \tag{4.11}$$

and accordingly

$$s_{ij,k} = \begin{cases} 1 & \text{if}\, \mathcal{A}_{\rm opt}(i,j;k) = \max_{1\le k'\le 3}\{\mathcal{A}_{\rm opt}(i,j;k')\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

Restriction to a linear gap penalty. To simplify the equations, we shall restrict ourselves from now on to a linear gap penalty function. We have already pointed out earlier (cf. Sec. 2.4), that in this case it is sufficient to check whether the insertion of the gap corresponds to an opening or an extension of a gap in order to assign the correct gap penalty. That is, to compute $\mathcal{A}_{\rm opt}(i,j;k=1)$ and $\mathcal{A}_{\rm opt}(i,j;k=3)$, we only need to check whether $s_{i,j-1,k=1}$ and $s_{i-1,j,k=3}$, respectively, have a value equal to one (in which case the insertion of the gap corresponds to an extension) or equal to zero (which implies the opening of a gap). The expressions for the $\mathcal{A}_{\rm opt}(i,j;k)$ can then be rewritten compactly in terms of the Potts spin variables as

$$\begin{aligned}
\mathcal{A}_{\rm opt}(i,j;k=1) &\equiv \mathcal{A}_{\rm opt}(i,j-1) - (1-s_{i,j-1,k=1})\, g_{\rm open} - s_{i,j-1,k=1}\, g_{\rm ext}, \\
\mathcal{A}_{\rm opt}(i,j;k=2) &\equiv \mathcal{A}_{\rm opt}(i-1,j-1) + s(A_j,B_i), \qquad (4.13)\\
\mathcal{A}_{\rm opt}(i,j;k=3) &\equiv \mathcal{A}_{\rm opt}(i-1,j) - (1-s_{i-1,j,k=3})\, g_{\rm open} - s_{i-1,j,k=3}\, g_{\rm ext}.
\end{aligned}$$

Energy function. Now we are in position to set up an energy function $E(\mathcal{S})$ of our system as desired for the application of the mean field annealing optimization technique. A suitable choice for this energy function that will be shown to match our optimization goal is

$$E(\mathcal{S}) = -\sum_{ij} \mathcal{A}_{\rm opt}(i,j) = -\sum_{ij}\sum_{k=1}^{3} s_{ij,k}\mathcal{A}_{\rm opt}(i,j;k). \qquad (4.14)$$

Then the local fields $E_{ij,k}$ of the Potts spin components $s_{ij,k}$ are, from Eq. (4.7) (for the moment being still with binary spins $\mathbf{s}_{ij}$) and using the expressions Eqs. (4.13),

$$E_{ij,k} = E|_{\mathbf{s}_{ij}=\mathbf{e}_k} - E|_{\mathbf{s}_{ij}=\mathbf{0}} = -\mathcal{A}_{\rm opt}(i,j;k), \qquad (4.15)$$

where we have dropped constant terms that are independent of i, j and k. This allows consistency with the reformulated Needleman–Wunsch recursion relation Eq. (4.10) if we identify $-\mathcal{A}_{\rm opt}(i,j)$ with a *local energy* E_{ij} of the node (i,j) such that we can write

$$E_{ij} = \sum_{k=1}^{3} s_{ij,k}E_{ij,k}. \qquad (4.16)$$

That is to say, we have motivated our particular choice of $E(\mathcal{S})$ by demonstrating that minimizing (note the this energy is essentially equivalent to the iteration of the Needleman–Wunsch recursion relation. Hence, the optimization goal has now been reformulated as the strive of each node to minimize its own local energy E_{ij}. From Eqs. (4.15) and (4.16) we notice that the expressions for the local fields $E_{ij,k}$ are directly obtained from Eqs. (4.13) by replacing $\mathcal{A}_{\text{opt}}(i,j;k)$ and $\mathcal{A}_{\text{opt}}(i,j)$ by $-E_{ij,k}$ and $-E_{ij}$; accordingly, the spin components $s_{ij,k}$ are then given by, cf. Eq. (4.12),

$$s_{ij,k} = \begin{cases} 1 & \text{if } E_{ij,k} = \min_{1\le k'\le 3}\{E_{ij,k'}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

4.2.3 Applying the mean field approximation

Introducing mean field spins. So far, we have only introduced a reformulation, Eq. (4.16), of the Needleman–Wunsch recursion relation in terms of energies and binary Potts spin variables. Now, let us finally use this as basis for the application of the mean field annealing method that replaces the strictly binary "winner-takes-all" spins $s_{ij,k}$, Eq. (4.17), by continuous "winner-takes-most" mean field variables $v_{ij,k}$. Following our treatment in Sec. 4.1.3, the final configuration of the spin variables $\mathbf{s}_{ij}$ that encodes the optimal alignment can be obtained by computing the thermal averages $\langle \mathbf{s}_{ij}\rangle_T$ and letting $T \to 0$, where $\langle \mathbf{s}_{ij}\rangle_T$ can be approximated by the mean field spin variables $\mathbf{v}_{ij}$ whose k-th component at temperature T is, from Eq. (4.6),

$$v_{ij,k} = \frac{e^{-E_{ij,k}/T}}{\sum_{k'=1}^{3} e^{-E_{ij,k'}/T}}. \tag{4.18}$$

The local fields $E_{ij,k}$ and energies E_{ij} are given by the same expressions as in the binary case, but now with the $s_{ij,k}$ replaced by the mean field spin variables $v_{ij,k}$. Additionally, we shall normalize all energies by the average sequence length $n = (M+N)/2$, since the energies are additive functions of the scores and would thereby aquire absolute values that depend upon the length of the sequences. This in turn would introduce an undesired dependency of the argument of the exponentials in Eq. (4.18) on the sequence length. With this normalization and the replacement $s_{ij,k} \to v_{ij,k}$ we obtain (cf. Eqs. (4.13))

$$\begin{aligned} E_{ij,k=1} &= E_{i,j-1} + \frac{1}{n}\Big[(1 - v_{i,j-1,k=1})\, g_{\text{open}} + v_{i,j-1,k=1}\, g_{\text{ext}}\Big], \\ E_{ij,k=2} &= E_{i-1,j-1} - \frac{1}{n} s(A_j, B_i), \\ E_{ij,k=3} &= E_{i-1,j} + \frac{1}{n}\Big[(1 - v_{i-1,j,k=3})\, g_{\text{open}} + v_{i-1,j,k=3}\, g_{\text{ext}}\Big], \end{aligned} \tag{4.19}$$

and

$$E_{ij} = \sum_{k=1}^{3} v_{ij,k} E_{ij,k}, \tag{4.20}$$

where the local energies associated with the nodes in the left and top margin of the alignment matrix are fixed by Eqs. (2.6), namely,

$$\begin{aligned} E_{00} &= 0, \\ E_{i0} &= \frac{1}{n} g_{\text{term}}(i), \qquad i = 1, \ldots, M, \\ E_{0j} &= \frac{1}{n} g_{\text{term}}(j), \qquad j = 1, \ldots, N, \end{aligned} \tag{4.21}$$

which for free terminal gaps ($g_{\text{term}} \equiv 0$) simplifies to

$$E_{i0} = E_{0j} = 0, \qquad i = 0, \ldots, M, \quad j = 0, \ldots, N. \tag{4.22}$$

Zero-temperature limit. In the limit $T \to 0$, the $v_{ij,k}$ turn into the binary variables Eq. (4.12) and the original Needleman–Wunsch algorithm is recovered. This can readily be seen by rewriting Eq. (4.18) as

$$v_{ij,k} = \frac{1}{1 + \sum_{k' \neq k} e^{-(E_{ij,k'} - E_{ij,k})/T}}. \tag{4.23}$$

Suppose

$$E_{ij,k} = \min_{1 \leq k' \leq 3} \{E_{ij,k'}\}, \tag{4.24}$$

then $E_{ij,k'} - E_{ij,k} > 0$ for the two $k' \neq k$ and the exponentials in the denominator vanish in the limit $T \to 0$, which implies $v_{ij,k} \to 1$; otherwise, there will be at least one $k' \neq k$ such that $E_{ij,k'} - E_{ij,k} < 0$ and hence,

$$e^{-(E_{ij,k'} - E_{ij,k})/T} \to \infty \tag{4.25}$$

as $T \to 0$, so $v_{ij,k} \to 0$.

Non-zero temperatures. For $T > 0$, each component of the spin variable $\mathbf{v}_{ij}$ will in general attain a non-zero value, where the component $v_{ij,k}$ with the largest value corresponds to the direction k for which the local field $E_{ij,k}$ attains its minimum. This is synonymous with the direction of propagation for which the largest alignment score is obtained. The particular values of the spin components indicate how large the differences in the resulting alignment scores among the three possible directions are. For example, if two components have similar values but the third component is close to zero, we can interpret this as the presence of two alternative directions of propagation that yield only a slightly different alignment score, thereby indicating a possible suboptimal alignment path. Remembering the probabilistic interpretation that can be attributed to the spin components, cf. Eq. (4.8), we can interpret the value $v_{ij,k}$ as the *probability* that an optimal alignment path passing through (i, j) will contain the node specified by the direction k. The more the temperature is increased, the more pronounced the contribution to spin components corresponding to non-optimal directions becomes, until in the infinite-temperature limit $v_{ij,k}(T \to \infty) \to 1/3$ for all k, and all information about a particular alignment path is lost; in this sense, the parameter T controls the fuzziness of the alignment path.

Algorithm. For a given temperature $T > 0$, the set of equations (4.19), (4.18) and (4.20) (in this order) is iterated for all nodes (i, j) until all spin variables have adopted fixed values. Since sequence alignment is a particularly simple optimization problem, usually a single iteration step is sufficient, given that the update is efficiently done. This can be achieved by proceeding through the alignment matrix row by row and column by column (as in the case of the application of the Needleman–Wunsch recursion relation) which ensures that new values are always computed with variables that have already been updated in the current iteration step. Once a steady state is reached, the temperature can be lowered according to $T \to \alpha T$ with $\alpha < 1$ (simulated annealing), and the iteration procedure can be repeated. Finally, when the $v_{ij,k}$ have converged close enough to their binary limits $s_{ij,k}$, Eq. (4.17), the resulting optimal alignment can be decoded from the spins in the usual way. Due to the relative simplicity of the optimization problem, this alignment is likely to coincide with the alignment that would have been obtained from the exact Needleman–Wunsch algorithm.

4.3 Obtaining a Reliability Measure

4.3.1 Monitoring mean field dynamics

Evaluating spin components. As we have pointed out in Chapter 3, the local thrustworthiness along the optimal alignment can be quantified by the extent to which suboptimal alternatives are locally present. In the preceding section, we have seen that the values of the mean field spin components can be interpreted as probabilities for a particular alignment path. A first approach towards a quantification of the local reliability of the optimal alignment could therefore be based on evaluating the values of the spin components that correspond to the optimal path at some temperature $T^* > 0$. The farther away from the limiting value of one the value of a component is found to be, the more likely suboptimal alternatives to the associated pair exist in the alignment and hence as the less trustworthy would the aligned pair be considered.

Arising difficulties. This idea had been pursued in the beginning of this thesis work. However, the choice of the particular temperature T^* at which the spin variables are evaluated turns out to be a severe problem. The evolution

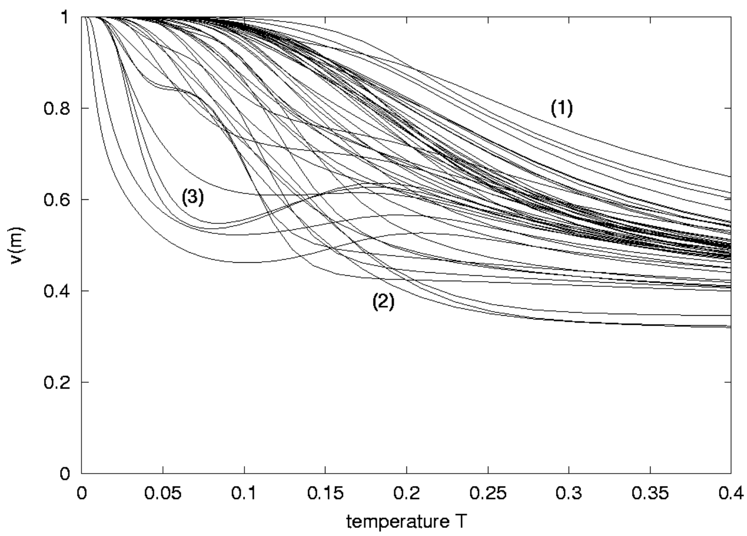

Figure 4.2: A typical evolution of the spin components corresponding to the optimal path. Some spins exhibit a fast decision towards the limiting value of one (1), representing a dominating, non-ambiguous local solution to the optimization problem. Other spins show only late convergence at low T (2), sometimes with additional oscillations (3); both indicates the presence of competing alternative solutions. Note that in the case of simulated annealing (lowering T), the development of the spins in time is given by reading from right to left, whereas for simulated smelting (increasing T) the directions of the time and the T axis coincide.

of the spin values depends on the problem under consideration, *i.e.* on the sequences to be aligned, and hence T^* would have to be chosen specifically for each new pair of sequences to obtain a meaningful reliability measure that allows comparison among different alignments. Such a dynamical choice is neither easy to accomplish (in particular with respect to the desired absolute reliability measure) nor very elegant as considerable arbitrarity in the selection remains. Moreover, strongly fluctuating components, actually indicating a particularly prominent existence of suboptimal solutions, might be evaluated at a temperature where their values happen to have only temporarily attained a value close to the binary limit of one; the corresponding pairs in the alignment would then be misclassified as reliably aligned despite the actual ambiguity of the solution.

Monitoring mean field dynamics. In the light of these difficulties and disadvantages, we have pursued a different approach that is based on monitoring the mean field *dynamics* of the system during the process of simulated annealing instead of evaluating spin components at a *fixed* temperature. In our description of the typical mean field dynamics in Sec. 4.1.3, we have pointed out that spin variables whose components are subject to oscillatory fluctuations and/or to a late onset of decision towards binary values usually correspond to the existence of a multiplicity of possible local solutions that make fast convergence to a unique solution difficult.

The central idea of our method is therefore based on monitoring the behavior of the spins that are part of the optimal alignment along the entire annealing process. A typical evolution of these spin components $v_{ij,k}$ as a function of T is shown in Fig. 4.2. All $v_{ij,k}$ on the optimal path will finally approach the limiting value of one (since they specify the optimal alignment), but the curves of evolution in the $v_{ij,k}$ vs. T diagram will enclose different areas with the T axis. Spin components that attain an early and final convergence will span a larger area than nodes that exhibit late convergence and/or oscillations.

Quantifying spin evolution. Quantifying these areas can simply be done by an integration of the $v_{ij,k}$ belonging to the optimal path over T. Denoting the initial (high) temperature in the process of simulated annealing by T_0 and the component of the spin variable corresponding to the m-th aligned pair on the optimal path by $v^{(m)}$, we shall therefore define a quantity $r(m)$ for the m-th pair in the optimal alignment as

$$r(m) = \frac{1}{T_0} \int_{\varepsilon \to 0}^{T_0} v^{(m)} \, dT. \tag{4.26}$$

Thus, $r(m)$ measures the ratio of the area that the curve $v^{(m)}$ encloses with the T axis to the largest area that the curve of a spin component could possibly enclose (which is the area $T_0 - \varepsilon \approx T_0$ enclosed by a curve with the constant value of one).

Defining a reliability index. Motivated by our remarks above, we shall define $r(m)$ as the *local reliability* of the m-th pair in the optimal alignment; this naming will be justified later. We shall report $r(m)$ as a *reliability index* taking an integer value between 0 (corresponding to $0 \leq r(m) < 0.1$) and 9 (corresponding to $0.9 \leq r(m) < 1.0$). We see that $r(m)$ can be computed both for residue–residue and residue–gap pairs in the alignment; only initial residue–gap pairs can not receive a reliability index by this method since the corresponding spin variables $v_{0j,0}$ and $v_{i0,2}$ are required to retain their initial values throughout the entire annealing process, which is of no relevance as assessing reliability to terminal gaps is anyhow a procedure of arguable biological meaning. Clearly, the choice of T_0 will influence the obtained reliability index, and we shall discuss below how to determine T_0 such that we can obtain an absolute measure that allows comparison among different alignments.

Discretized version. In the process of simulated annealing, the temperature is only changed in discrete steps $dT = |\alpha - 1|T$ (since $T \to \alpha T$ in each step), so we can replace the continuous integration by a summation over single discrete integration steps given by

$$r(m) = r(m) + |\alpha - 1| \frac{T}{T_0} v^{(m)}. \tag{4.27}$$

Discussion. We shall emphasize that our reliability index arises naturally from the dynamics of the optimization problem itself and does therefore not require the construction of a separate algorithm independent of the alignment task. The reformulation of the Needleman–Wunsch algorithm in terms of a mean field approach has provided the possibility to explicitely monitor the behavior of each part of the system while the optimization process is performed. This, in turn, allows to infer and quantify the existence of ambiguities and suboptimalities, which, as we have seen in Chapter 3, are typically used to deduce a measure for local reliability of a sequence alignment. Of course, so far this index value is just a mathematical definition motivated by an interpretative study of mean field dynamics. Hence we will have to verify experimentally whether $r(m)$ in fact mirrors the trustworthiness of an aligned pair and can therefore serve as a meaningful measure for local reliability.

4.3.2 Simulated smelting

Difficulties in the original approach. As it has been described above, the procedure for obtaining reliability indices exhibits some minor drawbacks that can easily be remedied. Firstly, the spins may evolve very differently dependent on the alignment problem under consideration, which requires a dynamical choice of the starting temperature T_0 based on the knowledge of the behavior of the spins to obtain absolute values of the reliability index, Eq. (4.26), that allow comparison among alignments; the evolution of the spins, however, is not known before the annealing process has actually been performed. Secondly, since in the mean field algorithm the optimal alignment is only determined when the annealing process is completed, it is not known from the beginning which nodes $v_{ij,k}$ will be part of the optimal path and hence require monitoring. Finally, it is not guaranteed (however likely due to the relative simplicity of the optimization problem) that the mean field algorithm will yield precisely the same optimal alignment as the exact Needleman–Wunsch algorithm, where the latter is of course the alignment we wish to obtain.

Modifying the procedure. Therefore, we shall introduce the following modified procedure. First of all, we calculate the optimal alignment with binary spin variables $\mathbf{s}_{ij}$, corresponding to the zero-temperature limit and thus recovering the exact Needleman–Wunsch algorithm. This provides us with a knowledge of the spin variables $v^{(m)}$ that are part of the optimal path and thus need to be taken into account for computing reliability indices. Consistency between the exact and the mean field solution is then achieved by taking the resulting set of spin variables $\mathbf{s}_{ij}$, local energies E_{ij} and local fields $E_{ij,k}$ as the initial configuration at $T = \varepsilon \to 0$ of a system to which now the mean field algorithm is iteratively applied while T is *increased* according to $T \to \alpha T$ with $\alpha > 1$ (a process for which we may coin the term *simulated smelting*), until the (now final) temperature T_0 is reached. Along this process, the step-by-step integration of the $v^{(m)}$ according to Eqs. (4.26) or (4.27) is performed.

We can of course not expect that the behavior of the mean field spin variables in a process of simulated smelting will be exactly the same as in a process of simulated annealing. However, it turns out that typical patterns of behavior which are significant in obtaining our reliability measure, such as strong fluctuations and late decisions, are retained in both possible directions of changing T.

4.3.3 Choice of integration limits

Lower limit. Obviously, the obtained value of the reliability index is influenced by the choice of the temperature parameters ε and T_0. The selection of ε is uncritical as long at is chosen small enough such that in a small region above this temperature the mean field spin variables will still match the binary variables with sufficient stability in their values. This will then ensure the desired consistency between the binary configuration of the $\mathbf{s}_{ij}$ obtained from the exact Needleman–Wunsch algorithm and the initial evolution of the mean field variables $\mathbf{v}_{ij}$.

Upper limit. In contrary, the choice of the parameter T_0 is non-trivial. If T_0 is selected too low, significant fluctuations of some $v^{(m)}$ at higher temperatures that can have important impact on the value of the integral and therefore on the resulting reliability index might get excluded from the integration. On the other hand, if T_0 is chosen too high, the integration might span a wide region of temperatures for which all or virtually all $v^{(m)}$ have attained stable values close to the value of the infinite-temperature limit of 1/3. The contribution of this region to the integration will then both blur out the differences between the reliability indices and uniformly decrease the absolute value of all reliability indices. Moreover, as pointed out before, since the behavior of the $v^{(m)}$ is dependent upon the alignment problem under consideration, a choice of the final temperature T_0 that is globally valid for all sequence pairs can not be expected to yield satisfying results.

Saturation. To circumvent these problems, our algorithm is based on a dynamic choice of T_0 that directly incorporates the current values of the evolving spin variables $v^{(m)}$. To this extent, we introduce the *saturation* $\Sigma(T)$ of a fuzzy alignment as

$$\Sigma(T) = \frac{1}{N_{\text{opt}}} \sum_{m=1}^{N_{\text{opt}}} v^{(m)}, \tag{4.28}$$

where N_{opt} is the number of nodes on the optimal path, excluding the constant spin components $v_{0j,0}$ and $v_{i0,2}$ that correspond to initial gaps in the alignment. We see that $\Sigma(T)$ is just the arithmetic average over all $v^{(m)}$. Neglecting degeneracies on the optimal path, $\Sigma(T) \to 1$ for $T \to 0$, since in this limit all $v^{(m)}$ approach the value one, corresponding to binary spin variables in the case for the exact Needleman–Wunsch algorithm. On the other hand, in the high-temperature limit $T \to \infty$, $v_{ij,k} \to 1/3$ for all k and therefore $\Sigma(T) \to 1/3$.

Accordingly, for a given temperature T, the saturation quantifies the degree of convergence of the spins, either towards their binary $T \to 0$ values or their $T \to \infty$ limits. For our purpose, it provides the criterion for determining the appropriate value of T_0 in the process of simulated smelting. When the saturation $\Sigma(T)$ has decreased to a predefined value Σ_0, the $v^{(m)}$ will have, in average, attained values suffiently close to the high-temperature limit of 1/3 with a certain degree of steadiness, and we can be widely confident that terminating the integration at this point will not cut off significant spin fluctuations. Hence, instead of specifying a fixed cut-off temperature T_0, we determine the point of termination dynamically from the current value of the saturation. Of course, the suitably balanced choice of Σ_0 that allows for the inclusion of meaningful fluctuations in certain $v^{(m)}$ at higher temperatures by simultaneously preserving clearly distinct values for the reliability index that span the entire range of possible values remains a matter of empirical studies. However, once a satisfying value has been found, it can globally be applied to the alignment of any two sequences.

4.3.4 Summary of the algorithm

The complete algorithm that performs an optimal global Needleman–Wunsch alignment of two given sequences and assigns a reliability index to each aligned pair (apart from initial gap-residue pairs) using the mean field annealing algorithm is summarized in Fig. 4.3.

As a first step, the optimal alignment is calculated in the $T = 0$ limit using binary spin variables, which is equivalent to the original Needleman–Wunsch algorithm. Then, the resulting configuration of the system is taken as the initial low-T state for an execution of the mean field annealing algorithm with continuous spin variables and increasing temperature. The evolution of the values of the spin components corresponding to all pairs in the optimal alignment is recorded by a series of discretized integration steps to compute the reliability index of each aligned pair. The process of simulated smelting

type	parameter	value
gap penalties	g_{open}	11.0
	g_{ext}	1.0
mean field annealing	ε	0.01
	Σ_0	0.55
	α	1.1

Table 4.1: The parameters used in the `MAXalign` program.

is terminated as soon as the saturation has decreased to a predefined value. Finally, we output the optimal alignment with the computed reliability indices for each aligned pair.

4.3.5 Implementation of the algorithm

The `MAXalign` program. The algorithm outlined above has been implemented in a computer program, `MAXalign`, to allow the evaluation of the performance of our reliability measure. The program has been written in the programming language `C` under the operating system `Linux`. A typical output of the program consisting of the optimal alignment with assigned reliability indices is shown in Fig. 4.4.

Parameters. The parameters used in the `MAXalign` program are summarized in Tab. 4.1. A linear gap penalty with $g_{\text{open}} = 11$ and $g_{\text{ext}} = 1$, values commonly used in sequence alignment algorithms, was chosen. Terminal gaps were not penalized to allow for a free translational shift of the sequences with respect to each other. As scoring matrix, a refined version of the PAM-250 matrix introduced by Gonnet *et al.* [8] was used, with matrix elements (similarity scores) ranging from -5.2, corresponding to the worst match of an aspartate to a tryptophan, up to $+14.2$ for the alignment of two tryptophans, the best possible match.

As pointed out, the precise choice of the value of the low-temperature limit ε in Eq. (4.26) is fairly uncritical with respect to the resulting values of the reliability indices, as long ε is selected sufficiently low such that at this temperature the system has not deviated much from its binary zero-temperature limit. On the other hand, ε has dominant influence on the number of annealing steps required to attain the final saturation Σ_0. To optimize the algorithmical speed, it is therefore desireable to choose this parameter as high as possible, by simultaneously retaining consistency with the $T = 0$ case. We used a value of $\varepsilon = 0.01$ as a suitable choice.

Regarding the saturation cut-off Σ_0, see Sec. 4.3.3, values between 0.45 and 0.55 yielded good results; the sensitivity of the reliability index to the precise

1. Perform a global (exact) Needleman–Wunsch alignment of the two given sequences:

   (a) For $i = 0, \ldots, M$ and $j = 0, \ldots, N$, initialize $\mathbf{s}_{i0}$, $\mathbf{s}_{0j}$ according to Eqs. (4.9), and E_{i0} and E_{0j} according to Eqs. (4.21) or (4.22).

   (b) By proceeding row by row $i \to i+1$, $i = 1, \ldots, M$, and in each row column by column, $j \to j+1$, $j = 1, \ldots, N$, calculate recursively for each node (i, j) in the given order:

      i. $E_{ij,k}$ from Eqs. (4.19) (with binary $\mathbf{s}_{ij}$ instead mean field $\mathbf{v}_{ij}$);

      ii. (binary) $s_{ij,k}$ from Eq. (4.17);

      iii. $E_{ij} = \sum_{k=1}^{3} s_{ij,k} E_{ij,k}$ (see Eq. (4.16)).

   (c) When finished, retrace the optimal path by starting from node (M, N) and following the directions encoded in the $\mathbf{s}_{ij}$ until the initial node $(0, 0)$ is reached.

   (d) Save resulting optimal alignment.

2. Compute a reliability index for every pair in the optimal alignment using mean field annealing:

   (a) Take the $\mathbf{s}_{ij}$, E_{ij} and $E_{ij,k}$ calculated from the Needleman–Wunsch algorithm in step 1 as initial configuration, with the binary $\mathbf{s}_{ij}$ replaced by continuous $\mathbf{v}_{ij}$. Initialize the temperature T close to zero, and all reliability indices as $r(m) = 0$.

   (b) Compute iteratively for each node (i, j) until convergence:

      i. $E_{ij,k}$ from Eqs. (4.19);

      ii. (fuzzy) $v_{ij,k}$ from Eq. (4.18);

      iii. $E_{ij} = \sum_{k=1}^{3} v_{ij,k} E_{ij,k}$ (see Eq. (4.20).

   (c) Using the values of the spin variables $v^{(m)}$ on the optimal path,

      i. perform a single discrete integration step, $r(m) = r(m) + |\alpha - 1| T v^{(m)}$ (cf. Eq. (4.27));

      ii. calculate the saturation $\Sigma(T)$ from Eq. (4.28).

   (d) If $\Sigma(T) > \Sigma_0$, increase $T \to \alpha T$ and go back to step 2b.

   (e) Normalize $r(m) \to r(m)/T$.

3. Output the optimal alignment together with the $r(m)$ (binned into reliability indices) for each aligned pair.

Figure 4.3: Summary of the algorithmical steps for the computation of the optimal alignment with reliability indices assigned to every aligned pair.

```
90% - 100% |      ** **                 ***          *                  *********
80% -  90% |    *******             *********    ***********        *  **********
70% -  80% | ************   **** ************   ************   * ** *************
60% -  70% | ************  ******************** ************  *******************
50% -  60% | ********************************************************************
40% -  50% | ********************************************************************
30% -  40% | ********************************************************************
20% -  30% | ********************************************************************
10% -  20% | ********************************************************************
 0% -  10% | ********************************************************************

             KSPEELKGIFEKYAAKEGDPNQLSKEELKLLLQTEFPSLLKGPSTLD--ELFEELDKNGDGEVSFEEF
             FSADELKKLF--KIADEDKEGFIEEDELKLFL-IAFAADLRALTDAETKAFLKAGDSDGDGKIGVDEF
```

Figure 4.4: A typical output from the `MAXalign` program. The optimal alignment of the two sequence strings is shown at the bottom, with the profile of reliability indices (here given as percentages) drawn above.

choice of this parameter has been found to be not as pronounced as one might expect. For the final implementation, we settled for a value of $\Sigma_0 = 0.55$ which seemed to perform best.

Finally, the annealing parameter α determines the size of temperature changes in the process of simulated smelting, cf. Eq. (4.27). Higher values of α will lower the number of required annealing steps and therefore speed up the computation. Proper functionality of mean field annealing is, however, only ensured if the temperature is changed sufficiently slowly. Values between 1.0 and 1.5 have been found to be a good compromise; for the `MAXalign` program, a value of $\alpha = 1.1$ was finally used.

Degeneracies. In the presentation of the original Needleman–Wunsch algorithm in Sec. 2.4.2, we have mentioned the possibility of a multiplicity of optimal alignments, that is, the presence of several, equally scoring optimal paths. In the language of our spin system this corresponds to two of the three spin components having an identical value of 0.5, with the third component necessarily being equal to zero (the case of all three components being equal to 1/3 is very unlikely and has never been observed). In the course of our studies, such degeneracies have been found to occur in nodes along the optimal path only for alignment ambiguities such as

$$\begin{array}{cc|cc|cc} \ldots & \mathrm{A} & \mathrm{B} & \mathrm{B} & \mathrm{C} & \ldots \\ \ldots & \mathrm{X} & \mathrm{Y} & - & \mathrm{Z} & \ldots \end{array} \qquad \text{vs.} \qquad \begin{array}{cc|cc|cc} \ldots & \mathrm{A} & \mathrm{B} & \mathrm{B} & \mathrm{C} & \ldots \\ \ldots & \mathrm{X} & - & \mathrm{Y} & \mathrm{Z} & \ldots \end{array} \tag{4.29}$$

Clearly, these two alternatives are indistinguishable from an evolutionary point of view and are of arguable structural difference. Therefore, we can, for the sake of simplicity, legitimately refrain from an explicit evaluation of possible multiple optimal alignments by choosing a trace-back direction at random whenever an ambiguous spin variable is encountered on the optimal path.

4.4 Validation Data

To establish the validity of our novel reliability measure, we need to compare the Needleman–Wunsch aligment of a pair of sequences with the correct reference alignment of the same pair of sequences to evaluate whether residues that have been aligned correctly by the algorithm correspond, in average, to a high reliability index, whereas misalignments have dominantly received low reliabilities. This will then allow us to demonstrate that our reliability index serves in fact as a meaningful measure for the local trustworthiness of an algorithmically obtained sequence alignment.

Reference alignments. The validation procedure outlined above requires the availability of a sufficiently large set of alignments that can be taken as the correct reference. For this purpose, we used the `3D_ali` database [20] that provides a broad collection of multiple sequence alignments (simultaneous alignments of several sequences) that have been obtained from thorough structural superpositions amongst proteins with similar three-dimensional properties. Referring to our definition of the standard of truth that we have adopted in Sec. 2.1, these alignments will serve as a set of correct reference alignments for the evaluation of our algorithm.

Structure and derivation of the database. The `3D_ali` database currently consists of 69 protein families, where each family contains multiple alignments of a set of sequences. Familial constituency was purely based on similarities in the main-chain fold, regardless of the distance between the resulting aligned sequences. This approach allows for very distant and yet reliable multiple alignments. The three-dimensional structures of one or several proteins in a family were explicitely known from a crystallographic analysis. These structures were superposed to obtain multiple structural alignments. Protein sequences taken from sequence databases were then associated with the multiple structural alignments, provided that they were at least 50% (group G-50) or 35% (group G-35) (alternatively) homologous in residue identity to a sequence of one of the proteins in the structural alignments, and at least in 50% of the residues alignable to the sequence in the structural alignment. The high thrustworthiness of these accurately derived alignments made the `3D_ali` database a frequently used tool for the evaluation of both sequence alignment algorithms and reliability measures.

Extraction of the validation data. From the multiple alignments in the database, pairwise alignments and the corresponding sequences were extracted to be used as a test set in the validation of our algorithm. From a family consisting of a multiple alignment of N sequences $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_N$, pairwise alignments $\mathbf{A}_i \| \mathbf{A}_j$ were produced by forming

$$\begin{array}{cccccc}
\mathbf{A}_1\|\mathbf{A}_2, & \mathbf{A}_1\|\mathbf{A}_3, & \mathbf{A}_1\|\mathbf{A}_4 & \ldots & \mathbf{A}_1\|\mathbf{A}_{N-1}, & \mathbf{A}_1\|\mathbf{A}_N, \\
 & \mathbf{A}_2\|\mathbf{A}_3, & \mathbf{A}_2\|\mathbf{A}_4 & \ldots & \mathbf{A}_2\|\mathbf{A}_{N-1}, & \mathbf{A}_2\|\mathbf{A}_N, \\
 & & & \ddots & & \\
 & & & & \mathbf{A}_{N-2}\|\mathbf{A}_{N-1}, & \mathbf{A}_{N-2}\|\mathbf{A}_N, \\
 & & & & & \mathbf{A}_{N-1}\|\mathbf{A}_N,
\end{array} \tag{4.30}$$

which yields a set of $N(N-1)/2$ pairwise alignments.

Classification of the validation data. Besides the separation into the two groups G-50 and G-35 of alignments that is already implemented in the original database, a further classification into subgroups of alignments with a defined range of similarity in the corresponding sequence pairs was performed to elucidate differences in the performance of our algorithm that are due to the degree of sequence similarity. Similarity is here defined as the *percent sequence identity* (PSI) of residue pairs in the optimal alignment, *i.e.* the fraction of identical residues pairs in the alignment computed by the Needleman–Wunsch algorithm.

Depending on the PSI value, alignments were assigned to one of three different *similarity classes* defined by 25–30%, 30–40% and 40–50% PSI, respectively. Sequence pairs with lower PSI values are found to usually not share relevant structural similarities, and algorithms normally fail to detect any possible relations [22]. On the other hand, algorithms align in general sequences with PSI values above 50% correctly in most of the residues. Accordingly, sequence pairs for which a PSI value lower than 25% or higher than 50% was obtained were excluded from our analysis.

Normalization of the data points. The population, *i.e.* the number of included sequences varies strongly among different families in the `3D_ali` database. Families such as globins and immunglobulins contain up to $n = 1,000$ sequences, whereas the majority of families consists of about $n = 5$ to $n = 50$ sequences each. Since the number of possible pairwise alignments is of $\mathcal{O}(n^2)$, a vast number of often very similar pairwise alignments will be obtained from more highly populated families. Attributing an equal statistical weight to all of these alignments in the evaluation of the complete set of families would accordingly lead to an unwanted, strong bias in the results relative to the influence from more sparsely populated families. To avoid this effect, the results obtained from the evaluation of one family were normalized by the number of sequence pairs taken into account.

Restricting the size of the data set. In general, groups of sequences within one family are related to each other and therefore yield often highly similar alignments. For highly populated families, frequently a large number of such very alike alignments will arise, and the evaluation of all these alignments instead of only a few representatives can therefore not be expected to yield improvements in the statistical significance of the results, particularly not since we normalize the results by the number of alignments taken into account. Hence, without loss in the validity of our results, we can greatly reduce the computational times required for a complete evaluation of our test data set by including only a limited number of alignments for a given family, group and class into the analysis; this limit was set to 40 alignments in all test runs.

Chapter 5

Results and Discussion

In this chapter, we shall present and discuss the results of the studies investigating the performance of our novel reliability measure. In Sec. 5.1, we will discuss the obtained relationships between the value of the reliability index and the likelihood for a correct alignment and compare our results to those of the work by Mevissen and Vingron [16] presented in Sec. 3.2.2. The required computational effort of our algorithm will be investigated in Sec. 5.2.

5.1 Reliability Index and Correct Alignment

Motivation. Most importantly, we would like to verify that the reliability index derived from our algorithm does indicate the likelihood that a particular residue–residue or residue–gap pair is actually correctly aligned. That is, we would like pairs that receive a high reliability index to be more likely correctly aligned with respected to the `3D_ali` reference than pairs with a lower index.

Results. Since the results obtained for the two data sets G-50 and G-35 were highly similar, averages were taken over the two sets. The results are shown in Fig. 5.1. The plots depict the relationship of the averaged values of the percentage of correctly aligned residue–residue or residue–gap pairs for each assigned reliability index, subdivided into the 3 different similarity classes (25-30%, 30-40% and 40-50% PSI). The data is based on an analysis of a total of 5382 `3D_ali` reference alignments (Tab. 5.1).

Standard deviations. Since the number of analyzed pairs receiving reliability indices < 5 was found to be much smaller than the number of pairs with higher index values, the standard deviations of the percentages of correct alignment can accordingly be expected to be definitely larger in the regions of lower reliability. Therefore, we should not attribute too great relevance to the precise percentage values given for low reliability indices. On the other

similarity class	number of analyzed alignments data set G-35	data set G-50
40-50% PSI	1435	751
30-40% PSI	1273	696
25-30% PSI	617	610
total	3325	2057

Table 5.1: Numbers of analyzed `3D_ali` reference alignments from the two data sets G-35 and G-50, sorted by similarity class.

hand, the amount of data analyzed makes the results for the range of higher reliability indices very trustworthy.

5.1.1 Similarity class 40–50%

Justification. Let us first turn to the results for the class of highest similarity (40–50%). From the plot we note, disregarding reliability index 1 (whose behavior will be explained in Sec. 5.1.3) for a moment, that over the full range of considered reliability indices the percentage of correctly aligned pairs grows with increasing reliability index. Since we have evaluated a large set of alignments, we can interpret this percentage as a probability for a correct alignment of the respective pair. This establishes the desired justification of our reliability index as a measure for the likelihood of a correct alignment as pairs with higher reliability indices are more likely to be correctly aligned than pairs that have a lower index value assigned.

Linear correspondence. But moreover, we observe an approximately linear growth of the percentage of correctly aligned pairs with the value of the reliability index, such that we can propose the direct relationship "reliability index times 10 $\approx$ probability for correct alignment in percent"! Therefore, our reliability index can *directly* be translated into a likelihood for correct alignment, thereby bypassing the need for further calibration with external data!

Comparison to the work by Mevissen and Vingron. In this respect, our method performs strinkingly better than the algorithm described in the work by Mevissen and Vingron [16]. Referring back to Fig. 3.2 showing the results of the latter method (also based on an analysis of `3D_ali` reference alignments belonging to the 40–50% similarity class), it can readily be seen that all reliability indices between 5 and 10 represent residue pairs with a likelihood to be correctly aligned nearly equal to 100%, as we have already pointed out earlier. Thus, the only conclusion that could be drawn from such

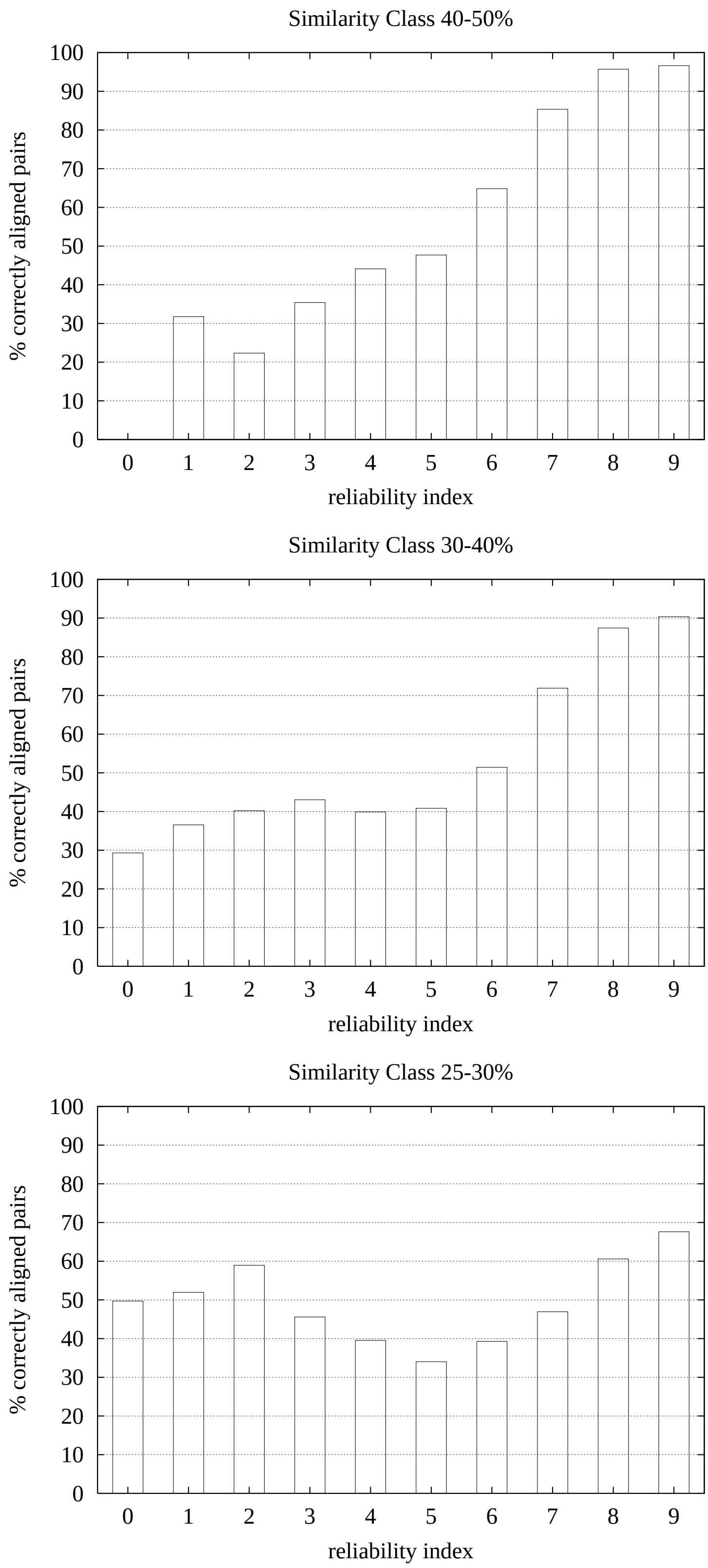

Figure 5.1: Percentage of correctly aligned pairs versus assigned reliability index for the three different similarity classes, averaged over the two data sets G-35 and G-50.

a reliability measure is that pairs which receive reliability indices equal or larger than 5 or so can be regarded as correctly aligned.

Hence, despite the existence of reliability indices ranging from 1 to 10, the corresponding likelihoods for a correct alignment are essentially identical for the majority of possible values, and no distinct interpretation can be attributed to two pairs that receive reliability indices of, say, 6 and 10, respectively. Even the lowest possible reliability index of 1 corresponds still to more than 50% probability that the corresponding residue pair is correctly aligned. This lack of distinction among the different values of the reliability index and the emerging inability to directly interpret the value of the reliability index as a likelihood for the correct alignment requires this method to be calibrated with statistical data obtained from an analysis of `3D_ali` reference alignments. Only this second additional step allows finally a translation of the algorithmically obtained reliability index into a meaningful measure whose absolute value corresponds to a likelihood. As shown above, our method does not exhibit this problem; the reliability index computed by the mean field algorithm can be given a direct probabilitistic interpretation.

Summary. We can conclude that in the class of sequences with 40–50% PSI, our algorithm exhibits outstanding performance. In particular, the direct correspondence between the value of the reliability index and the likelihood for a correct alignment, without the demand of further calibration, with fully distinct reliability indices over almost the entire range of possible values has, as to our knowledge, not been achieved by any other earlier method.

5.1.2 Classes of lower similarity

Class 30–40%. For the class of medium similarity (30–40%), we again observe an increasing likelihood for a correct alignment with growing values of the reliability indices. Only for the reliability indices 4 and 5 this trend is not completely fulfilled, but taking the standard deviations of the corresponding percentage values into account, we can safely claim that our reliability index is justified as a meaningful quantity for local reliability over the full range 0–9 of possible values. The approximate "1:10 correspondence" between the value of the reliability index and the percentage of correctly aligned pairs is successfully retained for reliability indices ≥ 5. Towards lower indices, the decrease proceeds more slowly and the percentage values remain in the 30–40% region.

Class 25–30%. For the class of lowest similarity, the method does not perform as well as for the two other classes. The desired growth of the percentage of correctly aligned pairs with increasing reliability index is observed for reliability indices ≥ 5, which justifies our reliability measure in this region. Here, the growth proceeds nearly linear, as wanted, although the corresponding percentage values are overally decreased such that the addition of an offset would be required to retrieve the 1:10 correspondence found for the other classes. For lower reliability indices the percentage of correctly aligned pairs exhibits both increases and decreases as the reliability index decreases from 5 to 0, which allows no meaningful interpretation of the reliability index in this range.

Comparison to the work by Mevissen and Vingron. As for the similarity class 40–50% discussed above, the method of Mevissen and Vingron [16] is again found to be not capable of establishing a direct relationship between the values of the reliability indices and the percentages of correctly aligned residues. In the 30–40% class, residue pairs with reliability indices between 5 and 10 have all probabilities close to one to be correctly aligned, in contrary to the virtually linear direct correspondence in this region of reliability indices that is observed in our results. For the 25–30% class, pairs with reliability indices 7–10 are correctly aligned in virtually all cases, whereas our method accomplishes again a linear correspondence for reliability indices between 5 and 10. Altogether, this demonstrates the better performance of our algorithm within the discussed range of reliability indices between 5 and 10 for the 30–40% and 25–30% similarity classes.

Towards lower reliability indices, the method of Mevissen and Vingron exhibits for both classes an only slow decrease in the percentages of correctly aligned residues, with the lowest possible reliability index corresponding to still 45–50% correctly aligned residues. This slow decrease is similarily found in our method, but with advantageously lower absolute percentage values at least in case of the class of 30–40% similarity.

Summary. The results establish the validity of our reliability index as a meaningful indicator of the trustworthiness of an aligned pair fully for the class of 30–40% and partly for the class of 25–30% similarity. For reliability indices ≥ 5, the linear correspondence between reliability index and probability for a correct alignments is successfully retained, which is clearly superior to the method by Mevissen and Vingron.

5.1.3 Influences on the performance

We have observed that the performance of our algorithm decreases 1) as the similarity between the sequences gets lower, and 2) as the reliability index decreases. In the following, we shall give possible explanations for this behavior.

Dependence of the performance on similarity. Issue 1) goes back to the general dependence of the performance of algorithms for sequence alignment on the similarity of the sequences. It is well known that these algorithms yield reasonable results only for pairs of sequences with similarity above the "twilight zone" of approximately 20–35% similarity [22]. Because our reliability measure is directly based on a reformulated sequence alignment algorithm, namely the Needleman–Wunsch algorithm, our reliability index also automatically inherits the weakness of the alignment algorithm itself regarding sequence pairs of low similarity. However, since the application of the sequence algorithm should anyhow be restricted to sequences of sufficient similarity, we should not attach too great importance to the somewhat reduced performance of our reliability algorithm in the low-similarity region. For sequences of established similarity ($\gtrsim$ 30%) for which the alignment by means of an algorithm is a biologically meaningful task, the reliability measure exhibits convincing validity.

Dependence of the performance on the reliability index. Observation 2), the constant or even increasing percentages of correctly aligned residues when going towards low reliability indices, might partly be explained in the same way as the first issue. Low reliability indices correspond to pairs whose alignment imposes a difficult task on the alignment algorithm, and the algorithm is therefore more likely to fail; as the reliability measure is in turn based on the alignment algorithm, one can expect weaker validity of the reliability index as its value decreases.

A second influence stems presumably from the method we have compared the alignments produced by the `MAXalign` program with the structurally correct reference alignments from the `3D_ali` database. Evaluating whether a particular residue–residue pair in the computed alignment is correctly aligned is an unambiguous procedure, since the letters can be uniquely indexed from their position in the sequence string. Gaps, however, are inserted in the alignment process, and therefore no particular order can be assigned to them. For example, suppose we would like to compare the two alignments

$$\begin{array}{cc|cc|cc} \ldots & \mathrm{A} & \mathrm{B} & - & \mathrm{C} & \ldots \\ \ldots & \mathrm{X} & - & \mathrm{Y} & \mathrm{Z} & \ldots \end{array} \quad \text{vs.} \quad \begin{array}{cc|cc|cc} \ldots & \mathrm{A} & - & \mathrm{B} & \mathrm{C} & \ldots \\ \ldots & \mathrm{X} & \mathrm{Y} & - & \mathrm{Z} & \ldots \end{array} \tag{5.1}$$

which are apparently different from both an evolutionary and a structural point of view. However, from a comparison of each individual aligned pair among the two alignments, we would infer that every pair is identically aligned in both alignments; assuming that, for example, the first alignment is the computed alignment, the second the standard of truth, such a pair-by-pair comparison would imply that every pair in the computed alignment is correctly aligned with respect to the reference.

A procedure that is capable of distinguishing the two alignments above would require not only validating that individual pairs are identically aligned, but also taking the alignment of adjacent pairs into account. Since an algorithmical implementation of this procedure would bring about significant complications, we have based our method on the straightforward pair-by-pair comparison, which implies however that a certain fraction of gap–residue pairs has probably been considered erroneously as being correctly aligned. This had surely no noticeable influence on our results for medium and high reliability indices (5–9) as here the number of corresponding residue–residue pairs exceeds by far the number of residue–gap pairs in the analyzed alignments. In contrary, very low reliability indices (0, 1) have virtually exclusively and low reliability indices (2–4) have dominantly been observed to be assigned to gapped regions. This implies that the obtained percentages of correct alignment for low reliability indices are presumably somewhat too high. In agreement with the observed behavior (Fig. 5.1), this influence of improperly evaluated gapped regions and therefore erroneously increased percentage values gets more pronounced as the similarity between the sequences decreases, because the alignment of sequences of lower similarity requires more often the insertion of gaps in general and of gapped regions of the type (5.1) bearing the danger of an ambiguous evaluation in particular.

5.2 Computational Effort

Now we shall investigate the computational effort required to calculate the optimal alignment with reliability indices assigned to every residue–residue or residue–gap pair. All given values were obtained from runs of the `MAXalign` program on a 800 MHz Pentium III with the same parameters (cf. Tab. 4.1) as used for obtaining the results presented in the preceding section. Sequences of pairwise identical length randomly drawn from the `3D_ali` database were used as data source.

5.2.1 Scaling of the computational effort

Calculating the reliability index for each pair corresponds essentially amounts to a multiple execution of the mean field Needleman–Wunsch algorithm. Therefore, the computational effort is expected to grow as the product of the sequence lengths times the number of annealing steps.

Number of annealing steps. The annealing process is terminated when the spin components corresponding to the optimal path have, in average, decreased to a value prescribed by the saturation cut-off Σ_0. The absolute value of the number of annealing steps required to reach this point is dominantly determined by the starting temperature ε and the annealing parameter α. Both parameters should therefore be chosen as large as possible with respect to the resulting performance of the reliability measure.

Assuming that a particular choice of parameters has been made, the only variable quantity remaining is the sequence length. Shorter sequences, in gen-

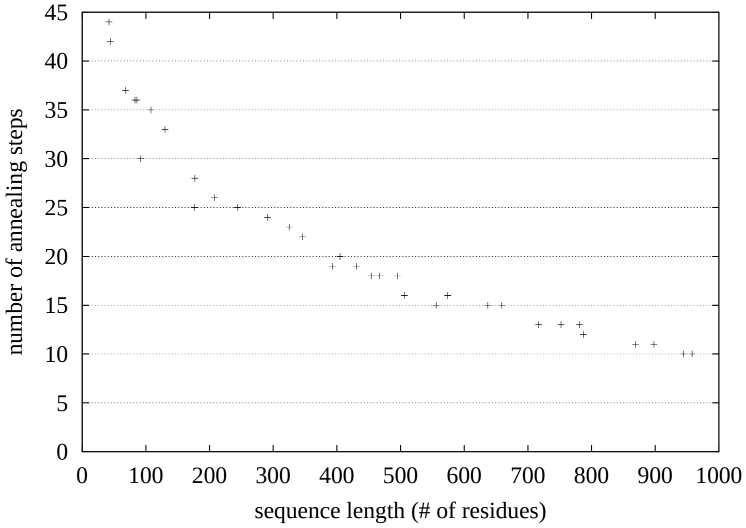

Figure 5.2: The number of required annealing steps as a function of sequence length.

eral, are easier to align, and hence we expect the spin components to attain the saturation cut-off value *later* (that is, at higher temperature T) with respect to the process of simulated smelting. This readily becomes clear by remembering that executing the mean field algorithm along a process of simulated annealing, *i.e.* by lowering T, will yield the optimal alignment; the easier the alignment problem, the *sooner* the spin components will converge towards their final value of one. In simulated smelting, however, the direction of changing T and therefore the orientation of the time axis is reversed.

The relationship between the number of annealing steps versus the sequence length is depicted in Fig. 5.2. We see that, in agreement to our expectation, the number of steps decreases as the sequence length increases, that is, as the alignment problem becomes more difficult. Some deviations in the resulting data points from a smoothly decreasing trend curve are observed; this is understandable from the fact that the difficulty of the alignment problem depends of course not only on the length of the sequences, but also on the particular composition of residues in the sequence, such that two sequence pairs of identical length can naturally not be expected to require the precisely same number of annealing steps.

Resulting scaling with sequence length. The resulting dependency of the computational times on the sequence length over the range of sequence lengths typically encountered in applications of sequence alignment algorithms is shown in Fig. 5.3. The result must be seen as a superposition of the two opposite scaling trends, namely the decreasing number of required annealing steps on one hand and the increased number of spins to be updated on the

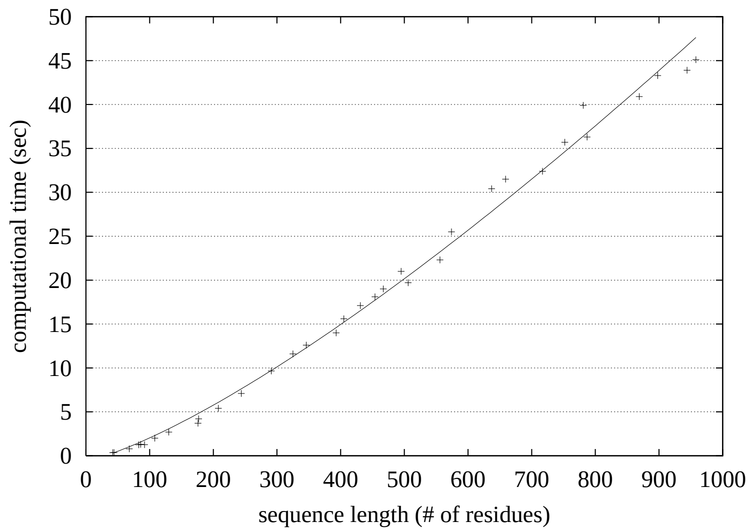

Figure 5.3: The required computational times for the calculation of the optimal alignment with assigned reliability indices as a function of the product of sequence lengths, with a fitting curve representing a N^γ power law where $\gamma \approx 1.3$.

other hand as the sequence length increases. The delineated relationship can be approximated by a power law N^γ where $\gamma \approx 1.3$. Without the influence of the number of annealing steps on the computational time, we would have expected $\gamma \approx 2$ since the execution time of the sequence alignment algorithm is of $\mathcal{O}(N^2)$.

Comparison to the work by Mevissen and Vingron. In contrary, the computation of the reliability index described in [16] scales as $\mathcal{O}(N^2)$, which is significantly worse compared to our algorithm, at least in the investigated typical range up to sequence lengths of 1,000 residues. Moreover, the need for a calibration of the reliability index requires the calculation and evaluation of a set of alignments of sequences from the `3D_ali` database which leads to additional computational effort.

5.2.2 Absolute computational effort

Actual times required to compute the optimal alignment with assigned reliability indices can be read off from Fig. 5.3. We see that our algorithm is very fast! For two sequences with lengths of around 1,000 residues, the entire computation is typically performed in less than one minute on a 800 MHz Pentium III.

Chapter 6

Summary and Outlook

In this section, we shall present a brief summary of the work presented in this thesis. Finally, we will give an outlook into possible future improvements.

Technique. We have introduced a novel approach towards quantifying the reliability of individual pairs in algorithmically obtained sequence alignments. The problem of pairwise global sequence alignment has been recasted in terms of an optimization problem mapped onto a spin system with a suitably chosen energy function which leads to an equivalent formulation of the classical Needleman–Wunsch algorithm for sequence alignment. Then, the binary "winner-takes-all" spins have then been replaced by continuous "winner-takes-most" spins in a mean field approximation.

This allowed the application of the mean field annealing technique as an alternative, approximative way to determine the optimal alignment, but with the important advantage of being able to explicitely study the behavior of the system during the optimization procedure. By monitoring these mean field dynamics, we inferred and quantified the presence of local suboptimal solutions for each pair in the optimal alignment. The resulting quantity assigned to each aligned pair was denoted as reliability index in anticipitation of serving as a measure of the local trustworthiness of the pair in question. That is, an aligned pair that receives a higher reliability index than another pair was expected to be in average more likely correctly aligned.

Evaluation. To verify our anticipitation, the algorithm outlined above was implemented in a computer program that allowed an automatized evaluation of a large set of structurally correct reference alignments from the `3D_ali` database. Thereby, we have been able to demonstrate that our reliability index is not only a meaningful indicator of the local trustworthiness of the alignment, but moreover can also be translated directly into an absolute probability for the correct alignment of the aligned pair over a wide range of reliability values, given a sequence pair of established similarity ($\gtrsim$ 30% PSI). This was found to be superior to the method suggested by Mevissen and Vingron [16], where only the inclusion of external reference data was able to yield an absolute measure.

The decreased performance of our algorithm for sequences of low similarity might be attributed to the underlying sequence alignment algorithm, but is nevertheless of only minor relevance since obtaining a meaningful alignment by means of an algorithm requires anyhow sequences of sufficient similarity. The partly undesireably high percentages of correctly aligned residues in the region of low reliability indices are presumably mainly due to our method of alignment comparison in the evaluation procedure and should therefore not necessarily be related to a weakness of the reliability algorithm itself.

Our reliability measure exhibits, with respect to [16], the clear advantage of being capable to be assigned not only to residue–residue pairs but also to gapped regions. The computational effort required to calculate the optimal alignment together with reliability indices assigned to every pair was over the investigated range of typical sequence lengths ($N = 10 \ldots 1,000$) found to be of only $\mathcal{O}(N^{1.3})$, which is significantly better than the $\mathcal{O}(N^2)$ algorithm of [16]. Finally, it should be stressed that our reliability measure evolves naturally from the dynamics of the sequence alignment task itself, instead of requiring the introduction of a separate algorithm distinct from the actual alignment procedure.

Outlook. Despite the very promising results of our algorithm, there are certainly a few issues that may require further investigation or improvements:

- First of all, the method for alignment comparison in the evaluation procedure should be extended such that gapped regions in the optimal alignment are only identified as correctly aligned when they precisely match the order provided by the reference alignment. This can be expected to significantly improve the results in the region of low reliability indices.

- In the evaluated alignments, we have noticed a strong dominance in the number of pairs with higher reliability indices. If desired, a broader and more even distribution over the range of possible values could presumably be obtained from a modification of the particular way the evolution of the spins is quantified. Possible changes applied to Eq. (4.26) may involve modifications of the integration parameters, a redefinition of the normalizing area or a functional transformation of the integration variable T in the integrand.

- The algorithmical speed may further be improved by increasing the parameters ε and α that dominantly govern the number required annealing steps. However, these modifications must be implemented with care in order to preserve the quality of the obtained reliability measure.

- The decreased performance of the algorithm for sequences of very low similarity (25–30% PSI) does not cry for remedy since application of an alignment algorithm should generally be restricted to sequences of established similarity. Furthermore, the degree of influence from incorrectly evaluated gapped regions on the observed weaknesses is unknown. If we have, however, shown that the poorer performance is in fact dominantly due to the method of computing the reliability measure, we may pursue improvements of our algorithm, although it might be argued whether a remedy by means of straightforward algorithmical modifications can be available as our reliability measures arises from the sequence alignment procedure itself which intrinsically exhibits weakness at low similiarities. Regardless, it will be surely worth trying to accomplish this goal, particularly in view of the sweeping success of our method up to the current point.

Acknowledgements

Firstly, I would like to express gratitude to my advisors Carsten Peterson and Mattias Ohlsson, who both took the best possible care of me. I was given the opportunity to gain a countless number of valuable insights that truly enhanced my perspective.

I also would like to thank all the other people of the Theoretical Physics group for their excellent company during this year. I have experienced a very personal atmosphere that contributed a lot to my sense of well-being here.

References

[1] Barton, G. J., and Sternberg, M. J. E. (1987). Evaluation and improvements in the automatic alignment of protein sequences. *Prot. Eng.* **1**, 89–94.

[2] Bellman R. (1958). On a routing problem. *Quarterly of Appl. Math.* **16**, 87–90.

[3] Byers, Th. H., and Waterman, M. S. (1984). Determining all optimal and near–optimal solutions when solving shortest path problems by dynamic programming. *Operations Research* **32**, 1381–1384.

[4] Chao, K.–M., Hardison, R. C., Miller, W. (1993). Locating well–conserved regions within pairwise sequence alignment. *Comp. Appl. Biosci.* **4**, 387–396.

[5] Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). A model of evolutionary change in proteins: Matrices for detecting distant relationships. In *Atlas of protein sequence and structure.* Dayhoff, M. O., ed., **5**, 345–358. National biomedical research foundation, Washington DC.

[6] Dayhoff M. O., Barker W. C., and Hunt L. T. (1983). Establishing homologies in protein sequences. *Meth. Enzymol.* **91**, 524–545.

[7] Ford, L. R., and Fulkerson, D. R. (1962). Flows in networks. Princeton University Press, 1962.

[8] Gonnet, G. H., Cohen, M. A., and Brenner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science* **256**, 1443–1445.

[9] Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.* **162**, 705–708.

[10] Häkkinen, J., Lagerholm, M., Peterson, C., and Söderberg, B. (1998). A Potts neuron approach to communication routing. *Neural Comput.* **10**, 1587–1599.

[11] Häkkinen, J., Lagerholm, M., Peterson C., and Söderberg B. (2000). Local routing algorithms based on Potts neural networks. *IEEE Transact. on Neural Networks* **11**, 970–977.

[12] Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.* **8**, 275–82.

[13] Kirkpatrick S., Gelatt C. D., and Vecchi M. P. (1983). Optimization by simulated annealing. *Science* **220**, 671–680.

[14] Kschischo M., and Lässing M. (2000). Finite-temperature sequence alignment. *Pacific Symposium on Biocomputing* **5**, 621–632.

[15] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller A. H., and Teller E. (1953). Introduction of the metropolis algorithm for molecular–dynamics simulation. *J. Chem. Phys.* **21**, 1087.

[16] Mevissen, H. Th., and Vingron, M. (1996). Quantifying the local reliability of a sequence alignment. *Prot. Eng.* **9**, 127–132.

[17] Miyazawa, S. (1995). A reliable sequence alignment method based on probabilities of residue correspondences. *Prot. Eng.* **8**, 999–1009.

[18] Naor D., and Brutlag, D. L. (1994). On near-optimal alignments of biological sequences. *J. Comp. Biol.* **1**, 349–366.

[19] Needleman, S. B., and Wunsch, Ch. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443–453.

[20] Pascarella S., and Argos P. (1992). A data bank merging related protein structures and sequences. *Prot. Eng.* **5**, 121–137.

[21] Peterson C., and Söderberg, B. (1998). Neural Optimization. To appear in *The Handbook of Brain Research and Neural Networks*, (2nd edition), Arbib, M. A. (ed.), Bradford Books/The MIT Press.

[22] Rost, B. (1999). Twilight zone of protein sequence alignments. *Prot. Eng.* **12**, 85–94.

[23] Saqi, M. A. S., and Sternberg, M. J. E. (1991). A simple method to generate non-trivial alternate alignments of protein sequences. *J. Mol. Biol.* **219**, 727–732.

[24] Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM J. Appl. Math.* **26**, 787–793.

[25] Smith, T. F., Waterman, M. S., and Fitch, W. M (1981). Comparative biosequence metrics. *J. Mol. Evol.* **18**, 38–46.

[26] Smith, T. F., and Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195–197.

[27] Smith, T. F., and Waterman, M. S. (1981). Comparison of biosequences. *Adv. Appl. Math.* **2**, 482–489.

[28] Taylor, W. R. (1986). Classification of amino acid conservation. *J. Theor. Biol.* **119**, 205–218.

[29] Vingron, M., and Argos, P. (1990). Determination of reliable regions in protein sequence alignments. *Prot. Eng.* **3**, 565–569.

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

[31] Vingron, M., and Waterman, M. S. (1995). Statistical significance of local alignments with gaps. In: *Bioinformatics: From nucleic acids and proteins to cell metabolism.* Schomburg D., and Lessel, U. (eds.), Weinheim.

[32] Vingron, M. (1996). Near-optimal sequence alignment. *Curr. Opin. Struct. Biol.* **6**, 346–352.

[33] Waterman, M. S., Smith, T. F., and Beyer, W. A. (1976). Some biological sequence metrics. *Advances in Mathematics*, **20**, 367–387.

[34] Waterman, M.S. (l983). Sequence alignments in the neighborhood of the optimum with general application to dynamic programming. *Proc. Natl. Acad. Sci. USA* **80**, 3123–3124.

[35] Waterman, M. S. (1995). *Introduction to Computational Biology: Maps, sequences and genomes,* (1st edition), Chapman & Hall/CRC Press.

[36] Zuker M. (1991). Suboptimal sequence alignment in molecular biology: Alignment with error analysis. *J. Mol. Biol.* **221**, 403–420.