FRAGMENT-BASED METHODS FOR STRUCTURE DETERMINATION WITH SPARSE DATA

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

IVAN KALEV

aus Kotel, Bulgarien

Tübingen, 2013

Ivan Kalev: *Fragment-based methods for structure determination with sparse data,* Dissertation © 2013

TAG DER MÜNDLICHEN QUALIFIKATION:
12.02.2014

DEKAN:
Prof. Dr. Wolfgang Rosenstiel

BERICHTERSTATTER:
1. Prof. Dr. Oliver Kohlbacher
2. Dr. Michael Habeck

Fragment-based methods for structure determination with sparse data

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Ivan Kalev

aus Kotel, Bulgarien

Tübingen
2013

Tag der mündlichen Qualifikation:	12.02.2014
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Oliver Kohlbacher
2. Berichterstatter:	Dr. Michael Habeck

ABSTRACT

The fragment assembly approach was initially conceived as an *ab initio* strategy for protein structure prediction; however, the use of fragment libraries is becoming an increasingly popular choice in protein structure determination as well. While most method development efforts have been rightfully directed towards optimization of the fragment assembly algorithms, we show that their performance is also strongly dependent on the quality of the underlying fragment libraries. Here we present an integrated, general-purpose framework for construction of dynamic fragment libraries and local structure prediction from sequence. Our method, HHfrag, uses sensitive local alignment of sequence profiles to detect recurrent protein motifs of local conservation and build fragment libraries of very high precision and truly dynamic fragment lengths, solving important limitations of earlier methods. We demonstrate that this approach improves the performance of traditional *ab initio* fragment assembly and introduce algorithms for fragment clustering, filtering and blind prediction of local motif conservation. The resulting confidence-guided framework for local structure prediction is a solid foundation for context-sensitive torsion angle prediction with higher accuracy. Finally, we discuss a new algorithm for detection of analogous fragments based on chemical shift similarity, optimal ways of mixing homologous with analogous fragments and their combined practical use in fully automated NMR structure determination from sparse data.

ZUSAMMENFASSUNG

Das Zusammenfügen von Peptid-Fragmenten wurde ursprünglich als Strategie für die *ab initio* Vorhersage von Proteinstrukturen erdacht. Die Nutzung von Katalogen solcher Peptid-Fragmente hat sich jedoch mittlerweile zu einer immer populäreren Möglichkeit für die Bestimmung von Proteinstrukturen entwickelt. Während die meisten methodischen Entwicklungen bloß darauf abzielen, die Algorithmen für das Zusammenfügen der Peptid-Fragmente zu optimieren, zeigen wir, dass die Leistungsfähigkeit all dieser Methoden maßgeblich von der Qualität der zugrundeliegenden Kataloge von Peptid-Fragmenten abhängt. Wir präsentieren eine integrative, breit einsetzbare Software-Architektur für die Konstruktion dynamischer Kataloge von Peptid-Fragmenten und die lokale Vorhersage von Proteinstrukturen ausgehend von der Aminosäure-Sequenz. Unsere Methode, die wir HHfrag nennen, benutzt empfindliche lokale Alignments

von Sequenz-Profilen, um wiederkehrende, homologe und lokal konservierte Motive innerhalb von Proteinen zu ermitteln, um damit schließlich Kataloge von Peptid-Fragmenten zu erstellen, die maßgeschneidert und dynamisch hinsichtlich ihrer Länge sind. Dadurch werden wichtige Limitierungen vergleichbarer Methoden überwunden. Weiterhin demonstrieren wir, dass unser Ansatz des Zusammenfügens von Peptid-Fragmenten in einer verbesserten Leistungsfähigkeit bei der *ab initio* Vorhersage von Proteinstrukturen resultiert, und stellen Algorithmen für das Clustern und Filtern von lokal konservierten Motiven in Proteinen vor. Die dadurch entstandene, konfidenz-geleitete Software-Architektur für die lokale Vorhersage von Proteinstrukturen erlaubt auch die kontext-empfindliche Vorhersage von Torsionswinkeln mit hoher Genauigkeit. Zuletzt behandeln wir einen von uns entwickelten Algorithmus für die Detektion von analogen Peptid-Fragmenten basierend auf der Ähnlichkeit von chemischen Verschiebungen aus NMR-Experimenten, optimale Wege für die Kombination homologer und analoger Peptid-Fragmente und deren gemeinsame praktische Anwendung in vollautomatisierter Strukturbestimmung durch NMR-Spektroskopie ausgehend von einer spärlichen Datenlage.

A programmer who subconsciously views himself as an artist
will enjoy what he does and will do it better.

— Donald E. Knuth

ACKNOWLEDGMENTS

I express my gratitude to Dr. Michael Habeck for his expertise, support and profound contributions to this research. I would also like to thank Prof. Oliver Kohlbacher for supervising this project along with Prof. Andrei Lupas for their constructive feedback and ideas.

Thanks to all contributors to the CSB open-source software and Dr. Moritz Ammelburg who helped me with the German translation.

This work has been funded by Deutsche Forschungsgemeinschaft (DFG) grant HA 5918/1-1, by contract research "Methoden in den Lebenswissenschaften" of the Baden-Württemberg Stiftung and by the Max Planck Society. Thank you for supporting our research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

HMMs	Pofile hidden Markov models
PSSM	Position-specific scoring matrix
PDB	Protein Data Bank
RMSD	Root-mean-square deviation
NMR	Nuclear magnetic resonance
NOE	Nuclear Overhauser effect
ppm	parts per million
TBM	Template-based modeling
FM	Free modeling
nrVASCO	Non-redundant VASCO
CSB	Computational structural biology toolbox
PDF	Probability density function

1 INTRODUCTION

The proteins observed in living organisms are complex and versatile molecules. The cellular function of each protein is directly determined by the way it folds in 3D space. Obtaining information about the structure of a given protein is therefore central to understanding its function. Protein structures can be determined using a number of experimental approaches, such as X-ray crystallography or NMR spectroscopy, but these methods require a significant amount of time and resources. Detailed understanding of the process of protein folding is a crucially important goal of modern science, because it would make protein structure determination more accessible, much faster and cheaper.

When Kendrew et al. determined the first protein structure in 1958, they were rather disappointed to encounter an overwhelming complexity: "*Perhaps the most remarkable features of the molecule are its complexity and its lack of symmetry. The arrangement seems to be almost totally lacking in the kind of regularities which one instinctively anticipates, and it is more complicated than has been predicated by any theory of protein structure.*". The complicated nature of protein structures proved to be really challenging, because more than 50 years later science is still as puzzled about the process of protein folding as Kendrew and co-workers were. But one thing we know for sure nowadays — regularities and common building patterns in protein structures *do exist*. Many evidences have been collected for the existence of recurrent motifs, shared among evolutionary unrelated proteins. The ability to detect such building blocks opened new possibilities for protein structure prediction from sequence and structure determination using the novel fragment assembly approach.

1.1 MOTIVATION

The success of the fragment assembly methods for protein structure prediction has prompted major research efforts for optimization and improvement of the fragment sampling algorithms and their related force fields and scoring functions. The topic of fragment optimality, however, has received significantly less attention during the past decade. Despite the importance of fragment quality for successful and efficient *ab initio* fragment assembly, existing fragment detection methods demonstrate relatively low precision, hurting the performance of folding algorithms. Little is known about the patterns of local motif conservation along the protein sequence and how the

ability to predict the exact locations of conserved recurrent fragments may influence the performance of local structure prediction. Existing methods for fragment detection conveniently define fragments as rigid sequence windows of fixed size. However, this definition is not supported by any biochemical evidences and neglects the highly polymorphic nature of structural motifs in proteins, which often vary in length and internal composition.

These and other open questions — such as the optimal way of combining dynamic fragment libraries with analogous fragments based on chemical shift similarity — motivated the development of our general-purpose framework for fragment detection, presented in the following chapters. We have addressed some important limitations of earlier methods and designed novel algorithms for dynamic fragment detection, putting a strong emphasis on accuracy, efficiency and minimalism. We describe the theoretical and technical background behind our framework, measure important aspects of its performance in standard benchmarks and provide practical examples of its applications for local structure prediction, *ab initio* tertiary structure prediction and NMR structure determination from sparse experimental data.

1.2 SYNOPSIS

This work is divided into multiple interrelated chapters, describing all different facets of our integrated fragment-detection approach.

We begin with a brief review of the existing literature (Chapter 2) and introduce the notion of recurrent structural motifs, detectable in sequence space by sensitive profile-comparison methods. Various concepts concerning the use of libraries of such motifs in protein structure prediction and NMR structure determination are explained in this background chapter.

Chapter 3 describes in detail the core foundation of our local structure prediction framework. We introduce HHfrag, an accurate method for detection of recurrent motifs from sequence. HHfrag improves on existing dynamic fragment libraries with significantly better precision and the ability to detect fragments of variable length or gapped nature. We show the utility of this method in Rosetta *ab initio* protein structure prediction experiments and demonstrate the improvement over the use of conventional fragment libraries.

This chapter concludes with the derivation of a novel algorithm for dynamic motif clustering and enrichment of fragment libraries, which is employed in HHfrag's fragment-filtering extension. We provide the groundwork for blind prediction and quantification of local motif conservation and demonstrate a practical application of our algorithm for prediction of torsion angles from sequence with high accuracy.

In Chapter 4 we propose a new method for detection of analogous fragments with compatible structure, based on chemical shift similarity. This method is useful for fragment detection in unconserved and low-accuracy regions, when basic experimental NMR data are available for the protein of interest. We introduce a flexible algorithm for incorporation of analogous fragments in standard HHfrag dynamic libraries, which increases their coverage in regions where sequence-based motif detection meets its physical limitations.

Chapter 5 brings practical applications of our analogous and homologous fragment-detection methods for the purpose of structure calculation from NMR data of arbitrary sparseness.

We conclude this work in Chapter 6 with a brief overview of the software architecture of the our framework, the availability of the HHfrag software and our contributions to the open-source community.

2 BACKGROUND

Deciphering the protein folding problem remains a fundamental challenge in structural bioinformatics and biochemistry to date. But although this process is still poorly understood, it is already known that protein sequences do not adopt unlimited varieties of global and local shapes. It has been determined that folding protein chains do not explore the complete conformational space exhaustively [68]. Rather, the local structure of each polypeptide segment is biased by the geometrical and chemical properties of its constituent amino acids [13]. This observation prompted the development of structural alphabets in an attempt to partition known protein structures into a dictionary of discrete motif prototypes [67]. It has been reported that such fragment libraries may be sufficient to describe all protein folds in terms of recurrent building blocks [28, 27].

The following sections serve as an introduction to these concepts and summarize the present state of knowledge about recurrent protein building blocks and the ways they can be detected or predicted.

2.1 PROTEIN STRUCTURE FUNDAMENTALS

2.1.1 *Protein sequence*

Proteins are large polymer molecules, built from amino acid monomer units. Individual amino acid residues are linked in a linear fashion to form unbranched polypeptide chains. The sequence of amino acids of a given protein, termed its *primary structure*, is genetically encoded and determines the folded state [9].

All amino acid residues share a common structural pattern (Figure 2.1): amino group (NH_2), carboxyl group ($COOH$) and a unique side chain (R) are attached to the central carbon atom (C_α). Polypeptide chains are formed by joining the terminal carboxyl group of the growing chain to the amino group of the next amino acid, thus forming a peptide bond. Each amino acid is uniquely identified by the chemical nature of its side chain group (R). Genetic code exists for 20 common (and 2 rare extra) amino acids. The amount of possible amino acid combinations in a single chain is therefore huge; e. g. for a small chain of size 100 residues there are at least 100^{20} different combinations. However, we can cluster all amino acids in a smaller number of groups based on their physicochemical properties, such as hydrophobicity, polarity and charge. Members of the same group are more likely to be observed as interchangeable in conserved protein

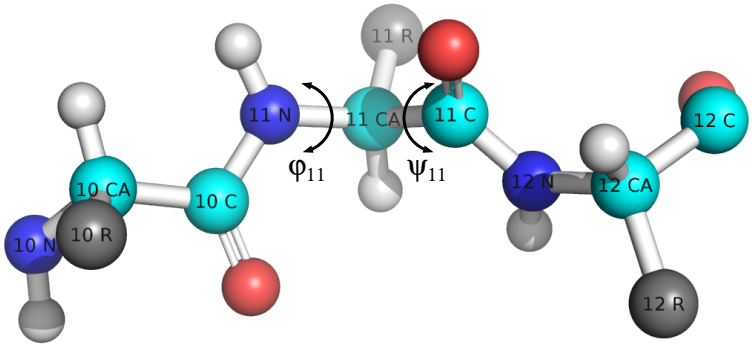


Figure 2.1: Primary structure. Shown is a small polypeptide segment (residues 10-12), extracted from PDB entry 2l09. A repeating pattern of NH–C_αH–C=O atoms forms the backbone of the chain. Amino acid side chains (R) have been condensed to simple spheres for clarity.

regions because they have similar properties and provide analogous chemical environment. Conversely, amino acid substitutions which involve residues from different categories are very likely to cause harmful conformational disruptions and loss of function.

Figure 2.1 demonstrates how protein chains can be represented as monotonic repetitions of a segment of 3 atoms:

$$\ldots - (N_i - {C^\alpha}_i - C_i) - (N_{i+1} - C^\alpha_{i+1} - C_{i+1}) - \ldots$$

When visualizing proteins, we often omit the side chains and use this representation (part of protein's *backbone*) to show the overall direction of the protein chain. The peptide bond, which links residues i and $i+1$, is partially double. This phenomenon restricts the possibilities for rotation around the peptide bond, leaving only two rotational degrees of freedom at each position: $N - C_\alpha$ (φ angle) and $C_\alpha - C$ (ψ angle) (Figure 2.1). Since the torsion angles φ and ψ are the only degrees of freedom, their values for every position in the protein chain are sufficient to describe the conformation of the backbone.

2.1.2 *Secondary structure*

During polypeptide synthesis, local interactions between neighboring amino acids lead to the formation of *secondary structure* elements. The most prominent forms of secondary structure observed in proteins are the *alpha helix* and the *beta strand* (Figure 2.2). By visualizing (φ, ψ) values from known proteins, Ramachandran et al. have shown that due to side chain clashes, not all (φ, ψ) combinations are physically possible. This restricts the Ramachandran plot [68] to a subset of sterically feasible regions, which correspond to well-known secondary structure elements such as the alpha helix and the beta strand.

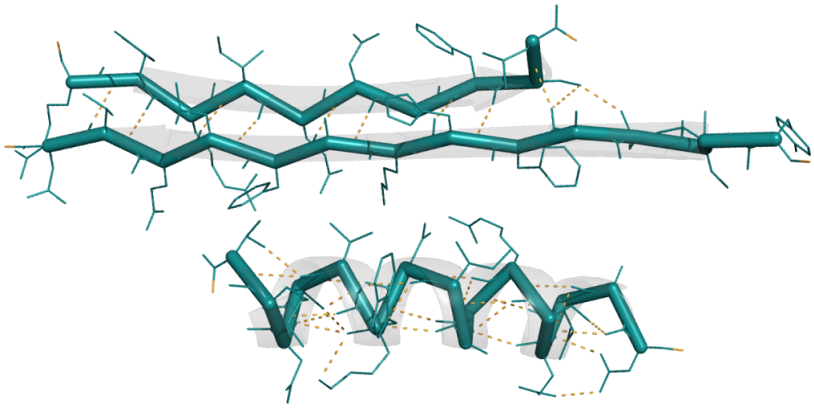


Figure 2.2: Secondary structure. Shown are two segments of secondary structure from PDB entry 1ath: antiparallel beta strands forming a sheet (top) and an alpha helix (bottom). The thick ribbon lines represent the backbone and side chains are shown as thin lines. These secondary structures are stabilized by hydrogen bonding, denoted by dashed lines.

The alpha helix is a more compact secondary structure element with periodicity of 3.6 residues per turn. It is stabilized by repetitive patterns of hydrogen bonds between backbone atoms. As intuition suggests, several amino acid residues are preferred in an alpha helical context, while other residue types may be undesirable for proper formation of an alpha helix [7, 63]. Beta strands are significantly more elongated forms of secondary structure (hence their name). They are frequently packed in parallel, antiparallel or mixed stacks of strands, called *beta sheets*. Beta sheets are stabilized by the formation of hydrogen bonding registers between the aligned strands (Figure 2.2).

Rigid regular secondary structure elements are usually connected by flexible loops of variable length. Loops tend to be exposed to the protein surface and often form binding sites for other molecules. Unlike regular secondary structure elements, loops exhibit higher sequence variability than the rest of the sequence. Note that such sequence and structural variability poses an immediate difficulty in modeling those regions of the protein molecule.

2.1.3 *Structural motifs*

Pairs of neighboring secondary structure elements, along with their flexible loop connectors, form higher level structural motifs called *supersecondary structure* elements. One such commonly found structural motif is the β *hairpin*, composed of two anti-parallel strands, connected by a very short *turn*. Some supersecondary structure elements demonstrate weak, but still detectable sequence conservation. This observation allowed the discovery of a number of ubiquitous supersecondary structural motifs, summarized in the I-Sites [10] frag-

ment library (2.4.1). Other elements [27] are entirely defined in terms if their local structure and may not have detectable sequence preferences (with the current state of sophistication of our sequence similarity detection techniques). Nevertheless, they still provide a further evidence that protein structures have modular nature and share a significant number of reusable design patterns. It has been proposed that some supersecondary structures may be part of a vocabulary of ancient peptides, from which present day folds have evolved [60].

2.1.4 *Domains*

Long-range interactions between multiple secondary structure elements and motifs lead to repeated bending of the polypeptide chain. This results in the formation of a compact, globular structure called *domain*. Domains are the manifestation of the third level of structural organization in proteins, termed *tertiary structure*. They are regarded as the main autonomously folding units in proteins. Polypeptide chains are not restricted to single domains however; a given chain may contain multiple domains. In that case each domain is charged with a clearly defined, specialized function.

2.2 HOMOLOGY

The advent of fast string matching algorithms allowed protein sequences from different organisms to be compared and analyzed for regions of similarity. This has lead to the discovery of families of proteins evolved from common ancestors. Such proteins, sharing a detectable degree of sequence similarity, are said to be *homologous*. A fundamental principle in the field of protein evolution states that homologous domains — or domains sharing a certain level of sequence identity in general — are very likely to have similar structures as well. This means that protein structures accumulate evolutionary changes more slowly than their corresponding sequences. Such claim makes intuitive sense given the observation that some amino acid substitutions are conservative, i. e. members of the same class of amino acids have a degree of equivalence and can be used "polymorphically" without causing functional damage.

When the evolutionary distance between two proteins is increased above a certain threshold, their corresponding sequences may have diverged so severely that sequence-based homology detection may no longer be possible. However, their 3D structures may still share visible similarity. In such cases the two proteins are *remote homologs*.

2.2.1 *Sequence alignment*

The goal of sequence-based homology detection is to compare a query sequence against a database of proteins, find the optimal alignment between each pair of sequences and report matches with significant degree of similarity. Such type of an algorithm is expected to maximize the amount of identical or similar (synonymous) residues, but also allow for gaps to be inserted in either sequence. Gaps in sequence alignments represent the mutational events of insertion and deletion.

Brute-force enumeration of all possible alignments between two sequences, in search for the optimal one, is computationally not feasible. This problem was solved with the discovery of quadratic-complexity dynamic programming algorithms for global [64] and local [83] alignment. The latter is the basis for the popular BLAST program [1], which allows very fast sequence database searches. All mentioned algorithms use matrices of log-odds scores and scalar gap penalties to compute the similarity between each pair of aligned residues. Several scoring matrices have been developed with BLOSUM62 [43] being one of the most popular choices.

Traditional sequence alignment methods successfully identify closely related homologous proteins, which share very similar or even identical structures. However, sequence-based homology detection becomes unreliable when the sequence identity between the proteins in comparison approaches the "twilight zone" ($< 25\%$) [74]. Remote homology detection is therefore not possible with conventional sequence alignment.

2.2.2 *Sequence profiles*

Since a single sequence represents only one member of a family of evolutionary related proteins, simple string matching of isolated sequences disregards the evolutionary history encoded in a protein family. This method is thus not sensitive enough to detect remotely homologous sequences, whose identity lies within the twilight zone.

In a successful attempt to extend the sensitivity of homology detection, sequence profiles have been introduced and subsequently implemented in programs such as PSI-BLAST [2]. The concept behind sequence profiles is rather intuitive, given the observation that amino acid preferences in proteins are position-(context) specific. For every position in the protein sequence of interest, its sequence profile contains a *frequency distribution* of amino acids, derived from multiple alignment of the protein with its evident homologs. The sequence profile is used to build a Position-specific scoring matrix (PSSM), which substitutes the original query sequence in iterative sequence database searches. The alignment of a PSSM against a sequence is technically not different from the algorithm for alignment of two sequences, with

the exception that all pairwise similarity scores are taken directly from the PSSM itself (rather than a generic scoring matrix).

As expected, this technique increases the homology detection capabilities of BLAST and makes remote homology detection possible.

2.2.3 *Hidden Markov models*

Pofile hidden Markov models (HMMs)[1] are an extension to the basic idea of sequence profiles. They improve the sensitivity of profile-based remote homology detection by incorporating information about position-specific insertions and deletions.

Note that if we extract all match states with their respective emission probabilities from an HMM, we will get a classic sequence profile (2.2.2).

A profile HMM is a probabilistic representation of a protein family, which captures the entire information contained in its underlying multiple sequence alignment. HMMs have layered structure (Figure 2.3). Each layer corresponds to a specific position in the original query sequence and contains a number of hidden states: match, insertion and deletion. Every hidden state is connected to a number of neighboring states via position-specific transition probabilities. The sum of all transitions leaving a given state is 1, thus forming a proper probability distribution. Deletion states are silent, while match and insertion states emit amino acids (observations) from their associated emission distributions. For insertion states, this is the standard distribution of background amino acid frequencies (i. e. the probability of observing amino acid A as a result of a random insertion event in the course of evolution). Match states however emit residues from a position-specific distribution, derived from the amino acid frequencies in the corresponding multiple alignment column. Figure 2.3 outlines the rules that govern transitions between different states and layers in a typical profile HMM.

The main purpose of building a profile HMM is the ability to search a database for sequences that are homologous or remotely homologous to the family represented by the HMM. The theoretical basis for the mechanism of this search is provided by the Viterbi algorithm [89], which can be used to compute the most probable alignment of a sequence with the HMM and its associated probability. Sequences that match the HMM better than a reference null model are regarded to be (remotely) homologous and reported as true positive hits. This approach is implemented in the popular HMMER homology detection package [29, 26].

2.2.4 *HMM comparison*

The final step in pushing the boundaries of remote homology detection was made possible with the introduction of the profile-profile

1 For brevity, the terms *HMM* and *profile HMM* will be used interchangeably throughout the text.

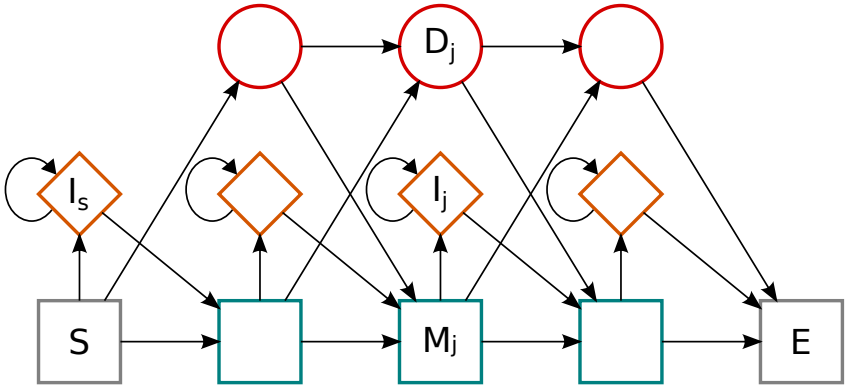


Figure 2.3: Layout of the profile HMMs in HHpred [84, 44]. M_j, I_j and D_j represent the match, insertion and deletion hidden states at a given layer j. S and E are the start/end states respectively and I_s is an insertion state attached to the start state. All possible transitions leaving a given state are designated by arrows.

comparison methods [75]. Pairwise comparison of sequence profiles remains the most sensitive remote homology detection strategy to date.

HHsearch [84] — the current *de facto* standard for homology detection — refines the idea of profile comparison further. More specifically, HHsearch performs local alignment between pairs of profile HMMs. For that purpose, both the query protein and the database must be represented in HMM format. The added computational time for building a database of HMMs is a good trade off, given the increased sensitivity and precision that this algorithm demonstrates over earlier remote-homology detection methods. Note that the ability to perform gapped local alignment between two profile HMMs is especially valuable, because it allows the detection of relatively short segments of remote homology (such as supersecondary structures and ancient peptides). The quality of the alignments produced by HHsearch is further improved with the incorporation of a secondary structure term into the scoring function. Each HMM in HHsearch is therefore built from two components:

1. Sequence profile, computed with multiple rounds of PSI-BLAST
2. Secondary structure: predicted with PSIPRED [49] (for the query sequence or non-PDB templates) or computed with DSSP [50] (for all PDB database templates [5])

For every pair of aligned profile layers, HHsearch computes the *column score*, which is proportional to the dot product of their emission distribution vectors:

$$S_{aa}(q_i, p_j) = \log \sum_{a=1}^{20} \frac{q_i(a)p_j(a)}{F(a)} \tag{2.1}$$

Here $q_i(a)$ and $p_j(a)$ are the emission probabilities for amino acid a in columns i and j. $F(a)$ is the background frequency for the same amino acid (the chance of observing such amino acid in natural proteins). The total profile alignment score is thus a sum of log-odds. To measure the significance of the obtained final score, HHsearch compares it to a distribution of scores for random alignments (not indicative of homology). This distribution appears to follow the Gumbel's extreme-value model. The *p-value*, reported for each alignment, is the probability of observing a random match with equal or better score; the e-value is the expected number of such random hits. Therefore, e-values significantly lower than 1 indicate alignments of homologous proteins.

When the secondary structure score is added to the total sum-of-odds score, the resulting distribution of random scores no longer fits the Gumbel model well. To account for this error, HHsearch provides an alternative to the *p-value* metric, called *HHsearch probability*:

$$P(S) = \frac{P_p(s = S)}{P_p(s = S) + P_n(s = S)} \tag{2.2}$$

where P_p and P_n are score distributions obtained for reference sets of positive (homologous) and negative (not homologous) alignments respectively. Since this approach incorporates the secondary structure score, it has better sensitivity and should be preferred over the use of e-values.

2.3 STRUCTURE PREDICTION STRATEGIES

The ability to predict the structure of a protein from its bare amino acid sequence is a fundamental, but still unresolved problem in structural bioinformatics. Little is known to date about the physical rules that govern protein folding. It is still not possible to simply "compute" the structure of a protein *de novo*.

Although a general physical theory on protein folding is yet to be discovered, several heuristic approaches have been developed to circumvent our lack of knowledge about folding proteins from first principles. Earlier in this chapter (2.2) we introduced the concept of homology and observed that:

1. homologous proteins have very similar structures
2. remotely homologous proteins, derived from the same evolutionary ancestor, are very likely do adopt similar folds
3. homology and traces of remote homology can be reliably detected with sequence and profile HMM-comparison methods (2.2.3)

These observations are the theoretical basis of all successful structure prediction efforts to date.

2.3.1 *Template-based modeling*

Homology modeling, also known as *comparative modeling*, is the most straightforward and robust structure prediction strategy. This method relies on sequence-based identification of homologous proteins from the Protein Data Bank (PDB) [5]. Since highly similar sequences imply identical structures, such homologous structures can be used as templates for direct modeling.

The PDB database is a central repository for experimentally determined protein structures. It already contains more than 80000 entries (and counting). Sequence databases however are growing at a much faster pace. Nevertheless, PDB has already reached a sufficient level of fold diversity — it is very likely that novel sequences in the public sequence databases would match existing experimental structures from PDB.

As we saw earlier, closely related homologs can be easily identified using conventional sequence alignment algorithms, such as BLAST [1] (2.2.1). The degree of sequence identity between the query sequence of unknown structure and its PDB template will determine whether homology modeling would succeed. Alignments with sequence identity of 40% or above will almost always produce very accurate homology models (i. e. RMSD of 2.0 Å or better) [74].

Note that with the advent of HMM-comparison homology detection methods, the traditional differentiation between homology modeling and threading is steadily getting less clear.

When the degree of sequence identity approaches the *twilight zone*, homology modeling becomes unreliable. At this level of sequence divergence we have reached the limits of conventional sequence alignment as a method for template selection. *Fold recognition*, also known as *threading*, is a natural extension to homology modeling, which makes use of multiple remotely homologous templates to achieve the exact same goal. An important improvement over standard homology modeling however is the possibility to combine multiple, not necessarily full-length templates. As discussed earlier (2.2.3, 2.2.4), remote homology detection is achieved through the use of sensitive profile-comparison methods. HHpred [44] is a classic threading program, which uses the HHsearch [84] remote homology detection algorithm for template selection. Highest ranking PDB homologs are simply used by HHpred as templates in standard homology modeling with Modeller [76].

2.3.2 *Fragment assembly*

Even if no remotely homologous templates can be detected for a given query sequence, it may still be possible to predict its 3D structure. *Ab initio* methods, namely the fragment-based *ab initio*, have been

recently developed. These algorithms do not use explicit, long templates, but rather rely on the fact that evolutionary unrelated protein structures often share common structural motifs (2.1.3). Some motifs can be detected using the same remote homology search methods used in threading. Based on this observation, a number of protein fragment libraries have recently emerged (2.4).

Note that ab initio is still different from de novo, i.e. these methods combine very short fragments ("templates") from existing structures as opposed to folding proteins from first principles.

The fragment-based *ab initio* strategy for structure prediction involves a few consecutive steps:

1. *Fragment selection*. Compatible structural fragments are picked from template protein structures using profile-profile alignment and secondary structure matching. This results in the preparation of a position-specific fragment library, tailored to the query sequence (2.4.2).

2. *Fragment sampling and assembly*. Starting from an elongated structure, fragments are randomly picked from the library and their torsion angles are inserted at the matching positions in the model. This procedure is repeated multiple times as part of a Monte Carlo conformational sampling protocol, guided by a combination of scoring functions [81, 72]. Since most scoring functions are very approximate and inferred by statistical methods [82], they do not necessarily represent the real energy landscape very accurately. The conformational search process is frequently trapped in local minima, which motivates the need to repeat the search multiple times with different starting conditions.

3. *Decoy selection and model optimization*. When a sufficiently large number of decoy structures have been generated, the entire pool of decoys is examined in order to select a final set of candidate models. The final candidate(s) may be a subject to optimization and high-resolution refinement using more detailed but computationally expensive force fields.

Rosetta *ab initio* [72, 55] is one of the most popular fragment assembly methods for protein structure prediction.

2.4 LOCAL STRUCTURE PREDICTION

Earlier in this chapter we saw how local hydrogen bonding leads to the formation of regions of regular secondary structure along the polypeptide chain (2.1.2). The DSSP [50] and STRIDE programs [32] are able to analyze the hydrogen bonding patterns in existing structures and compute their expected secondary structure, represented by an eight-state alphabet. However, proteins have traditionally been described as linear strings composed of segments adopting only two secondary structure conformations (helix (H) and strand (E)), connected by flexible coil or loop regions (C). Prediction of secondary structure

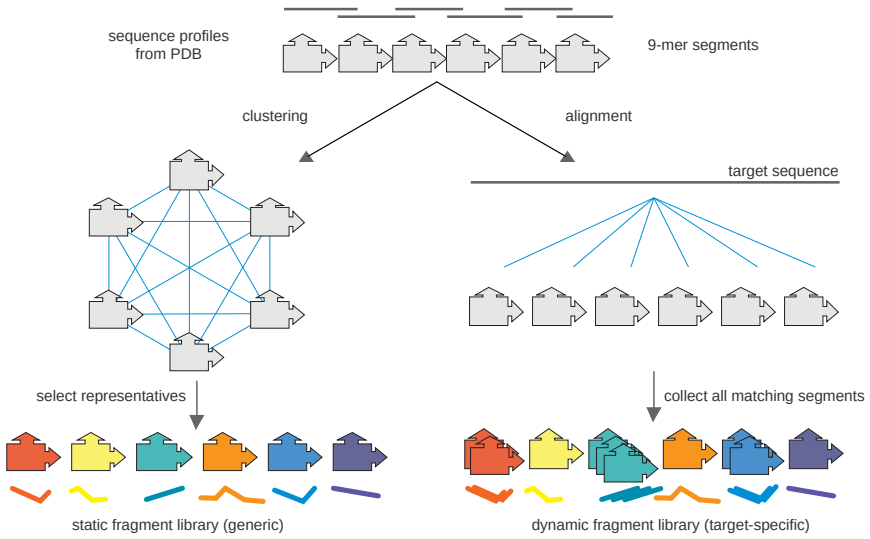


Figure 2.4: Fragment library concepts. The static approach (left) produces compact dictionaries of reusable motifs by clustering of fixed-length protein segments. The dynamic approach (right) compiles a comprehensive collection of target-specific fragments, which can be readily used in *ab initio* structure prediction.

from sequence is possible thanks to the development of methods like PSIPRED [49], which is able to perform three-state secondary structure prediction with high accuracy (70–80%).

However, the discovery of supersecondary structures (2.1.2) and the emerging hypothesis of a primordial fragment world [60] suggest that there is more information stored in the local structure of proteins than simple secondary structure elements. Protein structures are now viewed as combinations of reusable structural primitives, called *fragments*.

There are two main approaches to detecting fragments (Figure 2.4). The first approach focuses on the derivation of a compact, static structural alphabet. The second one aims at building comprehensive fragment libraries suitable for *ab initio* protein structure prediction.

2.4.1 *Static structural alphabets*

Protein chains from the PDB database can be discretized into compact alphabets of recurrent building blocks. Structural alphabets derived in this way are usually sufficient to describe virtually all known experimental structures (as simplified strings of motif identifiers). The discretization of the structural space is performed by partitioning existing protein structures into overlapping segments of short length. Clustering all fragment instances then allows to group analogous segments and select fragment representatives (Figure 2.4).

Several attempts have been made to develop structural alphabets using various fragment sizes and distance metrics for clustering:

- *Building Blocks*: 6-mers identified using k-means clustering of C_α Root-mean-square deviation (RMSD) values [88]
- *Local structural motifs*: 9-mer fragments, described in terms of torsion angles and identified by unsupervised learning [77]
- *Short Structural Building Blocks*: 4-mer C_α backbones (HMM) [15, 16, 14]
- *Protein Blocks*: 5-mer fragments, whose pairwise distance is defined by RMSD of angular values (self-organized map) [23, 22]
- *Small Libraries of Protein Fragments*: libraries of 4-, 5-, 6-, and 7-mer fragments, identified by k-means clustering of C_α RMSD values [57]
- *I-Sites*: overlapping fragments of 3 to 15 residues (Figure 2.5), identified using k-means clustering of sequence profiles and filtered by structural criteria (RMSD, maximum deviation in torsion angles) [10]

Most structural alphabets follow a straightforward approach of dividing and clustering the protein structure space into a reduced set of structural motifs. Fragment libraries designed upon this structure-oriented concept are used in algorithms for fast comparison of protein structures. Every structure can be encoded as a simple string of structural motifs and this representation allows efficient fold comparison using standard alignment algorithms and specialized scoring matrices [87]. However, libraries in this category have limited application in fold recognition and *ab initio* structure prediction. The first step in *ab initio* fragment assembly (2.3.2) is a sequence-based selection of compatible fragments from a fragment library. Only those structural alphabets which provide information about the sequence preferences of their fragment classes can be used for local structure prediction and *ab initio* 3D structure prediction from sequence. Several attempts have been made to infer the amino acid preferences of existing structural alphabets as their secondary property [23]. However, these approaches are currently not very accurate, which results in a prohibitively high false-positive rate of fragment assignment.

The first successful attempt to solve this limitation was the development of the I-Sites fragment library [10, 12]. The fragments in this dictionary have been discovered using clustering of sequence profile segments of fixed length. Detectable sequence conservation is therefore the main condition for defining a new I-Site. All fragments are subsequently examined for a stable sequence-to-structure correlation; fragment clusters that do not demonstrate conserved 3D shape are

Figure 2.5: Sample fragments from the I-Sites static fragment library. Segments of non-homologous, unrelated PDB structures, matching the sequence profiles of the sample I-Sites, have been extracted and superimposed. These I-Sites therefore represent recurrent building blocks in protein structures.

considered unreliable and excluded from the library. The final set of I-Sites consists of supersecondary structures which are characterized by the combination of two descriptors:

1. structural — torsion angles of the representative fragment in each cluster, termed the "paradigm" structure
2. sequence — average sequence profile of all segments in a given cluster

This property meets the requirements for *ab initio* protein structure prediction. Fragment assignment is performed using the nowadays well-established profile-profile alignment technique for remote homology detection. Matching fragments are kept and the torsion angles of their paradigms directly assigned to the query sequence of unknown structure [10, 11]. This method has very high accuracy, which ensures that most fragment assignments will lead to correct local structure prediction. However, it does not demonstrate sufficient sensitivity: only a small portion of the query sequence is typically covered with I-Sites assignments. This is a major flaw since high coverage is central to successful *ab initio* structure prediction.

The concept behind static fragment alphabets in general is very attractive. The ability to describe every protein structure in terms of a small number of structural motifs holds significant power. Unfortunately, this approach has unresolved intrinsic drawbacks, which restricts its practical applications in protein structure prediction mainly to the area of loop modeling [30].

2.4.2 *Dynamic fragment libraries*

The limitations of the structural alphabet approach have been overcome with the introduction of conventional fragment detection methods, which produce dynamic, *ad hoc* fragment libraries. These approaches do not attempt to describe and summarize all existing struc-

tural motifs. Instead, they serve a very practical purpose: ab initio protein structure prediction by fragment assembly (2.3.2). Their goal is straightforward: given a target sequence of unknown structure, build a fragment set tailored to the query sequence. Every fragment in a dynamic library is characterized by matching positions in the target sequence, compatible sequence profile, compatible secondary structure and associated list of (φ, ψ) torsion angle pairs.

The key properties of a dynamic fragment library are:

1. disposable — the fragment library exists only in the context of its associated target sequence

2. comprehensive — there must be at least one assigned fragment for the majority of target residues (ideally for all)

3. accurate — for each covered residue in the target, there must be at least one fragment in the list of candidates whose torsion angles are reliable and useful; that is, the 3D structure of most fragments should match the actual local structure of the target protein (ideally, the amount of incompatible fragments should be negligible)

Rosetta's NNmake fragment detection program [72, 36] is the first mainstream application of this approach. NNmake compiles dynamic libraries by excision of structural fragments from a non-redundant database of experimentally solved structures with high resolution. The fragment search process uses a sliding window of size 9 residues to match every sequence segment of the target protein against 9-mer segments from the database. The matching algorithm performs profile-profile comparison by computing the city-block distance between each pair of profiles (note the similarity with I-Sites[10]). The secondary structure — computed with DSSP [50] for all database entries and predicted with PSIPRED [49] for the target sequence — is also taken into account. All sequence profiles are computed with multiple iterations of PSI-BLAST [2]. Matching database 9-mers are ranked and top N segments per starting position are kept as the final list of candidates. The torsion angles of all candidate fragments are extracted from their respective PDB structures.

This fragment detection strategy is similar to the way the I-Sites motifs have been discovered. In fact this procedure routinely re-discovers existing I-Site fragments. However, it does not sacrifice profile sensitivity by averaging the sequence profiles over all fragment instances; NNmake simply keeps all such instances as useful fragments. As expected, this leads to significantly increased sensitivity and coverage, which makes the method very well suited for *ab initio* structure prediction. However, as it was the case with I-Sites, fragment libraries compiled with NNmake have fixed fragment size. This is a notable drawback which often hurts accuracy (Chapter 3).

2.5 NMR STRUCTURE DETERMINATION

Protein structures are traditionally determined by X-ray crystallographic methods. In recent years however, Nuclear magnetic resonance (NMR) spectroscopy has been developed for solving the structure of small proteins. NMR spectroscopy is currently the method of choice for studying the structure of proteins in solution.

2.5.1 *Chemical shifts*

NMR methods exploit the magnetic spin property of the atomic nuclei in protein molecules. When placed in a strong magnetic field and excited with radio frequency pulses, hydrogen atoms emit radio frequency radiation, which is registered by the instrument. The frequency of this radiation is characteristic to some extent for the nucleus that emits it — it depends on its chemical nature *and* concrete molecular environment [9]. This frequency (ν) is measured relative to a reference (ν_{ref}) to obtain a corresponding *chemical shift* (δ):

$$\delta = 10^6(\nu - \nu_{ref})/\nu_{ref} \tag{2.3}$$

expressed in parts per million (ppm).

Once the chemical shift values are assigned to the individual amino acids along the polypeptide chain, the information contained in them can be used for protein structure determination [91]. For example, chemical shift data has been successfully combined with *ab initio* protein structure prediction methods [17, 80]. Such methods rely on scanning known experimental structures for short analogous fragments with similar chemical shift patterns. Detected fragments are then used in a Rosetta-like fragment assembly folding protocol (2.3.2).

There is a well-pronounced correlation between the secondary structure of proteins and their chemical shifts [92]. This observation is used in programs like TALOS [21, 79] and DANGLE [18] to predict local structure (torsion angles and secondary structure elements). This is achieved by calculating the so-called *secondary chemical shifts* ($\bar{\delta}$) for each observed chemical shift (δ_{obs}):

$$\bar{\delta} = \delta_{obs} - \delta_{ref}(r, n) \tag{2.4}$$

where $\delta_{ref}(r, n)$ is the *random coil* chemical shift of nucleus n in amino acid type r. The reference chemical shift values in this equation are obtained for peptides in a "random-coil" conformation, which refers to an unfolded polypeptide chain with no secondary structure. The secondary chemical shift is a quantity that depends on the secondary structure of the protein. For example, in beta strands C-α atoms tend to have negative secondary shifts, while C-β atoms lean towards positive values.

2.5.2 *NOE spectra*

Nuclear Overhauser effect (NOE) is a phenomenon widely used in 2D or higher dimensional NMR experiments to obtain information about hydrogen atoms that are located close in space. The data from such experiments can be used to compute the 3D structure of proteins, since the pairs of contacting residues in NOE spectra are often located far apart in the amino acid sequence. These long-range contacts are converted into distance restraints and used in a structure calculation, based on the observation that every NOE peak corresponds to a pair of hydrogen atoms, separated by a distance of up to 5-6Å.

The main challenge with this method is the interpretation of the raw data. NOE spectra contain information about contacting protons and higher-dimensional experiments provide additional information about other chemical elements that are covalently linked to the interacting hydrogen atoms. However, NOE spectra do not specify where these atoms are located on the amino acid sequence. Software applications consuming NOE data must therefore guess the correct mapping (*cross-peak assignment*) — a non-trivial end error-prone task. The initial step in this process is matching the position of every NOE peak in frequency space (ω_1, ω_2) against the list of chemical shift values ω_i within a small tolerance $\Delta\omega$. Such strategy would rarely produce a single match for a given NOE peak, hence very few NOE peaks can be assigned to a pair of chemical shifts unambiguously.

Several algorithms have been developed in an attempt to automate the task of cross-peak assignment and cope with its inherent ambiguity. Early attempts focused on using unambiguous assignments only, which significantly hurts their performance (since the majority of peaks cannot be assigned unambiguously). This motivated the development of *ambiguous distance restraints* in ARIA [65, 66]. Furthermore, *iterative cross-peak assignment* was established as a computationally demanding, but successful strategy for NOE assignment and structure calculation. This approach is implemented in the widely used packages ARIA and CYANA [37, 39]. On each iteration, these programs compute a preliminary structure from the current list of assigned NOEs; then this structure is used to guide the NOE assignment on the next cycle.

The lack of accepted measures of quality for NMR models poses a second challenge for this method. NMR structure calculation can be envisioned as a process of fitting parameters (coordinates) to experimental data, but most structure calculation approaches do not provide quantitative measure of the *goodness of fit*. This issue was addressed in the Bayesian framework of ISD [70, 40, 71], which computes 3D coordinates along with their associated "error bars" and weights the experimental data optimally to avoid under or over fitting.

3 HHFRAG: DYNAMIC FRAGMENT LIBRARIES

3.1 INTRODUCTION

In Chapter 2 we introduced the concept of homology and discussed how methods for sensitive homology detection have made protein structure prediction possible. Homologous templates can be routinely selected from a growing collection of experimentally determined protein structures, stored in the Protein Data Bank [5]. The size and diversity of the PDB library are therefore critical factors, which determine the performance of structure prediction methods. This claim is intuitively valid for all comparative modeling algorithms (2.3.1), but also holds for *ab initio* approaches [81], which depend on fragment detection and fragment assembly (2.3.2).

Recent reports suggest that PDB has already grown to a level of diversity that is sufficient for practical structure prediction purposes [97]. New experimental structures are being published with a stable rate, however, the share of novel folds observed in new structures is getting surprisingly low. Even more peculiar is the evidence that the small number of newly discovered folds often reuse common structural motifs (2.1.3, 2.4.1), already seen in non-homologous PDB structures [28]. Novel folds tend to enrich the PDB library with new combinations of known building blocks and rarely introduce new motifs.

Our ability to detect homology has improved significantly with the development of methods for pairwise comparison of sequence profiles (2.2.2). Conventional sequence alignment is sufficient for template selection when direct homologs of the query protein are present in PDB (2.3.1). However, sensitive profile alignment methods succeed at template detection even when the sequence identity level is lower than the critical threshold of 30%, termed the *twilight zone* [74]. Since structures accumulate evolutionary changes more slowly than sequences, such *remotely homologous templates* are guaranteed to be useful for comparative modeling, although the percentage of sequence identity may indicate a massive sequence divergence.

But even when the most sensitive profile-profile comparison methods fail to identify a sufficiently long template for comparative modeling, there is still hope for successful structure prediction. Using local alignment of sequence profiles, we are often able to reveal relatively

Some of the material in this chapter has been previously published and adapted from Kalev *et al.* (2011) [51] and Kalev *et al.* (2013) [52]. Used with permission.

short segments local similarity, although the proteins in comparison may belong to different, unrelated superfamilies and adopt different folds. Remarkably, these short regions of profile similarity are often ubiquitous and may demonstrate conserved structural properties (local structure, geometry or contacts). The sequence-to-structure correlation of such recurrent motifs has been studied in the I-Sites fragment library [10], which has proven useful in protein structure prediction [11].

I-Sites is an early attempt to summarize known motifs into a compact, static structural alphabet (2.4.1) [67]. This approach to local structure prediction produces accurate results, but suffers from insufficient coverage. When the query sequences contain instances of any I-Sites motifs, the I-Sites scanning algorithm succeeds at their identification with acceptable precision. However, I-Sites motifs rarely span more than one third of the whole sequence. The remaining connecting regions would therefore remain unassigned, i. e. only the local structure of a small fraction of the query backbone can be predicted. Another important limitation of I-Sites (and static alphabets in general) is that all fragments have a fixed size. The associated local structure prediction algorithm must therefore use a sliding window of that size to assign fragments to the query. While many recurrent supersecondary structures demonstrate a conserved core region, matching a canonical I-Site, motif instances tend to vary at their termini and sometimes contain internal gaps or insertions. These two problems render the sliding window approach impractical. The first problem — the variability of fragments at their tails, was addressed with the preparation of complementary fragment libraries of increasing fragment length (3 to 15 residues). However, this approach is still inefficient in comparison to a truly dynamic, context-dependent fragment assignment, since it is not practically possible to capture the complete spectrum of mutational variability at the termini. The second problem — the presence of internal gaps and insertions, is less prevalent, but remains completely unaddressed by static approaches.

NNmake — the fragment detection module of Rosetta [72, 55] — solves the coverage problem by introducing the notion of dynamic fragment libraries (2.4.1). This method builds customized and comprehensive fragment sets, which are more suitable for *ab initio* structure prediction. In more conserved regions of the query (i. e. potential I-Site instances), we expect dynamic libraries to be as accurate as static methods for local structure prediction. However, in less conserved regions (i. e. variable I-Site-linking segments) the increased coverage of dynamic fragment search comes at the cost of significantly reduced precision. Another important limitation of this method is that dynamic fragments typically come in a fixed, predetermined size (usually 9 residues) [13, 81, 48]. This is a technical decision which simplifies the implementation, because it makes a sliding window

profile search possible. However, it does not appear to be supported by any biological justification. Moreover, the fixed fragment size hurts the precision of this method even in conserved regions, because not all paradigm motifs from I-Sites have core regions of size 9 residues. For example, whenever the prototype of a motif is shorter, NNmake may include unnecessary tails with incompatible torsion angles. Similarly to I-Sites, NNmake provides no answer to the internal gap or insertion issue, although it has the advantage of using a much bigger source library of fragments, thus theoretically increasing the chance of observing a fragment instance with the same set of internal mutations.

3.2 MOTIVATION

HHpred [44] has been one of the most successful comparative modeling servers in recent years. It outperforms many rival methods, thanks to its HHsearch module — a very sensitive algorithm for remote homology detection. This is achieved by pairwise comparison of profile HMMs — a concept employed by other static and dynamic fragment detection methods such as I-Sites and Rosetta NNmake, which compare conventional sequence profiles. However, unlike those methods, HHsearch uses a dynamic programming algorithm to perform actual alignment between each pair of profiles, thus detecting variably sized regions of local similarity.

This property reveals its potential application in fragment detection, where it can be used to address the limitations of existing approaches, one of which is the fixed fragment size. As already discussed, fragments of variable sizes are desirable, because most fragment instances deviate to a varying extent from the canonical I-Site paradigms at their termini, core, or both. Resolving this issue will therefore have an immediate positive impact on the performance of fragment detection. Given the random sampling nature of the current *ab initio* methods for structure prediction (2.3.2) and NMR-based determination (2.5.1), improving the quality of fragment detection is expected to boost the performance of fragment assembly. The use of fragments of higher quality will intuitively improve the decoys built with *ab initio* methods. When a fragment library is enriched with useful and accurate fragments, this also reduces the frequency of erroneous Monte Carlo moves while sampling the library, thus speeding up the conformational search.

With these observations in mind, we designed a novel fragment detection method, which combines the strengths of existing static and dynamic fragment libraries, while adequately resolving their limitations. Our method, called HHfrag [51], takes advantage of HHsearch's highly sensitive remote homology detection to discover local regions of structural similarity, shared across different folds. Instances

of such conserved motifs are directly excised from experimental protein structures. The resulting dynamic fragment libraries possess the following properties:

1. *dynamic* — customized to the query sequence and maximizing its coverage (i. e. no attempt is made to compile a structural alphabet), thus appropriate for *ab initio* protein structure prediction or determination by fragment assembly;
2. *flexible*, *context-specific* — fragments are variable in length and also allow for gaps and gapped fragment assignments, unlike earlier fragment detection methods;
3. *accurate* — fragments with maximum precision within and near conserved supersecondary structures (i. e. known I-Sites).

A fragment library that contains more variability than a static dictionary and that is, at the same time, more precise than the dynamic Rosetta approach, allows for more efficient sampling of the conformational space. This implies that an *ab initio* model of the target structure can be built in fewer trials and out of better decoys. Later in this chapter we demonstrate the application of our fragment library in *ab initio* structure prediction using a modified Rosetta *ab initio* protocol, adjusted to work with fragments of variable length.

3.3 THE FRAGMENT DETECTION ALGORITHM

HHfrag is a dynamic fragment detection method, which can be viewed as an extension to the HHsearch template selection algorithm [84]. While HHsearch runs in local alignment mode by default, it has been specifically developed for detection of longer threading templates. It has not been therefore optimized for fragment detection, although it is capable of identifying short conserved regions. Running HHsearch on a pair full-length HMMs is likely to produce a local alignment, but there is no guarantee that the alignment will be optimal in terms of locally conserved supersecondary structures. Very often HHsearch discards a single local match as insignificant, because a short supersecondary structure match does not provide enough evidence of homology between the proteins in comparison. To force HHsearch in "strictly local" mode and obtain a locally optimal alignment for a short motif, one has to restrict the aligned area to the region including the motif itself and its nearby residues. This goal is achieved in HHfrag by splitting the fragment search into two phases (Figure 3.1):

1. Detection of optimal query segments — the aim is to decompose the query sequence into segments which contain instances of known recurrent building blocks. This decomposition is very approximate, i. e. the boundaries of each segment are not optimal, but always exceeding or equal to the actual ones.

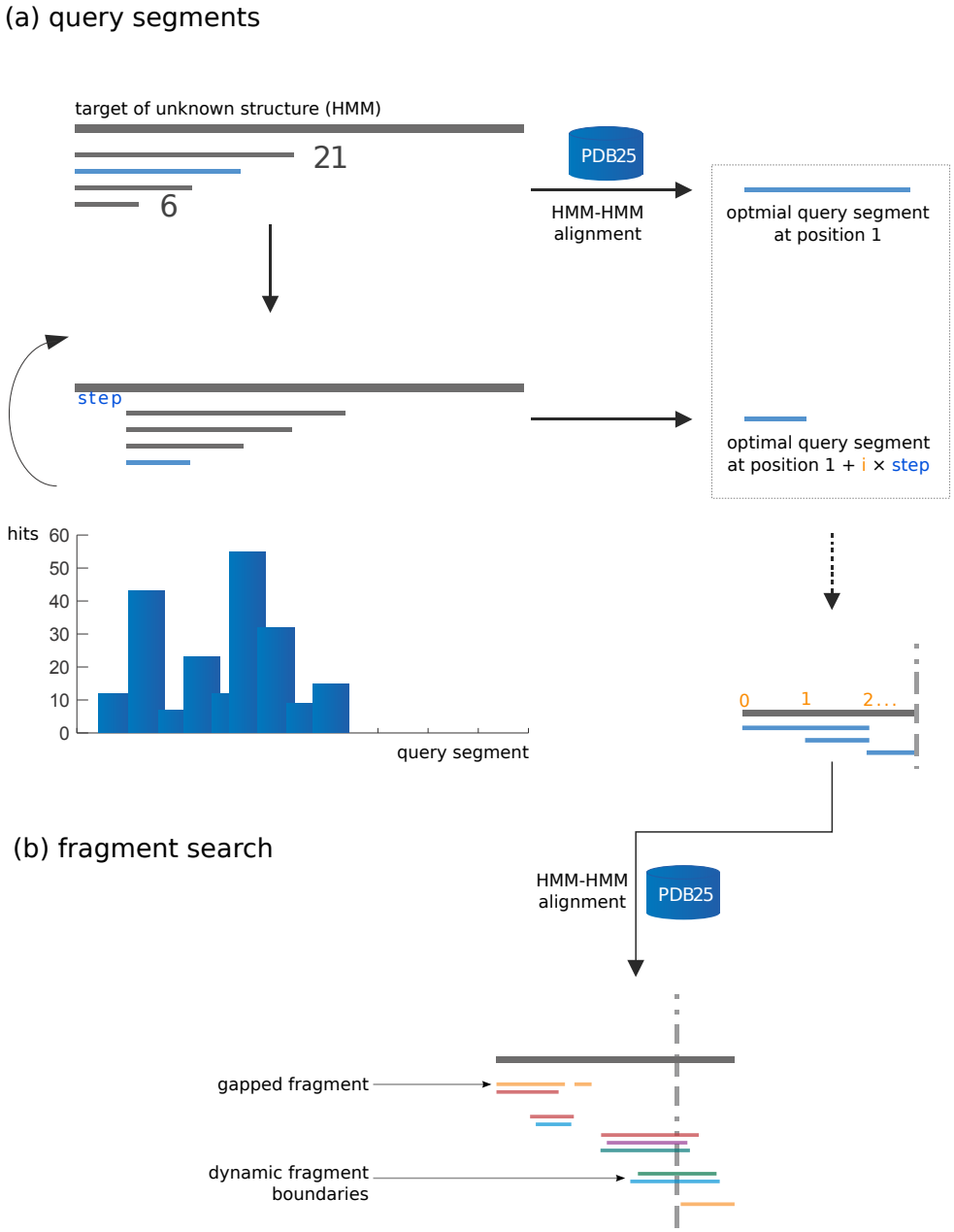


Figure 3.1: Outline of the HHfrag algorithm. During the first phase (a), HHfrag scans the query profile for conserved supersecondary structures and identifies their approximate boundaries. Excised query segments are then used to scan a library of experimental structures for matching motif instances (b). At this stage HHfrag determines the actual boundaries of each motif and compiles a dynamic fragment library.

2. Fragment extraction — at this stage HHfrag must determine the actual boundaries of each motif, find and excise instances of it from the library of experimental structures (PDBS25). This is achieved by performing local alignment of each query segment against the full length templates from PDBS25.

A flowchart of the full algorithm is shown in Figure 3.1. The key component of the fragment detection strategy is HHsearch [84], which is used for HMM-HMM alignment with pseudocounts and secondary structure scoring enabled.

3.3.1 *Preparation*

HHfrag uses HHsearch for profile alignment. All input sequences must therefore be converted to HHM (HHsearch HMM) format. We use the standard HHpred toolchain [44] to create all required profile HMMs. This involves generation of multiple alignments with 8 rounds of PSI-BLAST and inclusion e-value of 0.001 [2].

Secondary structure information is also included in the profile and used for fragment detection. For all database templates experimental structures are available, so we use DSSP [50] to compute the secondary structure. The query sequence does not have a 3D structure and thus PSIPRED [49] is used to obtain an approximate secondary structure prediction.

Each final HMM comprises amino acid emission probabilities and secondary structure propensities. To increase the sensitivity of the search, we also use emission and transition pseudocounts, computed with HHmake ([51] — supplementary material).

All fragments, contained in a dynamic HHfrag library, are real structural motifs, extracted from experimental PDB structures. The database of template structures, called PDBS25HMM or just PDB25 for brevity, is a compact, non-redundant subset of PDB. It is based on the April, 2010 build of PDBselect25 [35], which contains PDB structures filtered at 25% identity (4824 chains in total). Every entry in PDBS25 comprises of a profile HMM (built with the method outlined above) and a corresponding high-resolution structure.

3.3.2 *Motif decomposition*

During the first phase of the algorithm, HHfrag attempts to decompose the query profile HMM into conserved motifs and identify their approximate locations (Figure 3.1). As discussed earlier, HHsearch runs in "template homology detection" mode by default and dismisses many local matches as insignificant. To trigger the desired local search behavior, we must restrict the length of the aligned HMMs to the area in or around each motif.

This is achieved by chopping the query profile into a nested list of segments of increasing size, from 6 to 21 residues (Figure 3.1). The motif segmentation routine simply slices out all HMM layers within the selected segment. A new HMM is then created by copying the sliced out layers and the HMM is finalized by adding start and end states, connected to the first and last layers, respectively, with a transition probability of 1. At a given column c, the slicing procedure results in a list of candidate segments spanning layers $(c, c+6)$ to $(c, c+21)$ and anchored at the same starting position c. Note that this technique disregards any prior information about the location of the motif. Most of the resulting segments will hence contain either truncated motifs, or elements with extra flanking residues. In the event of truncation, it can be assumed that the resulting profile segment will be partially damaged and possibly dysfunctional. By comparing all candidate profile segments against the template profiles (PDBS25) we can therefore identify the segment(s) with preserved integrity: intact segments will produce more consistent and more abundant matches. The query segment, having the maximum number of hits, is believed to contain the intact motif and is chosen as the optimal query segment at the given position c.

After shifting the origin of segmentation by three residues downstream, we repeat the same procedure in order to obtain the next optimal query segment at position $1 + 3 \times i$. This is repeated until the C-terminus of the query sequence is reached. At the end of the last iteration, this yields a list of partially overlapping query segments with variable length and increasing start positions. Every optimal segment contains an intact motif, but its exact position within the segment will be determined in the next phase.

3.3.3 *Fragment extraction*

Once motif decomposition is completed for all $1 + 3 \times i$ iterations, HHfrag must identify the actual boundaries of each motif in the query profile and collect matching fragment instances from the PDBS-25 library. This is achieved by performing local alignment of each optimal query segment against the full-length template HMMs in PDBS25. As in the previous stage, HHsearch [84] is used for HMM-HMM comparison. This results in local HMM alignments that are equal to in length or shorter than the original query segments, possibly containing internal gaps as well. Using the information encoded in the alignments, HHfrags successfully achieves all of its final goals:

1. identifies the start and end positions of all conserved motifs in the query dynamically;

2. collects matching fragment instances from experimental PDB structures (PDBS25) and builds a dynamic fragment library;

3. collects gapped fragments, containing short internal insertions or deletions in the linkers, connecting the neighboring secondary structures.

The algorithm completes by excising all matching fragments from their respective PDB structures and building a position-specific fragment library in Rosetta [72] format. An HHfrag library however comprises of fragments of variable length, from 6 to 21 residues. The number of fragments assigned to each query position is also variable. For highly conserved regions, i. e. those corresponding to known I-Sites [10] or other abundant motifs, this number will be very high, often in the order of hundreds. For highly unconserved or linker regions HHfrag will extract few or no matches at all.

3.4 CHARACTERISTICS OF DYNAMIC FRAGMENTS

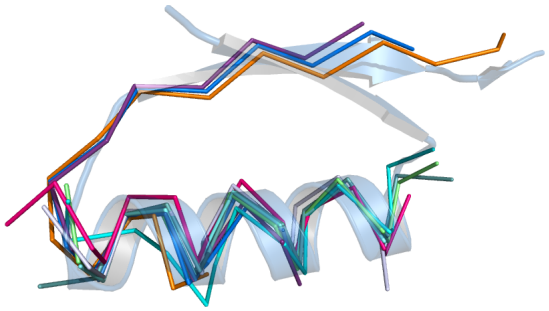
We define three intuitive performance metrics, which are sufficient to compare the performance of various fragment libraries and measure their quality objectively:

1. *global precision* — the percentage of true positives among all fragments assigned to the query;
2. *local precision* — the percentage of true positive assignments, which cover a specific region or a single residue;
3. *coverage* — the percentage of query residues that are covered by at least one true positive fragment [59].

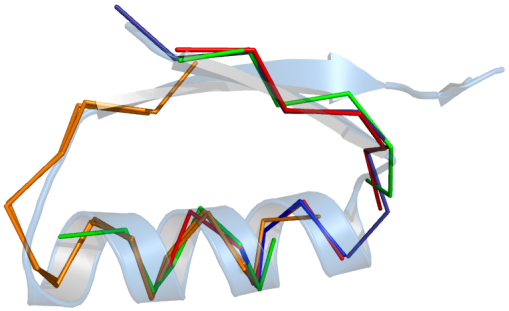
A fragment is considered to be a *true positive*, if it matches the local structure of its target protein accurately. Several metrics can be used to compare the structural similarity of short fragments [10, 57, 59]. We have chosen to use the popular C_α-RMSD and apply a threshold value of 1.5 Å, denoting a true positive (compatible) hit. The RMSD metric is length-dependent, which suggests that it might be adequate for comparison of constant length fragments only. Guided by this intuition, we considered the development of length-independent similarity metric, but found that the overall picture of our results did not change [51] (see supplementary data of [51] for detailed analysis). We therefore use the conceptually simpler definition of a true positive based on conventional RMSD, adopted as a structural alphabet similarity metric by other authors [57].

3.4.1 *Contextual variability*

The core mechanism behind HHfrag has emerged from an early attempt to rebuild the I-Sites [10, 12] fragment library using HHsearch [84] as a profile comparison backend. This approach, named StaticHH,



(a) I-Sites



(b) HHfrag

Figure 3.2: Modular fragment structure. Many fragments can be decomposed into elements of two or more connected sub-fragments. (3.2a) Superimposition of several overlapping I-Sites motifs. These fragments have been assigned to the same region of the query sequence, because their sequence profiles are very similar. (3.2b) Corresponding gapped fragments (green, red) and fragments with flexible boundaries, found by HHfrag.

successfully identifies most I-Sites. However, it suffers from the same limitations as other structural alphabets, which directed our efforts at dynamic fragment detection and lead to the development of HHfrag.

A closer look at Figure 3.2 reveals some of those limitations. During the development of StaticHH, we noticed that the most abundant fragments have core regions, which can be linked to corresponding I-Sites. The fragment core alone is unfortunately not sufficient to describe all instances of a fragment as their lowest common denominator. Most fragments tend to demonstrate hierarchical, modular arrangement. Proteins often omit arbitrary elements from these modular motifs, which renders their clustering and generalization difficult.

One typical modular fragment is shown in Figure 3.2. It is composed of two elements — an alpha helix and a beta strand, which are interestingly also defined as independent I-Sites paradigms. I-Sites therefore contains highly redundant and overlapping fragments, which is a legitimate attempt to address the hierarchical nature of the motifs. The protein structure shown on the figure contains a "full instance" of the fragment. Searching for instances of this fragment in

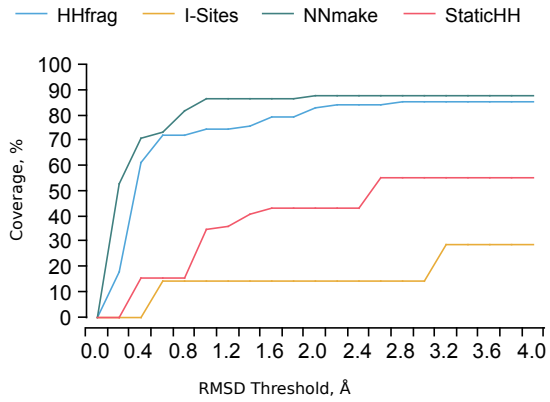
PDBS25 with HHsearch reveals a broad spectrum of variability. Some proteins contain only one of the two elements and these combinations are covered by I-Sites. Other structures however, like the one shown on the figure, contain an additional third element, which has no corresponding I-Site. A peculiar subset of the second group of matches is comprised of "full instances", which may contain a short internal insertion or deletion between the second and the novel third element. In addition, the tails of the shorter I-Sites fragments do not fit the overall shape of the actual motif very well. These two observations demonstrate the disadvantages of using rigid fragments of fixed length and suggest that there is no optimal length for fragment detection. A fully dynamic method for fragment assignment is expected to produce more accurate local structure predictions.

Our study shows that the prevalence of gaps in fragment instances is not very high. On average, only 9% of all HHfrag motifs contain any gaps and 8% of all best-fitting fragments are indeed gapped. This is a relatively small, but not negligible number. The gap-detection capabilities of HHfrag are therefore not essential when the template database is able to provide ungapped alternatives to all motifs. However, the ability to detect insertions and deletions could be crucial in the event that PDBS25 contains only very few instances of a given structural motif [51].

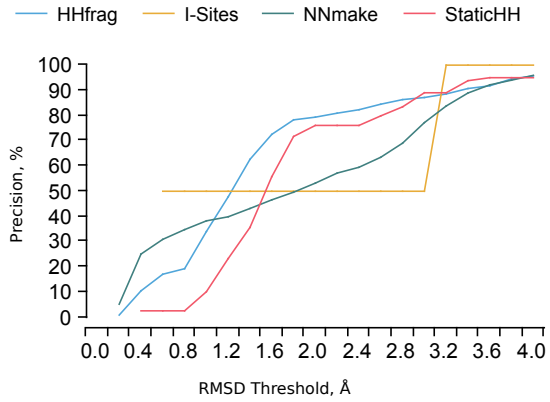
3.4.2 *Precision and coverage*

A key design goal behind HHfrag is establishing a balance between good precision and high coverage. As expected from a dynamic fragment detection method, HHfrag demonstrates significant improvement over structural alphabets in terms of coverage. Unlike earlier dynamic methods however, HHfrag does not sacrifice precision in exchange (Figure 3.3).

Figure 3.3a shows an example where HHfrag may look outperformed by Rosetta in terms of coverage: NNmake reaches 87% coverage at RMSD threshold of 1.4 Å, whereas HHfrag covers 76% of all target residues at the same cutoff. However, coverage alone can be a misleading metric, unless the precision of the fragment library is also taken into account. A high level of coverage ensures that most query positions are covered by at least one correct assignment. But it does not provide any guarantees that the correct fragments can be easily identified and extracted from the library. Most fragment assembly protocols rely on random sampling of fragments from each list of competing candidates. The frequency of picking a correct fragment will therefore have a significant impact on their performance. This frequency — the percentage of true positive fragments in a given library, also known as its *precision* — is typically reasonably high for static structural alphabets like I-Sites. It is surprisingly low, however,



(a)



(b)

Figure 3.3: Coverage and precision for benchmark protein 3nzl at increasing RMSD cutoffs.

for dynamic approaches such as Rosetta NNmake. Figure 3.3b shows that for the same target protein, HHfrag has 1.5-fold higher global precision than Rosetta.

We can get a better understanding of the physical meaning of these numbers by comparing the *local precision* of HHfrag and NNmake fragment libraries (Figure 3.4; complete set of diagrams can be found in [51], supplementary material). The residue-wise precision diagrams for both methods have clearly observable patterns of high-accuracy peaks, connected by regions of low motif conservation. However, the peaks found in all Rosetta NNmake diagrams have a very characteristic triangular shape. The precision of these libraries drops rapidly as we move away from the maximum. In comparison, HHfrag produces peaks which have more rectangular shape, keeping the area of high accuracy broader. These observations are easily explained if we consider the mechanism of action of both methods. NNmake uses a sliding window of fixed size for fragment detection, thus disregarding any contextual variability of the fragments (3.4.1). This approach leads to accumulation of a large number of truncated,

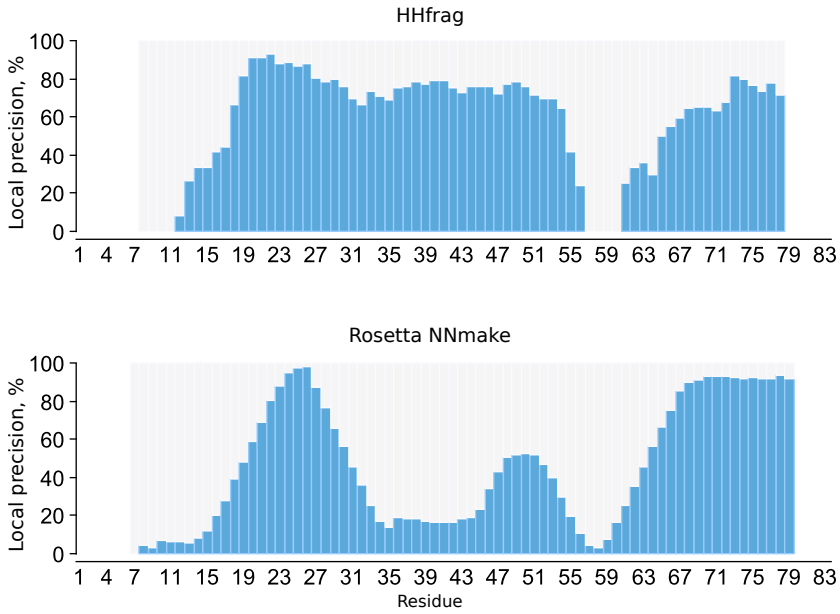
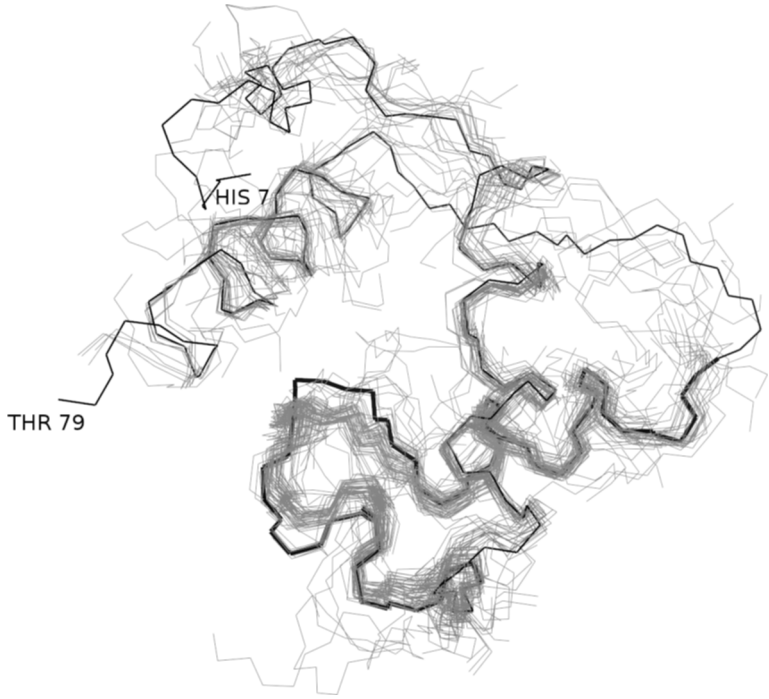


Figure 3.4: Local residue-wise precision of HHfrag and NNmake libraries at default RMSD cutoff (1.5Å), compiled for target 3nzl. Each blue bar indicates the percentage of true positive fragments, which cover a given target residue. The grey background corresponds to the false positive rate, and the white regions are completely unassigned.

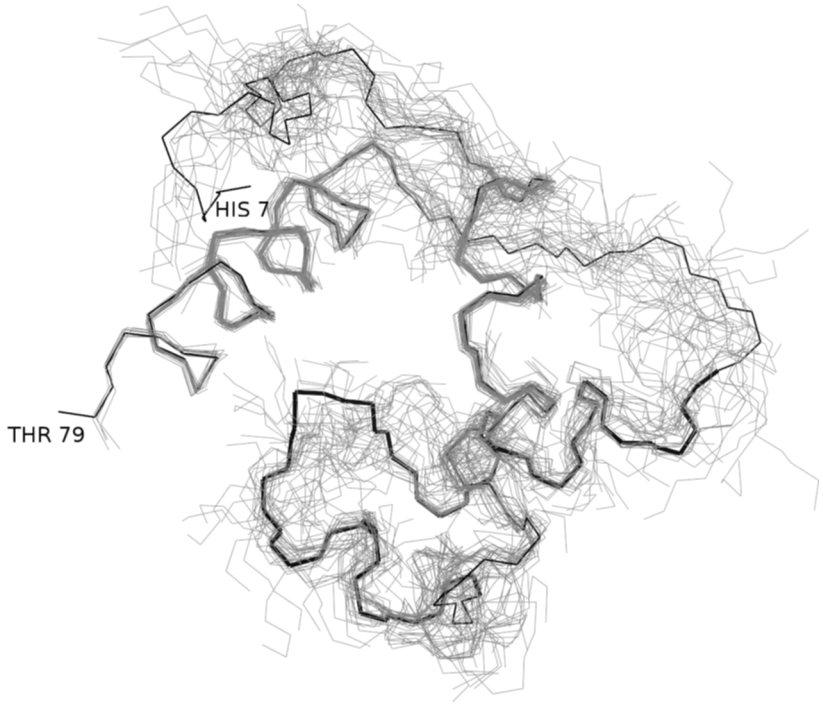
sub-optimal fragments. They may contain correct motif cores, but are often concatenated with unwanted tails, which extend from the core and exceed the boundaries of the actual motif. In contrast, HHfrag uses a local alignment algorithm to determine the correct boundaries of each fragment and rarely conquers unnecessary residues from neighboring linker regions (Figure 3.5). This is a key property of the HHfrag algorithm, which significantly improves the local precision of our method and contributes to the demonstrated high accuracy (3.5).

3.4.3 *Link to structural alphabets*

Dynamic fragment detection has been developed as an extension to the idea of structural alphabets, aiming at improved local structure prediction coverage, accomplished by fragment excision from less conserved or uncommon motifs. Dynamic fragment libraries can therefore be regarded as supersets of known structural alphabets, i. e. HHfrag and NNmake routinely re-discover known I-Sites prototypes. But what is the nature of the he high-accuracy peaks, observed on local precision diagrams (Figure 3.4)? Our study confirms the expectation that these regions are strongly correlated with the I-Sites motifs. The I-Sites profiles align to query regions where HHfrag assigns fragments with great precision ($80 \pm 18\%$ on average). This fact makes



(a)



(b)

Figure 3.5: Fragment maps, generated with HHfrag (3.5a) and Rosetta NNmake (3.5b). Shown are top 4 NNmake and all HHfrag fragments assigned to target 3nzl (thick backbone). All fragments were superimposed onto the native structure of the target. As already evident from Figure 3.4, residues 35-55 (thicker backbone) are covered with more accurate HHfrag motifs. Transitional regions, connecting the high-accuracy peaks, exhibit high variability and lower recurrence.

intuitive sense, since most I-Sites are highly recurrent motifs, which possess strong sequence profiles (making their detection relatively easy and reliable). Our data also confirm that HHfrag libraries are proper supersets of the I-Sites structural alphabet, thus fulfilling one of the major design goals of the project.

The unassigned and low-accuracy regions, seen on the local precision diagrams, are highly variable (Figure 3.5). These are regions of uncertainty, where no reliable local structure prediction is possible due to the lack of sequence conservation. Such parts of the protein molecules, often located in linkers and loops, need to be modeled using a brute-force approach during *ab initio* structure prediction. The fact that HHfrag does not assign fragments to uncertain regions ("white regions" in the fragment map) should be considered a feature rather than a shortcoming, because it encodes important information about the query protein. However, current structure prediction protocols such as Rosetta AbinitioRelax [55, 55] may not be able to take advantage of this information (discussed in detail in 3.6).

3.5 BENCHMARK

The performance of our dynamic fragment search method was tested on 105 target sequences, taken from the CASP9 competition [62]. The experimental structures of these proteins have been published after May, 2010 and therefore do not appear in our template library (PDBS25).

For each target, we compiled a dynamic fragment library with HHfrag and compared its performance to a reference Rosetta 9-mer fragment map. We used the same performance metrics described in detail earlier (3.4). To compute the coverage and precision of each library, we extracted the C_α backbones of all member fragments and superimposed them onto the native structure of the target at their respective positions, as determined by the corresponding fragment search method. Figure 3.5 shows the result of a sample superimposition.

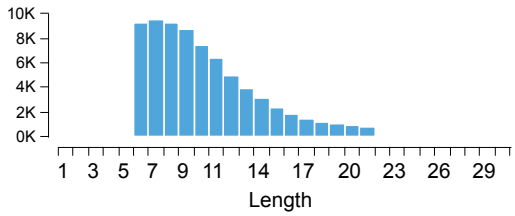


Figure 3.6: Length distribution of all fragments, extracted by HHfrag in the benchmark.

Figure 3.6 demonstrates the distribution of fragment lengths, found by dynamic HHfrag searches. This distribution peaks at a value of seven. The average fragment length is 10.3 ± 3.6 — a result in close

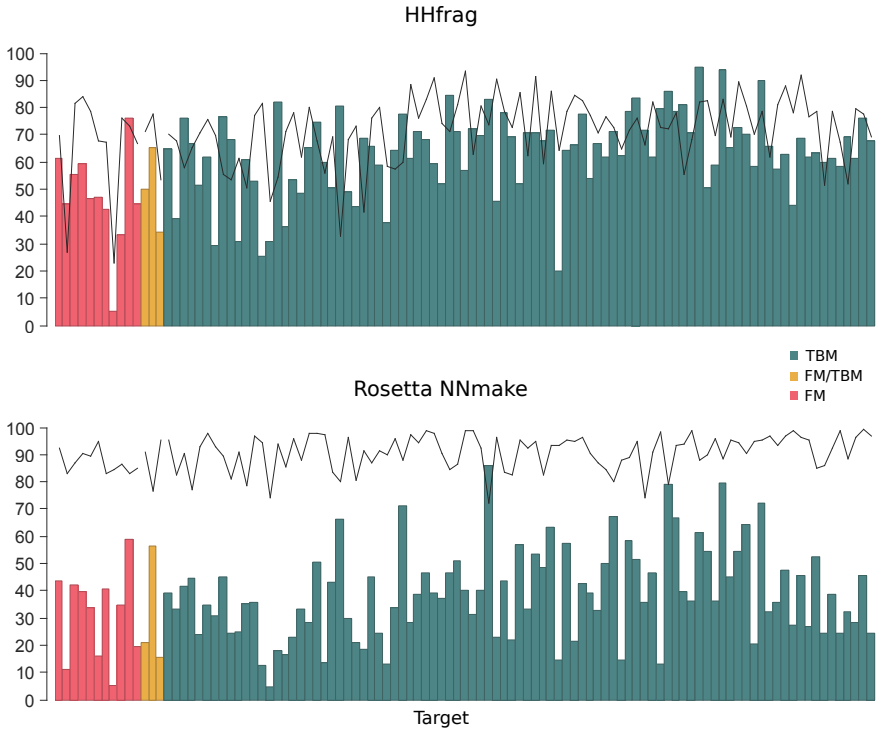


Figure 3.7: Global precision and coverage at RMSD cutoff of 1.5 Å. Each bar corresponds to a benchmark target, taken from CASP9. The height of the bar denotes the global precision of the corresponding library. All targets are ordered by decreasing modeling difficulty (red: FM, yellow: FM/TBM, green: TBM). The black curve indicates the coverage for each specific target.

accordance with earlier reports [13]. However, the distribution also demonstrates a significant probability for detecting much longer fragments. This result justifies the use of dynamic libraries of variable fragment length and confirms the validity of the method.

Figure 3.7 summarizes the performance of HHfrag and NNmake in this benchmark. HHfrag obtains an average precision of $62 \pm 16\%$, which is a significant improvement over NNmake ($38 \pm 17\%$). The overall improvement is two-fold on average and for some targets achieves a dramatic increase by a factor of 4 to 6. These results are consistent across all CASP target categories (Free modeling (FM) and Template-based modeling (TBM)).

The average sequence coverage achieved by HHfrag is $71 \pm 13\%$. When the analysis is restricted to residues, part of regular secondary structure elements, the coverage rises to 84 ± 14. The percentage of residues that remain completely unassigned (white regions) is $19 \pm 12\%$. This is a significant improvement over static libraries [10] and an acceptable loss in coverage compared to the dynamic NNmake method ($90 \pm 6\%$). This result highlights a fundamental property of our dynamic fragment detection method, compared to probabilistic fragment sampling from generative statistical models, such as Torus-

DBN [8]. In fragment detection by profile database searches, some residues will never receive fragment assignments because they are not part of conserved, recurrent motifs (i. e. in loops, linkers or flexible tails). In some cases the database of templates is simply not diverse enough to provide adequate fragment matches. In either case, however, the lack of fragment assignments is an important signal for the client application that reliable local structure prediction is theoretically not possible within the specified region of the protein. In probabilistic fragment sampling, on the other hand, there is always a non-zero chance that every residue will be covered as long as we sample long enough. With infinitely large number of samples drawn from the model, the coverage will approach 100%, but the precision may drop to prohibitively low values if the native fragment is not contained in the high probability density region.

Although 90.8% of all true positive fragments have uninterrupted structure, HHfrag has detected gapped fragments at least once for 98 out of 105 benchmark proteins. Most gaps in our dynamic fragments tend to be located in or around the central residue of each fragment. On average, about 70% of all gaps are concentrated in the central regions of the motifs, often located between two elements of a multi-segment motif. As expected, such gaps are always very short. 70% of all gaps span a single residue and the number of gaps spanning more than 2 residues is negligible. With the current degree of structural diversity of our PDBS25 library, gapped assignments did not influence the overall performance of HHfrag in a major way. However, 8.4% of all best-fitting fragments per query position contain gaps, which suggests that gap detection may be useful when the number of available local templates is limited.

3.6 AB INITIO STRUCTURE PREDICTION WITH HHFRAG

The advantages of our dynamic fragment detection method have been further demonstrated in *ab initio* protein structure prediction. A simple modification of the original Rosetta AbinitioRelax [72] application allows the use of variable length fragments along with this popular method. At this instance we would like to point out that potentially better results could be obtained by fragment assembly protocols, specifically optimized to utilize the extra information, encoded in HHfrag libraries (3.6.1). Nevertheless, the standard Rosetta protocol is sufficient for most practical applications.

3.6.1 *The impact of precision and coverage*

We have seen that HHfrag produces fragment libraries with the following properties:

- high accuracy — enriched with high-quality fragments;

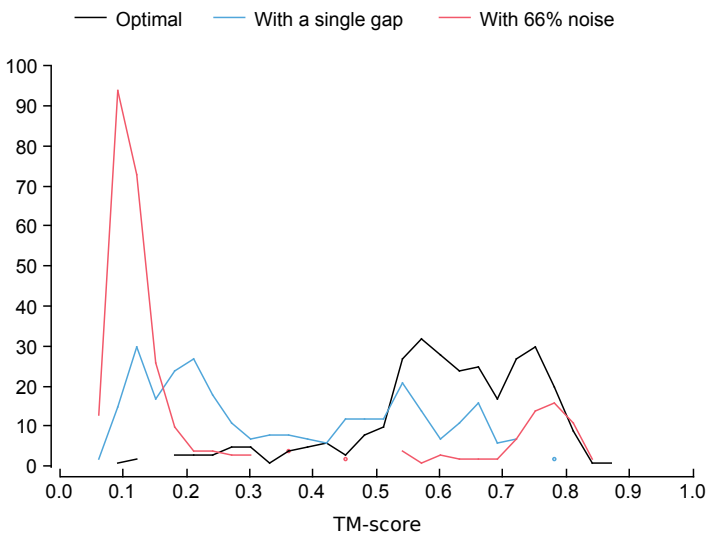


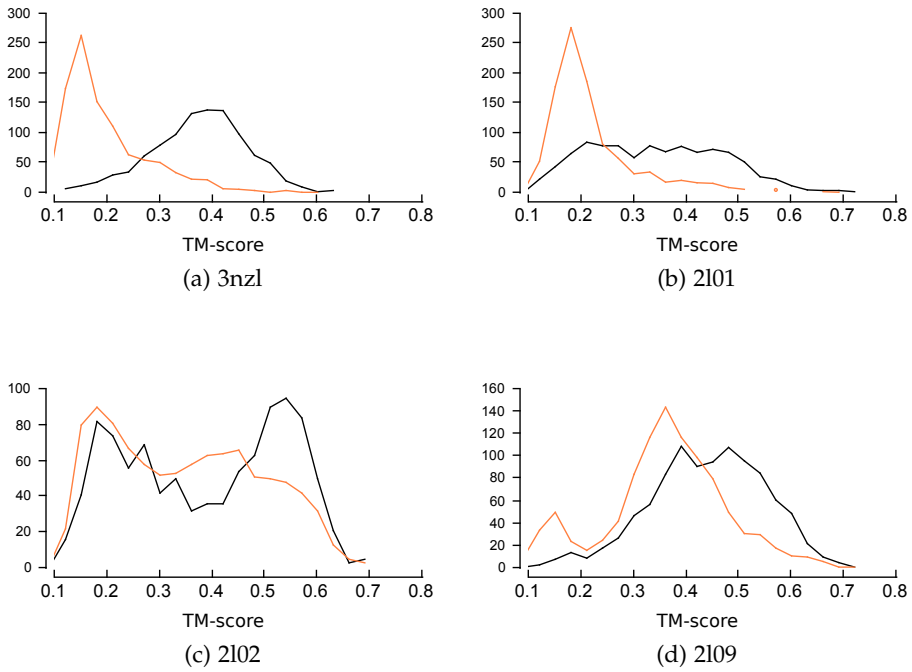
Figure 3.8: The impact of fragment precision and coverage on Rosetta modeling. Shown are the distributions of TM-scores for decoys built with Rosetta. Three different libraries have been prepared by excision of 9-mer fragments from the experimental structure of the target protein (2kxy): optimal, gapped and noisy. The percentage of good decoys was 90, 41 and 22% respectively. See 3.6.1 for discussion.

- discontinuity — some regions of the fragment map do not contain any assigned fragments by design;
- compactness — HHfrag libraries tend to contain lower number of fragments than their NNmake counterparts.

To study the effects of these properties on Rosetta *ab initio* modeling, we conducted a number of conceptual protein reconstruction experiments with idealized fragment libraries. The goal of each experiment was to test the ability of Rosetta to recover a protein structure from optimal fragments, i. e. using a library of 9-mers, extracted from the native structure of the protein. Figure 3.8 shows the decoy distributions, obtained in each experiment.

Rosetta successfully recovered the protein in the control test, performed with a library of maximal precision and coverage (90% good decoys). Intuitively, we found that mixing the optimal library with random fragments hinders the folding process, reflected by dramatic decrease in the number of near-native and good decoys (22%). This result confirms the expectation that the use of libraries of greater precision and smaller size results in much faster and more efficient conformational sampling.

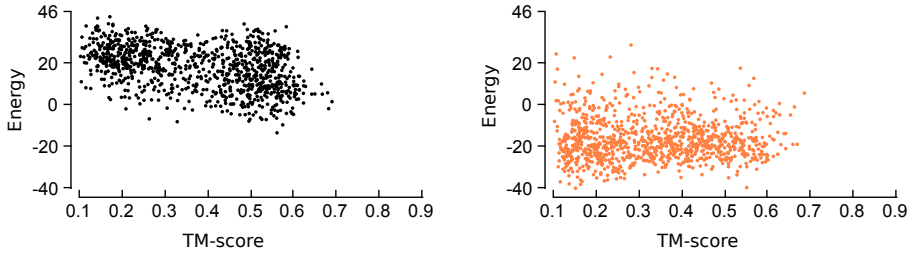
Next, we examined the impact of discontinuity on *ab initio* modeling and found that gaps in the fragment map have highly negative effects. We used an optimal fragment set, containing only a single-residue gap at an arbitrary loop position. Even with this simplified



(a) 3nzl (b) 2l01

(c) 2l02 (d) 2l09

Figure 3.9: Distribution of decoy TM-scores, built with HHfrag (black) and NNmake (orange) fragments. The decoys were generated with a modified Rosetta AbinitioRelax application. All resulting decoys were superimposed onto the native structure of the corresponding target using local fit and then the TM-score was calculated (higher is better). Structures with a TM-score of 0.4 or greater have correct fold.



(a) HHfrag (b) NNmake

Figure 3.10: Correlation between Rosetta energy and TM-score for decoys of target 2l02. The lowest-energy decoys, generated by each method, are shown in Figure 3.11.

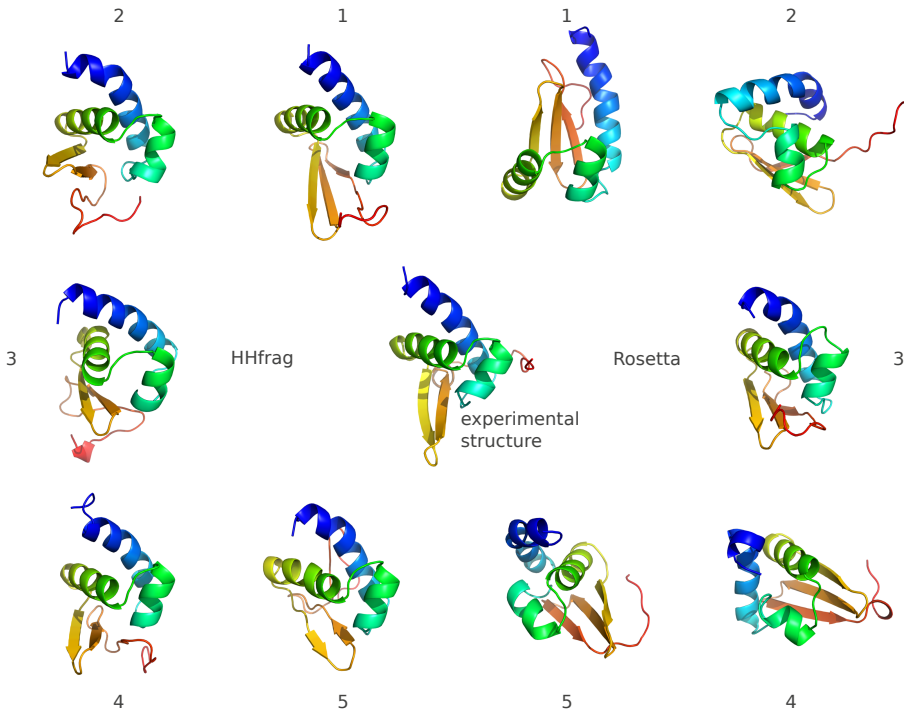


Figure 3.11: Lowest energy Rosetta decoys for target 2l02, built with HHfrag (left) and NNmake fragments (right). The native structure is shown at the center. The rank of each decoy is shown next to its structure. All decoys have been superimposed onto the native structure.

setup, the distribution of good decoys was shifted towards decoys of poor quality (41% decoys with correct fold). To work around this issue, HHfrag implements a "hybrid mode", in which gaps in the fragment map can be complemented with an arbitrary filling, such as structural alphabet prototypes or position specific NNmake fragments. Section 3.7 provides a more sophisticated complementation approach, which detects both unassigned and low-accuracy regions.

3.6.2 *Free modeling*

The performance of our high-precision dynamic fragment libraries was tested in Rosetta *ab initio* protein folding experiments (Figure 3.11; see also supplementary data of [51]). We modified the original Rosetta AbinitioRelax application to accept fragments of variable length and generated decoys for a subset of the proteins in our benchmark. Initially, we found 15 targets for which the BAKER-ROSETTASERVER [55] has submitted CASP9 models with comparable or better quality than HHpredA [44]. We picked the 11 shortest, single-domain targets of up to 150 residues to provide favorable input for Rosetta. 1000 decoys were generated for each target in this subset using default

parameters and stock NNmake fragments. In four cases Rosetta produced successful results under the following requirements:

- average TM-score [95, 96] to the native structure greater than the random (0.17);
- at least 2% of good decoys (TM-score > 0.4, i. e. models with correct fold).

The final benchmark set comprises of four targets: 3nzlA, 2l01A, 2l02A and 2l09A. 1000 decoys were generated for each target using variable length HHfrag libraries in place of the regular NNmake 9-mers, using the same set of input parameters and 3-mers. In all instances, we observed a positive shift in the distribution of TM-scores (Figure 3.9) and better energy funnels (Figure 3.10), confirming increased structure prediction accuracy (Figure 3.11). HHfrag shifts the position of the most populated TM-score bin and increases the fraction of good decoys (TM-score > 0.4) by 31, 26, 14 and 29%, respectively. Although the best decoys generated with both methods have practically the same TM-score, good decoys are produced 1.4–16.0 times more often when using dynamic HHfrag libraries.

3.7 FILTERING AND ENRICHMENT

Earlier in this chapter, we discussed how the local precision varies along the query sequence. We demonstrated the existence of high-accuracy regions, which generally correspond to conserved I-Sites motifs. The ability to identify those regions reliably holds a strong potential. Local and global structure prediction methods, which utilize fragments, may use this information to recognize high-quality, credible local structure predictions. Once identified, the most reliable fragments should receive higher weight and take precedence in the course of fragment assembly.

3.7.1 *The confidence score*

Dynamic fragment maps are composed of redundant fragments, covering identical target positions. Each fragment provides a list of torsion angles and C_α coordinate values, extracted from its parent PDB structure. All fragments, assigned to a given position, are at least partially overlapping. We can compute the structural divergence between every pair of fragments and build a corresponding adjacency matrix. Thus, for every target position we define a *fragment cluster*, which is a graph with the following properties:

1. *Recurrence* (r) — refers to the sequence conservation of a structural motif. The recurrence of a given fragment is measured

by counting the number of motif instances in a non-redundant database of templates (PDBS25). We compute the recurrence of a target position and its fragment cluster by simply counting the number of assigned fragments, covering this position, which is the number of vertices in the corresponding cluster.

2. *Consistency* (c) — reflects the structural homogeneity of the fragments. We measure the consistency of a cluster by calculating the percentage of structurally similar pairs of fragments. Two vertices are considered similar if the weight of the edge connecting them, measured by the C_α-RMSD distance between the two fragments, is lower than a critical threshold. As usual, we use a distance threshold set to 1.5 Å.

When these two properties are used in combination, we can obtain information about the reliability of the fragments, contained in a given cluster and covering a specific target position. This observation has several practical implications: (i) high-accuracy regions can be characterized quantitatively in terms of their fragment consistency and recurrence and (ii) each high-quality cluster can be compressed by exclusion of incompatible fragments. To quantify the goodness of a given fragment cluster, we compute the *confidence* of assignment C for its associated target position i:

$$C = c \log_{10} r = \frac{e}{|E|} \log_{10} |V| \tag{3.1}$$

where the recurrence $|V|$ is the number of fragments in the cluster, $|E|$ is the total number of edges and e is the number of edges below 1.5 Å. The recurrence term in this expression is a weighting factor. Highly conserved motifs like the *GD box* [3] have a recurrence of 50–100 or more and this impacts the confidence positively, multiplying the structural consistency term by a whole factor. Clusters of size greater than 10 are up-weighted since 10 is the critical number of HHsearch hits, below which the program switches to a less strict, greedy hit ranking algorithm [84]. At $r = 10$, the logarithm of the recurrence is 1 and the confidence is determined entirely by the degree of structural consistency. Clusters of size less than 10 are associated with increasing uncertainty and thus severely penalized. We can follow the same intuition to deduce the natural thresholds for the confidence metric:

1. $C < 0.8$: uncertainty. This confidence threshold is equivalent to a small fragment cluster (10 instances) at consistency equal to HHfrag's high precision for I-Sites (80%) or a highly recurrent motif (100 instances) at low precision (40%).

2. $C = 1$: transitional zone. Confidence value of 1 corresponds to a rare motif (10 instances) with maximum structural conservation

or a highly abundant motif (100 instances) at moderate consistency of 50%.

3. $C > 1$: credible local structure prediction, that is guaranteed to be accurate. For example, a confidence value of 1.4 can be obtained for a highly consistent cluster (70%) of size 100.

We measured the confidence values for all fragment clusters in the HHfrag benchmark and found good correlation between confidence and local RMSD (Figure 3.13). Detailed experimental confirmation of these predicted thresholds is presented in Section 3.7.3.

3.7.2 *The outlier rejection algorithm*

Each fragment cluster contains an arbitrary number of *outliers* — false positive fragments or fragment instances, whose local structure deviates from the canonical motif prototype. Here we introduce a greedy filtering algorithm, which is designed to identify and eliminate outliers reliably. This results in more homogeneous clusters, thus increasing the overall precision of our dynamic fragment libraries. The fragment with the lowest average distance to its adjacent vertices, termed the *centroid*, is finally selected as cluster's representative. By reducing each compact cluster to a single consensus fragment, HHfrag produces filtered fragment libraries of very low complexity. This property is highly desirable when fragment libraries are used for local structure prediction [57]. Non-redundant libraries however may also be used for very fast and efficient *ab initio* fragment assembly [51].

We represent each fragment cluster by a standard adjacency sets data structure, giving a space complexity of $O(V + E)$. The *BuildCluster* procedure (Algorithm 1) creates a sparse undirected graph G by computing the RMSD of each pair of overlapping fragments F, covering a given target position i. Each cluster keeps track of the total sum of pairwise distances (W); cluster members (v) also maintain an updated sum of all edges incident to them (W_v). Note that fragment clusters do not necessarily form complete graphs, because the sequence overlap between some pairs of fragments is shorter than required for a meaningful RMSD calculation.

The goal of the rejection algorithm is to enhance the structural consistency of a given cluster by performing a minimum number of vertex deletions. A given cluster is said to be *stable*, when the average RMSD distance between all adjacent vertices (D) is lower than the standard threshold of 1.5 Å:

$$D_s = \frac{1}{|E|} \sum_{(u,v)\in E} \omega(u,v) \leq 1.5 \tag{3.2}$$

The *RejectVertex* procedure (Algorithm 2) performs fragment rejections iteratively, until cluster stability is reached. On every iteration,

Algorithm 1 Cluster initialization.

```
1:  procedure BuildCluster(F, i)
2:      G = EmptyGraph()
3:      visited = ∅
4:      G.W = 0
5:      for each u ∈ F do
6:          if u.QStart ≤ i ≤ u.QEnd then
7:              for each v ∈ F do
8:                  if (v, u) ∉ visited then
9:                      visited = visited ∪ {(u, v)}
10:                     if u.QStart ≤ i ≤ u.QEnd then
11:                         if Overlap(u,v) ≥ 6 then
12:                             G.V = G.V ∪ {u, v}
13:                             δ = Distance(u, v)
14:                             G.Adj[u][v] = G.Adj[v][u] = δ
15:                             u.W = u.W + δ
16:                             v.W = v.W + δ
17:                             G.W = G.W + δ
18:                         end if
19:                     end if
20:                 end if
21:             end for
22:         end if
23:     end for
24:     return G
25: end procedure
26:
27: procedure Distance(u, v)
28:     return Cα RMSD of common residues
29: end procedure
```

we probe all vertices by calculating the average distance D'_v when vertex v is excluded from the cluster. This is given by the following greedy criterion:

$$D'_{opt} = \min_{v \in V} \frac{\sum_{e \in E} \omega(e) \; - \sum_{u \in N(v)} \omega(u, v)}{|E| - |N(v)|} \tag{3.3}$$

where $N(v)$ is the adjacency list of vertex v and $\omega(u, v)$ is the distance between fragments u and v. The vertex, whose exclusion leads to the most significant drop in D' towards stability ($D \leq 1.5$ Å), is selected for rejection and deleted. Each removal requires linear time of $O(|V|)$, needed to update all adjacency sets (linear complexity) and recalculate the sum of weights W_v of affected vertices (constant time per vertex). If no vertex removal is found to decrease the mean distance D, this cluster is not able to shrink further. Such clusters are said to be

Algorithm 2 Outlier rejection. See also Algorithm 3.

```
1:  procedure ShrinkCluster(G)
2:      while G.D > 1.5 and |G.V| > 1 do
3:          outlier = nil
4:          D'_opt = ∞
5:
6:          for each v ∈ G.V do
7:              D'_v = (G.W − v.W)/(|G.E| − |v.Adj|)
8:              if D'_v < D_opt then
9:                  outlier = v
10:                 D'_opt = D'_v
11:             end if
12:         end for
13:
14:         if D'_opt < G.D then
15:             RejectVertex(G, outlier)
16:         else
17:             Error("Diverging cluster")
18:         end if
19:     end while
20:
21:     return ComputeCentroid(G)
22: end procedure
```

diverging, which indicates heterogeneous aggregates of false positive fragments. The filtering process is terminated in such case, rendering the corresponding target position unassigned. The same negative result is also obtained in the event of cluster exhaustion before stability has been reached. The maximum number of iterations thus equals the number of vertices $|V|$. The amount of work performed on each iteration k is proportional to the number of nodes $|V_k|$ on iteration k (to compute the candidate for rejection) plus additional $|V_k|$ (to remove the candidate and update all adjacency sets and cached average distances). The worst-case running time of *RejectVertex* is hence given by:

$$\sum_{k=1}^{|V|} 2|V_k| = \Theta(|V|^2 + |V|) \tag{3.4}$$

However, most clusters reach stability much earlier than $k = |V|$ number of iterations, so the average running time is in practise better.

Once all outliers have been removed, the average RMSD distance in the cluster D_s is now 1.5 Å or less. We define the *representative* fragment as the centroid of the cluster (Algorithm 3), which is the vertex with the lowest average distance to all of its adjacent nodes. Since cluster vertices have variable number of incident edges, we require

Algorithm 3 Supporting procedures.

```
1:  procedure RejectVertex(G, outlier)
2:      G.V = G.V − {outlier}
3:      G.Adj = G.Adj − {outlier}
4:      for each v ∈ G.V do
5:          if outlier ∈ G.Adj[v] then
6:              G.W = G.W − ω(v, outlier)
7:              v.W = v.W − ω(v, outlier)
8:              G.Adj[v] = G.Adj[v] − {outlier}
9:          end if
10:     end for
11: end procedure
12:
13: procedure ComputeCentroid(G)
14:     centroid = nil
15:     D_min = ∞
16:     for each v ∈ G do
17:         if |G.Adj[v]|/|G.V| ≥ 0.5 then
18:             if v.D < D_min then
19:                 centroid = v
20:                 D_min = v.D
21:             end if
22:         end if
23:     end for
24:     return centroid
25: end procedure
```

the centroid to be a vertex, connected to a significant percentage of the nodes ($\geq 50\%$).

3.7.3 *Filtered fragment libraries*

The filtering algorithm and the confidence metric form the basis of an HHfrag extension, which is designed to build filtered fragment libraries of lower complexity and very high local precision. For each query position, HHfrag compiles a fragment cluster of all assigned fragments, as outlined in the previous sections. Incompatible fragments in each cluster are rejected and a single representative fragment per cluster is selected. After filtering out the outliers, the library is enriched with high-quality fragments. For high-accuracy regions, this always results in a centroid local precision of 100%, i. e. representative fragments in those regions are guaranteed to have compatible local structure. This is illustrated in Figure 3.12. After filtering the fragment library from Figure 3.4, we obtain a list of position-specific representative fragments. The local precision of the resulting library

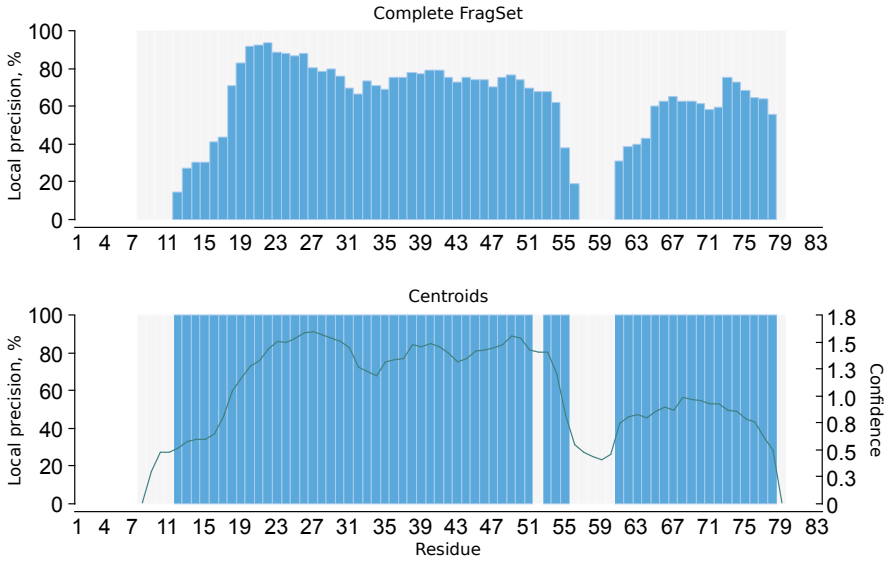


Figure 3.12: Local precision of representative fragments for benchmark target 3nzl. The fragment library, shown in Figure 3.4, was filtered using HHfrag's outlier rejection extension. The local precision of each representative fragment is shown as a bar, placed at its corresponding query position. The height of each bar was calculated as a binary measure: 100% if centroid's RMSD is at most 1.5 Å, 0% if greater. The confidence values for all query positions are shown as a green curve.

of centroids is 100% for all high-accuracy regions, observed on the original diagram (see Figure 3.12 for details on how this is calculated). The confidence curve correlates well with the observed local precision pattern, dropping rapidly in regions where inaccurate centroids have been selected. Similar results were obtained after filtering all remaining CASP9 targets from the standard HHfrag benchmark (per-target data available in [52], supplementary material).

Figure 3.13 shows the overall correlation between the local accuracy of all cluster centroids and their confidence in our benchmark. A weak confidence value (0.1–0.6) is a clear signal for the presence of a low-accuracy region. Higher confidence values (0.8–1.0) indicate generally conserved motifs, which sometimes cannot be predicted reliably. The overall centroid precision in this confidence interval is $80 \pm 17\%$ with an average RMSD to native structures of 1.0 ± 0.9 Å. Confidence greater than 1.0 guarantees an accurate and reliable local structure prediction with a very low chance for an error. The overall precision in such regions reaches $92 \pm 13\%$ with an average RMSD to native structures as low as 0.58 ± 0.57. These results confirm the theoretical confidence thresholds, derived in Section 3.7.1.

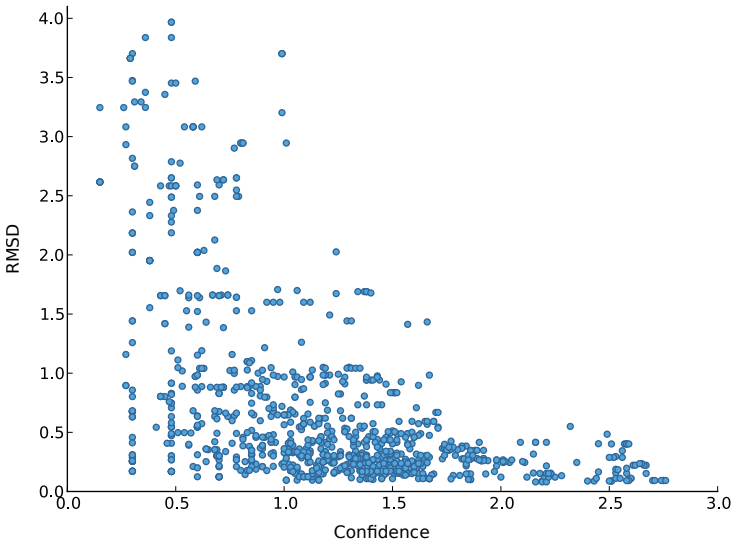


Figure 3.13: Reliability of the confidence metric. Shown is a correlation plot between the confidence of 1000 arbitrary fragment clusters and the C_α-RMSD to the native structures of their corresponding representative fragments.

3.7.4 *Confidence-guided prediction of torsion angles*

We use the filtered fragment libraries and their associated representative fragments for direct prediction of torsion angles from sequence. For each position i in a given query protein, we build a fragment cluster and compute the centroid fragment, as outlined earlier. The torsion angle pair (φ_i, ψ_i) of the representative fragment at target position i is extracted from centroid's experimental structure and directly reported as the final prediction at that position. Confidence values of 0.8 or greater indicate reliable predictions within a local region of high accuracy.

We used the familiar set of CASP9 [62] proteins to benchmark the accuracy of centroid-based torsion angle prediction. For each target, we obtained a prediction of its torsion angles with the procedure, described above. The prediction accuracy is measured by the mean absolute error (MAE) between the predicted (P) and experimental (E) torsion angle values:

$$MAE = \frac{1}{\sum_{i=1}^{N} L_i} \sum_{i=1}^{N} \sum_{j=1}^{L_i} |P_{ij} - E_{ij}| \tag{3.5}$$

where N is the number of proteins and L_i is the number of residues in protein i of confidence greater than a chosen cutoff ($C > x$). All predicted and experimental torsion angles are computed in degrees within the $[-180°, 180°]$ range. To keep the error values in that range as well, we apply the following rule when calculating the absolute angular errors $|AE_{ij}|$:

Method	Confidence	MAE (φ)	MAE (ψ)
TANGLE	0.8	31.9 ± 34.9°	90.7 ± 30.6°
ANGLOR	0.8	18.7 ± 25.8°	86.4 ± 43.0°
HHfrag	0.8	18.6 ± 27.0°	22.5 ± 36.2°
TANGLE	0.0	34.2 ± 36.4°	87.4 ± 32.3°
ANGLOR	0.0	23.5 ± 30.0°	84.7 ± 47.6°
HHfrag	0.0	25.4 ± 34.7°	34.9 ± 48.9°

Table 3.1: Torsion angle prediction performance. Shown are the mean absolute errors of φ and ψ torsion angle prediction for high-confidence ($C \geq 0.8$) and all residues ($C \geq 0$) in our benchmark.

$$|AE_{ij}| = \begin{cases} |AE_{ij} + 360| & \text{if } AE_{ij} < -180 \\ |AE_{ij} - 360| & \text{if } AE_{ij} > +180 \\ |AE_{ij}| & \text{otherwise} \end{cases} \tag{3.6}$$

We tested the performance of our centroid-based torsion angle predictor on 106 protein targets from the CASP9 competition [62]. The mean absolute errors (MAE) of predicted φ and ψ angles were compared against the values, obtained using two machine learning methods for torsion angle prediction from sequence: ANGLOR [94] and TANGLE [85]. The overall precision of HHfrag in comparison to these methods is reported in Table 3.1.

When regions of any confidence are considered, our method predicts φ angles with slightly lower accuracy than ANGLOR (2° higher MAE), but better than TANGLE. For ψ angles however, HHfrag is significantly more accurate, improving on both ANGLOR and TANGLE by a 50° lower MAE (Figure 3.14). The observed MAE of HHfrag is 25.4° for φ and 34.9° for ψ angles on average.

As expected, the quality of torsion angle prediction with HHfrag improves further when the confidence score of each query position is taken into account (Figure 3.15). In target regions of $C \geq 0.8$, the average MAE drops by 6.8° and 12.4° for φ and ψ angles respectively (Table 3.1). Generally, the MAE of HHfrag predictions gradually decreases as we discard regions of lower confidence (Figure 3.15). Such an improvement is less pronounced for φ angle predictions with ANGLOR (−4.8°) or TANGLE (−2.3°) and completely lacking when these methods are used to predict ψ angles (Table 3.1; Figure 3.15).

HHfrag does not always select optimal centroids in low-confidence regions ($C < 0.8$) as the lack of sufficient recurrence and consistency of such clusters hinders the filtering algorithm. However, in transitional zones ($C \geq 0.8$), the deviation from the optimal MAE becomes

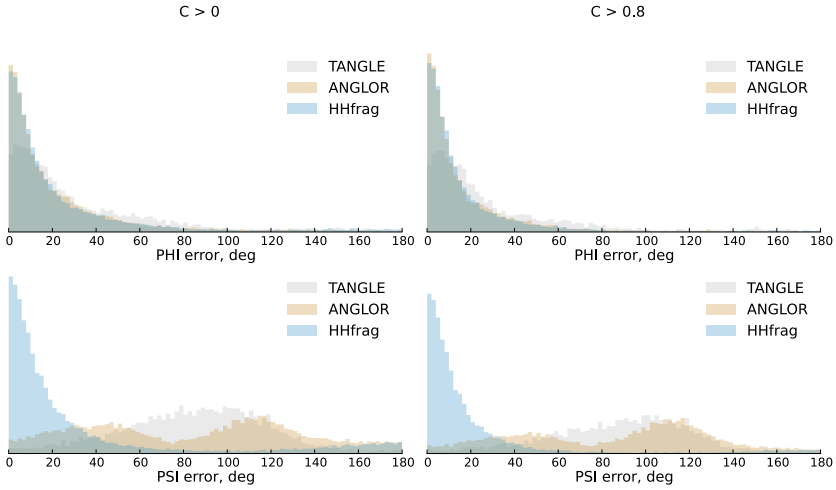


Figure 3.14: Distributions of the absolute errors of predicted torsion angles. Shown are the distributions of φ and ψ prediction errors for high-confidence (right) and all target residues (left) in our benchmark.

negligible and for high-confidence regions ($C \geq 1$) our method is guaranteed to extract torsion angles from the best-fitting fragment at each position (Figure 3.15). These results highlight the importance of taking the local conservation landscape into account and confirm the utility of our confidence-guided prediction strategy.

3.7.5 *Applications*

In the previous section we saw that the confidence metric is a reliable predictor for the locations of the high-accuracy regions, also called *high-confidence zones*. This enables client applications to use the confidence as a guide for reliability of local structure prediction and utilize fragments from confident zones with higher priority. Here we discuss some practical applications of filtered HHfrag libraries.

3.7.5.1 *Ab initio structure prediction*

In Section 3.6.1 we demonstrated that traditional *ab initio* fragment assembly methods such as Rosetta AbinitioRelax [72, 55] are designed for use with continuous fragment maps of maximum coverage. The interrupted nature of the HHfrag libraries may therefore be seen as a significant shortcoming. We resolve this incompatibility between HHfrag and Rosetta with the algorithm for preparation of hybrid, gapless fragment libraries. We start with a raw variable length fragment set, produced by a standard HHfrag run. As discussed earlier, HHfrag libraries demonstrate patterns of high-confidence motifs, connected by low-accuracy linkers (low number of assignments, mostly incompatible loop segments) or gaps (no assignments at all). Both

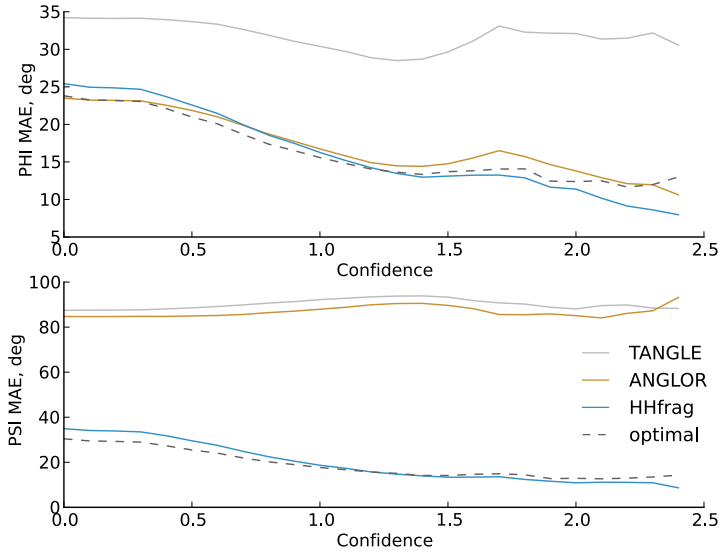


Figure 3.15: Torsion angle prediction accuracy at increasing confidence cutoffs. We measured the mean absolute error (MAE) of φ and ψ angle prediction at increasing confidence cutoffs. For each cutoff, we computed the φ and ψ MAE for all target residues in our benchmark, having a confidence greater or equal to the cutoff. The optimal curve shows the MAE calculated for the best-fitting fragments in each cluster.

unassigned and low-confidence regions can be envisioned as "gaps" in the fragment map, which require brute-force modeling. Using a confidence threshold of 0.6–0.7 (see Section 3.7.1 and Section 3.7.3 for justification), our algorithm can identify all low-accuracy query positions and mark them as eligible for complementation. All gaps are then filled with fragments from structural alphabets, chemical shift libraries (Chapter 4), or NNmake. Complementation with NNmake and CSfrag fragments has been implemented as a standard HHfrag extension in CSB (Section 6.3), which is designed to ensure compatibility of HHfrag with classic *ab initio* modeling applications (Section 6.3). We discuss the utility of this method in Chapter 5, where we describe a protocol for protein structure determination using hybrid fragment libraries.

3.7.5.2 *Local structure prediction*

The confidence metric provides a convenient and reliable framework for local structure prediction. We showed that cluster centroids, assigned to high-confidence regions ($C \geq 1$), are guaranteed to have accurate local structure. Client applications, such as ISD [70, 71], can use this information to increase the accuracy of local structure prediction. The CSB API [53], presented in Chapter 6, exposes the *Torsio-*

nAnglesPredictor class, which can be used to extract residue-wise torsion angle predictions from filtered HHfrag libraries and centroids, guided by the confidence score [52]. The centroid-based approach for torsion angle prediction from sequence is also implemented as an HHfrag extension in the latest version of the software (Section 6.3). We discuss a practical application of this method in Chapter 5, where the centroid-based torsion angle predictor is used to derive angular restraints for high-confidence regions in a protein structure determination protocol.

3.7.5.3 *Secondary structure prediction*

The fragment filtering procedure can also be used for three and eight-state secondary structure prediction. The most successful strategy in our experience involves a combined approach. In low-confidence regions ($C < 0.8$), we take a standard secondary structure prediction (PSIPRED [49]). In regions with higher confidence, we compute the consensus secondary structure among all fragments, survived the library filtering procedure.

3.8 CONCLUSION

In this chapter, we introduced the static and dynamic methods for fragment-based local and 3D structure prediction. We explained how the PDB database can be used as a comprehensive source of conserved supersecondary motifs, shared among proteins from different folds. Using the most sensitive methods for sequence profile comparison, such motifs can be detected and compiled in customized fragment libraries, suitable for *ab initio* protein structure prediction.

Building upon these concepts, we developed HHfrag — a novel method for profile HMM-based fragment detection, designed to combine the strengths of earlier static and dynamic approaches, while at the same time addressing their common limitations. HHfrag is the first method capable of detecting fragments of variable length and gapped nature, which leads to a significant improvement in local structure prediction accuracy. We showed that our method achieves a good balance between coverage and precision, improving the accuracy of fragment detection over dynamic methods like Rosetta NNmake, at the expense of $19 \pm 15\%$ loss in sequence coverage. Although the presence of unassigned fragment map regions is a disadvantage for traditional *ab initio* modeling, we showed that unassigned regions are usually part of unconserved segments, which need special treatment (i. e. sequence-based local structure prediction is not possible due to the lack of sequence conservation). We demonstrated that the locations of conserved motifs in a protein sequence can be predicted by examining the recurrence and structural homogeneity of detected fragments. The resulting confidence score correlates well with the

local RMSD of the representative fragments and allows prediction of torsion angles from sequence with better accuracy than existing machine learning methods. The ability to discriminate between low- and high-accuracy zones, along the development of filtered and low-complexity fragment libraries, opens interesting possibilities for use of HHfrag in local structure prediction and NMR structure determination.

Finally, the advantages of using libraries, enriched with high-quality fragments, were demonstrated in Rosetta *ab initio* folding experiments. By substituting the standard NNmake fragment detection module of Rosetta with HHfrag, we demonstrated that our dynamic fragment libraries improve the performance of traditional *ab initio* protein structure prediction. We observed enrichment of high-quality decoys, accompanied by faster sampling and improved energy funnels. This improvement is attributable to the use of dynamic fragment libraries of greater precision. We showed that HHfrag's ability to capture the contextual variability of detected motifs is one of the main contributing factors in this direction.

However, structure prediction by fragment assembly is not the only field of application of HHfrag. Fragments can be useful for many purposes. For example, fragment-based methods have been used in the recent structure determination of mitochondrial uncoupling protein 2 [4]. For such reasons, the main focus of our study was the development of a broader, general-purpose framework for accurate local protein structure prediction from sequence. In the following chapters we will see how HHfrag fragment libraries can be used in combination with sparse and low-quality experimental data for NMR structure determination.

4

ANALOGOUS FRAGMENT LIBRARIES

4.1 INTRODUCTION

HHfrag belongs to the family of traditional sequence-based methods for local structure prediction. All methods in this class rely on the observation that many reusable structural motifs in fact demonstrate a degree of sequence conservation. Our ability to detect these scarce sequence signals has recently improved significantly, thanks to the development of algorithms for pairwise alignment of profiles and HMMs (2.2) [84]. These algorithms have reached a level of sensitivity, sufficient enough to ensure detection of virtually all recurrent structural motifs, observed in experimental structures across different folds. It is therefore justified to assume that the identification of conserved supersecondary structures poses no principal challenge to modern dynamic fragment detection methods. However, the expectation that protein structures are simply combinations of conserved motifs has not seen experimental confirmation. While reusable sequence motifs clearly do exist [10], they rarely span protein sequences in their entirety. This important observation was discussed in detail as part of HHfrag's framework for filtering and enrichment (3.7). We showed that dynamic fragment libraries have a non-uniform precision along the query sequence, alternating between high- and low-confidence regions or gaps. Our analysis has previously confirmed that these are typically regions where no sequence-based fragment detection will ever succeed: loops, linkers or unstructured termini [51]. This is a fundamental limitation of all methods for fragment detection based on sequence profiles, which stems from the fact that some sequence regions are too variable, not part of regular secondary structure and therefore not instances of reusable structural motifs. Such regions have little chance to receive reliable local structure prediction.

Earlier we saw that fragment map interruptions have a strongly negative effect on the performance of Rosetta [72] fragment assembly (3.6.1). A high level of sequence coverage is in general a desirable property for any fragment library, regardless of its purpose. But how can we increase the coverage of a given fragment library, if some regions of the query cannot be detected, because they are naturally not part of conserved, remotely homologous motifs? To address this issue, we need to incorporate additional experimental information. Chemical shift data, obtained in NMR experiments, is an ideal candidate for this practical purpose. The correlation between local structure in proteins and secondary chemical shifts is a well-known phenomenon

(2.5.1). Algorithms for alignment of chemical shifts have been used for template selection [34], thus revealing their potential for detection of analogous structures. Recently, methods for *ab initio* fragment assembly have been successfully combined with chemical shift data, resulting in novel approaches to NMR structure determination [78, 80, 17]. This suggests that chemical shift data may be used for analogous fragment detection as a supplement to the inherently more reliable remote homology-based approach. When combined with sequence-derived fragment libraries, analogous fragments can be used to fill any gaps exposed in low-confidence regions and increase the overall coverage of the fragment libraries.

To address the limitations of sequence-based methods for fragment detection, we developed CSfrag — a method for construction of analogous fragment libraries, based on chemical shift similarity detection. This chapter begins with the derivation of a chemical shift scoring function, used to detect structural fragments with similar chemical shift patterns. The PDB library [5] contains an insufficient number of NMR structures at present; however, recent developments have made chemical shift prediction possible [42]. By matching experimental shifts of query segments against predicted shifts for PDB templates, CSfrag collects compatible fragments with analogous structure and compiles a fragment library. We also describe the algorithm behind new HHfrag extensions, designed to complement gaps and low-confidence regions in traditional HHfrag libraries with chemical shift (CS) fragments.

4.2 THE CHEMICAL SHIFT SCORING FUNCTION

We begin with a detailed description of our chemical shift scoring model and its derivation.

4.2.1 *Formal definition*

To evaluate the similarity between two structural segments of equal length L, we first calculate their *cumulative* chemical shift score C, which is the sum of all *pairwise* chemical shift scores for all L positions:

$$S(query, subject) = \sum_{i=1}^{L} \sum_{n \in \{nuclei\}} S_n(\Delta\bar{\delta}_{i,n}) \tag{4.1}$$

where $\Delta\bar{\delta}_{i,n}$ is the difference in secondary chemical shift values for nucleus n between the *query* and *subject* segments at alignment column i:

$$\Delta\bar{\delta}_{i,n} = \bar{\delta}_{i,n}^{\,query} - \bar{\delta}_{i,n}^{\,subject} \tag{4.2}$$

We consider the chemical shifts of 5 nuclei: CA, CB, C, N and HA. By convention, the chemical shift differences $\Delta\bar{\delta}_{i,n}$ are always calculated by subtracting the secondary shifts of the query from those of the subject. All secondary shifts are computed directly from their raw values by following the DANGLE approach [18]. We define the pairwise score S_n for a pair of residues q_i, s_i and a given nucleus type n as the likelihood ratio:

$$S_n(\Delta\bar{\delta}_{i,n}) = \log_2 \frac{P_n(\Delta\bar{\delta}_{i,n}|pos)}{P_n(\Delta\bar{\delta}_{i,n}|neg)} \tag{4.3}$$

i.e. the probability of observing a secondary shift difference $\Delta\bar{\delta}$ as part of a true positive match, divided by the corresponding probability for a negative (random) pair of residues. We take the logarithm base 2 of this ratio, thus the pairwise score is measured in bits. This is a familiar concept, resembling the scoring of amino acid pairs in classic sequence alignment [25]. The log-odds ratio is expected to be greater than zero for true positive pairs and lower than zero for true negatives. The total score for two structural segments of length L is obtained by summation of the pairwise bit-scores over all segment positions i and nuclei n (Equation 4.1). Segments with compatible structure are therefore expected to have positive total scores, while random matches and mismatches in general will lean towards negative values of the total chemical shift score C.

The following section outlines the derivation of the P_n densities.

4.2.2 *Model estimation*

To calculate the positive- and negative-pair probabilities P_n, we collected secondary shift differences from a large set of experimental protein structures and evaluated their empirical distributions. Experimental chemical shift data was taken from the latest release of the VASCO database [90, 69]. There are 408 VASCO entries included in the non-redundant PDB database PDBselect25 [35]. Recall that the same database has been previously used as the basis of PDBS25-HMM — HHfrag's library of fragment templates (3.3.1). We refer to the intersection between VASCO and PDBS25 as the Non-redundant VASCO (nrVASCO) database. Predicted chemical shifts for the entire PDBS25 library were additionally computed using the SHIFTX2 method [42].

To build the empirical distributions P_n, we calculated *experimental* versus *predicted* secondary shift differences $\Delta\bar{\delta}_n$ for each nrVASCO chain. This resulted in 5 sets of secondary shift differences and thus 5 different empirical distributions, one for each nucleus type (CA, CB, N, C and HA).

Positive residue pairs were extracted from structurally similar proteins — homologous or analogous — taken from our non-redundant

PDBS25 database. For every nrVASCO protein, we simply collected the full list of its DALI [47] structural neighbors, discarding those not found in PDBS25. Each structural neighbor was then aligned against its matching nrVASCO protein using TMalign [96]. Positive secondary shift differences were finally obtained from all *aligned* residue pairs part of alignments with a TM-score > 0.6.

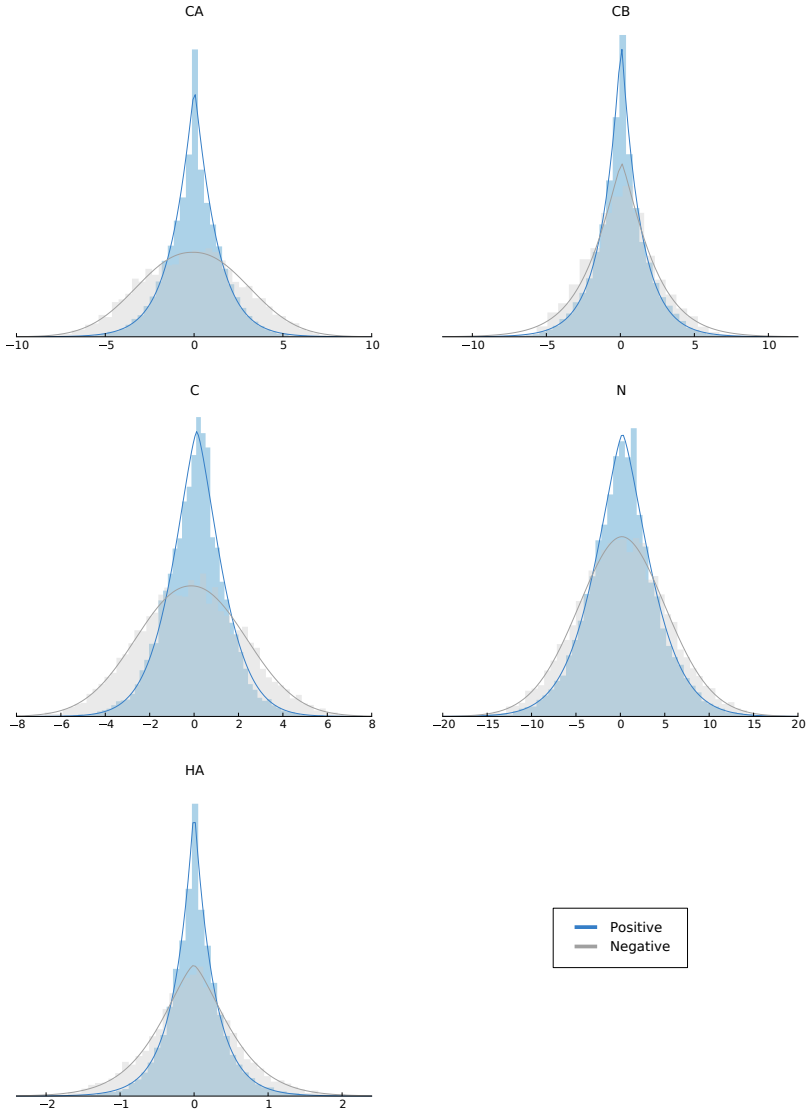


Figure 4.1: Empirical distributions of secondary chemical shift differences $\Delta\overline{\delta}_n$ for each nucleus type n. *Positive* is the distribution of matching (structurally similar) pairs $P_n(\Delta\overline{\delta}_{i,n}|pos)$ and *Negative* indicates mismatching (random) pairs — $P_n(\Delta\overline{\delta}_{i,n}|neg)$. The probability density functions of the corresponding Generalized Normal fits are shown as solid curves.

Negative residue pairs were extracted from random structural alignments. For each nrVASCO entry, we obtained structural alignments

Nucleus	Positive			Negative		
	μ	b	β	μ	b	β
CA	0.02	1.32	1.1	-0.08	4.23	2.2
CB	0.06	1.32	1.0	0.08	2.41	1.2
C	0.12	1.52	1.4	-0.13	3.42	2.1
N	0.23	4.39	1.4	0.17	7.08	1.9
HA	0.00	0.27	1.0	-0.01	0.66	1.4

Table 4.1: Estimated parameters of the Generalized Normal distribution for secondary shift differences $\Delta\bar{\delta}$. *Positive* is the distribution of matching (structurally similar) pairs and *Negative* indicates mismatching (random) pairs.

with 5 randomly selected PDB25 structures (TM-score < 0.2) and extracted negative secondary shift differences from all *unaligned* residue pairs.

The empirical distributions of positive and negative secondary shift differences are shown in Figure 4.1. In all instances, the histograms resemble heavy-tailed distributions, such as the Laplace distribution, but also demonstrate some Gaussian properties. These two models are combined in the flexible Generalized Normal distribution, which was found to approximate the empirical data well. Its Probability density function (PDF) is given by:

$$p(x|\mu, b, \beta) = \frac{\beta}{2b\Gamma(1/\beta)} e^{-(|x-\mu|/b)^{\beta}} \tag{4.4}$$

This PDF has three parameters:

- μ: location; this is the *median* of the chemical shift differences of the corresponding nucleus type;
- β: shape;
- b: scale; defined as:

$$b = \sqrt{\frac{\Gamma(1/\beta)}{\Gamma(3/\beta)} * \frac{1}{n} \sum_{1}^{n} (x_i - \mu)^2} \tag{4.5}$$

At $\beta = 1$, this density takes the form of a Laplace distribution and the Gaussian is defined at $\beta = 2$. We can estimate its parameters using the maximum-likelihood method. The estimated parameters for all 10 distributions of positive and negative pairs are summarized in Table 4.1 and the agreement between the estimated model and the empirical data is depicted in Figure 4.1.

4.2.3 *Performance*

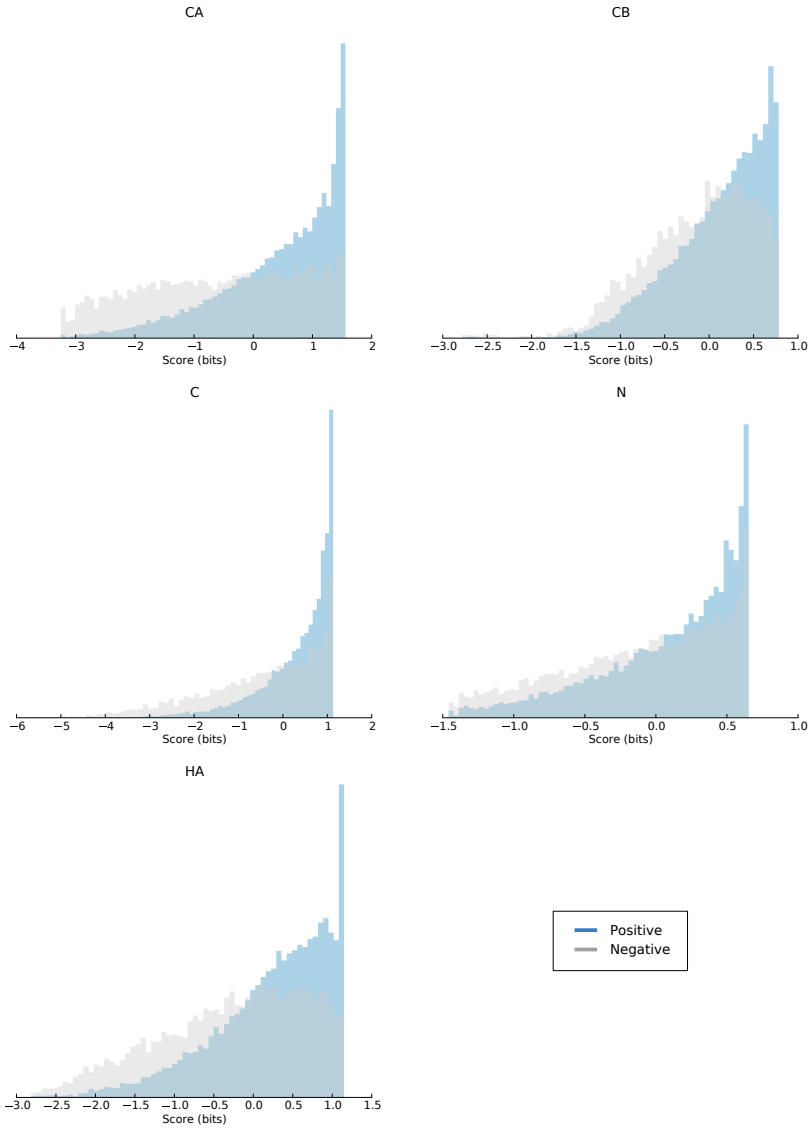


Figure 4.2: Distributions of the pairwise chemical shift score S_n for each nucleus type n. *Positive* is the score distribution of all matching (structurally similar) residue pairs from Figure 4.1 and *Negative* indicates mismatching (random) pairs. The average bit-score gain per residue for each set is given in Table 4.2.

The empirical distributions of positive (i. e. structurally compatible) residue pairs are generally more narrow than the corresponding negative distributions. The difference is more pronounced for CA and C nuclei, followed by HA. This suggests that the chemical shifts of CA and C nuclei will have significantly larger contribution to the total chemical shift score. To measure the ability of our scoring model to discriminate between structurally similar fragments, we evaluated all

positive and negative shift differences using the pairwise score from Equation 4.3. Thus, for each chemical shift type n, we computed the log-odds ratio $S_n(\Delta\bar{\delta}_n)$:

$$S_n(\Delta\bar{\delta}_n) = \log_2 \frac{P_n(\Delta\bar{\delta}_n|\mu^+, b^+, \beta^+)}{P_n(\Delta\bar{\delta}_n|\mu^-, b^-, \beta^-)} \tag{4.6}$$

where the model parameters, marked with "+", are the estimated parameter values for *positive* pairs and "-" indicates *negative*. The distribution of all pairwise scores, obtained in this way, is shown in Figure 4.2. Structurally similar residue pairs tend to achieve positive pairwise scores, while the score distributions for true negative pairs are shifted to the left and are less steep. These figures also confirm the initial observation that CA and C, followed by HA nuclei, have stronger potential to distinguish between compatible and structurally dissimilar residue pairs. The exact contributions by all chemical shift types are shown in Table 4.2.

Nucleus	Positive	Negative	Difference
CA	0.421	-0.535	0.956
CB	0.110	-0.125	0.235
C	0.332	-0.462	0.794
N	0.095	-0.099	0.193
HA	0.227	-0.265	0.492
sum	1.185	-1.486	2.670

Table 4.2: Total bit-score gain by chemical shift type. The *Positive* column contains the average of all pairwise bit-scores, obtained for residue pairs from similar 3D structures. *Negative* indicates the average bit-score for mismatching (random) pairs.

On average, true positive CA and C chemical shifts generate 0.4 and 0.3 bits per residue, while HA atoms contribute with roughly half of this amount (0.2). Considering all chemical shift types, the average yield per positive residue pair is greater than zero (1.185 bits), while structurally dissimilar residue pairs have a negative average bit-score (-1.486).

4.3 ANALOGOUS FRAGMENT PICKING

The chemical shift scoring model forms the basis of CSfrag — our method for detection of fragments with compatible (analogous) local structure. This is done by matching the experimental chemical shifts of the query protein against predicted chemical shifts of known protein structures. Sequence segments with similar chemical shifts are

expected to have similar local structure, so the structural fragments extracted in this way can be used to fill the gaps in remote homology-based fragment libraries. Detected fragments may have either homologous or analogous nature; their sequence similarity is irrelevant and therefore not measured.

4.3.1 *Preparation*

CSfrag uses the familiar non-redundant database of experimental structures as a source of fragments (Section 3.3.1). Recall that PDBS25 is derived from PDBselect25 [35], a database of PDB chains filtered at 25% sequence identity. We have found that this database represents the entire diversity of fragments in PDB and thus can be used as a PDB substitute for faster searching. However, less than 10% of all PDBS25 structures have been determined by NMR spectroscopy and provide chemical shift data. For such reasons, we need to approximate the chemical shifts for all remaining proteins by obtaining a prediction with SHIFTX2 [42]. Each PDBS25 entry is therefore characterised by:

1. experimental structure;
2. list of chemical shifts (CA, CB, C, N and HA).

All raw chemical shifts are converted to *secondary* shifts by subtracting them from the corresponding random coil reference values. This is done using the DANGLE method [18], as implemented in the *RandomCoil* CSB API [53]. More specifically, we compute sequence-corrected secondary chemical shifts, by subtracting each raw shift from the random coil value and then applying a sequence context-specific correction within a window of ± 2 residues.

4.3.2 *Fragment extraction*

The fragment extraction algorithm slices the query into a nested array of segments and matches them exhaustively against all templates in PDBS25. The simplest implementation uses a sliding window of short size, typically 7, 8 or 9 residues; we have identified 7 as a very good candidate. More sophisticated, variable-length fragment search is trivially implemented using a standard dynamic programming algorithm for sequence alignment, where the scoring matrix is substituted with the chemical shift scoring model from Section 4.2. As discussed earlier, the pairwise bit-score score is negative for "mismatches" and positive for "matches", which makes it ideal for inclusion in a sequence alignment algorithm. In this chapter, we stick to the sliding window approach. This is only a proof-of-concept implementation, i. e. we use a sliding window to simplify the interpretation of the results by excluding any confounding factors. However,

all practical implementations of CSfrag should prefer the alignment approach.

The sliding window-based implementation of CSfrag is straightforward. We obtain a nested array of all query segments of size 7, extract their experimental chemical shifts and compute corresponding secondary shifts for each nucleus type n in {CA, CB, N, C, HA}. For each segment, we use a sliding window of segment's length to scan every chain in PDBS25 for matching predicted secondary shifts. This is done by computing the experimental minus predicted secondary shift differences $\Delta\bar{\delta}_{i,n}$ on every position i in the 7-mer window, for each nucleus type n (where available). The total chemical shift score S_{seg} for this segment is computed from the extracted differences $\Delta\bar{\delta}$ according to Equations (4.1) and (4.6), where the Generalized Normal model is initialized with the estimated parameters from Table 4.1. The top 50 segments, whose score is greater than a cutoff, are kept as candidates. Finally, CSfrag collects all surviving candidates, orders them by score and builds a position-specific fragment library in Rosetta NNmake format [72, 55].

To define a meaningful chemical shift score cutoff, we used the mean bit-score yield of structurally similar residue pairs from Table 4.2. On average, positive pairs generate 1.034 bits per residue, considering all 5 chemical shift types. Chemical shift score cutoffs greater than $1.0 \times L$ (i. e. 7 bits for 7-mer fragments) are therefore good candidates. The standard cutoff in CSfrag is set to 1.1 bits per residue.

4.3.3 *Gap-filling with analogous fragments*

As discussed earlier, low-confidence and unassigned regions in HHfrag fragment maps correspond to unconserved segments, where remote homology detection efforts are in vain. This issue can be addressed by complementation of those regions with chemical shift fragments.

For regular HHfrag fragment maps, we propose the confidence-guided complementation procedure as already implemented for NNmake fragments (Section 3.7.5). We measure the confidence of each cluster along the query sequence and mark for complementation all positions with confidence of $C < 0.8$. All low-confidence positions are then filled with analogous fragments, centered around each marked position. This procedure ensures a very high coverage at the expense of a possibly reduced global precision.

Filtered HHfrag libraries are best complemented using a double filtering strategy. We first filter the analogous library using the standard outlier rejection algorithm (Section 3.7.2). Filtered chemical shift fragments are then mixed with the raw HHfrag fragment map. The mixed library is filtered once again to produce a final filtered frag-

ment map. This method increases the coverage and simultaneously preserves and even enhances the precision of filtered HHfrag libraries (Figure 4.4).

The gap-filling and double-filtering protocols are implemented in the latest version of HHfrag and CSfrag, available as part of the CSB toolbox (Section 6.3).

4.4 BENCHMARK

The performance of CSfrag libraries was evaluated on a set of 22 CASP9 targets. This is the number of HHfrag benchmark proteins (Section 3.5), for which experimental chemical shifts are available in VASCO [90, 69]. We used the same performance metrics as earlier — global precision, local precision and coverage (Section 3.5). For each target, we computed a constant-length library of analogous 7-mers, using the CSfrag procedure from Section 4.3.2. Next, we applied the double filtering algorithm from Section 4.3.3 to combine the existing HHfrag dynamic libraries with analogous fragments and once again measured the coverage and the precision of the resulting complemented fragment sets.

The sequence coverage of analogous libraries is very high and resembles the values, which we have seen for NNmake ($88 \pm 6\%$ over the 22 CASP9 targets). On average, CSfrag covers $90 \pm 14\%$ of the target residues. When all 50 fragments per position are considered, the coverage is close to optimal. Chemical shift-derived fragment libraries are therefore ideal candidates for complementation of the gapped HHfrag fragment maps, thus achieving their main design goal.

The chemical shift similarity, however, is a less reliable indicator of structural conservation. While sequence relatedness is usually a strong indicator of spatial similarity, matching secondary shift patterns do not necessarily translate to identical 3D structures. Similar chemical shifts generally imply similar secondary structure, but this does not always suffice to guarantee identity of two supersecondary structures in 3D, because the geometry of some fragments is determined by mid-range contacts, formed between individual secondary structure elements. The lack of sufficiently large number of experimental structures solved by NMR spectroscopy imposes the need to use crystal structures with predicted chemical shifts. This is an additional factor, which affects the precision of this method negatively.

For such reasons, analogous libraries, compiled with CSfrag, have lower precision than their HHfrag counterparts. The average precision of CSfrag is $36 \pm 14\%$ — a number, almost identical to the one obtained for NNmake (Section 3.5). These results make intuitive sense, given the observation that secondary shifts are known to correlate with secondary structure. This is further confirmed by examination of the local accuracy of CSfrag (Figure 4.3). The local precision pat-

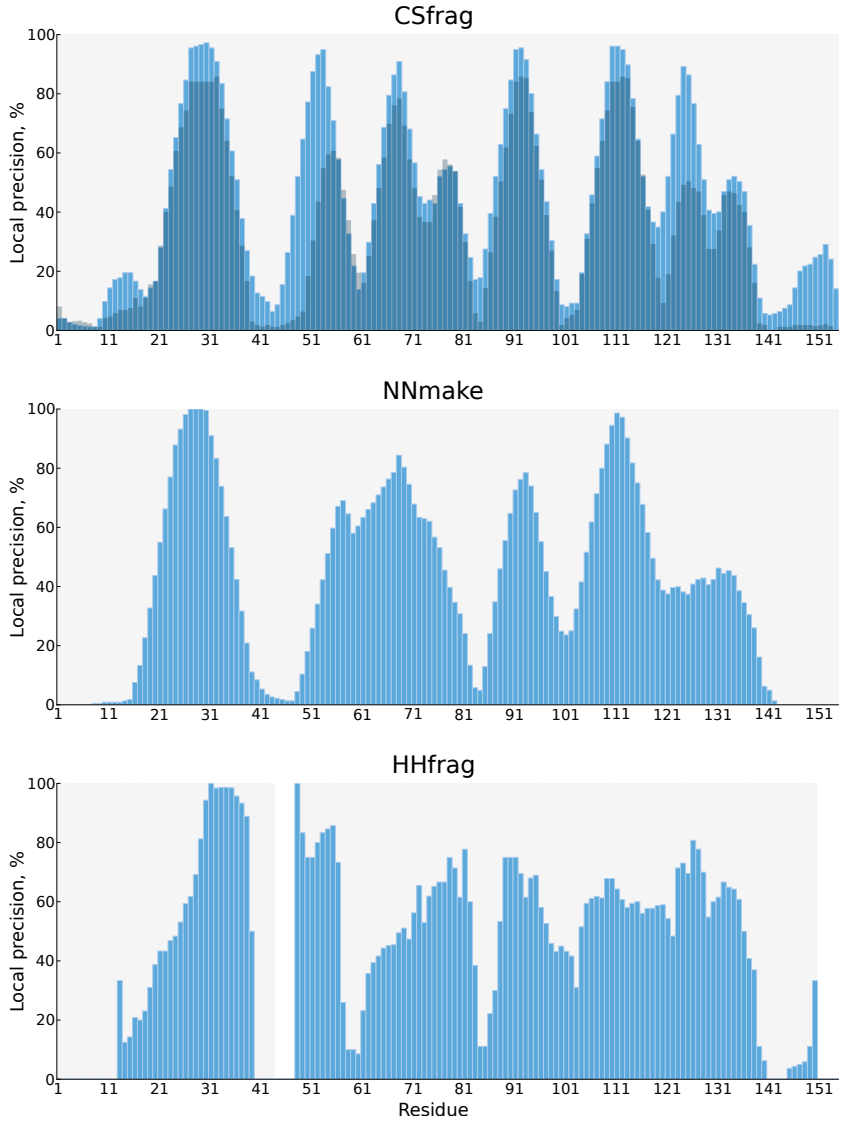


Figure 4.3: Local precision of CSfrag, NNmake and HHfrag libraries, compiled for target 2l06 at standard cutoff (1.5 Å). Each blue bar indicates the percentage of true positive fragments, which cover a given query residue. The grey background corresponds to the false positive rate, and the white regions are completely unassigned. The dark overlay on the first diagram represents the local precision of a 7-mer fragment library, compiled by matching the predicted secondary structure of the target against the computed secondary structures of all PDBS25 templates.

terns strongly resemble the ones, observed for NNmake and pure secondary structure-based fragment detection. However, CSfrag obtains higher global precision than the simple method of matching predicted versus computed secondary structure. This confirms the utility of using chemical shift-based detection of analogous fragments and justifies the added computational time.

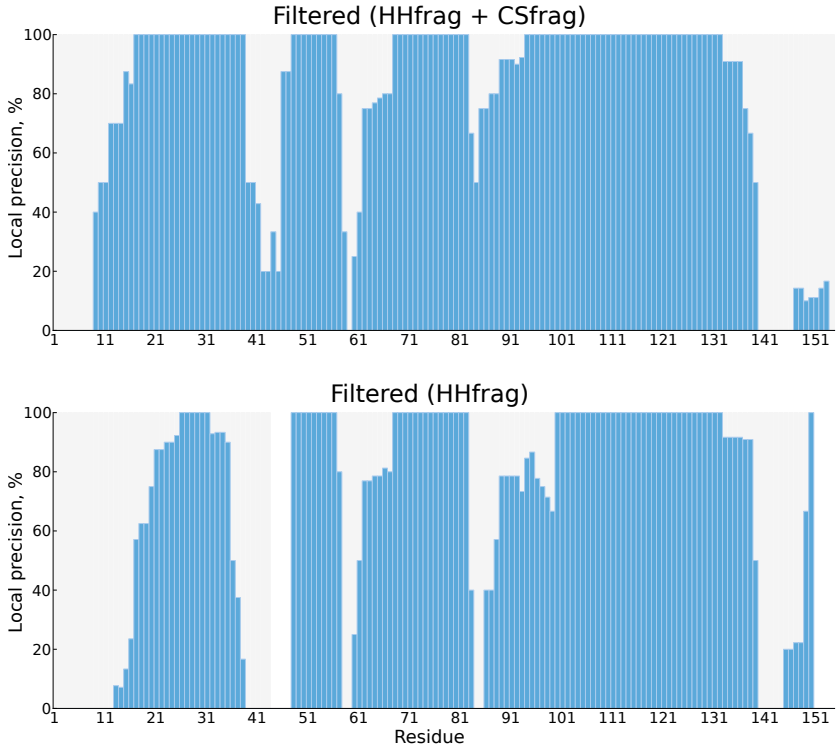


Figure 4.4: Local precision of the complemented double-filtered library for target 2l06. Shown are the standard filtered HHfrag library (bottom) and complemented, double-filtered library (top), prepared using the algorithm from Section 4.3.3. Note how the addition of analogous fragments and the double filtering approach lead to the closure of gaps at positions 40–50 and 80–90 and simultaneously increase the precision of the filtered library.

We found that analogous fragments can also be used to aid the preparation of filtered fragment libraries using HHfrag's filtering extension. However, chemical shift-derived libraries were shown to contain high rates of false positives, which hinders the filtering process. We address this issue with the double filtering algorithm, described in Section 4.3.3. Considering only the 22 NMR targets from CASP9, filtered HHfrag libraries achieve an average precision of $50 \pm 15\%$ and a mean coverage of $60 \pm 15\%$. After complementing the raw HHfrag libraries using double filtering, the average precision increases to $62 \pm 12\%$ and the coverage reaches $77 \pm 11\%$. While the double filtering procedure does not retain CSfrag's near-complete coverage, it increases the coverage of our high-precision filtered libraries significantly (see Figure 4.4 for a specific example).

4.5 CONCLUSION

We discussed the use of experimental NMR data for fragment detection. Secondary chemical shifts have been found to correlate with

local structure and this property can be exploited to extract analogous fragments with compatible structure. Building upon this concept, we developed a chemical shift scoring model, which can be incorporated in alignment algorithms and used to measure the similarity of structural fragments. Our fragment detection method, called CSfrag, compares segments of experimental chemical shifts against a non-redundant database of predicted chemical shifts. High-scoring segments from experimental structures are then excised to build a position-specific library of analogous fragments.

Analogous fragment libraries have significantly lower precision than the remote homology-based fragments, detected with HHfrag. However, the chemical shift matching approach has a clear advantage in low-confidence regions, because it does not rely on motif sequence conservation. Therefore, the analogous fragment libraries compiled with CSfrag are especially useful in HHfrag's procedure for complementation of regions of low accuracy and fragment map gaps. Recall that fragment map interruptions are highly undesirable when a given library is used in traditional *ab initio* fragment assembly. To address this problem, we proposed a simple and efficient method for gap-filling with analogous fragments. In addition, a flexible algorithm allows HHfrag to incorporate analogous fragments during the filtering phase, improving the coverage and precision of filtered fragment libraries significantly.

5

NMR STRUCTURE DETERMINATION WITH HHFRAG

5.1 INTRODUCTION

The fragment assembly approach to protein structure prediction is a flexible framework for 3D modeling, whose practical applications extend beyond the boundaries of *ab initio* structure prediction from sequence. Fragment libraries can be used in conjunction with experimental NMR data for fully automated protein structure determination [78, 80, 17]. This method relies on using traditional sequence profile fragments of detectable remote homology in combination with analogous fragments with compatible chemical shifts [38]. The process of structure calculation from homologous or analogous fragments is in fact identical to conventional *ab initio* structure prediction using fragment libraries. Differences may rather arise in the way the generated structures are scored and validated against the experimental data. The question of fragment optimality, however, is currently an underrepresented topic in this line of research. As a result, potentially useful contextual information such as fragment length variability, degree of local motif conservation and long-range contacts between pairs of co-occurring fragments may currently be unexplored.

Here we describe an application of our local structure prediction framework (Chapter 3, Chapter 4) in NMR structure determination from sparse data sets. Our method relies on HHfrag [51] as a source of remotely homologous fragments of variable length, which capture the contextual variability of detected motifs and their actual sequence boundaries accurately. We utilize our confidence-guided framework for local structure prediction [52] to identify regions of local motif conservation and incorporate this information as a valuable additional constraint in the structure calculation protocol. To increase the fragment library coverage at regions where no conserved motifs can be detected, we use the chemical shift-scoring facilities of CSfrag for extraction of analogous fragments with compatible structure. We conclude this chapter with practical examples of successful protein structure determination with this approach in combination with classic Rosetta *ab initio* fragment assembly [72].

5.2 STRUCTURE CALCULATION FROM SPARSE DATA

Our structure calculation protocol comprises of the following components: (i) dynamic HHfrag libraries of variable fragment length,

complemented with analogous chemical shift fragments; (ii) hybrid HHfrag centroids as a source of residue-wise torsion angle restraints in high-confidence regions; and optionally (iii) experimental 3D-NOE spectra as a source of distance restraints. Details on each individual component of the structure calculation protocol are provided next.

5.2.1 *Hybrid fragment libraries*

Hybrid fragment libraries were prepared using the gap-filling algorithm, described earlier (see Section 3.7.5 and Section 4.3.3).

For a given protein target, we first compiled a standard variable-length fragset with HHfrag. Recall that HHfrag [51] uses information from sequence profiles and predicted secondary structure to extract remotely homologous motifs from a non-redundant subset of the PDB database [5].

Next, we prepared a library of analogous 7-mers using assigned experimental chemical shifts as input. CSfrag computes secondary shifts from the input values for CA, CB, C, N and HA nuclei and scans the HHfrag database of templates (PDBS25) for 7-mer segments with matching predicted secondary shifts. We kept at most 25 of the best scoring fragments per starting position, having a cumulative chemical shift score of at least 7.7 bits (1.1 bits per residue). The complexity of the resulting library of analogous 7-mers was reduced by running CSfrag in filtering mode, which produces a compact library of centroids (using the familiar fragment clustering algorithm from Section 3.7).

The chemical shift-derived centroids were finally used as a filling for confidence-guided complementation with HHfrag (Section 3.7.5). Dynamic fragment map regions of confidence $C \leq 0.7$ were considered unreliable and complemented with CSfrag centroids to achieve a near-complete sequence coverage, without simultaneously increasing the complexity of the library significantly. We used the resulting hybrid library along with a modified Rosetta *ab initio* in place of the standard 9-mer fragset.

5.2.2 *Angular restraints*

Each hybrid fragment library was additionally used as a source of torsion angle restraints. The idea is to restrict the degrees of freedom of the folding protein chains in regions where HHfrag is guaranteed to produce near-native local structure prediction (high-confidence regions).

We used the centroid-based torsion angle predictor, part of HHfrag [52] (Section 3.7). After analyzing the structural consistency and recurrence of the fragments in a given complemented fragset, HHfrag extracts φ and ψ angle predictions from the representative fragments at each target position. We used all torsion angle predictions of very

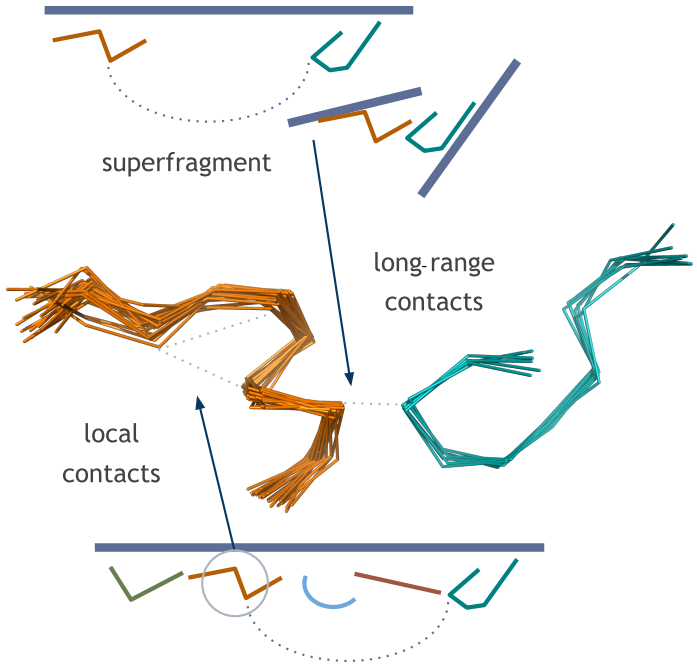


Figure 5.1: Extraction of predicted local (intra-fragment) and long-range (inter-fragment) contacts from dynamic fragment libraries.

high confidence ($C \geq 1$) as a direct source of angular restraints in Rosetta:

```
Dihedral C i-1 N  i   CA i   C i   CIRCULARHARMONIC phi 0.35
Dihedral N i   CA i+1 C  i+1 N i+1 CIRCULARHARMONIC psi 0.35
```

where *phi* and *psi* are the torsion angles of the representative fragment at target position i in radians and 0.35 is the σ parameter of the dihedral constraint in Rosetta:

$$f(x) = \left(\frac{NearestAngle(x, X_{rep}) - X_{rep}}{\sigma} \right)^2 \tag{5.1}$$

5.2.3 *Distance restraints*

Information about mid- and long-range contacts can be incorporated in the structure calculation protocol when experimental 3D NOESY data are available. We used a standard algorithm for reading unassigned NOE peaks and relating them back to the actual residues in the protein sequence [37]. For each proton endpoint of a given NOE peak, this procedure scans the list of assigned chemical shifts for entries within a small range of ± 0.2 ppm. Unfortunately, most NOE data sets contain extremely ambiguous proton-proton contacts, meaning that the endpoints of a given peak can be assigned to multiple chemical shift candidates. To derive an unambiguous list of contacts from a given NOE spectrum, we propose a basic filtering algorithm,

Target	Range	Best (frag.)	Best (superfrag.)	Selected
2kmmA	2-62	0.71	0.78	0.78
2kruA	6-51	0.68	0.73	0.73
2kj6A	34-95	0.54	0.88	0.54
2l9rA	11-60	0.78	0.55	0.78
2ln3A	1-76	0.73	0.41	0.73
2kifA	3-96	0.64	0.46	0.64
2la6A	14-99	0.53	0.57	0.57
2lojA	20-63	0.53	0.48	0.53
2lahA	12-160	0.69	0.29	0.69
2kpmA	23-82	0.50	0.47	0.50
2ltmA	11-107	0.38	0.58	0.58
2lciA	1-127	0.64	0.26	0.26
2ltlA	15-119	0.44	0.44	0.44
2kk1A	39-135	0.43	0.44	0.44

Table 5.1: TM-scores of the best Rosetta models, obtained with hybrid fragment libraries (denoted as *fragments*) and distance restraints derived from filtered 3D NOE spectra (*superfragments*). The best decoys from both sets were additionally tested for compatibility with the unfiltered NOE spectra. The decoy explaining a higher number of NOE peaks is indicated in the last column.

which validates the raw NOE contacts against a list of predicted intra- and inter-fragment contacts (Figure 5.1).

The extraction of inter-fragment contacts follows a straightforward procedure. We modeled the backbone of each fragment onto the target sequence using SCWRL4 [58] and protonated the resulting full-atom chains with Reduce [93]. Pairs of hydrogen atoms within an absolute distance of up to 6 Å were considered NOE-visible and extracted as short-range contacts.

To predict inter-fragment contacts, we inspected each possible pair of fragments in a given HHfrag library. For each pair, we scanned the PDBS25 database for structures in which the same pair of fragments co-occurs. This was performed by running HHsearch [84] with the HMM profiles of the fragments as queries and intersecting the resulting hit lists. Each time a pair of fragment instances in a given PDB chain were found to be in contact, i. e. having at least one pair of hydrogens within a distance of 6 Å, and separated by relatively conserved amount of residues in the primary structure of the chain, we nominated the fragment pair as a *superfragment*. All superfragment hydrogen-hydrogen contacts within a cutoff of 6 Å were finally extracted as NOE-visible long-range contacts.

To obtain the final list of filtered NOE contacts, we intersected the set of ambiguous NOEs with the set of all predicted fragment and superfragment contacts. Surviving NOEs were kept and incorporated in the structure calculation as *bounded* Rosetta distance constraints:

```
AtomPair Ni i Nj j NOE BOUNDED 1 6 0.5 0.5
```

where the minimum distance is set to 1 Å, the upper bound is 6 Å and 0.5 is the standard deviation parameter of the constraint. Since Rosetta has no built-in support for full-atom models and constraints, we considered only H and HA main-chain contact endpoints, while all side-chain contact endpoints were approximated as centroids (CEN).

5.3 PERFORMANCE

We used 14 protein targets from CASD [73] to evaluate the performance of our structure determination protocols. Table 5.1 summarizes the results of Rosetta structure calculations using hybrid fragment libraries (Section 5.2.1) with angular restraints (Section 5.2.2) and filtered NOEs (Section 5.2.3). We computed 200 decoys per target and superimposed them onto their respective native structures using a local fitting procedure. The quality of each decoy was evaluated by computing its TM-score [95]. Recall that a TM-score greater than 0.4 generally indicates a correct fold.

When using predicted torsion angles and HHfrag libraries complemented with analogous fragments, we were able to obtain decoys with correct fold in all instances (Table 5.1). The best decoys for 11 out of 14 targets had TM-scores greater than 0.5, which indicates structures of very high quality. These decoys were sampled with very modest computational resources (less than 200 trials per target). They can be further optimized using conventional NOE-based structure calculation programs, thus our protocol reduces the required computational time for structure calculation significantly.

Better decoys were obtained for 5 targets after the incorporation of distance restraints from filtered NOEs and superfragments (Table 5.1). However, the addition of distance restraints does not always increase the sampling rate of near-native decoys and the quality of the best decoys (Table 5.1). In some instances (2l9r, 2nl3, 2lah and 2lci), the incorporation of distance restraints in fact aggravates the performance. This is caused by highly noisy NOE spectra, which cannot be reliably filtered using predicted long-range contacts from superfragments. As a result, the filtered distance restraints lack some important long-range contacts, but include false-positive contacts at the same time. The combined action of these two negative factors may pose a significant challenge for the current Rosetta protocol.

To discriminate between these two extreme cases and blindly eliminate all NOE-derived decoys of low quality, we examined the compat-

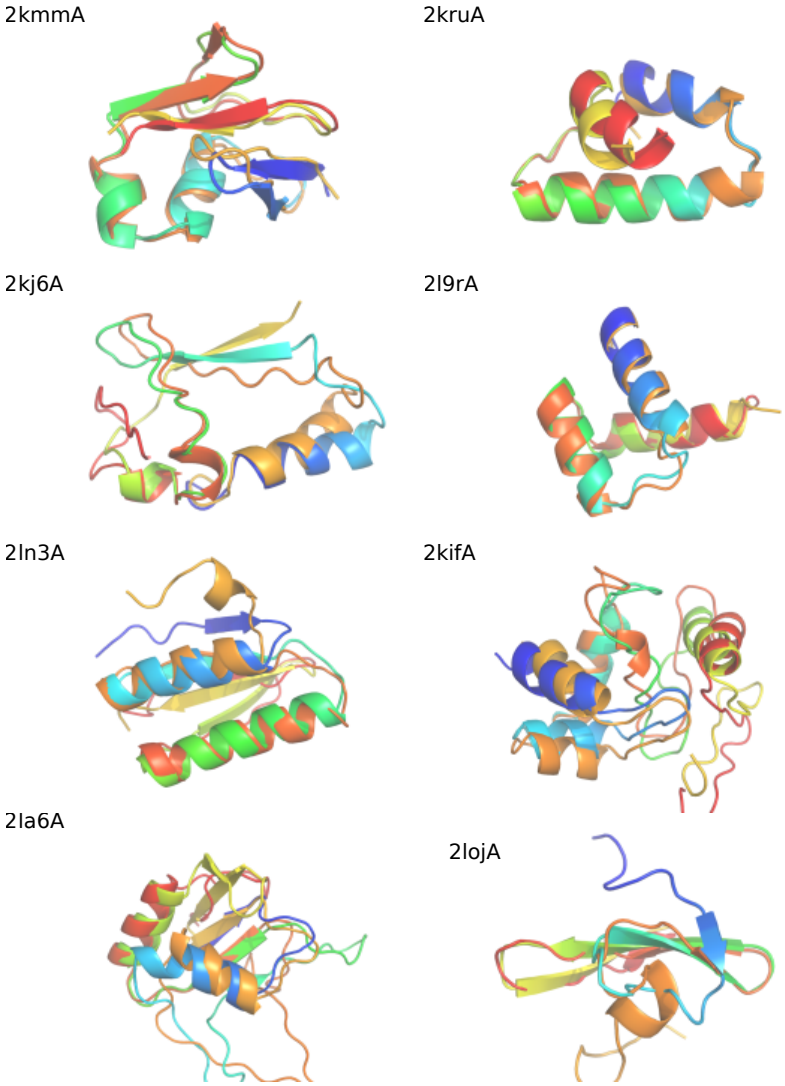


Figure 5.2: Superimposition of the final Rosetta models from Table 5.1 onto their corresponding experimental structures.

ibility of each pair of alternative decoys (with and without distance restraints) with the experimental data. After building protonated full-atom models, we extracted all pairs of NOE-visible contacts using a distance cutoff of 6 Å. This procedure is identical to the one used for the extraction of long-range contacts from superfragments. The decoy containing a higher number of unfiltered NOE peaks is selected as the final, winning model (see Table 5.1 and Figure 5.2).

5.4 CONCLUSION

We introduced a practical approach to automated protein structure determination using sparse experimental data (chemical shifts and 3D NOE spectra) and the HHfrag local structure prediction framework, described in previous chapters. Our structure calculation proto-

col is based on the classic Rosetta *ab initio* fragment assembly and uses variable-length sequence motifs (from HHfrag) in combination with analogous fragments (from CSfrag) and angular restraints, derived from confidence-guided local structure predictions with HHfrag. We demonstrated that this protocol can reduce the computational time needed to obtain initial models of high quality, and can be used as a solid starting point in fully automated NMR structure determination.

6 THE CSB OPEN-SOURCE PROJECT

This chapter is a brief overview of the software behind all algorithms, which have been described so far. We discuss the architecture of our software tools and present some key elements of their design and application programming interface. The software, which has emerged as part of our work, has been released to the pubic domain free of charge as part of the CSB open-source project. Developers seeking detailed documentation and support, or those willing to contribute to our community effort, are invited to visit the homepage of the project at CodePlex.

6.1 INTRODUCTION

The Python programming language is becoming an increasingly popular choice in research. Python's comprehensive numerical libraries and its dynamic type system render this platform an attractive environment for rapid application development. The rapid prototyping paradigm has seen wide adoption in scientific projects, because it facilitates experimentation with new techniques or features with minimal effort. However, the systematic use of *ad hoc* scripting soon turns into a burden, preventing efficient code reuse and hindering further development and agility. The industry-standard solution to these problems is the use of continuously developed, well-abstracted and tested software libraries. Productivity in building solid, reliable and extensible bioinformatics applications could significantly benefit from the practice of using carefully engineered libraries.

Here we introduce the Computational structural biology toolbox (CSB) — a Python library, designed for solving problems in the field of structural bioinformatics [53]. The project was conceived as a trivial separation of HHfrag's executable components from its generic abstractions; however, it has quickly expanded as a full-fledged Python framework. CSB APIs have been designed to meet the following design goals: (i) reusability and extensibility, (ii) clean interfaces, (iii) granular and well-encapsulated abstractions, (iv) use of classic design patterns, (v) preference of obvious, self-documenting design and (vi) testability.

All algorithms, described in earlier chapters, are implemented as thin client applications (sometimes called *protocols* or *apps*), which

Some of the material in this chapter has been previously published and adapted from Kalev *et al.* (2012) [53]. Used with permission.

consume the *core library* APIs. The CSB library improves over existing packages, such as Biopython [20], with its granular, consistent and extensible object model, but also provides new features like a comprehensive statistical API and support for new abstractions and file formats. The current version provides mature APIs for working with biological macromolecular structures, sequences, sequence profiles and fragment libraries, but also involves a significant amount of statistical modules, including many probability distributions and samplers. We put a strong emphasis on quality and reliability achieved through continuous attention to good design and best practices in test engineering.

6.2 API OVERVIEW

CSB is composed of several highly branched, hierarchical Python packages. The core library can be divided into bioinformatical (*csb.bio.**) and statistical (*csb.statistics.**) APIs.

6.2.1 *Core abstractions*

All fundamental biological abstractions are part of the *csb.bio* namespace. For example, *csb.bio.sequence* defines the base *AbstractSequence* and *AbstractAlignment* interfaces and provides a number of useful implementations of these abstractions, such as *Sequence*, *SequenceAlignment*, and *StructureAlignment*. As suggested by its name, *csb.bio.hmm* deals with HHpred and its profile HMMs [84], while *csb.bio.fragments* contains all supporting objects behind our fragment detection algorithms. A package of central importance, part of the *csb.bio* namespace, is *csb.bio.structure*, which defines a common infrastructure for all remaining modules with essential objects like *Structure*, *Chain*, *Residue* and *Atom*.

The architecture of the *Structure* abstract model in CSB is shown in Figure 6.1. *Structure* instances are hierarchical objects, which implement the Composite pattern [33]. Each level in the composite tree (*Structure*, *Chain*, *Residue* or *Atom*) is represented by a class, derived from the base *AbstractEntity*, and supports iteration over its immediate sub-entities. Every entity thus exposes a standard set of operations, such as *AbstractEntity.transform()* and *AbstractEntity.items*, which automatically propagate down the tree when invoked at an arbitrary level. All members of the composite data structure can therefore be treated polymorphically via a uniform interface, which allows the implementation of flexible composite iterators [33]. Clients are also free to define their own, pluggable *AbstractEntity* implementations.

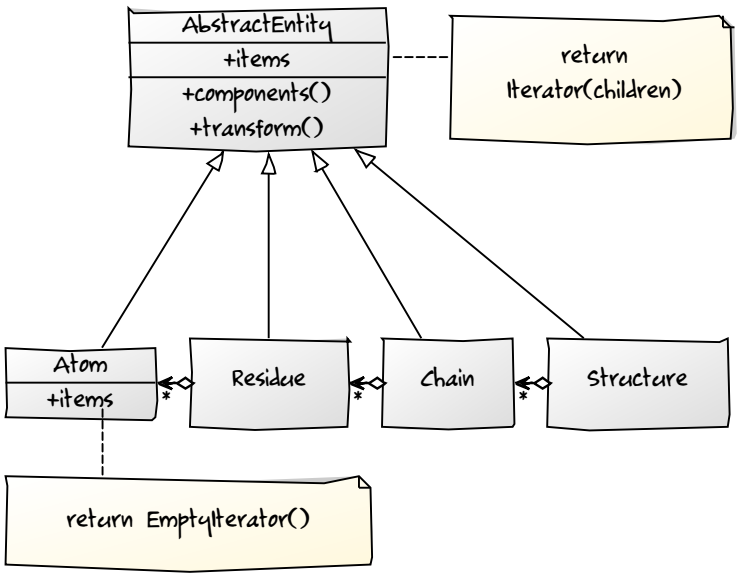


Figure 6.1: The *Structure* model in CSB. *Structure* is the root entity in a multi-level composite aggregation. Each node in the composite implements a uniform *AbstractEntity* interface and supports iteration over its sub-entities. The leaf nodes (*Atom*) have no children and thus return a null iterator.

6.2.2 *I/O*

CSB exposes an I/O API for a broad variety of biological file formats (*csb.bio.io*). For example, *csb.bio.io.hhpred* is the first publicly available Python module to date for working with HHpred's HMM and result files [84]. Another module, *csb.bio.io.mrc*, contains cryo-electron density map processing utilities, while *csb.bio.io.clans* provides readers and writers for CLANS files [31]. Extensive PDB file manipulation is supported using our PDB API, which is part of *csb.bio.io.wwpdb*.

CSB contains a fast, reliable and extensible PDB parser model with novel features. The architecture of our PDB parsers is detailed in Figure 6.2. The base *AbstractStructureParser* is a TemplateMethod [33], which defines a common backbone for all PDB parsing schemes. Concrete parser implementors must define how the PDB header will be handled through a special hook method, in order to complete the implementation. The main PDB parser in CSB —- *RegularStructureParser*, differs significantly from existing solutions such as Biopython. *RegulafStructureParser* reads and initializes all residues from SEQRES, rather than the ATOM fields in the file. ATOM records are subsequently mapped to the residue objects using a simple and very fast alignment algorithm. Therefore, the resulting *Chain* products always contain the complete primary structure of the PDB chain, as governed by the SEQRES fields. This feature eliminates the need to relate the PDB atoms back to the real sequence of the protein in question –

a process which is often difficult and error-prone. A fallback parser for PDB files with no header fields is also provided — *LegacyStructureParser*. The PDB parsers in CSB are typically consumed through a dedicated *StructureParser* factory, which transparently determines which parser implementation to instantiate, given a specific PDB file.

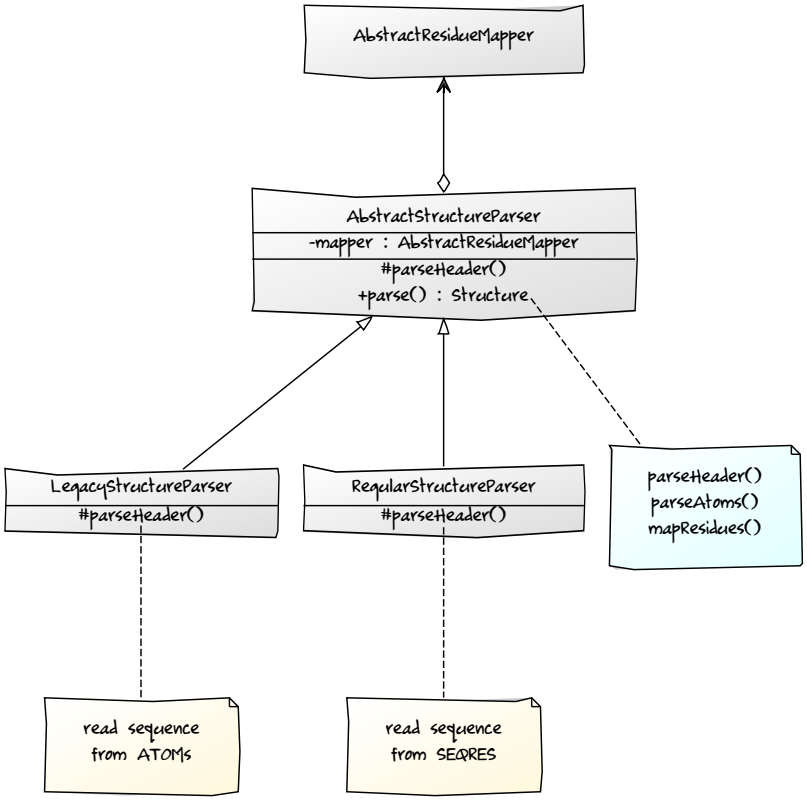


Figure 6.2: Basic architecture of the PDB parsing model in CSB. The base *AbstractStructureParser* is a TemplateMethod (*parse*), which defers the PDB header handling and sequence initialization to implementing subclasses through a hook (*parseHeader*). Implementors may fine-tune the mechanism of structure parsing by overriding a number of granular hook methods (not shown). Each parser maintains an instance of a concrete *ResidueMapper* Strategy, which is used to relate the ATOM records back to the sequence and assemble the final structure.

The residue mapping algorithm is properly abstracted in a dedicated *ResidueMapper* strategy [33], thus allowing alternative mapping strategies to be employed and exchanged at runtime. When benchmarked over the complete PDB database, the standard SEQRES mapping algorithm *FastResidueMapper* fails for about 250 structures. This is frequently an indication of a PDB format issue, which is an unrecoverable error. In this case, *AbstractStructureParser* will immediately switch to a less strict *RobustResidueMapper*, based on a classic Needleman–Wunsch global alignment [64]. This algorithm is 100% fail-safe, but unlike *FastResidueMapper*, it has quadratic running time

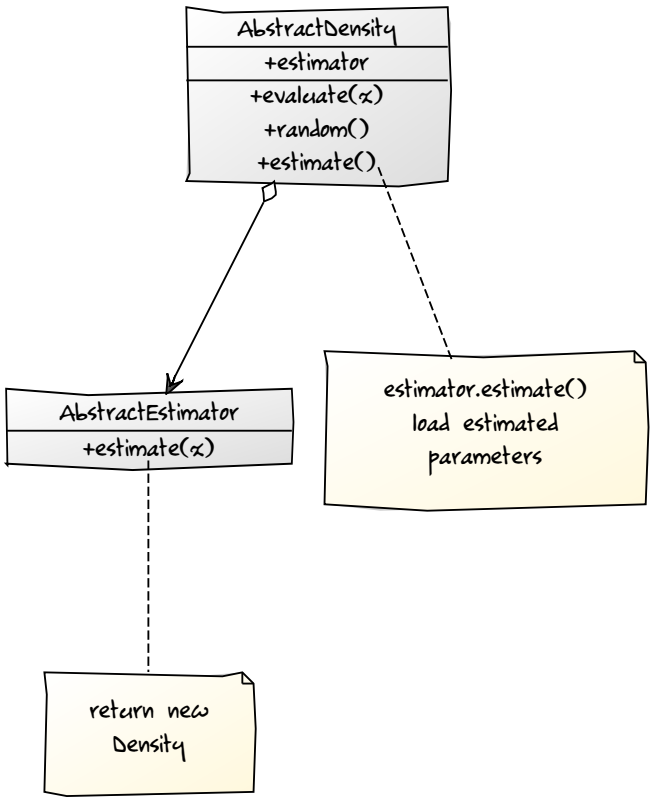


Figure 6.3: Architecture of the probability density functions in CSB.

and space requirements, which accounts for a noticeable overhead. However, the PDB parser context rarely switches to this mapping strategy, thus achieving excellent balance between speed and uncompromising precision.

We compared the performance of *RegularStructureParser* with PDB I/O modules from alternative libraries: Biopython [41], PyCogent [19] and the C++-based OpenStructure [6]. As expected, OpenStructure was the fastest and parsed 4000 PDB entries with 0.09s per structure. CSB is positioned between Biopython (0.19s) and PyCogent (0.43s) with 0.32s per structure, which suggests that the SEQRES mapping feature comes with an acceptable performance overhead.

6.2.3 *Statistics API*

CSB is bundled with a collection of statistical models in the *csb.statistics* namespace. All probability distributions are derived from a common *AbstractDensity* object. The *evaluate* hook controls how implementors compute their corresponding PDF values. This model also supports the notion of pluggable estimators through a classic Strategy [33] (Figure 6.3).

Among the implemented density functions are standard uni- and multivariate probability distributions such as the Normal and the

Gamma distribution, but also more exotic distributions such as the Multivariate Normal Inverse Gaussian distribution, used to model multivariate heavy-tailed data. Several estimators based on maximum likelihood and Gibbs sampling are implemented. We also provide a general framework for Markov chain Monte Carlo simulation and implementation of standard schemes such as random walk Metropolis Hastings, Hamiltonian Monte Carlo [24] and replica-exchange Monte Carlo [86].

6.3 CSB APPS

CSB comes with a simple framework for writing console applications (*csb.apps*). These applications could be seen as short protocols, built on top of the core library and consuming its APIs. The main concept behind this framework is to allow rapid application development, while simultaneously ensuring reusability and sharing of app modules and components. Each CSB app is designed to operate seamlessly in two alternative contexts:

A. executable — as a standard console application;

B. component — as a regular Python object, which can be instantiated and reused without side effects, such as causing unexpected system exits or standard output stream writes.

Each release is bundled with a number of pre-installed, open-source applications. For example *csb.apps.hhfrag* provides HHfrag [51], the dynamic fragment detection and confidence-guided torsion angle prediction method, discussed in Chapter 3. Two supplementary apps are provided for building sequence profiles (*csb.apps.buildhmm*) and measuring the local precision of fragment libraries (*csb.apps.precision*). Our chemical shift-based method for detection of analogous fragments (Chapter 4) is fully implemented in *csb.apps.csfrag* and provides a compatible interface. Both HHfrag and CSfrag are tightly integrated and can be used in conjunction to replicate the various fragment filling and filtering protocols described so far, without the need of consuming the Python API directly. The supporting PDBS25 database required to run HHfrag and CSfrag can be obtained from the official release package at:

csb.codeplex.com/releases

BFit is another CSB app, which can be used to perform robust superposition of protein structures [61]. Every release package contains also EMBD, an application for sharpening of cryo-electron microscopy maps [46] using non-negative deconvolution, and Promix, an application implementing Gaussian mixture models for identifying rigid domains in structure ensembles [45].

6.4 DEVELOPMENT

CSB is being developed under a continuous integration model. The reliability of each production release is controlled by CSB's built-in high-coverage unit test framework. Stable builds are regularly released to the Python Package Index (PyPi), CodePlex and Debian repositories. Portability is also an essential design goal, so CSB works without modification on every major platform (Windows, Linux, Mac) and any modern Python interpreter (version 2.6 or higher, including Python 3).

Our package is distributed under a permissive MIT license, which allows direct integration in other open-source or proprietary software projects. Detailed tutorials, technical API documentation, complete source code and release packages can be obtained from the web site of our project at:

csb.codeplex.com.

7

OUTLOOK

This work describes a comprehensive framework for local structure prediction from sequence and chemical shift data with a strong emphasis on fragment precision and correct detection of motif boundaries. We demonstrated that our dynamic approach to fragment detection improves the performance of existing fragment assembly algorithms for protein structure prediction or structure determination. We provide feature-complete, unit-tested, open-source APIs to facilitate the adoption of our algorithmic contributions. HHfrag was additionally designed as a stand-alone fragment extraction module, which can be readily incorporated into existing protein structure prediction and structure determination protocols.

While keeping compatibility with earlier methods, our work introduces an assortment of novel features, whose potential may be revealed by further development of the fragment assembly approach. For example, the current Rosetta folding protocol is optimized for fragments of constant length and cannot take advantage of the rich information encoded in HHfrag's gapped fragments and discontinuous fragment maps. Especially interesting are the prospects of tight integration between the ISD method for structure determination and HHfrag's confidence-guided framework for prediction of local motif conservation.

In conclusion, with this work we have overcome important limitations of the sequence-based fragment detection approach and contributed valuable new developments in the field of local structure prediction.

PUBLICATIONS

Some ideas and figures have previously appeared in the following publications:

- *Kalev I* and Habeck M. Confidence-guided local structure prediction with HHfrag. *PLOS ONE*. 2013.
  - Text and figures from this manuscript appear in Chapter 2, Chapter 3 and Chapter 5.
- *Kalev I*, Mechelke M, Kopec KO, Holder T, Carstens S and Habeck M. CSB: a Python framework for structural bioinformatics. *Bioinformatics*. 2012.
  - Text and figures from this manuscript appear in Chapter 6.
- *Kalev I* and Habeck M. HHfrag: HMM-based fragment detection using HHpred. *Bioinformatics*. 2011.
  - Text and figures from this manuscript appear in Chapter 3.

BIBLIOGRAPHY

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, Oct 1990.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, Sep 1997.

[3] V. Alva, S. Dunin-Horkawicz, M. Habeck, M. Coles, and A. N. Lupas. The gd box: A widespread non-contiguous supersecondary structural element. *Protein Science*, 18(9):1961–1966, 2009.

[4] M. J. Berardi, W. M. Shih, S. C. Harrison, and J. J. Chou. Mitochondrial uncoupling protein 2 structure determined by NMR molecular fragment searching. *Nature*, 476:109–113, 2011.

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

[6] M. Biasini, V. Mariani, J. Haas, S. Scheuber, A. D. Schenk, T. Schwede, and A. Philippsen. OpenStructure: a flexible software framework for computational structural biology. *Bioinformatics*, 26(20):2626–2628, Oct 2010.

[7] M. Blaber, X. J. Zhang, and B. W. Matthews. Structural basis of amino acid alpha helix propensity. *Science*, 260(5114):1637–1640, Jun 1993.

[8] W. Boomsma, K.V. Mardia, C.C. Taylor, J. Ferkinghoff-Borg, A. Krogh, and T. Hamelryck. A generative, probabilistic model of local protein structure. *Proc. Natl. Acad. Sci. U.S.A.*, 105:8932–8937, 2008.

[9] Carl-Ivar Branden and John Tooze. *Introduction to Protein Structure*. Garland Publishing, second edition, January 1999. ISBN 0815323050.

[10] C. Bystroff and D. Baker. Prediction of local structure in proteins using a library of sequence-structure motifs. *J. Mol. Biol.*, 281: 565–577, Aug 1998.

[11] C. Bystroff and Y. Shao. Fully automated ab initio protein structure prediction using I-SITES, HMMSTR and ROSETTA. *Bioinformatics*, 18 Suppl 1:54–61, 2002.

[12] C. Bystroff and B. J. Webb-Robertson. Pairwise covariance adds little to secondary structure prediction but improves the prediction of non-canonical local structure. *BMC Bioinformatics*, 9:429, 2008.

[13] C. Bystroff, K. T. Simons, K. F. Han, and D. Baker. Local sequence-structure correlations in proteins. *Curr. Opin. Biotechnol.*, 7:417–421, Aug 1996.

[14] A. C. Camproux and P. Tuffery. Hidden Markov model-derived structural alphabet for proteins: the learning of protein local shapes captures sequence specificity. *Biochim. Biophys. Acta*, 1724 (3):394–403, Aug 2005.

[15] A. C. Camproux, P. Tuffery, J. P. Chevrolat, J. F. Boisvieux, and S. Hazout. Hidden Markov model approach for identifying the modular framework of the protein backbone. *Protein Eng.*, 12 (12):1063–1073, Dec 1999.

[16] A. C. Camproux, R. Gautier, and P. Tuffery. A hidden markov model derived structural alphabet for proteins. *J. Mol. Biol.*, 339 (3):591–605, Jun 2004.

[17] A. Cavalli, X. Salvatella, C. M. Dobson, and M. Vendruscolo. Protein structure determination from NMR chemical shifts. *Proc. Natl. Acad. Sci. U.S.A.*, 104(23):9615–9620, Jun 2007.

[18] M. S. Cheung, M. L. Maguire, T. J. Stevens, and R. W. Broadhurst. DANGLE: A Bayesian inferential method for predicting protein backbone dihedral angles and secondary structure. *J. Magn. Reson.*, 202(2):223–233, Feb 2010.

[19] M. Cieslik, Z. S. Derewenda, and C. Mura. Abstractions, algorithms and data structures for structural bioinformatics in PyCogent. *J Appl Crystallogr*, 44(Pt 2):424–428, Apr 2011.

[20] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, Jun 2009.

[21] G. Cornilescu, F. Delaglio, and A. Bax. Protein backbone angle restraints from searching a database for chemical shift and sequence homology. *J. Biomol. NMR*, 13(3):289–302, Mar 1999.

[22] A. G. de Brevern. New assessment of a structural alphabet. *In Silico Biol. (Gedrukt)*, 5(3):283–289, 2005.

[23] A. G. de Brevern, C. Etchebest, and S. Hazout. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins*, 41(3):271–287, Nov 2000.

[24] S. Duane, A. D. Kennedy, B. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222, 1987.

[25] R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998. ISBN 9780521629713.

[26] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9): 755–763, 1998.

[27] N. Fernandez-Fuentes, B. Oliva, and A. Fiser. A supersecondary structure library and search algorithm for modeling loops in protein structures. *Nucleic Acids Res.*, 34(7):2085–2097, 2006.

[28] N. Fernandez-Fuentes, J. M. Dybas, and A. Fiser. Structural characteristics of novel protein folds. *PLoS Comput. Biol.*, 6:e1000750, Apr 2010.

[29] R. D. Finn, J. Clements, and S. R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.*, 39 (Web Server issue):29–37, Jul 2011.

[30] L. Fourrier, C. Benros, and A. G. de Brevern. Use of a structural alphabet for analysis of short loops connecting repetitive structures. *BMC Bioinformatics*, 5:58, May 2004.

[31] T. Frickey and A. Lupas. CLANS: a Java application for visualizing protein families based on pairwise similarity. *Bioinformatics*, 20(18):3702–3704, Dec 2004.

[32] D. Frishman and P. Argos. Knowledge-based protein secondary structure assignment. *Proteins*, 23(4):566–579, Dec 1995.

[33] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional, 1995.

[34] S. W. Ginzinger and J. Fischer. SimShift: identifying structural similarities from NMR chemical shifts. *Bioinformatics*, 22(4):460–465, Feb 2006.

[35] S. Griep and U. Hobohm. PDBselect 1992-2009 and PDBfilterselect. *Nucleic Acids Res.*, 38:D318–319, Jan 2010.

[36] D. Gront, D. W. Kulp, R. M. Vernon, C. E. Strauss, and D. Baker. Generalized fragment picking in Rosetta: design, protocols and applications. *PLoS ONE*, 6(8):e23294, 2011.

[37] P. Guntert. Automated NMR structure calculation with CYANA. *Methods Mol. Biol.*, 278:353–378, 2004.

[38] P. Guntert. Automated structure determination from NMR spectra. *Eur. Biophys. J.*, 38(2):129–143, Feb 2009.

[39] P. Guntert, C. Mumenthaler, and K. Wuthrich. Torsion angle dynamics for NMR structure calculation with the new program DYANA. *J. Mol. Biol.*, 273(1):283–298, Oct 1997.

[40] M. Habeck, W. Rieping, and M. Nilges. Weighting of experimental evidence in macromolecular structure determination. *Proc. Natl. Acad. Sci. U.S.A.*, 103(6):1756–1761, Feb 2006.

[41] T. Hamelryck and B. Manderick. PDB file parser and structure class implemented in Python. *Bioinformatics*, 19(17):2308–2310, Nov 2003.

[42] B. Han, Y. Liu, S. W. Ginzinger, and D. S. Wishart. SHIFTX2: significantly improved protein chemical shift prediction. *J. Biomol. NMR*, 50(1):43–57, May 2011.

[43] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.*, 89(22):10915–10919, Nov 1992.

[44] A. Hildebrand, M. Remmert, A. Biegert, and J. Söding. Fast and accurate automatic structure prediction with HHpred. *Proteins*, 77 Suppl 9:128–132, 2009.

[45] M. Hirsch and M. Habeck. Mixture models for protein structure ensembles. *Bioinformatics*, 24(19):2184–2192, Oct 2008.

[46] M. Hirsch, B. Schölkopf, and M. Habeck. A blind deconvolution approach for improving the resolution of cryo-EM density maps. *J. Comput. Biol.*, 18(3):335–346, Mar 2011.

[47] L. Holm and P. Rosenstrom. Dali server: conservation mapping in 3D. *Nucleic Acids Res.*, 38(Web Server issue):W545–549, Jul 2010.

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

[49] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, 292:195–202, Sep 1999.

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

[51] I. Kalev and M. Habeck. HHfrag: HMM-based fragment detection using HHpred. *Bioinformatics*, 27(22):3110–3116, Nov 2011.

[52] I. Kalev and M. Habeck. Confidence-guided local structure prediciton with HHfrag. *PLOS ONE*, Aug 2013.

[53] I. Kalev, M. Mechelke, K. O. Kopec, T. Holder, S. Carstens, and M. Habeck. CSB: a Python framework for structural bioinformatics. *Bioinformatics*, 28(22):2996–2997, Nov 2012.

[54] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–666, Mar 1958.

[55] D. E. Kim, D. Chivian, and D. Baker. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res.*, 32: W526–531, Jul 2004.

[56] Donald E. Knuth. Computer Programming as an Art. *Communications of the ACM*, 17(12):667–673, December 1974.

[57] R. Kolodny, P. Koehl, L. Guibas, and M. Levitt. Small libraries of protein fragments model native protein structures accurately. *J. Mol. Biol.*, 323:297–307, 2002.

[58] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins*, 77(4):778–795, Dec 2009.

[59] S. C. Li, D. Bu, X. Gao, J. Xu, and M. Li. Designing succinct structural alphabets. *Bioinformatics*, 24:i182–189, Jul 2008.

[60] A. N. Lupas, C. P. Ponting, and R. B. Russell. On the evolution of protein folds: are similar motifs in different protein folds the result of convergence, insertion, or relics of an ancient peptide world? *J. Struct. Biol.*, 134(2-3):191–203, 2001.

[61] M. Mechelke and M. Habeck. Robust probabilistic superposition and comparison of protein structures. *BMC Bioinformatics*, 11:363, 2010.

[62] J. Moult, K. Fidelis, A. Kryshtafovych, and A. Tramontano. Critical assessment of methods of protein structure prediction (CASP)–round IX. *Proteins*, 79 Suppl 10:1–5, 2011.

[63] V. Munoz and L. Serrano. Intrinsic secondary structure propensities of the amino acids, using statistical phi-psi matrices: comparison with experimental scales. *Proteins*, 20(4):301–311, Dec 1994.

[64] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453, Mar 1970.

[65] M. Nilges. Calculation of protein structures with ambiguous distance restraints. Automated assignment of ambiguous NOE crosspeaks and disulphide connectivities. *J. Mol. Biol.*, 245(5): 645–660, Feb 1995.

[66] M. Nilges, M. J. Macias, S. I. O'Donoghue, and H. Oschkinat. Automated NOESY interpretation with ambiguous distance restraints: the refined NMR solution structure of the pleckstrin homology domain from beta-spectrin. *J. Mol. Biol.*, 269(3):408–422, Jun 1997.

[67] B. Offmann, M. Tyagi, and A. G. de Brevern. Local protein structures. *Current Bioinformatics*, 2:165–202(38), 2007.

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

[69] W. Rieping and W. F. Vranken. Validation of archived chemical shifts through atomic coordinates. *Proteins*, 78(11):2482–2489, Aug 2010.

[70] W. Rieping, M. Habeck, and M. Nilges. Inferential structure determination. *Science*, 309(5732):303–306, Jul 2005.

[71] W. Rieping, M. Nilges, and M. Habeck. ISD: a software package for Bayesian NMR structure calculation. *Bioinformatics*, 24(8): 1104–1105, Apr 2008.

[72] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker. Protein structure prediction using Rosetta. *Meth. Enzymol.*, 383:66–93, 2004.

[73] A. Rosato, J. M. Aramini, C. Arrowsmith, A. Bagaria, D. Baker, A. Cavalli, J. F. Doreleijers, A. Eletsky, A. Giachetti, P. Guerry, A. Gutmanas, P. Guntert, Y. He, T. Herrmann, Y. J. Huang, V. Jaravine, H. R. Jonker, M. A. Kennedy, O. F. Lange, G. Liu, T. E. Malliavin, R. Mani, B. Mao, G. T. Montelione, M. Nilges, P. Rossi, G. van der Schot, H. Schwalbe, T. A. Szyperski, M. Vendruscolo, R. Vernon, W. F. Vranken, S. d. Vries, G. W. Vuister, B. Wu, Y. Yang, and A. M. Bonvin. Blind testing of routine, fully automated determination of protein structures from NMR data. *Structure*, 20(2):227–236, Feb 2012.

[74] B. Rost. Twilight zone of protein sequence alignments. *Protein Eng.*, 12(2):85–94, Feb 1999.

[75] R. I. Sadreyev, D. Baker, and N. V. Grishin. Profile-profile comparisons by COMPASS predict intricate homologies between protein families. *Protein Sci.*, 12(10):2262–2272, Oct 2003.

[76] A. Sali and T. L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, 234(3):779–815, Dec 1993.

[77] J. Schuchhardt, G. Schneider, J. Reichelt, D. Schomburg, and P. Wrede. Local structural motifs of protein backbones are classified by self-organizing neural networks. *Protein Eng.*, 9(10):833–842, Oct 1996.

[78] Y. Shen, O. Lange, F. Delaglio, P. Rossi, J. M. Aramini, G. Liu, A. Eletsky, Y. Wu, K. K. Singarapu, A. Lemak, A. Ignatchenko, C. H. Arrowsmith, T. Szyperski, G. T. Montelione, D. Baker, and A. Bax. Consistent blind protein structure generation from NMR chemical shift data. *Proc. Natl. Acad. Sci. U.S.A.*, 105(12):4685–4690, Mar 2008.

[79] Y. Shen, F. Delaglio, G. Cornilescu, and A. Bax. TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. *J. Biomol. NMR*, 44(4):213–223, Aug 2009.

[80] Y. Shen, R. Vernon, D. Baker, and A. Bax. De novo protein structure generation from incomplete chemical shift assignments. *J. Biomol. NMR*, 43(2):63–78, Feb 2009.

[81] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.*, 268:209–225, Apr 1997.

[82] M. J. Sippl. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *J. Mol. Biol.*, 213 (4):859–883, Jun 1990.

[83] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147(1):195–197, Mar 1981.

[84] J. Söding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21:951–960, Apr 2005.

[85] J. Song, H. Tan, M. Wang, G. I. Webb, and T. Akutsu. TANGLE: two-level support vector regression approach for protein backbone torsion angle prediction from primary sequences. *PLoS ONE*, 7(2):e30361, 2012.

[86] R. H. Swendsen and J.-S. Wang. Replica Monte Carlo simulation of spin glasses. *Physical Review Letters*, 57:2607–2609, 1986.

[87] M. Tyagi, P. Sharma, C. S. Swamy, F. Cadet, N. Srinivasan, A. G. de Brevern, and B. Offmann. Protein Block Expert (PBE): a web-based protein structure analysis server using a structural alphabet. *Nucleic Acids Res.*, 34(Web Server issue):W119–123, Jul 2006.

[88] R. Unger, D. Harel, S. Wherland, and J. L. Sussman. A 3D building blocks approach to analyzing and predicting structure of proteins. *Proteins*, 5(4):355–373, 1989.

[89] A Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. ISSN 00189448. doi: 10.1109/TIT.1967.1054010.

[90] W. F. Vranken and W. Rieping. Relationship between chemical shift value and accessible surface area for all amino acid atoms. *BMC Struct. Biol.*, 9:20, 2009.

[91] D. S. Wishart and D. A. Case. Use of chemical shifts in macromolecular structure determination. *Meth. Enzymol.*, 338:3–34, 2001.

[92] D. S. Wishart, B. D. Sykes, and F. M. Richards. Relationship between nuclear magnetic resonance chemical shift and protein secondary structure. *J. Mol. Biol.*, 222(2):311–333, Nov 1991.

[93] J. M. Word, S. C. Lovell, J. S. Richardson, and D. C. Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J. Mol. Biol.*, 285(4):1735–1747, Jan 1999.

[94] S. Wu and Y. Zhang. ANGLOR: a composite machine-learning algorithm for protein backbone torsion angle prediction. *PLoS ONE*, 3(10):e3400, 2008.

[95] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57:702–710, Dec 2004.

[96] Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, 33: 2302–2309, 2005.

[97] Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl. Acad. Sci. U.S.A.*, 102:1029–1034, Jan 2005.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/