Tel Aviv University
The Raymond and Beverly Sackler
Faculty of Exact Sciences
School of Computer Science

The Common Point Set Problem with Applications to Protein Structure Analysis

THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

By Maxim Shatsky

Under the supervision of Prof. Haim Wolfson and Prof. Ruth Nussinov

Submitted to the senate of Tel Aviv University
June 2006

Acknowledgments

I would like to express my deepest gratitude to my advisors, Prof. Ruth Nussinov and Prof. Haim Wolfson, for their wise guidance, support and encouragement throughout my research.

I wish to thank Noga Alon for the help with approximation algorithms. I thank Danny Halperin for the excitement discussions on computational geometry. I want to thank my friends and collaborators from Tel-Aviv University for the great partnership: Alexandra Shulman-Peleg, Oranit Dror, Dina Schneidman-Duhovny, Keren Lasker, Yuval Inbar, Inbal Halperin, Shira Mintz, Aviad Tsherniak, Nelly Bluvshtein, Oded Schwartz, Meytal Cohen and Hadar Benyamini. I thank Gunasekaran Kannan for the fruitful suggestions.

I deeply appreciate the financial support from the Council of Higher Education (VATAT) and I thank the Yeshaya Horowitz association for the fellowship in "Complexity Science".

Abstract

In this thesis we study computational problems related to the analysis of protein 3D folds and protein binding sites. Specifically, we deal with one of the fundamental problems in the field of Structural Proteome research, the problem of common spatial pattern detection in a set of protein structures, protein binding sites and protein-protein interfaces.

From the algorithmic standpoint we deal with the problems of multiple common point set detection under different metrics. We prove NP-Hardness results even for the case of practically defined geometrical constraints on the input point sets. While the generally defined problem and sub-problems of common point set detection are hard to approximate, under the practical biological constraints we present polynomial time approximation algorithms.

On the practical side, we present novel computational methods for multiple alignment of protein structures, structure based multiple sequence alignment, multiple binding site and multiple protein-protein interface alignment. Due to the hardness of the computational problems, we apply a combination of heuristic, approximation and branch-and-bound techniques in order to achieve a trade-off between practical efficiency and accuracy of the biological results.

The results of this thesis have been published in [107, 108, 105, 129, 110, 109, 111, 112, 118].

Contents

Chapter 1

Introduction

The decision to initiate the Structural Genomics project, following the completion of the first Human Genome draft, has demonstrated the crucial significance of the knowledge of protein structures both for theoretical research and Computer Assisted Drug Design. Through molecular structures and their complexes (protein-drug, protein-protein, etc.) the scientific community aims to get an insight into protein function and mis-function.

Proteins play major functional roles in the cells. Their diverse functionality ranges from catalytic activity to structural and mechanical roles, e.g. skeleton formation, transportation of various molecules, immune response, replication of the DNA, protein biosynthesis and degradation, etc.

The major determinants of protein function are the primary amino acid content and the geometry of the folded protein 3D structure. Protein 3D structure serves two interrelated tasks. First, it has a pure mechanical role, i.e. protein size and shape determine its mobility inside the cell, construction properties of larger protein-protein complexes, e.g. long fibrils and virus scaffolds, flexibility, e.g. in muscle contraction, etc. Protein 3D fold has a phenomenal stability against mutations. A substantial number (more than 70%) of amino acids can be often mutated, while preserving the overall structural fold. Amino acid mutations along the protein 3D fold stability provide a basis for enhancement and extension of various protein functions. However, mutation of even one key core residue may be fatal for the correct protein folding.

Consequently, during the course of evolution the protein 3D fold is more preserved than its primary sequence. Therefore, analysis of protein sequences alone is not sufficient. Study of the protein structures provides a deeper insight into their function and evolution.

A second, interrelated task of protein 3D structure is to provide a proper geometrical configuration for its functional binding sites. Mutation of key residues at protein binding site may change or destroy its function, even without altering the overall fold. Protein binding site functions well as long as its side chains are able to reach the required (for binding) 3D configuration. Indeed, binding sites with similar function may be found in proteins with different 3D fold arrangements and consequently with different overall primary amino acid sequence. Only the key residues at the binding site have to maintain a similar biochemical and geometrical properties. Therefore, the functionality of a protein molecule is a unison of its primary sequential amino acid content and its folded 3D structure.

In this thesis, we study computational problems related to the analysis of protein 3D folds and protein binding sites. Specifically, we deal with one of the fundamental problems in the field of Structural Proteome research, the problem of common spatial pattern detection in a set of protein structures, protein binding sites and protein-protein interfaces.

Detection of multiple protein structure alignments is important for reasons similar to those of multiple protein sequence alignments. Conservation of a region in many molecules bears higher significance as compared to its recurrence in only two. Multiple structure alignment is essential for (1) detection of structural motifs common to proteins that share the same function or binding property. This allows derivation of residue conservation in analogous positions in 3D space, such as in protein cores or in protein binding sites[81, 22, 10]; (2) protein classification[96, 29]; (3) evolutionary relations between proteins, e.g. identification of a consensus (sub)structure which can serve as a model of potential ancestor[45]; and (4) structure prediction[3]. Homology modeling methods utilize multiple structure alignment to increase the accuracy of hypothetical templates[47]. Therefore, study of the protein backbone alignment may give answers related to the second task of the protein 3D configuration - presentation

of proper geometrical configuration for protein functional binding sites. However, the phenomena of similar functioning proteins with different 3D folds opens new challenges for structural analysis. One assumption is that proteins with similar function have similar, in terms of geometry and biochemical properties, binding site patterns. Consequently, identification of common binding patterns has applications for drug design, e.g. design of improved or new inhibitors, recognition of possible side-effects, etc. In general, methods for protein backbone alignment are not suitable to carry out the task of protein binding site alignment. Therefore, novel methods have to be developed.

Despite the computational difficulty of the multiple structure alignment problem, we propose practically efficient methods for (1) multiple protein backbone alignment, including structure derived multiple sequence alignment, (2) multiple protein binding site alignment, and (3) multiple protein-protein interface alignment. In practice, the proposed methods solve the required biological tasks.

The thesis is organized as follows:

Chapter 2: Largest Common Point Set Problem

We start from a theoretical study of the spatial pattern detection between a set of structures. The problem of pattern detection between two structures answers the following question. Given two point sets A and B, find a subset of A that is similar (according to some similarity metric) to some subset of B. The optimization problem is to maximize the cardinality of similar subsets. One common way to define the similarity between two point sets is by the *bottleneck* metric[1, 37]. Similar subsets are called ε-congruent if the maximal distance between the matched points is not greater than ε. The optimization problem, called the *Largest Common Point Set* (LCP) problem, involves finding a transformation, e.g. Euclidean motion, that maximizes the size of two ε-congruent sub-sets. In the biological community, the LCP problem (under different metrics) is also called the *structure alignment* problem. For the *bottleneck* metric in 3D the LCP problem can be solved in $O(n^{32.5})$ time[7], where $n = max(|A|, |B|)$. Obviously, such time complexity is not practical even for small point sets. Therefore, more efficient methods are required.

Numerous heuristic methods have been developed to solve the structure alignment problem for a pair of protein structures (for more details see review by Eidhammer at el.[38]). Many of these heuristic approaches apply problem domain knowledge to improve the results as well as to speed up the computation times. Fast approximation algorithms have also been introduced to solve the more general geometrical problem. The approximation is given either to the similarity threshold or to the size of a common pattern[48, 1, 18].

The problem to detect a common point set for a set of K structures is the natural extension of the pairwise LCP problem. We denote this problem as the multiple LCP problem, or shortly mLCP. Due to the high practical importance, several heuristic solutions have been proposed[45, 3, 102, 124, 95, 76, 108, 31] even before the theoretical study of computational aspects of the mLCP problem by Akutsu and Halldorson[2]. It has been shown that the mLCP problem is NP-Hard even in the one dimensional space for the case of exact congruence ($\varepsilon = 0$), where the number of structures, K, is a parameter[2]. However, the mLCP problem is further complicated by the fact that in practice it is impossible to work with *zero-congruence*. In the case of $\varepsilon > 0$, the pairwise LCP problem is polynomial[7], while for three structures its complexity has been unknown. In this work we show that the mLCP problem in the case of $\varepsilon > 0$ is NP-Hard even for three structures. The problem is hard even in 2D, while in 1D we give a simple polynomial time algorithm.

In some applications, the Euclidean superposition between the input point sets may be given as the input, i.e. point sets are fixed in space. Therefore, under such conditions, the general mLCP problem is reduced to computing the largest common pattern without optimizing the molecular placement in space. We denote this problem as multiple similarity measure, or shortly mSim. There are several practical applications for this problem:

(I) Pharmacophore Extension. A set of drug molecules (or other molecules) may share an *a-priori* known common set of features which is responsible for the drug function. Such a common set is also called a *pharmacophore*. Therefore, a multiple superposition between the structures can be computed based on a given set of corresponding points - a *pharmacophore*. Still, there is an important question - what are

the additional corresponding (common) features that are shared between the set of drug molecules. Another, relevant question - is there a subset of molecules that share a larger pattern then the common pattern between all the input molecules? We call this problem *Family mLCP*.

(II) Ligand Based Multiple Binding Site Alignment. Consider a set of protein-ligand complexes, which consist of different proteins bound to the same ligand molecule. A common superposition between protein binding sites can be computed according to the common ligand molecule. Therefore, detection of a largest common protein binding site pattern is reduced to the mSim problem (though, as we discuss in Section 5.3.4, in some cases such an approach may be inferior to solving the general mLCP problem).

(III) Practical Approximation Algorithms. Another application of the mSim problem comes from the approximation algorithms used to solve the pairwise and multiple LCP problem. Such practically efficient methods compute a polynomial set of possible Euclidean transformations. Hence, given a bounded set of transformations the original mLCP problem is reduced to solving the number of the mSim problems.

For computational and practical reasons, we study two versions of the mLCP/mSim problem. In the first version, we require that the bottleneck metric is satisfied between all structure pairs. In the second version, we relax the similarity definition and require that the bottleneck metric is satisfied only between a *pivot* and each other structure. We call this version pmLCP/pmSim problem. The pmLCP/pmSim problem is computationally simpler than mLCP/mSim. In addition, the pmLCP/pmSim definition is more appealing for application in approximate alignment techniques, which are used in practice. In Chapter (5), we propose a practical method for multiple protein binding site alignment, using the pmLCP definition.

Tables 1.1 and 1.2 summarizes the new results presented in this work. Partial results of this Chapter have been published in [111, 112].

	$\mathbb{R}^1$	$\mathbb{R}^2$
$\varepsilon = 0$		
mSim/pmSim	$O(K \cdot n \cdot log(K \cdot n))$	
mLCP/pmLCP	NP-Hard[2] $O(n^{D \cdot K})$	
$\varepsilon > 0$		
ε-K-tuple	$O(K \cdot n \cdot log(K \cdot n))$	NP-Complete
mSim	$O(K \cdot n \cdot log(K \cdot n))$	NP-Hard (K≥3) $O(n^{1.5} log(n))$[37] (K=2)
pmSim	$O(K \cdot n \cdot log(K \cdot n))$	NP-Hard (K≥4) $O(n^3)$(K=3, solved by network flow)
mLCP	$O(K^2 n^{2K})$	NP-Hard (K≥3) $O(n^{32.5})$[7] (K=2)
pmLCP	$O(K^2 n^{2K})$	NP-Hard (K≥4) Poly(n) (K=3)
$\varepsilon = 0$		
Family mSim/pSim	NP-Hard $2^K Poly(n)$	

Table 1.1: mLCP(pmLCP) - multiple Largest Common Point set problem (pmLCP multiple LCP with pivot), both problems involve optimization on Euclidean transformations. mSim/pmSim - multiple similarity measure of point sets which are fixed in space; n - size of the largest point set; K - number of point sets. The unreferenced results are new and presented in this thesis.

	$\mathbb{R}^2/\mathbb{R}^3$	**Generalized Problems**
$\varepsilon > 0$, bounded vertex degree		
ε-K-tuple	γ-additive-approximation PTAS, Poly(n,K)	Maximum independent set in degree 3 graphs is APX-Complete[15]
mSim/pmSim	δ-approximation PTAS, Poly(n)	K-dimensional Matching in Hyper-graphs, $K = 3$, is APX-Hard[68, 55]
mLCP/pmLCP	δ-(γ-additive)-approximation PTAS, Poly(n)	$\varepsilon = 0$, cannot be approximated within $n^{1-\delta}$, for any $\delta > 0$[2].

Table 1.2: Approximation results of the geometrical problems studied in this thesis versus the generalized problems. Bounded vertex degree denotes a constant number of points from point set A inside a circle/sphere of radius ε around some point $a \in A$, for all $a \in A$ and for all input point sets. As it can be seen from the table, the geometrical properties of the studied problems allow development of polynomial-time approximation schemes (i.e. the problems adopt PTAS), while the general graph optimization problems are hard to approximate (a relation between the ε-K-tuple and the independent set problems can be seen in Section 2.3). The unreferenced results are new and presented in this thesis.

Chapter 3: Multiple Protein Structure Alignment

Recognition of a structural core common to a set of protein structures serves as a basic tool for the study of protein evolution and classification, analysis of similar structural motifs and functional binding sites, and for homology modeling and threading. Despite the computational theoretical infeasibility of the general mLCP problem, we present an efficient method, MultiProt, for the multiple protein structure alignment.

There are a number of practical requirements for a multiple protein structure alignment method. It should be able to detect (1) partial solutions, i.e. common domains or motifs which can be only parts of the proteins; (2) sequential and non-topological alignments, i.e. those alignments that do and do not follow the natural protein amino acid sequence order; and (3) alignments between a subset of molecules that are more similar than the whole input set. The fourth requirement is efficiency, i.e. low running times.

Our proposed method, MultiProt, was devised to support all the above requirements. Due to the NP-Hardness of the multiple alignment problem, discussed in Chapter (2), the method utilizes several heuristics, which are based on the problem domain knowledge.

The main idea of the MultiProt approach is its ability to efficiently compute a large number of local non-gapped multiple structure alignments. Essentially, a local multiple alignment is computed for each possible fragment of the input molecules. Such local alignments serve as a basis for the extension to the larger partial multiple alignments. In addition, in order to detect subset alignments, i.e. alignments composed of different molecules, we score multiple alignments separately according to the protein molecule composition. For example, a scoring of alignment between proteins $\{a, b, c\}$ does not effect a ranking of an alignments between proteins $\{a, b, d\}$.

We present case studies that demonstrate the ability of MultiProt to answer the practical requirements given above. Despite the complex task, MultiProt is extremely efficient and is suitable for simultaneous comparison of up to tens of proteins.

The results of this Chapter have been published in [107, 108, 105, 129, 109].

Chapter 4: Structure-Derived Multiple Sequence Alignment

Sequence alignment methods may produce inaccurate alignments due to low sequence identity. For proteins with solved 3D structure a structural superposition provides a basis for a more robust assessment of evolutionary relationships between amino acids. Yet, a structural 3D superposition does not uniquely define an alignment between protein sequences. There can be several choices for sequence alignments that are consistent with the structural superposition. Obviously, we would prefer an alignment with less gaps that also places more similar amino acid types, according to some substitution matrix, in the same column. Therefore, we face an optimization problem which is similar to the multiple sequence alignment problem but has additional spatial constraints.

We propose to perform a multiple sequence alignment that unifies structural information, derived from a multiple structure alignment, with amino acid substitution matrices. Since a protein structure is generally more conserved than sequence, we propose to perform an optimization of the multiple alignment, first, according to structure and then according to amino acid types combined with 3D information. Namely, we propose the following scheme: Given a set of protein structures, first, perform a multiple structural alignment. Second, based on the multiple structure superposition, perform a multiple sequence alignment optimizing our newly defined sequence-structure unified scoring function. Therefore, the presented optimization method, STACCATO, unifies sequence and structure information. The alignment score is based on standard amino acid substitution probabilities combined with newly computed 3D structure alignment probabilities. The advantage of our alignment scheme is in its ability to produce more accurate multiple alignments.

The presented test cases include comparison with the HOMSTRAD[89] benchmark of multiple structure based sequence alignments. We argue that our approach produces more accurate alignments. We present some applications of STACCATO, which include analysis of loop motion in Tyrosine Kinase and improving the accuracy of protein-protein docking methods.

The results of this Chapter have been published in [110, 109].

Chapter 5: Recognition of Common Binding Patterns

Binding sites with similar physico-chemical and geometrical properties may perform similar functions and bind similar binding partners. Such binding sites may be created by evolutionarily unrelated proteins that share no overall sequence or fold similarities. Their recognition has become especially acute with the growing number of protein-ligand complex structures determined by the Structural Genomics project. Multiple alignment of binding sites that are known to have similar binding partners allows recognition of the physico-chemical and geometrical patterns that are responsible for the binding. These patterns may help to understand and predict molecular recognition. Moreover, multiple alignment of binding sites allows analysis of the dissimilarities of the binding sites which are important for the specificity of drug leads.

We present an efficient, practical, method, MultiBind, for identification of common protein binding patterns by solving the multiple structure alignment problem. The problem we aim to solve is NP-Hard, therefore our goal is to find a trade-off between practical efficiency and theoretical bounds of solution accuracy, while, most importantly, validating the biological correctness of the results. We represent the protein binding site as a set of 3D points that are assigned a set of physico-chemical and geometrical properties important for protein-ligand interactions. The implementation of our method includes three major computational steps. The first one is a generation of 3D transformations that *align* the molecular structures. Here we apply the time efficient Geometric Hashing method[131, 130]. The advantage of this method is that it enables to avoid processing of points that can not be matched under any transformation. In other words, its time complexity is proportional to the number of potentially matched points included in the defined set of transformations. The second step is a search for a combination of 3D transformations that gives the highest scoring common 3D core. For this step we provide an algorithm that guarantees to find the optimal solution by applying an efficient filtering procedure which practically overcomes the exponential number of multiple combinations. The final step is a computation of matching between points under multiple transformations, namely solving the mSim problem. Here, we give a fast approximate solution with factor K.

The overall scheme guarantees to approximate the ε-congruence as well as the cardinality of multiple alignment. We apply MultiBind to some well studied biological examples such as estradiol, ATP/ANP and transition state analogues binding sites. Our computational results agree with the available biological data.

The results of this Chapter have been published in [111, 112, 118].

Program Availability

All the developed methods are freely available for the biological community by web server access (MultiProt) or by downloading the programs (Staccato, MultiBind, MAPPIS):

- http://bioinfo3d.cs.tau.ac.il/MultiProt/
- http://bioinfo3d.cs.tau.ac.il/staccato/
- http://bioinfo3d.cs.tau.ac.il/MultiBind/
- http://bioinfo3d.cs.tau.ac.il/mappis/

Chapter 2

Largest Common Point Set Problem

2.1 Introduction

Below we review the progress made in Computational Geometry in studying the pattern detection problem. We start with a description for the case of two structures and continue to multiple structures. Our emphasis is only on subjects related to molecular structures. First, we give a brief introduction to the computational problem and in the next section we introduce the formal definitions and known computational results.

The problem of pattern detection answers the following question. Given two point sets A and B, find a subset of A that is similar to some subset of B. The optimization problem of maximal *similarity* is to maximize the cardinality of similar subsets. One common way to define the similarity between two point sets is by the *bottleneck* metric[1, 37]. Similar sub-sets are called ε-congruent if the maximal distance between the matched points is not greater than ε. The optimization problem, called the *Largest Common Point Set* (LCP) problem, involves finding a transformation, e.g. Euclidean motion, that maximizes the similarity between A and transformed B.

In this work we consider the similarity problem and the LCP problem under the *bottleneck* metric only, if not specified otherwise. The similarity problem between

two point sets is equivalent to solving the maximal matching in a bipartite graph, where A and B define two partitions and edges are created between ε-close points. Indeed, a matching in such a graph defines a one-to-one correspondence between point pairs which are ε-close. A maximal bipartite matching can be solved in $O(n^{2.5})$[61], where $n = max(|A|, |B|)$. However, considering the geometrical properties of such bipartite graphs the maximal matching can be solved in 2D in time $O(n^{1.5}log(n))$ and in 3D in time $O(n^{\frac{11}{6}+\delta})$, for any $\delta > 0$[37]. Denote by $dens(\varepsilon)$ a number of points inside a sphere of radius ε, for any possible location in $\mathbb{R}^3$. Consider a practical case where $dens(\varepsilon)$ of A and B is a small constant (this is a valid assumption in case of molecular structures and practically meaningful values of ε). Under these conditions the maximal vertex degree in a bipartite graph is bounded by $dens(\varepsilon)$, i.e. constant. Then the number of edges is $O(n)$, so the running time of the maximal bipartite matching is only $O(n^{1.5})$[i].

The optimal solution to the LCP problem in 3D can be computed in $O(n^{32.5})$ time[7]. Obviously, such time complexity is not practical even for small point sets. Therefore, more efficient methods are required.

Approximation techniques can significantly reduce time complexity at the price of solution accuracy. A simple *alignment technique*[64] constructs a finite set of transformations by *aligning* each non-colinear triplet of points from the first structure with each ε-congruent triplet from the second one. For each transformation we can apply the maximal bipartite matching algorithm to compute a bijective mapping of points that are within ε distance from each other. Such *alignment technique* guarantees to find the LCP under 8ε-congruence of cardinality at least of the LCP under ε-congruence. The time complexity is $O(n^{8.5})$. This technique was first developed for the Hausdorff distance[48] and later applied for the *bottleneck* distance [1]. Instead of approximating ε-congruence the same technique can be applied to approximate the LCP size[18]. Interestingly, an optimal algorithm for solving the LCP problem for a group of transformations limited to rotations only, can improve the approximation factor of the LCP problem for general Euclidean transformations from 8ε to 2ε, while

[i]We will use the constant $dens(\varepsilon)$ assumption below to develop fast approximation schemes for the general LCP problem between K point sets.

preserving the complexity of $O(n^{8.5})$[18]. However, an implementation of such a technique is more complicated and the constant factors of the time complexity become larger.

Extension of the problem to detect a common point set between a set of K structures (from now on the term *point set* and *structure* will be used alternatively) has many important applications for the analysis of protein and drug molecules. However, even in the one dimensional space for the case of exact congruence ($\varepsilon = 0$) the problem is NP-Hard and it is hard to approximate within the factor of $n^{1-\delta}$, for any $\delta > 0$, where n is the size of the smallest structure[2]. The problem is further complicated by the fact that in practice it is impossible to work with *zero-congruence*. Therefore, we face another combinatorial sub-problem. Namely, given a set of superimposed structures, compute the largest common ε-congruent sub-set. We will call this problem *ε-K-partite* matching (in 2D or in 3D). While for two structures ($K = 2$) it can be solved by bipartite matching, for $K > 2$ structures it can be solved by *K-partite* matching. However, this problem is known to be NP-Hard even for $K = 3$ in graphs and hyper graphs[43, 55]. Here, we show that the *ε-3-partite* matching problem even in 2D is NP-Hard. As a result, we show that the common point set problem for $\varepsilon > 0$ is NP-Hard even for three structures. When K is a free parameter, even the most basic primitive of the *ε-K-partite* matching problem, i.e. detection of at least one ε-K-tuple, is NP-Hard. However, the geometrical properties of the studied problems allow development of polynomial-time approximation schemes (i.e. the problems adopt PTAS), while the general graph optimization problems are hard to approximate.

2.2 Preliminaries and Definitions

Below we list some of the notations which we alternatively use throughout the thesis:

Structure	Point set. We assume that a point set is unordered if not specified otherwise.
K	Number of input point sets.
n	Number of points in one point set. We assume that the input point sets are roughly of the same size, i.e. O(n).
Transformation	Euclidean rigid motion (rotation + translation).
Multiple structure alignment	Common point set detection.
1D 2D 3D	$\mathbb{R}^1$ $\mathbb{R}^2$ $\mathbb{R}^3$.
$dens(\varepsilon)$	Point density. Upper bound on the number of points inside a circle/sphere of radius ε. The upper bound is taken over all possible circle/sphere locations.

In this work we use the *bottleneck* metric to measure the similarity between point sets[1, 37]. The *bottleneck* distance between equally sized point sets S_1 and S_2 is less than ε if there exists a bijective mapping $m : S_2 \rightarrow S_1$ such that for each point $s \in S_2$, $d(m(s), s) \leq \varepsilon$, where $d(.,.)$ is the Euclidean metric. Two point sets S_1 and S_2 are ε similar if there exists an Euclidean transformation such that the *bottleneck* distance between S_1 and $T(S_2)$ is less than ε. For simplicity of other notations, which will be introduced shortly, we use slightly different definitions of distance and similarity, which are equivalent to the *bottleneck* distance and a similarity under the *bottleneck* metric. First, we define how we measure closeness and congruence between two point sets:

Definition 2.1. ε-closeness. *Two equally sized point sets $S_1 = \{a_1, ..., a_n\}$ and $S_1 = \{a_1, ..., a_n\}$ are* **ε-close** *if $d(a_i, b_i) \leq \varepsilon$ for each $i = 1, ..., n$, where $d(.,.)$ is the*

Euclidean metric.

Definition 2.2. **ε-congruence.** *Two equally sized point sets $S_1 = \{a_1, ..., a_n\}$ and $S_1 = \{a_1, ..., a_n\}$ are* **ε-congruent** *if there exists an Euclidean transformation T such that S_1 and $T(S_2)$ are ε-close.*

Next, we define optimization problems that search maximal size subsets that are ε-close or ε-congruent.

Problem 1. *Similarity Measure between 2 Sets.* Given $\varepsilon > 0$, and two point sets S_1 and S_2, find equally sized ordered subsets $S'_i \subseteq S_i$ (i=1,2) of maximal cardinality such that S'_1 and S'_2 are ε-close.

Problem 2. *Largest Common Point Set (LCP) between 2 Sets.* Given $\varepsilon > 0$, and two point sets S_1 and S_2, find equally sized ordered subsets $S'_i \subseteq S_i$ (i=1,2) of maximal cardinality such that S'_1 and S'_2 are ε-congruent.

The defined similarity measure problem is equivalent to solving the maximal matching in a bipartite graph, where S_1 and S_2 define two partitions and edges are created between ε-close points. A matching in such a graph defines a one-to-one correspondence between point pairs which are ε-close, i.e. it defines an order of ε-close subsets. An optimal solution can be computed in $O(n^{\frac{11}{6}+\delta})$, for any $\delta > 0$[37]. The LCP problem is complicated by an additional free parameter - the Euclidean transformation, still it can be exactly solved in $O(n^{32.5})$ time[7].

The idea behind the methods that solve exactly the LCP problem, can be best explained in case of Euclidean transformations limited to translations only. Let us define a space of all relevant translations that bring the points of S_2 into ε proximity of points from S_1. Let $T_{ij} = \{t \ : \ |(t + q_j) - p_i| \leq \varepsilon\}$, $p_i \in S_1$ and $q_j \in S_2$. T_{ij} is a ball of radius ε centered at $(p_i - q_j)$ and it defines all translations that bring point q_j to ε proximity of p_i. Consider an arrangement, A, defined on $\{T_{ij}\}$, $i = 1, ..., |S_1|$ and $j = 1, ..., |S_2|$. A cell, c, of this arrangement is the intersection of some $\{T_{i_k j_k}\}$. Any translation from c defines exactly the same bipartite graph of ε close points, consequently, any translation from c gives the same answer to the LCP problem. Therefore, it is enough to consider only one representative translation from each cell

of $A(\{T_{ij}\})$, and for each representative translation we solve the similarity measure problem, i.e. the maximal bipartite matching. The number of cells in A is $O(n^4)$ in 2D and $O(n^6)$ in 3D. For the Euclidean transformations the idea is essentially the same, though computing an arrangement in higher dimensions becomes more complicated. To conclude, while the LCP optimization problem involves an unbounded number of possible transformations, only a polynomial number induce combinatorially different instances of the LCP problem.

In this work we extend the similarity measure and the LCP problem to multiple sets and we call it the mSim and mLCP problem. We also define the multiple Sim/LCP problem with respect to a *pivot* structure and we call it the pmSim/pmLCP problem.

Problem 3. (mSim). *Similarity Measure between K Sets.* Given $\varepsilon > 0$, and K point sets S_i, $i = 1, ..., K$, find equally sized ordered subsets $S'_i \subseteq S_i$ $(i = 1, ..., K)$ of maximal cardinality such that each pair $(S'_i,\ S'_j)$ is ε-close $(i \neq j,\ i, j = 1, ..., K)$.

Problem 4. (pmSim). *Similarity Measure between K Sets with a Pivot Set.* Given $\varepsilon > 0$, a pivot set S_1 and $K - 1$ point sets S_i, $i = 2, ..., K$, find equally sized ordered subsets $S'_i \subseteq S_i$ $(i = 1, ..., K)$ of maximal cardinality such that each pair $(S'_1,\ S'_i)$ is ε-close $(i = 2, ..., K)$.

For two point sets the similarity measure problem is solvable by the maximal bipartite matching algorithm. Similarly, the multiple similarity requires solving the K-partite-matching problem in a K-partite graph defined on 3D structures, where edges between the nodes from different partitions (structures) are created if and only if the distance between the nodes is not greater than ε. Below we extensively use the notion of K-partite graph and K-partite-matching, therefore, let us give an equivalent definition to the mSim/pmSim problems using graph theoretic notations:

Definition 2.3. (ε-K-partite graph). *Given $\varepsilon > 0$ and K point sets S_i, $i = 1, ..., K$, an ε-K-partite graph $G(S_1, ..., S_K) = (V, E)$ is defined as $V = \cup_{i=1}^{K} S_i$ and $E = \{(p, q) : p \in S_i, q \in S_j, i \neq j, d(p, q) \leq \varepsilon\}$. A matching in an ε-K-partite graph is a set of disjoint K-tuples $\{(p_{t_1}, ..., p_{t_K}) : p_{t_i} \in S_i, p_{t_j} \in S_j, (p_{t_i}, p_{t_j}) \in E\}$.*

Definition 2.4. (ε-K-partite-pivot graph). *Given $\varepsilon > 0$ and K point sets S_i, $i = 1, ..., K$, of which S_1 is the pivot, an ε-K-partite-pivot graph $G(S_1, ..., S_K) = (V, E)$ is defined as $V = \cup_{i=1}^{K} S_i$ and $E = \{(p, q) : p \in S_1, q \in S_j, j > 1, d(p, q) \leq \varepsilon\}$. A matching of an ε-K-partite-pivot graph is a set of disjoint K-tuples $\{(p_{t_1}, ..., p_{t_K}) : p_{t_1} \in S_1, p_{t_j} \in S_j, (p_{t_1}, p_{t_j}) \in E\}$.*

In general graphs the K-partite-matching problem is NP-Hard even for three sets[43, 55]. Below, we show that it is still NP-Hard even for ε-K-partite and ε-K-partite-pivot graphs.

In case that K, the number of input structures, is a free parameter, we have another sub-problem of detecting at least one K-tuple of points which are pairwise ε-close:

Problem 5. (ε-K-tuple). Given $\varepsilon > 0$, and K point sets S_i, $i = 1, ..., K$, answer whether there exists a K-tuple $(p_1, ..., p_K)$, $p_i \in S_i$, $i = 1, ..., K$, such that $\forall i, j \quad |p_i - p_j| \leq \varepsilon$.

The ε-K-tuple problem appears as a basic part of the general optimization, ε-K-partite matching problem. For the case of ε-K-partite-pivot matching, detection of at least one ε-congruent K-tuple is trivially solved, by verifying that for each pivot point whether there is at least one ε-close point from each other structure.

Now, let us define the multiple LCP problems that include the optimization on Euclidean transformations.

Problem 6. (mLCP). *Largest Common Point Set between K Sets.* Given $\varepsilon > 0$, and K point sets S_i, $i = 1, ..., K$, find transformations T_i $(i = 2, ..., K)$ and equally sized ordered subsets $S'_i \subseteq S_i$ $(i = 1, ..., K)$ of maximal cardinality such that each pair $(T_i(S'_i), T_j(S'_j))$ is ε-close $(i \neq j,\ i, j = 1, ..., K)$, where T_1 is the identity transformation.[ii]

[ii]Notice that it is possible to use an alternative definition that looks for K subsets that are ε-congruent. Assume that the following pairs (A, B) (B, C) (A, C) are ε-congruent with their corresponding transformations T_{AB}, T_{BC} and T_{AC} that satisfy their ε-closeness. However, $T_{AB}(B)$ and $T_{AC}(C)$ are not necessarily ε-close. Therefore the two definitions are different. In this work we consider only the first definition.

Problem 7. (pmLCP). *Largest Common Point Set between K Sets with a Pivot Set.* Given $\varepsilon > 0$, a pivot set S_1 and $K-1$ point sets S_i, $i = 2, ..., K$, find transformations T_i $(i = 2, ..., K)$ and equally sized ordered subsets $S'_i \subseteq S_i$ $(i = 1, ..., K)$ of maximal cardinality such that each pair $(S'_1, T_i(S'_i))$ is ε-close $(i = 2, ..., K)$.

In case that K is a free parameter, not surprisingly, the mLCP and pmLCP problems are NP-Hard, even in the one dimensional space for the case of exact congruence, i.e. $\varepsilon = 0$ [2].

Before we give the complexity analysis of the multiple similarity measure and multiple common point set problems, we discuss the approximation algorithms for the LCP problem between two point sets.

2.2.1 LCP δ-(γ-additive)-approximation

Here we define approximation versions to the all optimization problems defined above, i.e. similarity measure and largest common point set problems.

Definition 2.5. δ-(γ-additive)-approximation, $\delta \geq 1$, $\gamma \geq 0$. If there exists a solution of size L with error ε, then a δ-(γ-additive)-approximation algorithm guarantees to return in polynomial time a solution of size at least $\frac{1}{\delta} \cdot L$ with error at most $(\varepsilon + \gamma)$.

We use the same definition for two point set as well as for multiple set versions of the problem.

Polynomial time γ-additive-approximation to the problem of LCP between two point sets is based on a simple alignment technique and works as follows. For each non-colinear triplet of points from S_1 and for each almost-congruent triplet from S_2 construct a 3D transformation that *aligns* the second triplet with the first one (it is enough to consider only pairs of triangles with maximal triangle side difference $\leq 2\varepsilon$). Apply this transformation to S_2 and construct a bipartite graph where the vertices are the points of S_1 and of transformed S_2, and edges are created between points with distance not greater than $\varepsilon + \gamma$. Apply a maximal bipartite matching algorithm[37] to compute the largest set of aligned points. The algorithm works in

$O(n^3 \cdot n^3 \cdot n^2) = O(n^8)$. For this method the approximation ratio depends on the alignment rule for the construction of a 3D transformation based on two triplets of points (p_1, p_2, p_3) and (q_1, q_2, q_3). To define the alignment rule we need to define appropriate local reference frames and their alignment.

Local Reference Frame, $LRF(p_1, p_2, p_3)$: Define a local right hand coordinate system s.t.: $p_1 = (0, 0, 0)$, $(p_2 - p_1)/|(p_2 - p_1)| = (1, 0, 0)$, $(p_2 - p_1) \times (p_3 - p_1)/|(p_2 - p_1) \times (p_3 - p_1)| = (0, 0, 1)$.

Alignment Rule: Define a transformation T' that superimposes $LRF(q_1, q_2, q_3)$ onto $LRF(p_1, p_2, p_3)$, i.e. $p_1 = T'(q_1)$, $(p_2 - p_1)/|p_2 - p_1| = (T'(q_2) - T'(q_1))/|T'(q_2) - T'(q_1)|$ and $sign((p_2 - p_1) \times (p_3 - p_1)) = sign((T'(q_2) - T'(q_1) \times (T'(q_3) - T'(q_1)))$

This alignment rule gives a $7 \cdot \varepsilon$-additive-approximation to the LCP problem [48, 1]. Instead of considering one transformation for each triangle pair, we can construct a larger set of transformations that approximate better an optimal transformation. The following simple method unifies several ideas for the solutions of the LCP problem in 2D. The first idea is based on an exact solution to LCP in case of rotations around some fixed point in 2D[18]. The second idea is based on the γ-additive-approximation, for any $\gamma > 0$, in case of Euclidean transformations[56]. Here we extend it to 3D.

Construct a 3D grid, G_1, around point p_1 with side length $O(\gamma)$ (for the sake of clear description we neglect small constant factors). Consider only the grid points that are $\varepsilon + \gamma$ close to p_1. The number of grid points of G_1 is $O((\frac{\varepsilon}{\gamma})^3)$. Then, for each grid point $g_1 \in G_1$ consider a sphere, $S_{|q_1 - q_2|}(g_1)$, of radius $|q_1 - q_2|$ centered at g_1. Construct a 2D grid, $G_2(g_1)$, that covers the part of sphere $S_{|q_1 - q_2|}(g_1)$ that is inside the sphere $S_{\varepsilon+\gamma}(p_2)$. The distance between the grid points of G_2 is $O(\gamma)$. The number of grid points of $G_2(g_1)$ is $O((\frac{\varepsilon}{\gamma})^2)$. For each $g_1 \in G_1$ and each $g_2 \in G_2(g_1)$ create a transformation that aligns q_1 with g_1 and q_2 with g_2. The total number of combinations of grid point pairs is $O((\frac{\varepsilon}{\gamma})^5)$. One degree of freedom is left unspecified, i.e. rotation around the axis (q_1, q_2). To solve this, we use the optimal method similar to the one used for a rotation in 2D[18]. Arbitrarily select some vector perpendicular

to (q_1, q_2) to indicate a reference point (i.e. zero angle) for rotations around (q_1, q_2). For each pair (s_1, s_2), $s_1 \in S_1$ and $s_2 \in S_2$, compute a pair of angles $[\theta_1, \theta_2]$, such that when rotating s_2 around (q_1, q_2) point s_2 enters the sphere $S_{\varepsilon+\gamma}(s_1)$ at angle θ_1 and exists the sphere at angle θ_2. The arrangement of all such pairs on a 1D fragment $[0, 2\pi]$ defines all relevant rotational angles. The number of points in this rotational arrangement is $O(n^2)$. Therefore it is enough to consider only $O(n^2)$ rotation angles to solve the LCP problem. Instead of applying the maximal bipartite matching from scratch for each rotation angle, notice the following fact. Each time we move from one cell of the arrangement to it's neighbor cell, only one edge in the bipartite graph is changed. In such a case, updating the maximal matching is equivalent to finding one augmenting path, which costs $O(n \cdot log(n))$[37]. Therefore solving the LCP for rotations around (g_1, g_2) takes $O(n^3 log(n))$, which is composed of (1) $O(n^2 log(n))$ computing angle events and sorting them, (2) $O(n^2)$ for finding the initial bipartite matching at the zero angle, (3) updating the bipartite matching $O(n^2)$ times, when each time costs $O(n \cdot log(n))$.

Theorem 2.6. *In 3D the γ-additive-approximation to the LCP problem between two point sets can be computed, for any $\gamma > 0$, in time $O((\frac{\varepsilon}{\gamma})^5 n^7 log(n))$.*

Notice, that the exact as well as the approximate algorithms use an approach that decompose the LCP problem to (i) computing a polynomial number of relevant transformations and (ii) given a superposition solving the similarity measure problem.

2.3 ε-K-tuple Problem

In one dimensional space the ε-K-tuple problem can be efficiently solved by finding the maximal number of ε-K-tuples, i.e. solving the optimization problem of 1D-ε-K-partite matching (see Section 2.4.1). However, in 2D the problem is NP-Complete.

Theorem 2.7. *The ε-K-tuple problem in 2D is NP-Complete.*

Proof. The problem is in NP, since verification that some K-tuple is ε-K-tuple can be done efficiently. We make a reduction from the 3-SAT problem. First, we briefly

sketch the reduction from 3-SAT to the *Clique* problem (for more details see [43]), and then we explain the necessary changes for the 3-SAT to ε-K-tuple reduction.

An instance of the 3-SAT problem includes a set of variables $U = \{u_1, u_2, ..., u_n\}$ and a set of clauses $C = \{c_1, c_2, ..., c_m\}$. Each clause contains three literals of variables U. A graph $G = (V, E)$ is constructed as follows. For each variable $u_i \in U$ create two vertices $v^U_{u_i}$ and $v^U_{\bar{u}_i}$. For each clause $c_j = (a_j, b_j, c_j)$ create three vertices $v^C_{a_j}$, $v^C_{b_j}$ and $v^C_{c_j}$. Therefore in total, there are $2n + 3m$ vertices in graph G. The edges are created as follows:

$$\begin{aligned} E &= \{(v^U_{u_i}, v^U_{u_j}) : i \neq j\} \\ &\cup \{(v^U_{\bar{u}_i}, v^U_{\bar{u}_j}) : i \neq j\} \\ &\cup \{(v^U_{u_i}, v^U_{\bar{u}_j}) : i \neq j\} \\ &\cup \{(v^C_{a_i}, v^C_{b_j}) : i \neq j\} \\ &\cup \{(v^U_{a_i}, v^C_{b_j}) : a_i \neq b_j\}, \end{aligned}$$

i.e. the created graph G is a complement[iii] of a graph $\bar{G}$ with edges created between (i) $v^U_{u_i}$ and $v^U_{\bar{u}_i}$; (ii) vertices from the same clause c_j and (iii) variable v^U and its corresponding literals from the clauses v^C. Therefore, C is satisfiable if and only if G has a clique of size $n + m$. Such a clique will contain exactly one vertex from each $\{v^U_{u_i}, v^U_{\bar{u}_i}\}$ and exactly one vertex from each $\{v^C_{a_j}, v^C_{b_j}, v^C_{c_j}\}$. The unselected vertices of type v^U correspond to the truth assignment in C. Alternatively, C is satisfiable if and only if the graph $\bar{G}$ has a vertex cover of size $n + 2m$[43], when selected vertices of type v^U correspond to the truth assignment.

Now, our aim is to transform the graph G to a 2D-ε-K-partite type graph, so that the original edges are preserved. First, we group the vertices into partitions. Each variable and its complement, $\{v^U_{u_i}, v^U_{\bar{u}_i}\}$, create a partition. Each clause $\{v^C_{a_j}, v^C_{b_j}, v^C_{c_j}\}$ create a partition. Therefore in total, there are $K = n + m$ partitions. Notice, according to the definition of ε-K-partite graph, no edges are created between the

[iii] We consider the complement graph $\bar{G}$ since, originally, it is used in the reduction of 3-SAT into the *Node Cover* problem. Then, the *Node Cover* is reduced into the *Clique* problem[43].

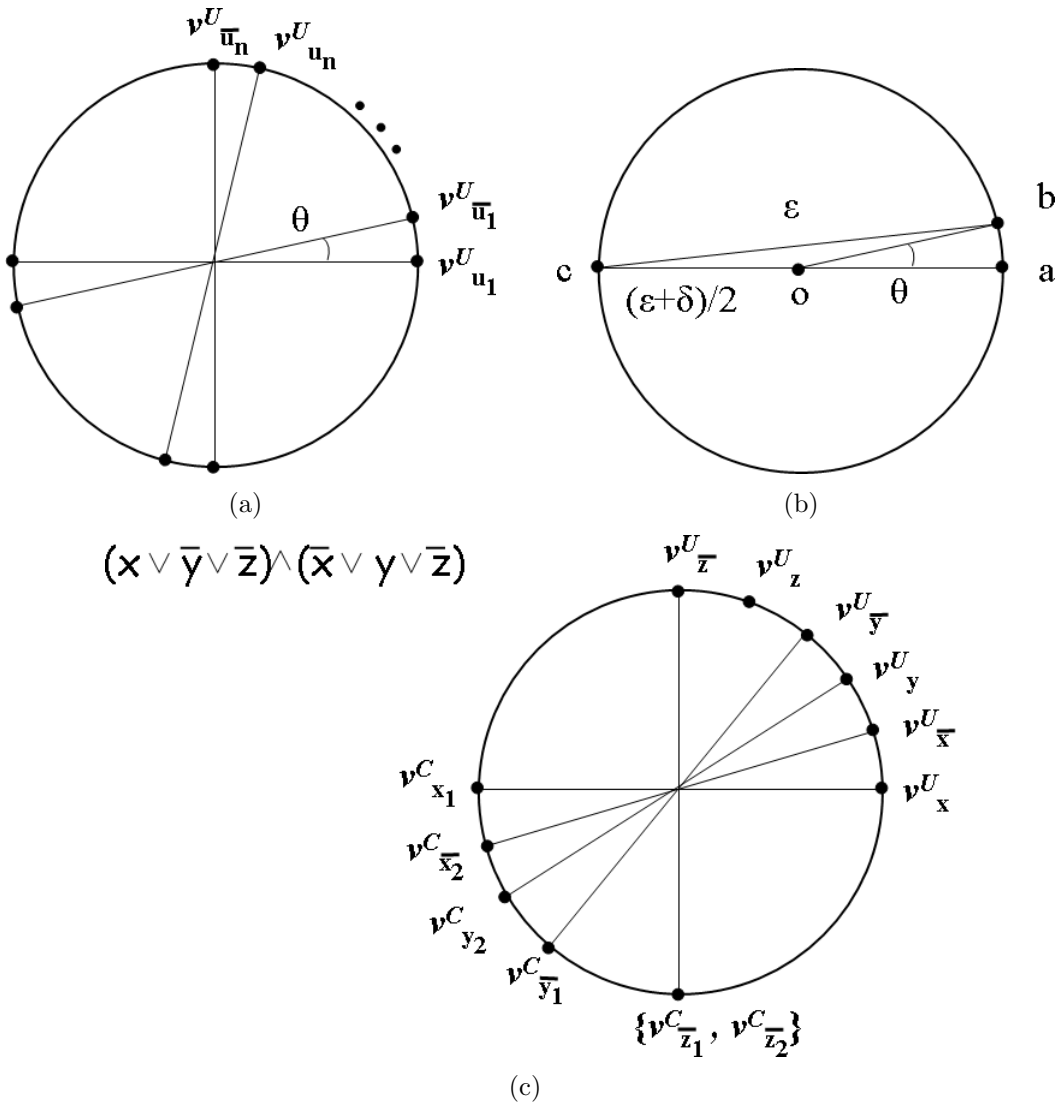


Figure 2.1: Reduction from 3-SAT to ε-K-tuple. (a) Construction of an ε-K-partite graph. The vertices are positioned on a circle with diameter $\varepsilon + \delta$, $\delta > 0$. The pairs $\{(v^U_{a_i}, v^C_{b_j}) : a_i = b_j\}$ are placed diametrically opposite. If a variable appears in several clauses, the equal literals are assigned the same coordinate. (b) The circle diameter is set to $\varepsilon + \delta$, $ao = bo = co = (\varepsilon + \delta)/2$ and $bc = \varepsilon$. Given θ and ε we compute the desired δ from the cosine rule, i.e. $\varepsilon^2 = (\varepsilon + \delta)^2(1 - cos(\pi - \theta))/2$. (c) Example of a geometrical construction.

vertices from the same partition, even if the vertices are ε-close.

The geometrical construction is depicted in Figure 2.1. The vertices are positioned on a circle with diameter $\varepsilon + \delta$, $\delta > 0$. The pairs $\{(v^U_{a_i}, v^C_{b_j}) : a_i = b_j\}$ are placed diametrically opposite. If a variable appears in several clauses, the equal literals are assigned the same coordinate. Therefore, since the circle diameter is greater than ε, there are no edges between the variables and its occurrences in the clauses, exactly as in the original graph G.

Specifically, we assign the coordinates to the vertices as follows. We uniformly map vertices $\{v^U\}$ into the circle arc $[0, \pi/2]$, i.e. vertex $v^U_{u_i}$ is mapped at angle $(2i-2)\theta$ and vertex $v^U_{\bar{u}_i}$ is mapped at angle $(2i-1)\theta$. Then, the corresponding literals are mapped at the diametrically opposite side. Our aim is to select θ and δ so that only the diametrically opposite point pairs are located at distance greater than ε and each other pair is located at distance less or equal to ε. The following assignment satisfies such requirements:

$$\begin{aligned} \delta &= \varepsilon\left(\sqrt{\frac{2}{1 - sin(\theta)}} - 1\right), \\ \theta &= \frac{\pi/2}{2n-1}. \end{aligned}$$

Therefore, this geometrical construction, along with the vertex partition, preserves the original edges of the graph G. Consequently, the 3-SAT instance C is satisfiable if and only if the constructed 2D-ε-K-partite graph has an ε-K-tuple. Such a ε-K-tuple corresponds to a clique of size $n + m$ in the original graph G. □

Computing exactly an ε-K-tuple is hard, therefore we are interested in looking for polynomial time algorithms that compute approximate solutions. The ε-K-tuple problem can be considered as a constrained case of a more general problem - largest clique detection. If we are able to compute the largest clique then we are able to solve the ε-K-tuple problem as well. The largest clique is hard to approximate, essentially it cannot be approximated within the factor of $n^{1-\delta}$, for any $\delta > 0$ [54]. In the above reduction, the problem may look simpler, since each vertex is connected to almost any

other vertex. However, for the above reduction we can use an NP-Complete variant of the 3-SAT problem where each variable is restricted to appear at most three times and each literal at most twice. Hence, in the above reduction the maximal vertex degree in graph $\bar{G}$ is three. Even considering such constraints the maximum independent set in degree 3 graphs is APX-Complete[15]. Instead of approximating the size of an ε-K-tuple, we define another approximation criterion, which will allow us to devise a polynomial time approximation scheme.

Definition 2.8. ε-K-tuple γ-additive-approximation. If there exists an ε-K-tuple, then a γ-additive-approximation algorithm guarantees to return in polynomial time an $(\varepsilon + \gamma)$-K-tuple.

An algorithm that achieves an ε-additive-approximation is based on a simple fact that a maximal distance between any two points in a circle/sphere of radius r is at most $2r$. Therefore, it is enough to detect a circle/sphere of radius ε that encloses K points from different point sets. If such enclosing circle/sphere exists, then, the detected K-tuple is a 2ε-K-tuple. Obviously, if an ε-K-tuple exists, then the algorithm detects at least one 2ε-K-tuple. The algorithm works as follows. Construct a circle/sphere of radius ε around each point. Compute an arrangement of these circles/spheres. A cell of this arrangement contains an intersection of some circles/spheres. If some cell contains an intersection of K circles/spheres each coming from a different point set, then a 2ε-K-tuple is detected. Construction and traversal of an arrangement of circles (spheres) takes $O(K^2n^2)$ ($O(K^2n^2\lambda_6(K \cdot n))$, where $\lambda_6(K \cdot n)$ is nearly $O(K \cdot n)$) time [52] and this is the running time of the algorithm.

The ε-additive-approximation can be improved by noticing that a smaller circle/sphere can be used such that it guarantees to enclose any ε-K-tuple. The maximal enclosing circle/sphere is achieved in case when some three/four points from a ε-K-tuple are located at distance ε one from each other, i.e. they form an equilateral triangle / tetrahedron (for details see Appendix 2.8). Therefore, the diameter of a circle/sphere that encloses an equilateral triangle / tetrahedron with side lengths ε is $\frac{2}{\sqrt{3}}\varepsilon$ and $\frac{\sqrt{6}}{2}\varepsilon$ correspondingly.

Theorem 2.9. *In 2D the $0.16 \cdot \varepsilon$-additive-approximation to the ε-K-tuple problem*

can be computed in 2D in time $O(K^2n^2)$. In 3D the $0.225 \cdot \varepsilon$-additive-approximation to the ε-K-tuple problem can be computed in time $O(K^2n^2\lambda_6(K \cdot n))$, where $\lambda_6(K \cdot n)$ is nearly $O(K \cdot n)$.

Next, we show that the ε-K-tuple problem adopts PTAS (there is a polynomial γ-additive-approximation algorithm for any $\gamma > 0$), and the approximation algorithm is fixed-parameter tractable (FPT), i.e. polynomial in both n and K.

Theorem 2.10. *The γ-additive-approximation to the ε-K-tuple problem can be computed, for any $\gamma > 0$, in 2D in time $O(n \cdot min[2^{(\varepsilon/\gamma)^2}, K(\varepsilon/\gamma)^{2K}] \cdot ((\varepsilon/\gamma)^2 + K))$, and in 3D in time $O(n \cdot min[2^{(\varepsilon/\gamma)^3}, K(\varepsilon/\gamma)^{3K}] \cdot ((\varepsilon/\gamma)^3 + K))$.*

Proof. For each point $s \in S_1$ perform the following steps. Consider only the points located in the circle of radius ε around the point s. From the first partition consider only s. Transform the selected points so that s is at the origin.

For the case of 2D, construct a grid with side length $\gamma/\sqrt{2}$, such that the grid vertices have coordinates $(i \cdot \gamma/\sqrt{2}, j \cdot \gamma/\sqrt{2})$, $i, j \in \{-\lceil\varepsilon\sqrt{2}/\gamma\rceil, ..., 0, ..., \lceil\varepsilon\sqrt{2}/\gamma\rceil\}$. Map each input point to the nearest grid vertex. Each point position is changed by at most $\gamma/2$. Consider each possible combination of the grid vertices. There are $O(2^{(\varepsilon/\gamma)^2})$ combinations. For each combination verify whether each grid vertex pair is within $\varepsilon + \gamma$ distance and whether the grid vertices have at least one point from each partition S_i, $i = 1...K$. If yes, then return an $(\varepsilon + \gamma)$-K-tuple. Given a combination of grid vertices, the verification time is $O((\varepsilon/\gamma)^4 + (\varepsilon/\gamma)^2K)$. However, since the vertex combinations can be constructed iteratively, so that only one vertex is changed between each combination, then the verification time is only $O((\varepsilon/\gamma)^2 + K)$. Clearly, if an ε-K-tuple exists, then the algorithm detects at least one $(\varepsilon+\gamma)$-K-tuple.

In case that K is small relative to $(\varepsilon/\gamma)^2$, then it is enough to verify only the combinations of K or less grid vertices, since each grid vertex should contribute at least one point to a $(\varepsilon + \gamma)$-K-tuple. Therefore, the number of combinations is $min(O(2^{(\varepsilon/\gamma)^2}), O(K(\varepsilon/\gamma)^{2K}))$.

In 3D, we construct a grid with side length $\gamma/\sqrt{1+\sqrt{2}}$. The number of vertices is $O((\varepsilon/\gamma)^3)$. The rest is the same as in the case of 2D. $\square$

2.4 ε-K-partite Matching

Here, we prove that the problems of ε-K-partite and ε-K-partite-pivot maximal cardinality matching are NP-Hard, even when K is a small constant. For the case of 3D-ε-3-partite graphs, we give an explicit geometrical construction of the reduction. Next, we show an even stronger result that the problems of 2D-ε-3-partite and 2D-ε-4-partite-pivot maximal cardinality matching are NP-Hard even in 2D. However an explicit geometrical construction becomes more complicated, therefore we provide only a schematic proof. Before we proceed to the 2D/3D cases, we consider the 1D case, which turns out to have a polynomial time complexity for any K.

2.4.1 1D-ε-K-partite-(pivot) Matching

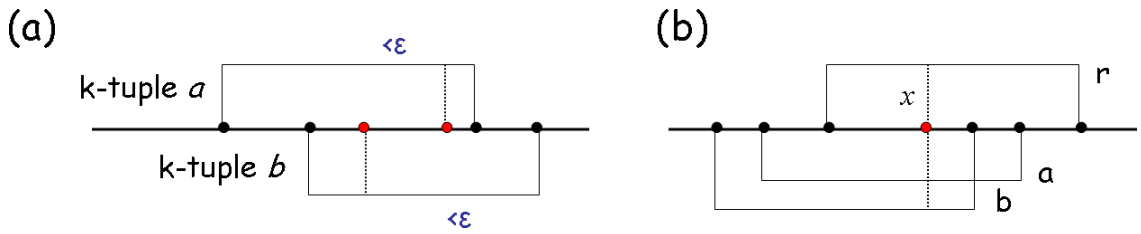


Figure 2.2: (a) a and b are overlapping K-tuples. Black points can be any color besides red. Since a has selected a red point with a coordinate larger than a red point from b, then these K-tuples are not sequential. The points that contradict sequentiality can only be located in the overlapping region. Therefore these points can be exchanged between a and b while not invalidating the ε proximity constraint. (b) K-tuple r has the largest coordinates in all its colors. If K-tuples a and b are sequential and belong to the maximal matching then a point x of red color cannot be selected by r and by b at the same time (though it can be selected by r and by a).

To simplify the notations, when referring to a K-tuple we will assume that all its points satisfy the ε proximity constraint. Each partition can be associated with a different color. When referring to two points with the same color we mean that they belong to the same partition. We make the following observation:

Definition 2.11. A matching $M = \{(p_{t_1}, ..., p_{t_K})\}$ is called *sequential* if for each pair of K-tuples $(p_{t_1}, ..., p_{t_K}) \in M$ and $(p_{s_1}, ..., p_{s_K}) \in M$ either $\forall j\ p_{t_j} \leq p_{s_j}$ or $\forall j\ p_{t_j} \geq p_{s_j}$.

Lemma 2.12. *For any matching there exists a sequential matching of the same size.*

Proof. Consider any two K-tuples that contradict the sequential constraint. The contradiction can occur only in the overlapping region (a *region* of a K-tuple is defined between its leftmost and rightmost points, see Figure 2.2(a)). Since the overlapping region belongs to both K-tuples it fulfills the ε proximity constraint for both K-tuples. Thus, we can exchange between the points from the overlapping regions, satisfying the sequential constraint. In this way we may satisfy the sequential constraint for any given K-tuple pair. However, satisfying the K-tuples a and b, and then b and c may result that a and b are again unsatisfied. Therefore, it is left to show that this exchange process for all K-tuple pairs will stop. Consider a K-tuple, which contains the rightmost point. After performing all exchanges of this K-tuple with others, all of its K points will have the largest coordinate. In the same manner repeat the process for the rest of the K-tuples. □

Lemma 2.13. *Let r be the K-tuple such that its points have the largest coordinate. For any maximal matching M there exists a matching of the same size, which includes r.*

Proof. According to Lemma 2.12 we assume that M is sequential. At least one of the points of r is selected by M, otherwise M is not maximal. Assume that K-tuples a, b from M share some points with r, and the coordinates of the corresponding points of a are larger than of b (see Figure 2.2(b)). We will show that r can share points with at most one K-tuple, namely a, thus exchanging a with r proves the lemma. Consider the red point x from the K-tuple r (see Figure 2.2(b)). Assume to the contrary that x is selected by b. Since K-tuple a is to the right of b, then a should select some red point which has larger coordinate than x. Thus, it contradicts that r has selected a red point with the largest coordinate. Therefore, no two K-tuples from a maximal sequential matching can share points with r. □

According to Lemma 2.12 and Lemma 2.13 we can use the following simple algorithm to compute the largest matching. Iteratively select the rightmost K-tuple until no selection is possible. We will maintain pointers to the rightmost unselected point

of each sorted set S_i, $i = 1, ..., K$ (sorting takes $O(K \cdot n \cdot log(K \cdot n))$). At each step we consider K unselected points, one from each set. To answer whether these points are within ε distance we need to know the rightmost and the leftmost point, therefore we maintain a *heap* data structure, such that inserting a new point costs $O(log(K))$ and querying the rightmost and the leftmost points costs $O(1)$. If the K points are within ε distance, we select them, and move the pointers in each set S_i to the next, left point. Otherwise, we discard the rightmost point (among the current K points) and move the pointer in the corresponding set to next point. Each such step will cost $O(log(K))$ to update the heap. Therefore, during the whole course of the algorithm for each point we perform $O(log(K))$ operations.

The above lemmas are also valid for the 1D-ε-K-partite-pivot matching. The iterative algorithm is similar to the previous one. Instead of verifying the distance between the leftmost and rightmost point for the current iterating K points, $(p_1, ..., p_K)$, we check the distance to the current pivot point p_1. In case that the leftmost, non pivot, point from $(p_1, ..., p_K)$ is at a distance larger than ε from p_1 and is located to the left of p_1 we discard p_1 and select the next, left pivot point. Otherwise, we consider the rightmost non pivot point p_i from $(p_1, ..., p_K)$. If its distance to the pivot is not greater than ε, then select the current K-tuple and consider next K points to the left. Otherwise, select next p_i and repeat the process. This way we spend time proportional to the total number of points multiplied by $O(log(K))$ to maintain the leftmost and the rightmost point.

Theorem 2.14. *The maximal 1D-ε-K-partite/1D-ε-K-partite-pivot matching can be computed in $O(K \cdot n \cdot log(K \cdot n))$ time.*

2.4.2 3D-ε-K-partite Matching

Theorem 2.15. *The perfect 3D-ε-3-partite matching problem is NP-Complete.*

Proof. First, we briefly present a reduction from *3-SAT* to *3-partite matching* in general graphs (for more details see [43]), and then extend it to the instances of 3D-ε-K-partite graphs.

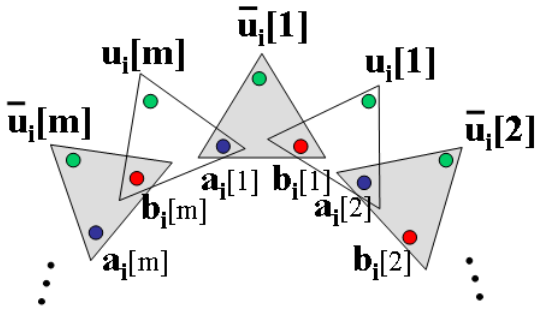


Figure 2.3: Schematic representation of the component T_i[43]. Any perfect matching will have to include either all gray triangles, and thus setting u_i to true, or all white triangles, and thus setting u_i to false.

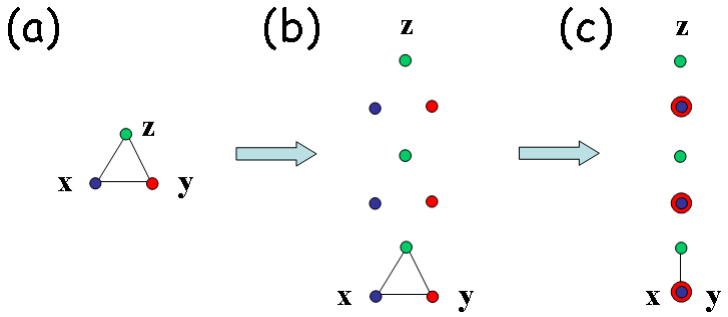


Figure 2.4: (a) $z \in S_1$, $x \in S_2$ and $y \in S_3$. The distance between each pair of points is ε. (b) *Long-distance-edge* gadget. Original triplet (x,y,z) can be elongated to any distance by adding additional triplets at ε distance. The construction preserves the matching choices of the original triplet. (c) Simplified *long-distance-edge* gadget. The nodes of partitions S_2 and S_3 can be assigned the same Euclidean coordinate. This can be done only for nodes from the components C and G.

An instance of the 3-SAT problem includes a set of variables $U = \{u_1, u_2, ..., u_n\}$ and a set of clauses $C = \{c_1, c_2, ..., c_m\}$. Each clause contains three literals of variables U. The goal of the reduction is to construct three disjoint sets S_1, S_2 and S_3 of equal cardinality, and a set of edges $M \subseteq S_1 \times S_2 \times S_3$ such that M contains a perfect matching if and only if C is satisfiable.

Three classes of edges are created, T - "truth setting and fan-out", C - "satisfaction testing" and G - "garbage collection". The components of T are constructed for each variable u_i. Denote $u_i[j]$ to be a variable u_i in clause j.

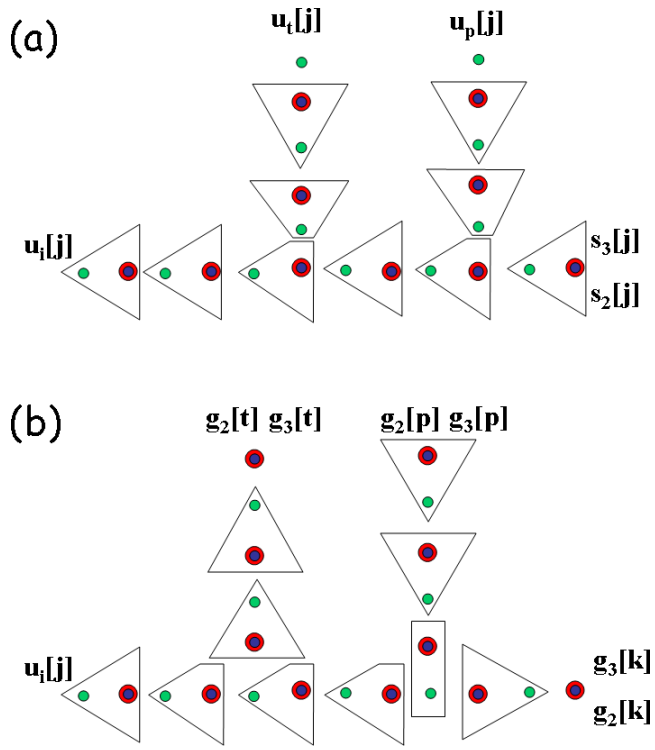


Figure 2.5: (a) *Split* gadget. It guarantees that a pair (s_2, s_3) or (g_2, g_3) will select exactly one node u. One such selection is depicted. The pair $(s_2[j], s_3[j])$ selects $u_i[j]$ while the nodes $u_t[j]$ and $u_p[j]$ should be selected by someone else. (b) *Join* gadget. It guarantees that a node $u[j]$ can be selected by at most one pair of type (s_2, s_3) or (g_2, g_3).

$$
\begin{aligned}
T_i^t &= \{(\bar{u}_i[j], a_i[j], b_i[j]) : 1 \le j \le m\} \\
T_i^f &= \{(u_i[j], a_i[j+1], b_i[j]) : 1 \le j \le m-1\} \cup \{(u_i[m], a_i[1], b_i[m])\} \\
& \quad \bar{u}_i[j], u_i[j] \in S_1, \quad a_i[j] \in S_2, \quad b_i[j] \in S_3
\end{aligned}
$$

The component T forces a matching to choose between setting u_i true and setting u_i false. Any perfect matching will have to include either all triplets from T_i^t or all triplets from T_i^f, see Figure 2.3. Next, for each clause c_j a component C_j aims to select a truth setting for one of its three literals:

$$\begin{aligned} C_j &= \{(u_i[j], s_2[j], s_3[j]) : u_i \in c_j\} \cup \{(\bar{u}_i[j], s_2[j], s_3[j]) : \bar{u}_i \in c_j\}, \\ & s_2[j] \in S_2, \quad s_3[j] \in S_3. \end{aligned}$$

Thus, only one triplet can be contained in any matching assigning the clause c_j to true setting.

Finally, the "garbage collection" component aims to compensate the unequal number of nodes created so far in S_1 and in the other two partitions S_2 and S_3:

$$\begin{aligned} G &= \{(u_i[j], g_2[k], g_3[k]), (\bar{u}_i[j], g_2[k], g_3[k]) : \\ & 1 \leq k \leq m(n-1), 1 \leq i \leq n, 1 \leq j \leq m\}, \\ & g_2[j] \in S_2, \quad g_3[j] \in S_3. \end{aligned}$$

To summarize, the edges are defined as: $T = \cup_{i=1}^{n}(T_i^t \cup T_i^f)$, $C = \cup_{j=1}^{m} C_j$, $M = T \cup C \cup G$. This completes the reduction from *3-SAT* to *3-partite matching*. Next, we adapt the above reduction for 3D-ε-3-partite type graphs.

Notice that the constructed graph M does not belong to the 3D-ε-3-partite type of graphs. Only the component T can be drawn in 2D to satisfy this property, i.e. only the point triplets from T can be placed within ε distance one from each other (see Figure 2.3)[iv]. The problem is that the nodes of type s_2, s_3 and g_2, g_3 can not be placed in 3D so that their distance from the different nodes of type u_i is not greater than ε. To resolve this problem we introduce the *long-distance-edge* gadget. The basic principle is illustrated in Figure 2.4. The edge (x, y, z) can be elongated to any distance preserving the property for matching.

The second gadget aims to *split* edges going from nodes of type s_2, s_3 (g_2, g_3). Assume we have three edges $(u_i[j], s_2[j], s_3[j])$, $(u_t[j], s_2[j], s_3[j])$ and $(u_p[j], s_2[j], s_3[j])$. Figure 2.5(a) illustrates how these three edges can be constructed. Triangles illustrate

[iv]In the *3-SAT* to *3-partite matching* reduction, the definition of a hypergraph edge as a triplet of points (a,b,c) is equivalent to three edges (a,b), (a,c) and (b,c) in a regular graph.

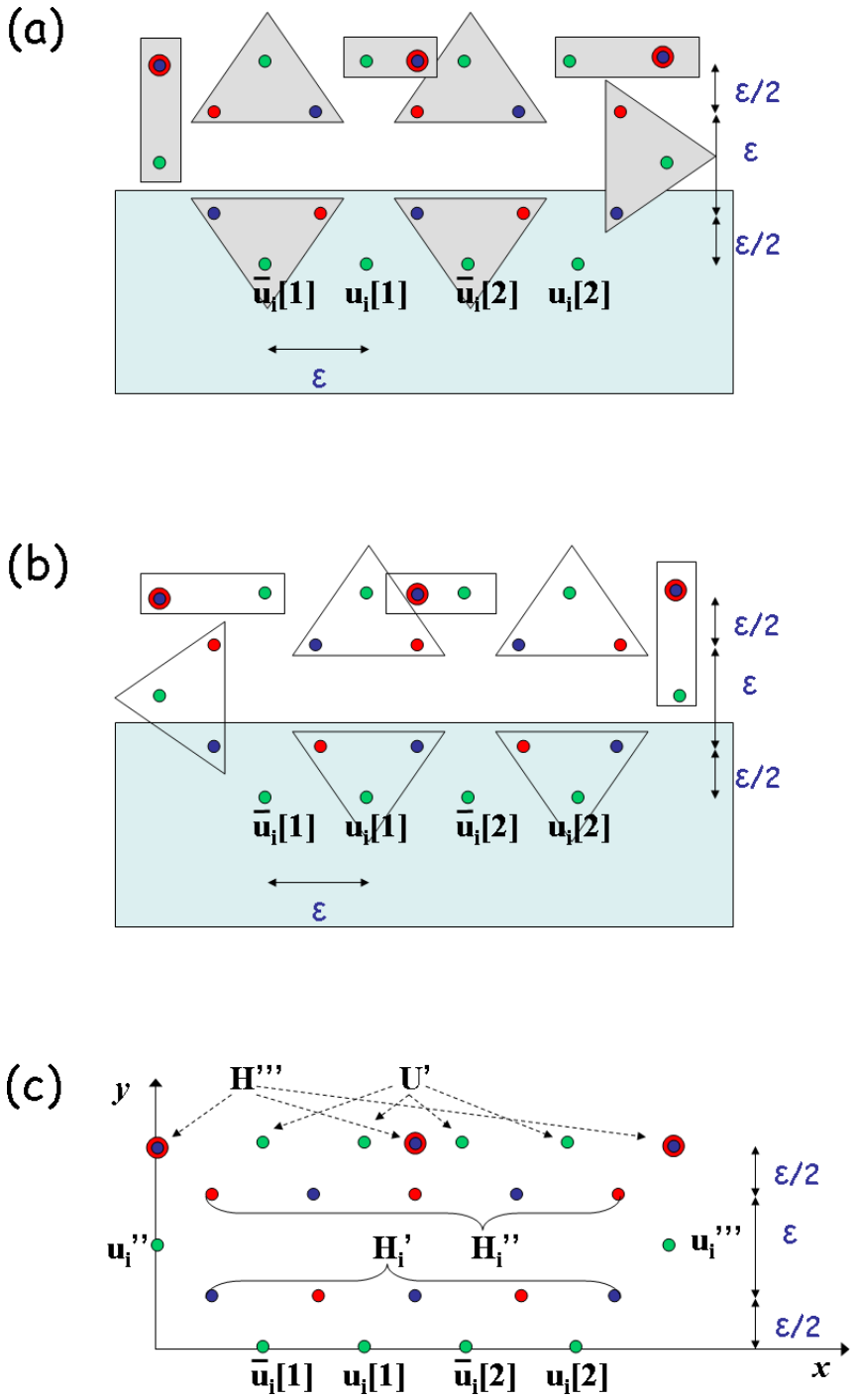


Figure 2.6: Component T_i is extended with additional nodes. The original nodes $\bar{u}_i$ and u_i are placed on line $(x, 0, 0)$. Figures (a) and (b) represent the only two choices for matching. The original rule of truth assignment for $\bar{u}_i$ and u_i nodes is preserved. (c) New node name assignments.

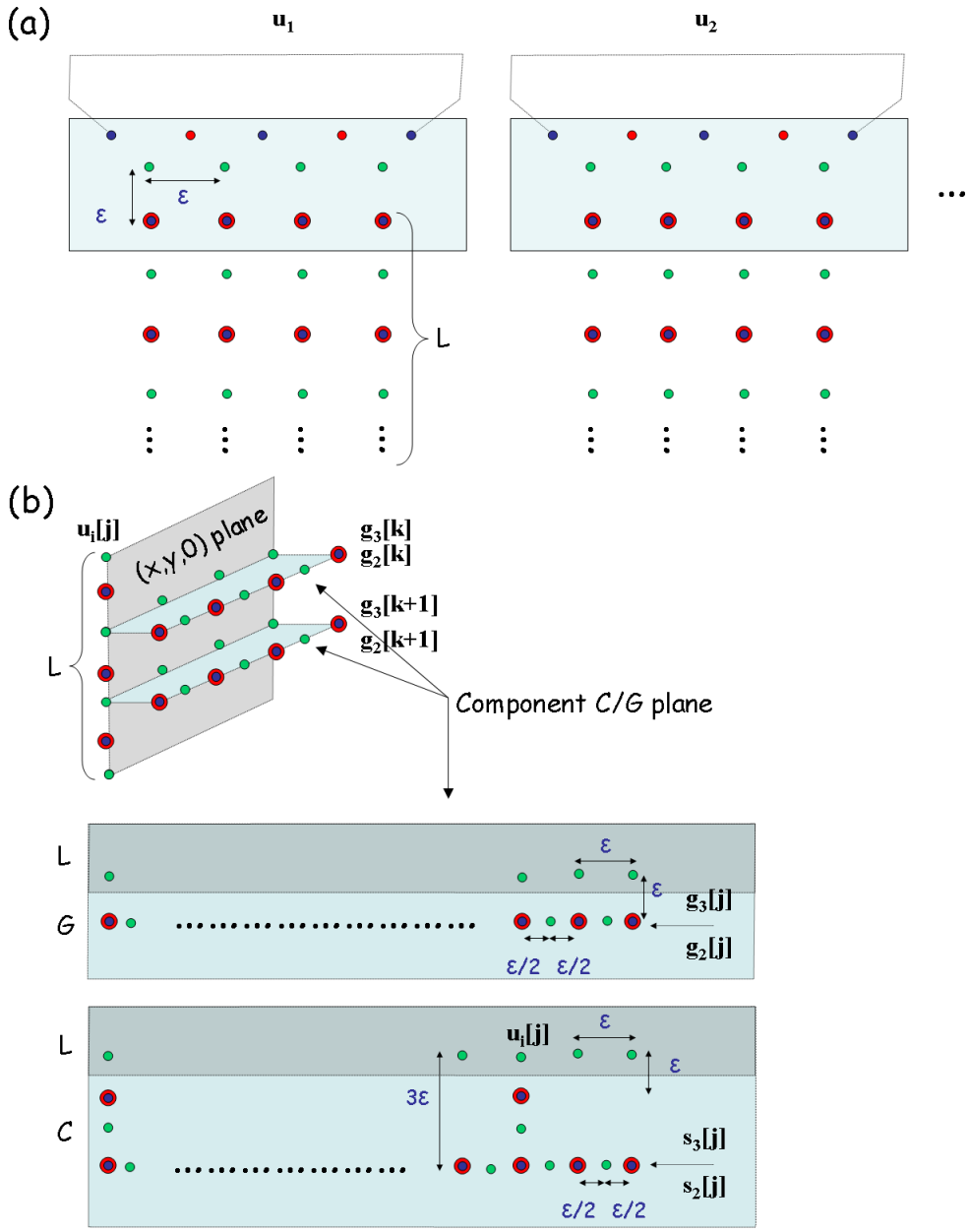


Figure 2.7: (a) The components T_i are placed in the plane $(x, y, 0)$. The *long-distance-edges* L connect nodes $\bar{u}_i$ and u_i with the components C_j and G_k. (b) The components C_j and G_k are placed in parallel planes which are perpendicular to the plane $(x, y, 0)$.

possible matching. This gadget guarantees that any perfect matching will select only one node $u[j]$ with combination of nodes from this gadget. However, there are many *long-distance-edges* coming to nodes of type $u[j]$ and there is a need to join them. The *split* gadget is not suitable for this task, therefore we introduce a *join* gadget (see Figure 2.5(b)). The *join* gadget guarantees that any perfect matching will connect a node $u[j]$ to only one pair of type s_2, s_3 (g_2, g_3). A direction of triangles (towards a node of type S_1) uniquely identifies a selection path.

To complete the construction we need to show how to place in 3D all the *long-distance-edges* and connections between them. The idea is to place the component T in the plane $(x, y, 0)$, components C_j and G_k on planes that are perpendicular to the plane $(x, y, 0)$. The planes of C_j and G_k are parallel. The *long-distance-edges* are constructed between the components like water pipes. Below we give details of a 3D coordinate assignment for each component.

The component T as depicted in Figure 2.3 is not convenient for a coordinate assignment. We transform it into a new structure as shown in Figure 2.6. Now, all the nodes that represent variables $u_i[j]$ are located on the 1D line $(x, 0, 0)$. The T_i components are composed from the same set of points that are shifted with the following vector: $Shift(i) = ((i-1)(2m+2)\varepsilon, 0, 0)$.

The component T_i, $1 \leq i \leq n$, is constructed from the following points (below, all the nodes denoted by $u/a/b$ belong respectively to $S_1/S_2/S_3$):

$$
\begin{aligned}
U_i^t &= \{\bar{u}_i[j] = ((2j-1)\varepsilon, 0, 0) + Shift(i) : \ 1 \le j \le m\} \\
U_i^f &= \{u_i[j] = \bar{u}_i[j] + (\varepsilon, 0, 0) : \ 1 \le j \le m\} \\
u_i'' &= (0, \varepsilon, 0) + Shift(i) \\
u_i''' &= ((2m+1)\varepsilon, \varepsilon, 0) + Shift(i) \\
U' &= \{u'_i[j] = (j\varepsilon, 2\varepsilon, 0) + Shift(i) : \ 1 \le j \le 2m\} \\
H' &= \{a'_{i,j} = (0.5\varepsilon + 2(j-1)\varepsilon, 0.5\varepsilon, 0) + Shift(i) : \ 1 \le j \le m+1\} \\
&\cup \ \{b'_{i,j} = (1.5\varepsilon + 2(j-1)\varepsilon, 0.5\varepsilon, 0) + Shift(i) : \ 1 \le j \le m\} \\
H'' &= \{a''_{i,j} = (1.5\varepsilon + 2(j-1)\varepsilon, 1.5\varepsilon, 0) + Shift(i) : \ 1 \le j \le m\} \\
&\cup \ \{b''_{i,j} = (0.5\varepsilon + 2(j-1)\varepsilon, 1.5\varepsilon, 0) + Shift(i) : \ 1 \le j \le m+1\} \\
H''' &= \{a'''_{i,j} = b'''_i[j] = (2.5j\varepsilon, 2\varepsilon, 0) + Shift(i) : \ 0 \le j \le m\}
\end{aligned}
$$

Figure 2.7(a) depicts *long-distance-edges* that connect T_i components to the components C/G. All these edges are placed in the same plane $(x, y, 0)$ as T_i:

$$
\begin{aligned}
L &= \{a^L_{i,j,k} = b^L_{i,j,k} = (j\varepsilon, (1-2k)\varepsilon, 0) + Shift(i) : \ 1 \le j \le 2m, 1 \le k \le mn\} \\
&\cup \ \{u^L_{i,j,k} = (j\varepsilon, -2k\varepsilon, 0) + Shift(i) : \ 1 \le j \le 2m, 1 \le k \le mn\} \\
&\quad 1 \le i \le n
\end{aligned}
$$

Figure 2.7(b) depicts components C/G that are placed perpendicular to the plane

$(x, y, 0)$. The component C consists of the following points:

$$
\begin{aligned}
C \;=\; & \{a^C_{j,k} = b^C_{j,k} = (j\varepsilon, -2k\varepsilon, 3\varepsilon) : \; 1 \le j \le (2mn + 2(n-1)), 1 \le k \le m\} \\
\cup \; & \{u^C_{j,k} = (1.5j\varepsilon, -2k\varepsilon, 3\varepsilon) : \; 1 \le j \le (2mn + 2(n-1) - 1), 1 \le k \le m\} \\
\cup \; & \{ \; \{a^C_{i,j} = b^C_{i,j} = ((2j-1)\varepsilon, -2j\varepsilon, \varepsilon) + Shift(i)\} \\
& \quad \cup \{u^C_{i,j} = a^C_{i,j} + (0, 0, \varepsilon)\} \\
& \quad : \bar{u}_i \in c_j, 1 \le j \le m\} \\
\cup \; & \{ \; \{a^C_{i,j} = b^C_{i,j} = (2j\varepsilon, -2j\varepsilon, \varepsilon) + Shift(i)\} \\
& \quad \cup \{u^C_{i,j} = a^C_{i,j} + (0, 0, \varepsilon)\} \\
& \quad : u_i \in c_j, 1 \le j \le m\} \\
& 1 \le i \le n
\end{aligned}
$$

The component G consists of the following points:

$$
\begin{aligned}
G \;=\; & \{a^G_{j,k} = b^G_{j,k} = (j\varepsilon, -2k\varepsilon, \varepsilon) : \; 1 \le j \le (2mn + 2(n-1)), (m+1) \le k \le mn\} \\
\cup \; & \{u^G_{j,k} = (1.5j\varepsilon, -2k\varepsilon, \varepsilon) : \; 1 \le j \le (2mn + 2(n-1) - 1), (m+1) \le k \le mn\}
\end{aligned}
$$

The whole construction requires a polynomial number of points.

□

2.4.3 2D-ε-K-partite-(pivot) Matching

In this section we want to show a schematic proof that the maximal cardinality 2D-ε-3-partite and 2D-ε-4-partite-pivot problems are NP-Hard.

Lichtenstein[79] introduced an approach to prove NP-Completeness results of planar instances of some of the graph related problems. He demonstrated that any 3-SAT formula can be reduced to an equivalent 3-SAT formula with the following planar property. Define a bipartite graph where the nodes of the first partition represent variables from the 3-SAT formula and the second partition nodes represent clauses. An edge is defined between a variable and a clause if the variable appears,

complemented or non-complemented, in the clause (see Figure 2.8(a)). Given a 3-SAT formula there exists an equivalent formula for which the above defined bipartite graph is planar. This property of 3-SAT formula allows to prove NP-Completeness of planar *node cover* problem, *directed Hamiltonian circuits* problem, etc[79].

Consider the original 3-partite matching problem. Given a hypergraph with the edges defined between triplets of nodes, we will define a 4-partite graph in the following manner. If there is an edge $e = (a, b, c)$ then the nodes a, b and c are added to the first, second and third partition accordingly and the node e is added to the fourth partition. The edges (a, e), (b, e) and (c, e) are created (see Figure 2.8(b)). Actually, the final graph is bipartite (there are no edges between the first three partitions), but we will consider it as a 4-partite graph (we will use the fourth partition as a *pivot*). Dyer and Frieze[35] demonstrated that even for instances of the perfect 3-partite matching problem for which the associated graph defined above is planar, the problem is NP-Complete. The proof relies on the result of Lichtenstein[79]. In contrast to the associated graph of a 3-SAT instance, the associated 4-partite graph uniquely defines the original instance of the 3-partite hypergraph.

We will transform the associated planar 4-partite graph to the 2D-ε-4-partite-pivot graph. Figure 2.8(c) shows how to apply *long-distance-edges* to connect between the nodes. The construction preserves the original matching choices. To assure that no unwanted relations are created between the nodes within ε distance (for example, we want to avoid that two *long-distance-edges* are too close to each other) a geometrical scaling can be performed on the original (schematic) planar graph. In the construction of Dyer and Frieze[35] the maximal degree of each node is only 3, therefore we do not need *split* or *join* gadgets. The extensions we perform to the graph add an equal number of nodes for each partitions. The original graph had the equally sized first three partitions. Assume that after our extension the size of the first three partitions equals to N. If we define the fourth partition as a *pivot* partition then we conclude with the following theorem:

Theorem 2.16. *The problem of existence of a 2D-ε-4-partite-pivot matching of size N is NP-Complete.*

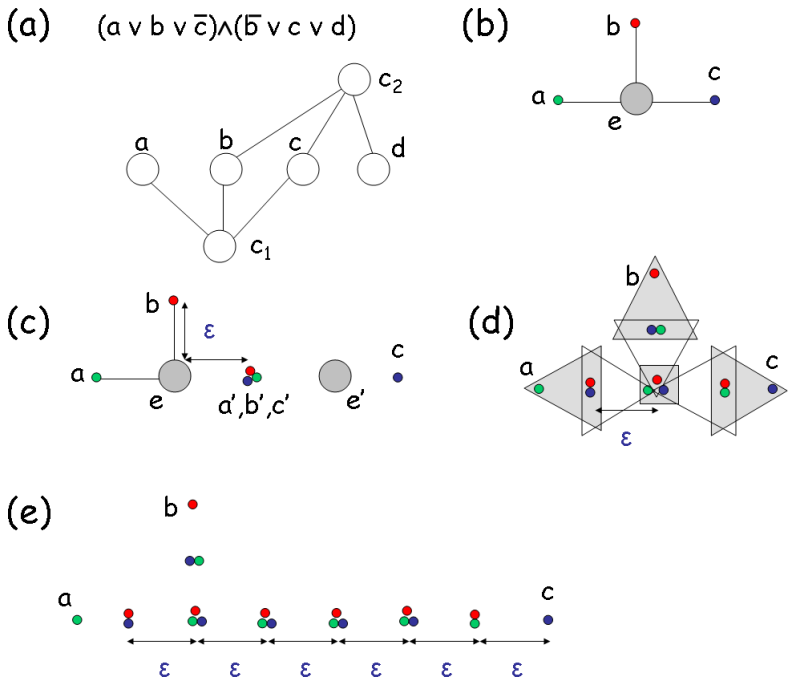


Figure 2.8: (a) A bipartite graph associated with the 3-SAT formula[79]. (b) An edge $e = (a, b, c)$ is represented by four nodes a, b, c and e. (c) A *long-distance-edge* is added between e and c. This extension does not change the original matching choices, i.e. either e selects a, b and c or selects none. The extended component has also two equivalent choices. According to the reduction, all nodes from the first three partitions should be selected (a, b, c, a', b' and c'). First option, e selects a, b and c', then e' has to select a', b' and c. The second and the third option, e or e' selects none, then a', b' and c' have to be selected by e' or e respectively. (d) The 4-partite component shown in (b) is replaced by a 3-partite component. There are only two options for perfect matching, either in gray (equivalent to selecting (a, b, c)) or in white (equivalent to not selecting (a, b, c)). (e) An example of a *long-distance-edge* added to the component from (d). The matching options are preserved.

Notice, that the maximal cardinality 3-partite-ε-pivot matching problem is solvable in polynomial time by solving a maximal network flow[v]. In case we require that all the *pivot* nodes should be matched, then the ε-K-partite-pivot matching problem, for any K, is also solvable in polynomial time by a maximal network flow[vi].

The above construction can be reduced to a 3-partite graph while preserving the same options for matching of the original triplets. In similar manner we can add *long-distance-edges*. See Figure 2.8(d), (e). Thus, we obtain the following theorem:

Theorem 2.17. *The perfect 2D-ε-3-partite matching problem is NP-Complete.*

Notice, that in all the above reductions the maximal node degree is a small constant. In the case of 3D-ε-3-partite/2D-ε-4-partite-pivot/2D-ε-3-partite reduction, the maximal degree is 6/9/8, and if we consider only edges between any two partitions then the maximal node degree is only 3/3/4. In practical applications, which are discussed in Chapters (3) and (5), the node degree can be even larger, though it is bounded by a small constant ($\varepsilon = 3.0$Å). For practical node degree estimates see Section 5.2.2. Alternatively, we can assume that the point distribution is not too dense as a function of ε. Formally, we assume that the number of points inside any circle in 2D (sphere in 3D) of radius ε is bounded by a small constant d ($dens(\varepsilon) \leq d$). Therefore, the NP-Completeness results still hold, however, we can utilize the geometric properties to devise an approximation algorithm which is fix parameter tractable (FPT). The algorithm is polynomial in n.

Theorem 2.18. *Let n be the size of the smallest partition. The size of the ε-K-partite/ε-K-partite-pivot matching can be approximated in 2D within* $1/(1 - 4 \cdot \varepsilon/c)$,

[v]Connect the *source* node to all the points/nodes of the second point set. Make directed edges from the second point set to the corresponding ε close points of the *pivot*/first point set. Then, direct the edges from the *pivot* nodes to the corresponding ε close points of the third point sets. Finally, connect all third set points to the *sink* node. Set all the edge capacities to one. The maximal flow in this graph corresponds to the maximal 3-partite-ε-pivot matching.

[vi]Connect the *source* node to all the points/nodes of the *pivot*/first point set. Set the minimum and maximum capacity of these edges to K-1 (this will guarantee that each *pivot* point is selected). Then, for each *pivot* node p add new K-1 nodes $\{v_2, ..., v_K\}$ and add edges $\{(p, v_i) : i = 2, ..., K\}$ with capacity one. These edges will guarantee that at most one node from the point set i will be matched with p. Then, direct one unit capacity edges from the node v_i to the points from the set i that are ε close to p. Finally, connect nodes from the sets $2, .., K$ to the *sink* node and set the maximum capacity of these edges to one. The maximal flow in this graph (if a flow exists) will connect each *pivot* point with exactly one point from each other point set.

$\forall c > 4\cdot\varepsilon$, in time $O(n\cdot d^{(c/\varepsilon)^2\cdot d\cdot K})$, and can be approximated in 3D within $1/(1-6\cdot\varepsilon/c)$, $\forall c > 6\cdot\varepsilon$, in time $O(n\cdot d^{(c/\varepsilon)^3\cdot d\cdot K})$.

Proof. The general idea of the approximation algorithm is to construct a grid that covers the input points and then to solve optimally the matching problem separately for each grid face.

Consider the 2D case. We construct a finite size grid, g_0, with spacing c such that it covers all points of the graph. It is enough to cover points and their ε surrounding only from the smallest point set, hence, the number of grid faces is less than $O(n)$. The number of points that are covered by each grid face is at most $O(K\cdot d\cdot(c/\varepsilon)^2)$. If a point is located on a grid edge we arbitrarily associate it to one of the adjacent faces. The grid can be constructed in time $O(K\cdot n\cdot log(n))$.

Given a grid face, consider its covered points. Apply an exhaustive search to detect the largest matching. A search can be done by iterative selection of ε-K-tuples according to the vertices from the smallest partition. For each given vertex there are $O(d^{K-1})$ ε-K-tuples in which it can participate. There are $O((c/\varepsilon)^2 d)$ vertices from the smallest partition and this is the depth of the iterative search. Thus, the exhaustive search takes $O(d^{(c/\varepsilon)^2\cdot d\cdot K})$ time.

Define by $M(g)$ the union of optimal matchings from all the grid g faces, detected by the exhaustive search in each grid face. Notice that the union is a valid matching since no point is shared between the grid faces.

Let M^{OPT} be the optimal ε-K-partite matching. Define by $M^{OPT}(g)\subseteq M^{OPT}$ a set of ε-K-tuples that its edges intersect with the grid g edges. Therefore, $|M^{OPT}| - |M^{OPT}(g)| \leq |M(g)|$.

To reduce the maximal size of $M^{OPT}(g)$ we consider $(c/\varepsilon)^2$ grids constructed at slightly different locations, namely, $G^* = \{g_0 + (i\cdot\varepsilon, j\cdot\varepsilon) : 0\leq i,j\leq\lceil c/\varepsilon\rceil\}$, where the additive operator is a translation applied to all grid elements.

Consider some ε-K-tuple $k\in M^{OPT}$. Any ε-K-tuple can be covered by a circle with diameter less than $2\cdot\varepsilon$ (for the tight bound see Appendix 2.8). The distance between grid parallel edges is larger than $4\cdot\varepsilon$. Hence, only one vertical/horizontal line from g_0 can intersect a circle of diameter $< 2\cdot\varepsilon$. Since all the grids are generated at steps ε a vertical/horizontal line from g_0 can intersect a circle at most twice.

For each intersection there are (c/ε) different grids in G^*. Counting separately for vertical and horizontal lines there are at most $4 \cdot (c/\varepsilon)$ grids from G^* that can intersect with k. Summing for all grids and all $k \in M^{OPT}$ there are at most $4 \cdot (c/\varepsilon)|M^{OPT}|$ intersected ε-K-tuples. Thus, there exists $g \in G^*$ with at most $4 \cdot (c/\varepsilon)|M^{OPT}|/(c/\varepsilon)^2$ intersected ε-K-tuples from M^{OPT}. Hence, the algorithm computes g and $M(g)$ such that $|M^{OPT}|(1 - 4 \cdot \varepsilon/c) \leq |M(g)|$. □

2.5 Largest Common Point Set Problem Between K Point Sets

2.5.1 mLCP/pmLCP in 1D

For the case of 1D and a constant number of structures we give a polynomial solution to the pmLCP problem.

First, we start with the algorithm for the pairwise case, $K = 2$. Consider an arrangement defined on translations between points of $S_1 = \{p_i\}$ and $S_2 = \{q_j\}$. $T_{ij} = \{t \ : \ |(t + q_j) - p_i| \leq \varepsilon\}$. T_{ij} is a contiguous fragment and can be defined by its two extreme points, $T_{ij} = [p_i - q_j - \varepsilon, p_i - q_j + \varepsilon]$. All relevant translations belong to $T^2 = \{T_{ij}\}$. Arrangement of $\{T_{ij}\}$, $A(T^2)$, is defined by the fragment end points. The arrangement faces correspond to contiguous fragments between the nodes. The translations that belong to some face define exactly the same combinations of possible matched points. Therefore, to solve the pmLCP problem between S_1 and S_2 it is enough to consider only one translation from each arrangement face. Such translation representative can be chosen as a middle point of the face fragment. The complexity of the arrangement $A(T^2)$ is $O(n^2)$. For each face we solve the pmSim problem which takes $O(n\ log(n))$ for two structures. Thus the total time complexity is $O(n^3 log(n))$.

pmLCP in 1D

First, we compute $A(T^i)$ $(i = 2, ..., K)$, which are the translation arrangements between S_1 and S_i $(i = 2, ..., K)$. Each face of $A(T^i)$ defines an unique solution to the pmLCP problem with respect to S_1 and S_i. Therefore, each face (K-1)-tuple, consisting of one face from each $A(T^i)$, defines a multiple superposition between all K structures. Traversing all combinations of $K-1$ face tuples gives all relevant multiple superpositions. For each $K-1$ face tuple we solve the pmSim problem in time $O(K \cdot n \cdot log(K \cdot n))$, and output the largest solution. There are $O(n^{2(K-1)})$ different $K-1$ face tuples. Therefore, the running time is $O(n^{2K-1}\ K \cdot log(K \cdot n))$.

mLCP in 1D

Notice, that in the pmLCP problem, given a $K-1$ tuple of faces, there are $K-1$ degrees of freedom between relative positions of structures S_i and S_j $(i \neq j,\ 1 < i, j \leq K)$. In the case of the pmLCP problem these degrees of freedom are not relevant, since only the position relative to S_1 is taken into account. However, for the mLCP problem we need to consider relative position between all the structures. For simplicity, consider the case of three structures, S_1, S_2 and S_3. As previously, for structures S_1 and S_2 we define an arrangement $A(T^{1,2})$. In contrast to pmLCP, it is not enough to consider only one representative from a face of $A(T^{1,2})$, since the relative position of S_2 and S_3 will change the combinations of possibly matched point triplets. A face of $A(T^{1,2})$ is defined by two fragment endpoints $[a, b]$. Consider a set $S_{1,2}(t) = S_1 \cup (S_2 + t)$, $t \in [a, b]$, where the additive operator, $+$, is a translation applied to each point of S_2. Let $A(T^{1,2,3}(t))$ be the translation arrangement between $S_{1,2}(t)$ and S_3. Clearly, some nodes (points) of $A(T^{1,2,3}(t))$ are static, i.e. do not depend on t, and some node positions are a function of t. Let us denote by V the static set of nodes and by $V(t)$ the second set. As t continuously changes in the range $[a, b]$, the combinatorics of $A(T^{1,2,3}(t))$ can also be changed. The combinatorics of $A(T^{1,2,3}(t))$ can be changed only when some point from $V(t)$ intersects some point from V. Since the points of V are static and points of $V(t)$ are all moving with the same velocity, the intersection events, $\{t'\}$, can be computed in time $|V|\ |V(t)|\ log(|V|\ |V(t)|) \in$

$O(n^4 log(n))$. Each event t' defines a translation of point set S_2. It also defines, one, new face of $A(T^{1,2,3}(t'))$, i.e. a translation t'' for S_3. Notice, that at each event t' we don't need to update $A(T^{1,2,3}(t'))$. Therefore, for each t' we compute mSim of S_1, $S_2 + t'$ and $S_3 + t''$.

Therefore, the algorithm is as follows. For each face $[a, b]$ of $A(T^{1,2})$ we compute all combinatorially different arrangements $A(T^{1,2,3}(t))$, $t \in [a, b]$. For each, combinatorially different, face of $A(T^{1,2,3}(t))$ we solve the mSim problem. The total running time is $O(n^2 n^4 log(n^4)(3n \cdot log(3n))) = O(n^7 log^2(n))$.

For K structures we apply a recursive algorithm as follows. Similar to the three set problem, the arrangement $A(T^{1,2,...,K}(t_2, ..., t_{K-1}))$ is defined between two point sets $S_1 \cup (S_2 + t_2) \cup ... \cup (S_{K-1} + t_{K-1})$ and S_K. This arrangement contains one static point set and $K - 2$ point sets which can be independently translated within some range. The number of combinatorially different arrangements, as a function of $(t_2, ..., t_{K-1})$, is not larger than $(n^2)^{K-1}$. Let $T(k)$ denote the time to compute combinatorially different arrangements for k point sets. Then, $T(k) = T(k-1) + (k-2) \cdot n^2 \cdot (n^2)^{k-2} \leq k^2 n^{2K-2}$. The number of faces in an arrangement is $O((K - 1) \cdot n^2)$. For each face of combinatorially different arrangement we solve the mSim problem. Therefore, the running time to solve the mLCP problem is $T(K) + (K - 1) \cdot n^2 \cdot K \cdot n \cdot log(K \cdot n)$.

Theorem 2.19. *The mLCP/pmLCP problem in 1D is solvable in polynomial time for any fixed K.*

2.5.2 mLCP/pmLCP in 2D

Akutsu el al.[2] have demonstrated that the mLCP problem between K point sets in 1D is NP-Hard even for $\varepsilon = 0$. For a constant K, however, there is a polynomial solution. Here we show that in the case of $\varepsilon > 0$ even for three point sets in 2D (4 point sets for the pmLCP problem) the mLCP problem is NP-Hard. In our proof we use the NP-Completeness results of the matching problem for the ε-K-partite graphs.

Theorem 2.20. *The mLCP problem in 2D is NP-Hard for K=3 and $\varepsilon > 0$.*

Proof. We will reduce the 3-partite-ε matching problem to the mLCP problem. Assume that we are given an ε and three point sets (S_1, S_2, S_3). The idea of the reduction

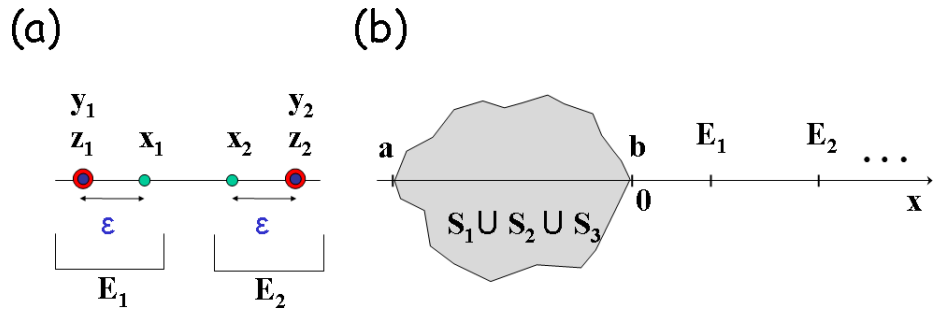


Figure 2.9: (a) $E_1 = \{x_1, y_1, z_1\}$ and $E_2 = \{x_1, y_1, z_1\}$ define two triplets of points. Points $\{x_1, x_2\}$, $\{y_1, y_2\}$ and $\{z_1, z_2\}$ are added to point sets S_1, S_2 and S_3 respectively. The points y_1 and z_1 (y_2 and z_2) are assigned the same coordinate. The identity transformation and 180° rotation around $(x_1+x_2)/2$ are the only two transformations that can be applied to S_2 (S_3) that can match with ε-congruence both triplets E_1 and E_2. (b) The triplets E_i, $i = 1, ..., 4N$, are placed on the axis of diametrically opposite points (a, b) of the point set $S_1 \cup S_2 \cup S_3$.

is to add some new points to the original sets so that the largest common point set can be obtained only with identity transformations (in the mLCP problem Euclidean transformations are applied to the point sets S_2 and S_3 in order to maximize the matching).

Consider two sets of three points E_1 and E_2 as depicted in Figure 2.9(a). We add points $\{x_1, x_2\}$, $\{y_1, y_2\}$ and $\{z_1, z_2\}$ to S_1, S_2 and S_3 respectively. The identity transformation and 180° rotation around $(x_1+x_2)/2$ are the only two transformations that can be applied to S_2 (S_3) that can match the two triplets E_1 and E_2. Other transformations will match at most one triplet. Let $N = min(|S_1|, |S_2|, |S_3|)$. Let (a, b) be a pair of points from the set $S_1 \cup S_2 \cup S_3$ with the maximal distance, i.e. $(a, b) = arg\ max_{a,b \in S_1 \cup S_2 \cup S_3}\ |a - b|$. Transform the coordinate system so that a and b lie on the x-axis, b is located at $(0, 0)$ and a to the left of b (see Figure 2.9(b)). Let $d = |a - b|$, $L_0 = 2(d + 3\varepsilon)$, $L_1 = L_0(2N)^2$. We define $4N$ sets E_i, $i = 1, ..., 4N$, ($2N$ sets of type E_1 and $2N$ sets of type E_2), all located on the x-axis (in this construction we use an idea borrowed from Akutsu el al.[2]) . The leftmost point of E_1 is located at distance L_0 from point b, i.e. $y_1 = z_1 = (L_0, 0)$, $x_1 = (L_0 + \varepsilon, 0)$. The set E_2 is defined as $x_2 = (L_1 + 3L_0, 0)$, $y_2 = z_2 = (L_1 + 3L_0 + \varepsilon, 0)$. The set E_{2i-1}, $i = 2, ..., 2N$, is defined as $E_{2i-1} = E_1 + ((2i - 2)L_1 + L_0 \sum_{k=2}^{2i-1} k, 0)$. The set E_{2i}, $i = 2, ..., 2N$, is

defined as $E_{2i} = E_2 + ((2i-2)L_1 + L_0 \sum_{k=3}^{2i} k, 0)$. For simplicity, consider that the leftmost point of each set E_i is located at $(L_0 + (i-1)L_1 + L_0 \sum_{k=2}^{i} k, 0)$, $i = 2, ..., 4N$.

Given three point sets (S_1, S_2, S_3) we extend it with $\cup_{i=1}^{4N} E_i$. Let us call the extended sets (S_1^*, S_2^*, S_3^*). Let M^{max} be the size of the maximal 3-partite-ε matching of the graph $G(S_1, S_2, S_3)$. From the construction it is clear that for the identity transformations the size of a common point set of (S_1^*, S_2^*, S_3^*) is $M^{max} + 4N$. We want to show that for the non identity transformations the size of any common point set of (S_1^*, S_2^*, S_3^*) is not larger than $3N$. We examine the following properties of the construction.

Property I. Notice that for any transformation applied to S_2 (S_3), points from at most one set E_i can be matched with transformed points of S_2 (S_3) (the distance between any E_i and E_j, $i \neq j$, is larger than the diameter of $S_1 \cup S_2 \cup S_3$).

Property II. For any given matching, denote by I the matched triplets consisting of points from the same E_i, $i = 1, ..., 4N$, i.e. triplets (x_i, y_i, z_i). From the construction it is clear that $|I| = 4N$ only for the identity transformations, and $|I| \leq 2N$ for any other transformations (a small shift by less than ε along the x-axis will match either even or odd sets E_i).

Property III. For any given matching, denote by I^* the matched triplets consisting of points coming from at least two different sets, E_i and E_j, $i \neq j$, i.e. triplets (x_i, y_j, z_k), $x_i \in E_i$, $y_j \in E_j$ and $z_k \in E_k$, such that either $i \neq j$ or $i \neq k$ or $j \neq k$. We want to show that for any transformation $|I^*| \leq 2$. Consider a not trivial matching I^* of size larger than one. Assume that a point $x_t \in E_t$ is matched with $y_{t'} \in E_{t'}$ and $x_j \in E_j$ (from another triplet) is matched with $y_{j'} \in E_{j'}$. Therefore,

$|x_t - x_j| = |y_{t'} - y_{j'}| + \delta$, where $\delta \in [-2\varepsilon, 2\varepsilon]$.

Cases where $t \neq j$ and $t' = j'$ (or $t = j$ and $t' \neq j'$) are trivially invalid due to the construction. Assume that $t > j$, then

$(t-j)L_1 + L_0 \sum_{k=j+1}^{t} k = |t'-j'|L_1 + L_0 \sum_{k=j'+1}^{t'} k + \delta + \delta^*$, where $\delta^* \in \{-2\varepsilon, 0, 2\varepsilon\}$.

Since $L_1 > L_0 \sum_{k=j+1}^{t} k$, it follows that $(t-j) = (t'-j')$. Since $L_0 > |\delta + \delta^*|$, then

$L_0 \sum_{k=j+1}^{t} k = L_0 \sum_{k=j'+1}^{t'} k$.

It follows that $t = t'$ and $j = j'$. Therefore, $|I^*| \leq 1$ in case of $t > j$ and $t' > j'$. In case that $t > j$ but $t' < j'$ then $t = j'$ and $j = t'$ (e.g. 180° rotation around

$(x_t + x_{t'})/2)$, therefore, $|I^*| \leq 2$, since according to the construction the equality $t = j'$ and $j = t'$ cannot hold for more than two triplets.

These construction properties give the following result. For identity transformations the common point set has the size $M^{max} + 4N$. If at least one of the transformations is not identity then the common point set size is less than $N + 2N$ (the upper bound of the mLCP of (S_1, S_2, S_3) plus the upper bound of a common point set of $\cup_1^{4N} E_i$ in case of non identity transformation). Therefore, the maximal 3-partite-ε matching of (S_1, S_2, S_3) is of size M if and only if the mLCP size of (S_1^*, S_2^*, S_3^*) is $M + 4N$. □

The above reduction works as well for the pmLCP problem with four point sets. The additional elements E_i are extended with points, t^1 and t^2, which will be added to the fourth point set, such that $y^1 = z^1 = t^1$ and $y^2 = z^2 = t^2$.

Theorem 2.21. *The pmLCP problem in 2D is NP-Hard for K=4 and $\varepsilon > 0$.*

As we have discussed above in Section 2.2, the optimization LCP problem can be separated into two sub-problems, transformation search and the mSim/pmSim problem. For any fixed number of point sets, K, the multiple transformation search can be computed in polynomial time, either by applying the optimal methods (that construct arrangement of transformations in high dimensions) or by applying a γ-additive-approximation method. The optimal method can be extended for any K similar to the method described for the 1D case. Since the optimal algorithm is not practical even for two point sets, here we do not consider its extension. Instead, we consider the approximation technique. Compute all relevant pairwise transformations between S_1 and S_i, $\forall i = 2, ..., K$, that guarantee the γ-additive-approximation to an optimal transformation. For each combination of $K - 1$ transformations, one for each point set, solve the mSim/pmSim problem. In case of the pmSim problem the γ-additive error remains the same as in the pairwise case. In the mSim case, the algorithm gives a $2 \cdot \gamma$-additive approximation and this error does not depend on K.

Theorem 2.22. *For a constant number of point sets, the δ-(γ-additive)-approximation to the mLCP/pmLCP problem in 2D/3D can be computed in polynomial time.*

2.6 Largest Common Point Set Problem Between Point Set Families

Above we have discussed the problem of computing the largest point set that is shared between all input structures. In practice, several structures may be outliers or the input may consist of several structurally similar families of point sets. Therefore, in real applications it is important to detect families of structures that may share significantly larger common patterns than a common pattern between all the input point sets. Essentially, this is a generalized version of the mLCP problem.

Problem 8. Family mLCP. The input consists of $\varepsilon > 0$, and K point sets $S = \{S_i\}_{i=1}^{K}$. Denote by $F^t = \{S_{i_1}, ..., S_{i_t}\}$ a family consisting of t point sets, $S_{i_j} \in S$. Find, for each $t = 2, ..., K$, a family F^t_{opt}, such that $mLCP(F^t_{opt})$ is the largest over all families of t point sets, i.e. $\forall F^t\ mLCP(F^t_{opt}) \geq mLCP(F^t)$.

For a few input structures it is possible to compute an mLCP (optimal or approximate solution) for each possible family in polynomial time. However, for large K, since the *Family mLCP* problem is a generalization of the mLCP problem, it is NP-Hard in 1D even for $\varepsilon = 0$ [2]. Let us consider the multiple similarity problem extended to the *Family* requirements, i.e. the *Family mSim* problem. Then the mSim in 1D, $\varepsilon = 0$, is trivially solvable by sorting points and detecting points with the same coordinate. However, we conclude that the *Family mSim* problem is NP-Hard.

Theorem 2.23. *The Family mSim problem in 1D is NP-Hard for $\varepsilon = 0$.*

Proof. We make a reduction from the maximum edge cardinality biclique problem (*MBP*) [98, 26]. Given a bipartite graph $G = (V \cup U, E)$, a subgraph $BC = (V' \cup U')$, $V' \in V$ and $U' \in U$, is a biclique if for every $v \in V'$ and $u \in U'$ there is an edge $(v, u) \in E$. Maximum edge cardinality biclique in G is a biclique with a maximum number of edges.

Given a bipartite graph $G = (V = \{v_1, ..., v_n\} \cup U = \{u_1, ..., u_m\}, E)$ we associate each node v_i with a point set S_i and each node u_j with a coordinate (j) on a line. First, we define the total set of coordinates and then we assign points to these coordinates.

The coordinates are $\{p_j = (j) \mid j = 1, ..., |U|\}$, i.e. each node u_j from U is mapped to a coordinate (j). A point set S_i, $i = 1, ..., |V|$, is defined as $S_i = \{p_j \mid (v_i, u_j) \in E\}$.

According to the construction, a biclique defines a family of point sets and their common point set (not necessarily the largest), and the opposite, a family of point sets with their common point set correspond to a biclique (not necessarily the largest). A solution to the Family mSim problem, for each $t = 2, ..., |V|$ returns the largest biclique $BC_t = (V' \cup U')$ so that $|V'| = t$. Therefore, selecting the largest biclique from $\{BC_t\} \cup \{$*bicliques with only one node from* $V\}$ gives the largest biclique in the graph G. □

Notice, that in case $\varepsilon = 0$, the Euclidean dimensionality of the mSim (or *family mSim*) problem is irrelevant. It is always possible to arbitrarily change a coordinate of the coincide points without effecting the size of a common point set. Therefore, the input point sets can be represented as a pure bipartite graph as defined in the above proof.

In case of $K = 3$ and $\varepsilon > 0$ the Family mLCP (mSim) problem is hard in 2D due to the NP-Hardness of the mLCP (ε-K-partite matching) in 2D for three point sets.

2.7 Conclusions

We have studied the computational complexity of the multiple common point set problem. We considered the problem of multiple similarity measure, where the point sets are fixed in space (mSim and pmSim problems), as well as, the general optimization version where points are allowed to undergo an Euclidean transformation (mLCP and pmLCP problems). Several parameters play a role in determining the complexity of these problems, these are the dimensionality of the Euclidean space; ε - the error of congruence between the common point sets; K - the number of input point sets; point set density in a circle/sphere of radius ε. Even the most simply defined problem of detecting at least one K-tuple of ε-close points (ε-K-tuple), which serves as a basic primitive of the multiple similarity measure, is NP-Hard. However, while the generally defined combinatorial problems are hard to approximate, the geometrical

constraints of the studied problems allowed us to devise polynomial time approximation schemes which are fixed-parameter tractable. The results are summarized in Tables 1.1 and 1.2.

2.8 Appendix: Max-Min Enclosing Circle/Sphere

Consider the 2D case. For an ε-K-tuple k denote by $r(k)$ the minimum radius of a circle that encloses all points of k. Let r_{max} denote the maximum value over all possible values of $r(k)$. In other words, for any ε-K-tuple k, and for any $K \geq 3$, there is a circle of radius r_{max} that encloses the points of k. Without loss of generality, assume that $\varepsilon = 1$. The radius of a circle that encloses an equilateral triangle with unit side lengths is $\frac{1}{\sqrt{3}}$. Hence, $r_{max} \geq \frac{1}{\sqrt{3}}$. Let us prove that $r_{max} = \frac{1}{\sqrt{3}}$.

Assume that $r_{max} > \frac{1}{\sqrt{3}}$, therefore there are r and k such that $r(k) > \frac{1}{\sqrt{3}}$. Denote the minimum enclosing circle of k (with radius r) by C_k. From the points of k that lie on C_k select two points, a and b, with the maximal distance. Let $x = |a - b|$, $c = (a + b)/2$, o - center of C_k, d - as depicted in Figure 2.10.

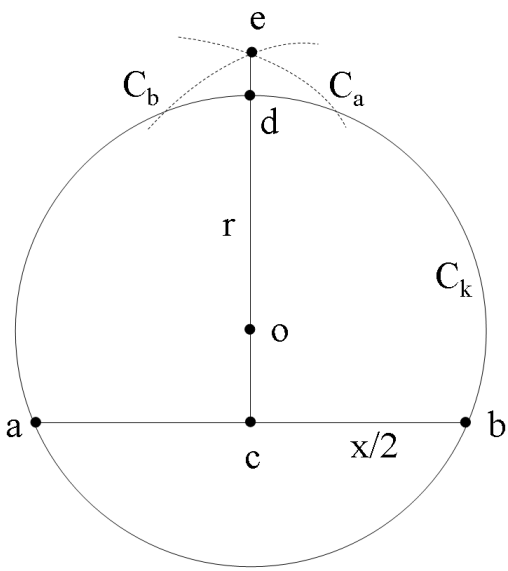


Figure 2.10: For details see the text in the Appendix.

Points a and b are not diametrically opposite, since otherwise $r < \frac{1}{\sqrt{3}}$, therefore,

$|o - c| > 0$. Consider a point $e \in C_a \cap C_b$ such that it is located at the side of the positive direction of vector $(d - c)$ as depicted in Figure 2.10. There are two cases, either $|e - o| < r$ or $|e - o| \geq r$.

Consider the first case, $|e - o| < r$. It follows that there is no other point from k on a larger circle arc defined by (a, b). Since there are more than 180 degrees in the larger arc (a, b) then the circle C_k is not a minimal enclosing circle of k, which is a contradiction.

Consider the case $|e - o| \geq r$. Then we obtain the following inequality:

$$\begin{aligned} |c - b|^2 + |d - c|^2 = |d - b|^2 &\leq x^2 \\ (\frac{x}{2})^2 + [r + \sqrt{r^2 - (\frac{x}{2})^2}]^2 &\leq x^2. \end{aligned} \tag{2.1}$$

Since $r > \frac{1}{\sqrt{3}}$ and $r > \frac{1}{\sqrt{3}} > \frac{x}{2}$ then:

$$(\frac{x}{2})^2 + [\frac{1}{\sqrt{3}} + \sqrt{(\frac{1}{\sqrt{3}})^2 - (\frac{x}{2})^2}]^2 \leq x^2$$

It follows that $x \leq x^2$. Since $0 < x \leq \varepsilon = 1$, then x should be equal to one. Substituting x with one into equation 2.1 we get $r \leq \frac{1}{\sqrt{3}}$ and this leads to a contradiction.

With similar reasoning it can be shown that in 3D the maximal radius of all minimal enclosing spheres of ε-K-tuples is $\frac{\sqrt{6}}{4}\varepsilon$, and this radius is reached if a ε-K-tuple contains an equilateral tetrahedron with side lengths ε.

Chapter 3

Multiple Protein Structure Alignment

3.1 Introduction

The increasing number of determined protein structures opens new horizons for studies of protein function. There are numerous examples of similar functioning proteins, e.g. Isomerases, Cytokines, Myoglobins, Immunoglobulins, Transferases, with similar 3D structure but less than 25% sequence identity. Therefore, in order to study relationships between such proteins sequence analysis alone is not sufficient. While methods for sequence analysis have significantly advanced in the past years, methods for structural analysis are still at an earlier, exploratory stage. Here, we address one of the most basic structure related problems, the problem of multiple protein structure alignment.

A number of methods have been proposed to solve the problem of structural alignment between a pair of proteins, e.g. VAST[82], Geometric Hashing (GH)[11], CE[115], DALI[29], and others[38]. Obviously, multiple structure alignment can provide much more information. Recognition of a structural core common to a set of protein structures has many applications in the studies of protein evolution and classification[96, 29], analysis of similar functional binding sites and protein-protein interfaces[81, 22, 10], homology modeling and threading[3, 47] etc. However, despite

this need, the multiple structure alignment problem has not been extensively studied and, consequently, there are very few available methods that solve this task.

Let us formulate a list of some principle requirements for a multiple structure alignment method:

- Partial Alignment
- Subset Alignment
- Sequential Alignment
- Sequence Order Independent Alignment
- Time Efficiency

Partial Alignment. There might be only a sub-structure (motif, domain) that is similar between a set of molecules, for example, in a set of multi-domain proteins having one or several common domains. Another example is alignment between multi-protein complexes having some structurally similar combination of molecules. Thus, a detection of all common motifs, domains or multi-protein combinations may be required for a multiple structure alignment method. We consider a *local* alignment as a special case of *partial* alignment. For example, a partial alignment may consist of several locally matched structural elements that can be aligned under the same Euclidean transformation.

Subset Alignment.[i] An important aspect of any multiple, sequence or structure, alignment is a detection of a subset of molecules that are more similar than the whole input set. For example, consider an input set of 10 proteins from one family and 5 proteins from another family. Assume that the proteins in each family are structurally similar, but there is little similarity between any two proteins from the first and second family. A multiple alignment between these 15 molecules would probably detect at most one common secondary structure element. Therefore it is very important for a

[i]Above, in Section 2.6 we denoted this problem by *Family mLCP* and showed that it is NP-Hard even in 1D and even in case of $\varepsilon = 0$.

multiple alignment method to be able to automatically distinguish between two such subsets.

To demonstrate the significance of the *partial* and *subset* alignment ability consider a schematic example in Figure 3.1 (a). Three proteins share a common small pattern, and each pair of the proteins share additional, larger, patterns. The desired goal of a multiple structure alignment method is to detect all four patterns. An additional example with real protein structures is given below in Section 3.4. It should be clear, that the number of all possible solutions that may be also biologically meaningful, could be exponential in the number of input molecules. For example, consider proteins which contain a large number of α-helices. Each pair of α-helices could be structurally superimposed (at least partially, if they are different in their lengths). Any multiple combination of α-helices from different proteins results in some multiple alignment. Obviously, the number of such multiple alignments is exponential. Therefore, even if an algorithm is capable of detecting all such combinations, it is not practical to report them.

Sequential and Sequence Order Independent Alignment. Sequence alignment methods naturally produce alignments that follow the protein sequence order, i.e. aligned amino acids indices are always in increasing order. However protein evolution imposes less constraints on the sequence than on the structural properties. Consequently, proteins may have a similar function but topologically different 3D structure ([134] and references therein). In Section 3.4 we consider one such example.

Time Efficiency. Optimal pairwise structural alignment can be solved in polynomial time, however it is still computationally expensive and currently not practical for an implementation[7]. Approximation techniques can significantly reduce time complexity with relatively small degradation in solution accuracy[1, 72]. However, even for three structures the multiple alignment problem is NP-Hard[112]. While the worst case scenario may be computationally infeasible for the detection of an exact solution, considering specific geometrical properties of the protein molecules can, in practice, significantly reduce the computational cost. Examples of such properties, which are far from resembling a random point distribution, include sequentiality of

the protein backbone, secondary structure element composition and protein compactness. Therefore, "smart" heuristic methods that utilize such properties, in practice, may give results that are sufficient for a biological research. An excellent example of a heuristic method for multiple sequence alignment is MUSCLE[36]. Still further research, theoretical and practical, is required for the multiple structural alignment problem.

Below we briefly review available methods for the multiple structure alignment task and try to correlate them with the list of requirements defined above.

A *center-star* approach is one of the efficient ways to compute a multiple sequence alignment. Analogously, it can be applied for multiple structure alignment. A *center* structure is selected which is most similar to the rest of the molecules. Then, iteratively, all other structures are joined into a multiple alignment based on their *pairwise* alignments with the *center* structure[45, 3]. Alternatively, one can apply a *tree-progressive* approach, where a multiple alignment is created according to some distance tree [102, 124, 95]. Therefore, a *tree-progressive* alignment first aligns similar proteins, then proceeds to more distant relationships. An advantage of such an approach is its ability to detect subset alignments of structurally different families.

The *center-star* and the *tree-progressive* approaches are essentially based on some pairwise alignment method that is iteratively applied for the construction of a multiple alignment. Therefore, such technique is less suitable for detection of small structurally similar motifs since at each stage of the iterative alignment only one, the best, solution is selected. Figure 3.1 (a) shows a simple example where a straightforward application of a pairwise alignment method will fail to recognize a pattern common to more than two sequences/structures.

The MALECON method[94] aims to avoid the shortcomings of the iterative pairwise approaches and considers all possible combination of input molecules. When the number of input proteins is large, such a combinatorial approach becomes exponential, therefore, the method considers at least all possible protein triplets while other proteins are progressively added to the aligned triplets. Consequently, the advantage of the method is in detection of subset alignments. However, since only one solution is considered for any given combination of proteins, some smaller local alignments

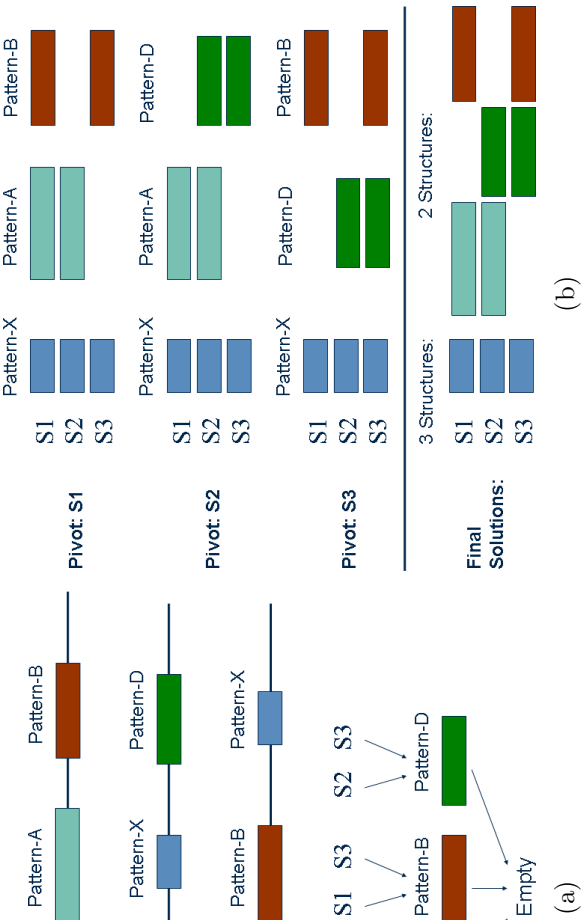


Figure 3.1: (a) A schematic example of three proteins that share a common pattern X. Applying a pairwise alignment method that detects the most similar common pattern will result in pattern A for S1 and S2, pattern B for S1 and S3, and pattern D for S2 and S3. Therefore, no common pattern can be derived from the patterns A, B, and D. One possible solution is to store two (or more) high scoring solutions for each pairwise comparison. However, in this case the number of iterations to compare all pairwise results to detect the best combination of multiple alignments becomes exponential. (b) The goal of the MultiProt method is to detect the local multiple alignments of all four patterns. The depicted alignments are detected while selecting each structure as a pivot, e.g. patterns X, A and B are detected when molecule S1 is selected as a pivot. Finally, pattern X will appear in the multiple alignment of 3 molecules. Patterns A, B and D will appear in the set of alignments consisting of 2 molecules.

can be missed. The method produces sequential alignments.

The *MUSTA* algorithm[76, 75] computes a common geometric core that appears simultaneously in all the input molecules, thus avoiding the shortcomings of the iterative pairwise approaches. The method applies the *Geometric Hashing* technique[130], which allows detection of the sequence order independent alignments. This technique was successfully applied in a number of pairwise structure alignment methods[93, 106]. Since the method requires that all input molecules participate in the multiple structural alignment, the drawback of this method is inability to distinguish outliers. It is sufficient that one structure is very distinct from the others to result in an empty alignment. Consequently, this method cannot detect subset alignments. Second, its efficiency limits practical application for only 10-15 molecules.

Another approach, SPratt2[67], aims to detect small common, local structural motifs of size 3-20 amino acids. The method describes each residue as a short string of its spatial neighbors. Then, an efficient sequence pattern discovery technique is applied to detect sets of residues with common environmental descriptors. The computed alignments are sequential. The method is efficient and allows subset alignments.

Protein representation by secondary structure elements significantly reduces the problem input size[88, 49, 9, 100, 90, 65, 4, 60, 51, 17, 119, 133, 5, 114, 23]. If an average protein has about 300 amino acids, then the number of secondary structures is only about 10. Recently, a new multiple structure alignment method, MASS[31, 30], has been proposed. The method utilizes the secondary structure information (SSE's) to reduce the computational cost of initial common core detection. Therefore, it requires that at least two pairs of SSE's be multiply aligned. The method requires that α-helix is aligned with α-helix, β-strand with β-strand. This natural assumption is sufficient for most cases. Though, there are proteins for which alignment between an α-helix and β-strand has a biological meaning. One such example is given below in Section 3.4.2. The MASS method is capable of detecting partial, subset, sequential and non-sequential alignments.

In order to produce structure-based multiple sequence alignment, the recently developed method 3DCoffee[97] incorporates spatial weights into the multiple sequence alignment method TCoffee. The spatial weight of an amino acid pair is defined as

a positive large constant number, when this pair is structurally aligned according to some *pairwise* structural alignment method. Therefore, the method does not distinguish between amino acids that are structurally aligned at different distances. Since the method applies information only from the best pairwise structure alignment the weighting decision may be inaccurate for the multiple alignment problem.

Here, we propose a method that aims to solve the multiple structural alignment problem with the support of the above defined list of requirements. One of the advantages of our method is its ability of subset and partial alignment. This is illustrated in Figure 3.1 (a,b). Consider the set of proteins from Figure 3.1 (a). The goal of our method is to detect local multiple alignments of all four patterns. This is achieved by performing all possible local multiple alignments of ungapped fragments. The final solutions are constructed from these locally aligned multiple fragments. This makes our approach different from most existing methods, which generally derive a multiple alignment from the high scoring pairwise superpositions. If a pattern appears more than once in some protein, our method recognizes only one combination of this pattern from all possible appearances, however, all sets of possible combinations are reported by the program. Our method is extremely efficient and is suitable for simultaneous comparison of up to tens of proteins.

3.2 The MultiProt Algorithm

In **MultiProt** we try to give an efficient heuristic solution to the multiple structure alignment problem, which we define as:

(*) *Given m molecules, a parameter κ and a threshold value ε, for each r ($2 \leq r \leq m$), find the κ largest ε-congruent multiple alignments containing exactly r molecules.*

Here an *ε-congruent multiple alignment* is defined as: given a pivot molecule M_1 and $r-1$ molecules $(M_2, ..., M_r)$, we define an ε-congruent multiple alignment as a set of $r-1$ 3D transformations $(T_2, ..., T_r)$ (T_j is a transformation which superimposes molecule M_j onto M_1) and a set of K r-tuples (aligned points) $\{(v_{i_k^1}, v_{i_k^2}, ..., v_{i_k^r})\}_{k=1}^{K}$, $v_{i_k^j} \in M_j$, such that $\forall k \forall i_k^j \; ||v_{i_k^1} - T_j(v_{i_k^j})|| \leq \varepsilon$, i.e. the matched points are within ε distance from the appropriate pivot molecule point.

This definition is based on the selection of the pivot molecule. In our algorithm, in order not to be dependent on the choice of the pivot, we iteratively choose every molecule to be the pivot one.

The above multiple alignment problem definition (*) is general and can be applied for comparison of any 3-D objects represented as unconnected point sets in 3-D space. However, we wish to utilize the fact that a protein structure can be represented as an ordered set of points, e.g. by the sequence of the centers of the C_α atoms. Thus, we exploit the natural assumption that any solution for the multiple structure alignment of proteins should align, at least short, contiguous fragments (minimum 3 points) of input atoms. For example, these fragments could be secondary structure elements which could be aligned between the input molecules. First we detect all possibly aligned fragments of maximal length between the input molecules. Then, we select solutions that give high scoring global structural similarity based on the (*) definition. Aligning protein fragments is not a new idea. It has been previously applied in several methods for pairwise protein structural alignment[127, 115]. In our method we use an algorithm which detects structurally similar fragments of maximal length, i.e. fragment pairs which cannot be extended while preserving ε-congruence[106, 113].

For any multiple alignment problem we can always assume that the selected pivot molecule has to participate in all the alignments. If all the molecules are iteratively selected as pivots, then all solutions can be detected. Therefore, our method is based on the pivoting technique, i.e. the rest of the molecules are aligned with respect to the pivot molecule.

Input: m molecules $S = \{M_1 ... M_m\},\ \varepsilon \geq 0$
$for\ i = 1\ to\ m - 1$
$\quad M_{pivot} = M_i$
$\quad S^p = S \setminus M_{pivot}$
$\quad FragmentPairs = Detection\ of\ Fragment\ Pairs(M_{pivot}, S^p)$
$\quad Cuts = Multiple\ Fragment\ Alignment(FragmentPairs)$
$\quad Global\ Multiple\ Alignment(Cuts)$
end

The input is m protein structures, $\{M_i\}_{i=1}^m$, each represented as a sequence of the centers of the C_α atoms. In addition, the input contains a parameter $\varepsilon \geq 0$ which is the distance threshold between the matched C_α atoms. The goal of the algorithm is to compute, for each $r = 2, ..., m$, the largest multiple alignments consisting of exactly r structures. Practically, the number of multiple alignment solutions computed for each r is a user defined parameter.

First, we pick a *pivot* structure and require that it is included in all multiple alignments. In order to prevent dependency on a *pivot* structure, all input structures are iteratively selected to be a *pivot* one.

The MultiProt algorithm consists of three major stages. We call two sequential (without gaps) fragments of the same length to be ε-congruent if there exists a Euclidean 3D transformation that superimposes both fragments with rmsd (root mean square deviation) less than ε. In the first stage (*Detection of Fragment Pairs*), all ε-congruent fragment pairs are efficiently detected between the *pivot* and all other structures.

At the next stage (*Multiple Fragment Alignment*), we compute all possible combinations of ε-congruent multiple (sub)-fragments. This stage is analogous to the detection of all non-gaped local multiple alignments. To prevent an exponential number of multiple local alignments we do not compute them explicitly but rather store all possible alignments by means of combination sets (*Cuts*). Such set consists of one fragment from the *pivot* structure and its ε-congruent fragments from other structures, therefore such set may include several fragments from some molecule. The algorithm requires that the pivot molecule participates in the combination sets (*Cuts*), but it does not require that all input molecules from set S^p are included in the *Cuts*.

Finally (*Global Multiple Alignment*), for each local multiple alignment set we heuristically select (the problem of selecting the optimal combination is NP-Hard[2]) a combination of fragments, one fragment from each structure. Once a unique combination is selected we compute a global multiple correspondence between the C_α atoms. At this stage, we have a choice (user defined parameter) whether to compute a sequential alignment or a non-sequential one.

The main idea of the MultiProt approach is its ability to efficiently compute a

large number of local non-gapped multiple structure alignments. Essentially, a local multiple alignment is computed for each possible fragment of the input molecules. Such local alignments serve as a basis for the extension to the larger partial multiple alignments. In addition, in order to detect subset alignments, the solutions are scored separately according to protein composition, i.e. a scoring of alignment between proteins $\{a, b, c\}$ does not effect a ranking of an alignment between $\{a, b, d\}$ (the application for this requirement is demonstrated in Figure 3.4.2).

3.2.1 Stage 1. *Detection of Fragment Pairs.*

Given a pivot molecule M_p, for each molecule M_k from the set S^p we detect all structurally similar fragment pairs between M_p and M_k. Namely, a structurally similar (or ε-congruent) fragment pair is defined as $F_i^p F_j^k(l)$ (a fragment starts at point i (j) in molecule M_p (M_k) and has length l). It also satisfies the following condition: $RMSD_{opt}(F_i^p F_j^k(l)) \leq \varepsilon$. $RMSD_{opt}$ is defined as:

$$RMSD_{opt}(F_i^p F_j^k(l)) = min_T RMSD(F_i^p(l), T(F_j^k(l))),$$

where T is a rigid 3-D transformation.

To calculate all ε-congruent fragment pairs two options are implemented in the program. One can use an exact algorithm[ii], but in order to achieve a favorable running time of the program we can apply the same efficient greedy method as in the *FlexProt* algorithm[106]. We start by aligning a single matching atom pair (v_a, u_b), where $v_a \in M_p$ and $u_b \in M_k$. Now, we iteratively try to extend the initial match-list. We do this by adding one matching atom pair to the left and to the right (following the backbone direction) of the current fragment alignment. This is done iteratively, until the RMSD of the fragment alignment exceeds a predefined threshold. That is, we stop when the match list cannot be extended neither to the left, nor to the right. Given $F_i^p F_j^k(l)$, the next alignment is initiated at $(v_{i+(l+1)}, u_{j+(l+1)})$. The process can be viewed as proceeding along the diagonals of the 2D matrix, which represents the

[ii] All ε-congruent fragments, $\{F_i^p F_j^k(l)\}$, can be obtained by an exhaustive verification in polynomial time.

indices of M_p and M_k. Since RMSD can be continuously updated by a constant number (O(1)) of operations at each step, the time complexity of computing $F_i^p F_j^k(l)$ is only $O(l)$ and our greedy iterative approach takes only $O(|M_p| \cdot |M_k|)$. All presented experimental results are computed with the greedy method.

At the end of this step we have a set of congruent fragment pairs, $\{F_i^p F_j^k(l) : k \neq p,\ RMSD_{opt}(F_i^p F_j^k(l)) \leq \varepsilon\}$. The experimental number of such fragment pairs is $O(m \cdot n^2)$. The computation takes $O(m \cdot n^3)$ time.

Fragment Pair Clustering. Consider a 2D matrix that represents the indices of two proteins M_p and M_k. The detected congruent fragment pairs can be depicted as sub-diagonals in this matrix (Figure 3.2). In practice, it can be observed that the many fragment pairs are located very closely one to each other. This happens due to repetitive structural elements, like alpha-helices, that allow many structural matches with small shift in indices. In practice, it produces many spurious fragment pairs that slow down the program running times at advance stages. To detect such fragment pairs we perform the following clustering procedure.

Two fragment pairs (sub-diagonals), $F_i^p F_j^k(l)$ and $F_{i'}^p F_{j'}^k(l')$, are considered similar iff (1) the distance between corresponding diagonals is at most δ_1, i.e. $||i-j| - |i'-j'|| \leq \delta_1$ and (2) the overlap on both axes is at least some fraction δ_2 of their length, i.e. $min(i+l, i'+l') - max(i, i') \geq \delta_2 \cdot min(l, l')$ and $min(j+l, j'+l') - max(j, j') \geq \delta_2 \cdot min(l, l')$. For the clustering procedure we applied a simple iterative method. We pick some elements and create a cluster. Then, we add other elements that are similar to all elements in the cluster. When no element can be added, we initiate another cluster. For each completed cluster we select the longest fragment pair, the rest elements in the cluster are discarded.

To prevent missing the true-positive solutions, we selected very tight constraints, $\delta_1 = 5$ and $\delta_2 = 0.8$. On average, the number of fragment pairs after the clustering procedure is reduced by 39%.

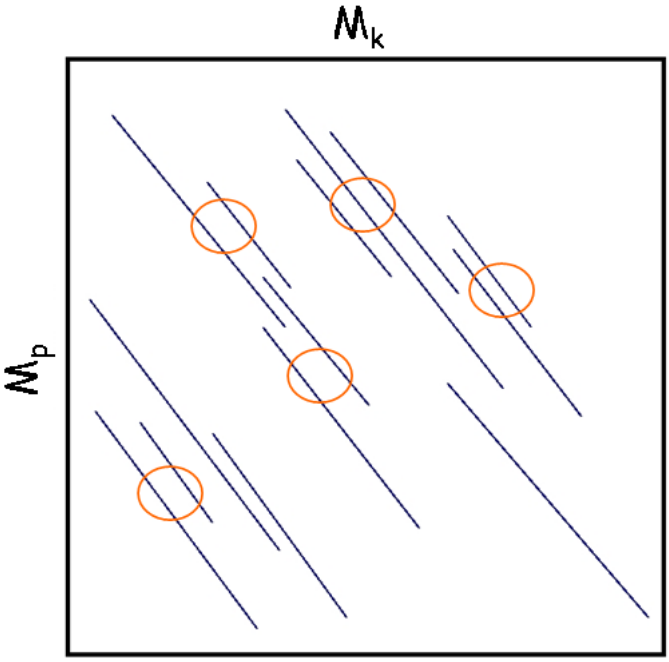


Figure 3.2: Fragment pair clustering.

3.2.2 Stage 2. *Multiple Fragment Alignment.*

Let us represent the above set by a 2 dimensional plot. The x-axis represents the sequence of the pivot molecule M_p. The y-axis is divided to bins, one bin for each molecule from set S^p. Fragment $F_j^k(l)$, from the pair $F_i^p F_j^k(l)$, is plotted in bin $M(k)$ (y-axis) and aligned to $F_i^p(l)$, i.e. its projection onto the x-axis is exactly the set of the corresponding (according to the $F_i^p F_j^k(l)$ alignment) points of the $F_i^p(l)$. The order of the fragments inside the $M(k)$ bin (on the y-axis) is arbitrary. See Figure 3.3.

Drawing two vertical lines at points α and β defines a *cut* on the interval $[\alpha, \beta]$. A fragment belongs to the *cut* if the interval $[\alpha, \beta]$ lies inside the fragment, i.e. $F_i^p F_j^k(l)$ is in the *cut* if and only if $i \leq \alpha$ *and* $\beta \leq (i + l - 1)$. Since several fragments from the same molecule might participate in the *cut*, such a *cut* provides us with a set of multiple choices for the alignment. Choosing from the *cut* only one fragment for each molecule gives us some multiple alignment. The number of choices equals $\prod_i k_{M_i}$ where k_{M_i} is the number of fragments from molecule M_i in the *cut*. Thus, the number

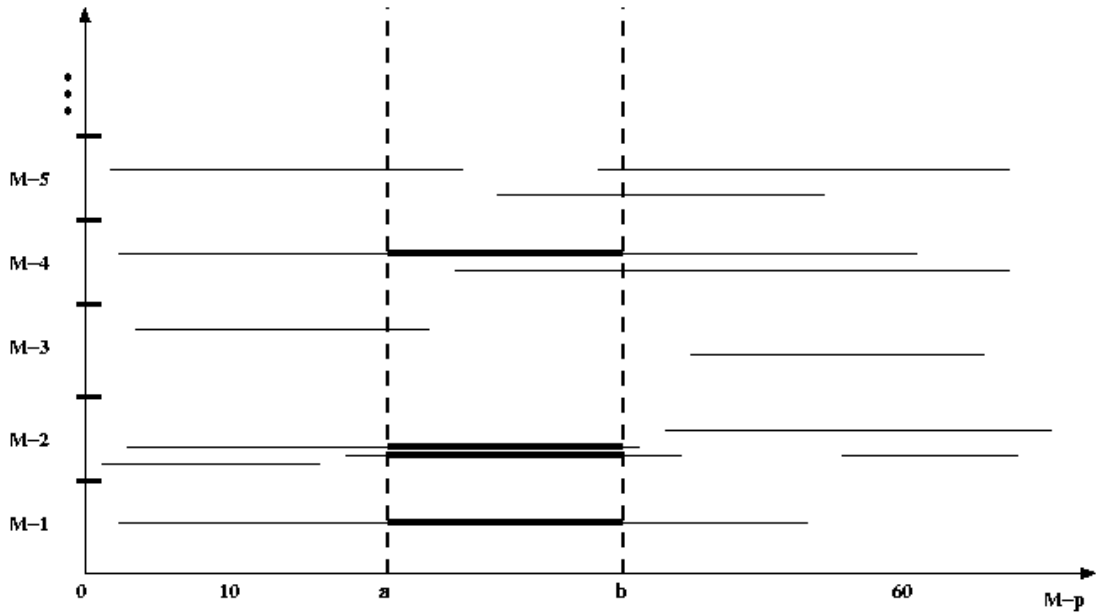


Figure 3.3: *Cuts.* The x-axis is the sequence of M_p. Molecules $M_1, ..., M_m$ are assigned into bins on the y-axis. Fragment pairs that include completely fragment $[\alpha, \beta]$ (shown in bold) are incorporated into $Cut[\alpha, \beta]$.

of possible multiple alignments (for the given *cut*) might grow exponentially with the number of molecules. This is the nature of the *multiple alignment* problem (there is an exponential number of choices to align α-helices from, for example, all-alpha-proteins). We shall return to this problem later in *Stage 3* after we explain how to detect the *cuts*.

From Figure 3.3 we can observe that it might be possible to extend a given *cut* to the left and to the right so that the *cut* contains the same fragments. Thus, we define a locally maximal cut $Cut[\alpha, \beta]$ as an interval $[\alpha, \beta]$ such that for any $\delta > 0$, $Cut[\alpha - \delta, \beta]$ and $Cut[\alpha, \beta + \delta]$ contain different fragments than $Cut[\alpha, \beta]$. It is obvious that any $Cut[\alpha, \beta]$ starts at the beginning of a fragment and ends at the end of a (possibly, another) fragment. From now on, we use the terms *cut* and $Cut[\alpha, \beta]$ interchangeably.

All possible *cuts* can be detected efficiently by a simple algorithm which is similar to the sweeping technique from Computational Geometry[27]. The idea is to represent

the projections on the x-axis of the fragment start(end)-points as events on the x-axis. The fragments left (right) end-point is the *start* (*end*) event. Starting from the first point (C_α atom) of the pivot molecule, M_p, we move along the x-axis (M_p sequence) with a vertical line and generate *cuts*. At every moment (i.e. position α of the vertical line) we remember the fragments that the vertical line passes through. At every *start* event we add its fragment to the list of current fragments and generate new *cuts*. These new *cuts* start at the encountered *start* event and end at *end* points of the fragments from the current list.

At an *end*-event we just remove the fragment from the current list of fragments. It is easy to see that the described algorithm generates all possible *cuts* according to the definition of $Cut[\alpha, \beta]$.

The complexity of the *cut* detection stage is linear in the number fragment pairs, i.e. $O(m \cdot n^2)$, where m is the number of input molecules and n is the size of the longest molecule.

3.2.3 Stage 3. *Global Multiple Alignment.*

Consider one of the previously detected *cuts* $Cut[\alpha, \beta]$. One of the possible approaches is to leave the *cuts* as is, i.e. not to choose the multiple alignment(s) from the set of possible ones of the specific *cut* (several fragments of the same molecule might be included in the *cut*). These complete *cuts* are included in the output of the program. Thus, an end-user can apply additional criteria to filter out the non-relevant fragments (or 3D transformations).

However, our goal is to detect the best multiple alignments based on the global structural similarity, as defined in (*). As we have already pointed out, this is a hard problem[iii], so we provide only a heuristic solution.

In this step, we select from every *cut* only one fragment for each molecule. We aim to perform this selection, in a way that the resulting multiple alignment would

[iii] In the first two stages we compute a set of transformations. However, even computing all possible transformations will not reduce the complexity of the multiple structure problem. The problem is NP hard even in the case of exact congruence ($\varepsilon = 0$). When $\varepsilon = 0$ all possible transformations can be computed in polynomial time (in 3D it is enough to match all possible triplets of points). Therefore to select the correct set of transformations is still a NP hard problem.

give, possibly, the highest score. Namely, given a $Cut[\alpha, \beta]$ containing $\{F_i^p F_j^k(l) : i \leq \alpha, (i + l - 1) \geq \beta\}$, for each different k (for each molecule M_k) select the fragment (if there is more than one fragment from molecule M_k) so that its transformation gives the largest, global, structural (pairwise) alignment with the pivot molecule (M_p). Let us explain this step in more detail. Given a fragment pair $F_i^p F_j^k(l)$ and the transformation T_{opt} that optimally superimposes $F_j^k(l)$ onto $F_i^p(l)$, we apply the transformation T_{opt} on molecule M_k. Now, when the two molecules are aligned, we calculate the size of their maximal structural alignment which preserves ε-congruence. For more details see Appendix 3.6.1.

At this stage of the algorithm every solution for the multiple structure alignment problem has a non-ambiguous representation, i.e. each solution contains at most one representative from each molecule. Now, the task is to score these solutions based on the size of the multiple alignment. Notice, that the transformation for each molecule is now fixed, thus we only need to calculate the *multiple correspondence* size. Computing multiple correspondence is NP-Hard (Section 2.4.2). We apply a very efficient procedure which guarantees m approximation to the optimal multiple alignment size. For the details of this procedure see Appendix 3.6.2. In case of small ε value (< 1.7Å), for which there is at most one C_α-atom from M_k inside a sphere of radius ε around C_α-atom from M_p, then this procedure gives the optimal result.

To enlarge the obtained solutions we apply an iterative improvement procedure as follows. For each solution, after the *multiple correspondence* between the pivot molecule with the other molecules is established, we apply a rigid transformation that minimizes the RMSD between the matching points. Then, we compute the *multiple correspondence* once again and repeat this procedure (the default number of iterations is 3).

Solution Scoring. When a common geometric core is detected, we compute the multiple *RMSD* (*mRMSD*) of the alignment. It is computed as an average of the *RMSD* values between the geometric core of the pivot molecule M_p with the corresponding geometric core of each molecule from the multiple alignment. Thus, solutions are grouped according to the number of aligned molecules and each group is sorted according to the size of the alignment and according to the *mRMSD*, giving

priority to the alignment size.

Optimization Schemes and Bio-Core detection. As described above we treated the problem as a pure geometrical structural alignment. We can apply a somewhat different scoring scheme, which requires that aligned points are of the same biological type (still, the points should be close enough in 3-D space). In our method the input points can be either positions of C_α or C_β atomic centers, or geometric centers of amino acids, or residue specific points (see description in Results section for the case of G-proteins). Therefore, each point represents a specific amino acid and thus, we can require that only points with similar characteristics be aligned. For instance, we can require residue identity matching, but it is usually too restrictive. Thus, we adopted the following classification: *hydrophobic (Ala, Val, Ile, Leu, Met, Cys), polar/charged (Ser, Thr, Pro, Asn, Gln, Lys, Arg, His, Asp, Glu), aromatic (Phe, Tyr, Trp), glycine (Gly).* Let us name this the *bio-core* classification.

At *Stage-2* the method detects a set of possible multiple transformations, while at *Stage-3* the solutions are scored based on the size of the multiple structural alignment. Therefore, we can apply various scoring schemes, like the *bio-core* classification, to *Stage-3* to obtain different solution ranking.

One of the ways used to measure sequence similarity of several proteins is to compute an average sequence identity. However this technique is based on pairwise properties and is dependent on gap penalty parameters. The *bio-core* classification possibly provides a more robust sequence characteristic than an average pairwise identity, since it is based on multiple structural alignment itself. In case that the *bio-core* is small relative to pure structural alignment size, then the common sequence properties are evolutionary distant. In addition, an optimization according to the *bio-core* classification may give a different structure superposition, which may be more appropriate in some cases. In the *Results* section we calculate the pure geometrical structural alignment as well as the *bio-core* alignment.

3.2.4 Complexity and Running Time Analysis

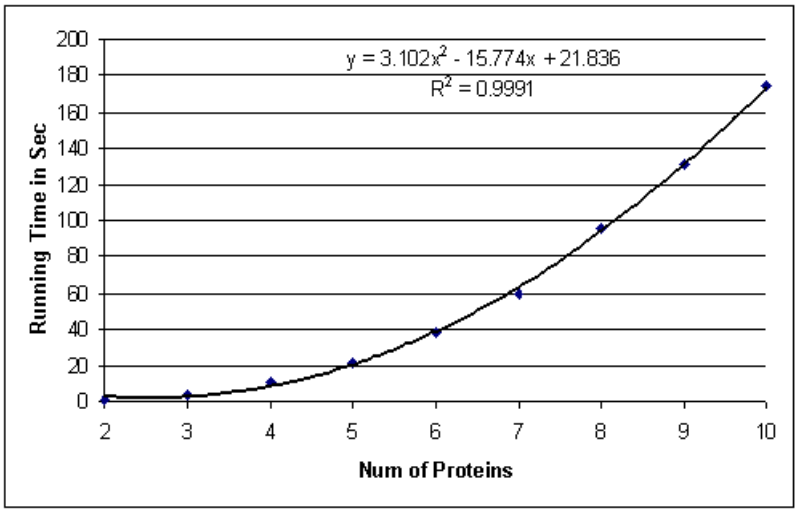
Given a pivot molecule we compute $O(m \cdot n^2)$ ε-congruent fragment pairs. This computation takes $O(m \cdot n^3)$ time. The stage of cut detection takes time linear in the number of ε-congruent fragment pairs, i.e. $O(m \cdot n^2)$. The number of computed cuts is $O(n^2)$. For each cut we spend $O(m \cdot n)$ to compute multiple alignment. There are m iterations over all possible pivots, therefore the total complexity is $m \cdot (O(m \cdot n^3) + O(m \cdot n^2) + O(n^2) \cdot O(m \cdot n)) = O(m^2 \cdot n^3)$.

In order to verify the MultiProt time complexity, $O(m^2 \cdot n^3)$, we conducted two experiments. First, we tested that the time complexity is proportional to $O(m^2)$. This is demonstrated in Figure 3.4. The quadratic running time behavior holds for average protein set (Figure 3.4 (a)) , as well as for proteins with the fixed size (Figure 3.4 (b) and Figure 3.5 (a)). In the second experiment, we verified that the MultiProt time complexity is proportional to $O(n^3)$. To demonstrate this, it is enough to run pairwise alignments for different protein sizes. For this task we selected Transferase, 1lxa, with Superhelix structural fold. This protein has an extremely repetitive structure created by beta-strands, see Figure 3.9 (a). Therefore, comparing this protein with itself creates near maximal number of ε-congruent fragment pairs, which represents the worst case scenario for our method. Figure 3.5 (b) shows cubic running time behavior. All experiments were conducted on a standard PC with Pentium(R) 4 2.00GHz.

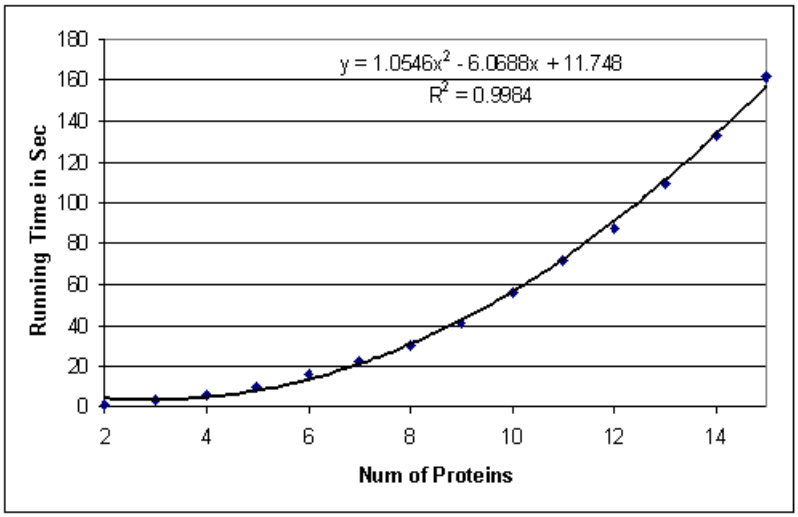
However, the practical running times are quite low. Structural alignment of up to seven average size proteins takes less than a minute (Figure 3.4 (a)).

3.3 Multiple Alignment Significance

To compute a significance of a multiple structural alignment we apply a simple estimation by the means of p-value. Naturally, the p-value depends on the number of input proteins, m, and their sizes. Therefore, we computed the multiple alignment size distribution for different values of m (practically only for $m = 2...10$). We selected a representative set of 5674 protein structures from the SCOP database (1.65)

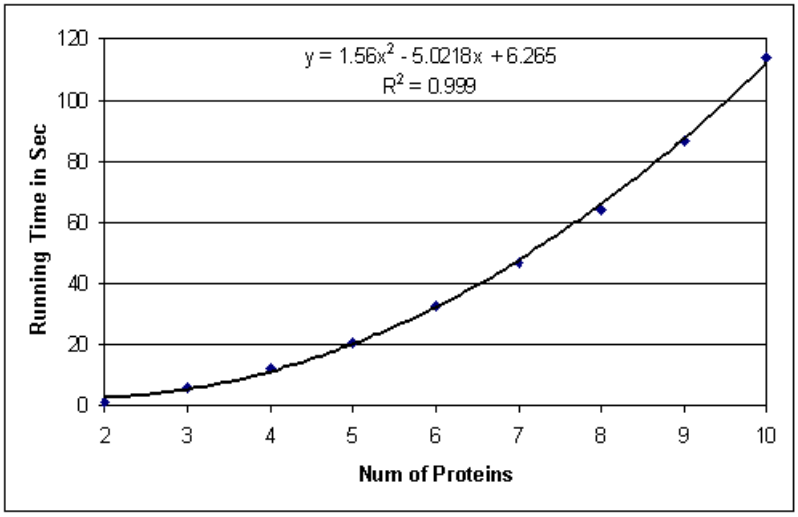


(a)

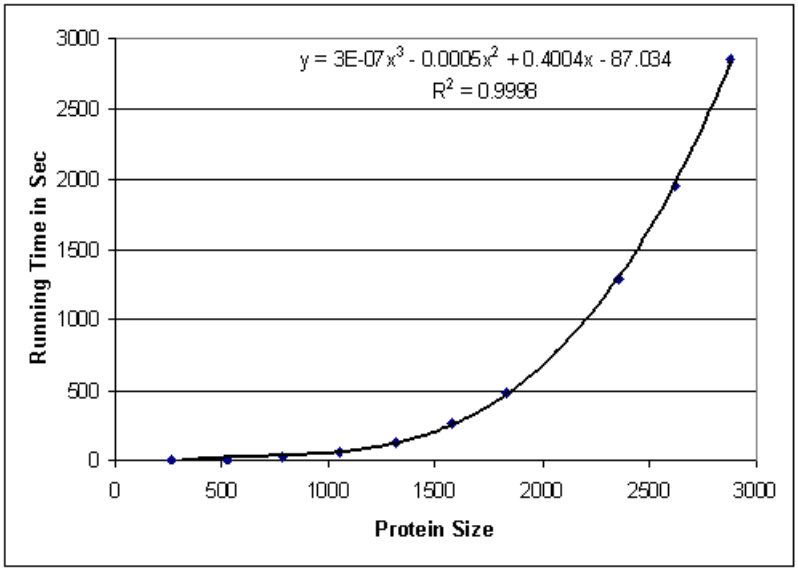


(b)

Figure 3.4: Two graphs show MultiProt quadratic running time behavior in the number of structures. (a) Average running time computed based on the random alignments of SCOP representatives from Section 3.3. Protein average size is 179 amino acids, (b) A protein structure (Transferase, 1lxa, $n = 262$, Superhelix structural fold) is compared against its own copies.



(a)



(b)

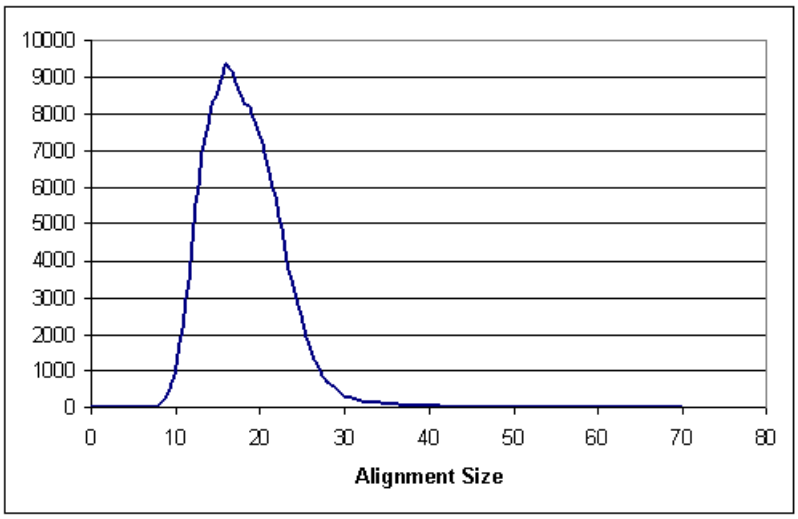
Figure 3.5: (a) Myoglobin, 5mbn, $n = 153$, is compared against its own copies. (b) Cubic running time as a function of protein size. Pairwise self-alignment of Transferase, 1lxa, with Superhelix structural fold. In each experiment the protein is extended with its own copy.

[92] which have less than 40% of pairwise sequence identity (this data set is provided by ASTRAL [20]). This resulted in 2304 protein domains (according to the SCOP classification). For each domain we arbitrarily selected only one structure. For each m, the number of structures, we applied MultiProt on m randomly selected structural domains. In total, we performed 10000 such random alignments for each m. We computed these distributions separately for sequential and non-sequential alignments. Example of multiple alignment size distribution is given in Figure 3.6 (a). Figure 3.6 (b) displays multiple alignment significance thresholds as a function of the number of input structures. The significance threshold is selected as 5% of the largest size alignments from the randomly picked protein domains. As it can be observed from the graph a chance for a large (around 100 amino acids) non-sequential pairwise alignment is relatively high. As expected, more structures impose tighter geometrical constrains and for more than five structures the sequential or non-sequential multiple alignment size larger than 20 C_α atoms is statistically significant.

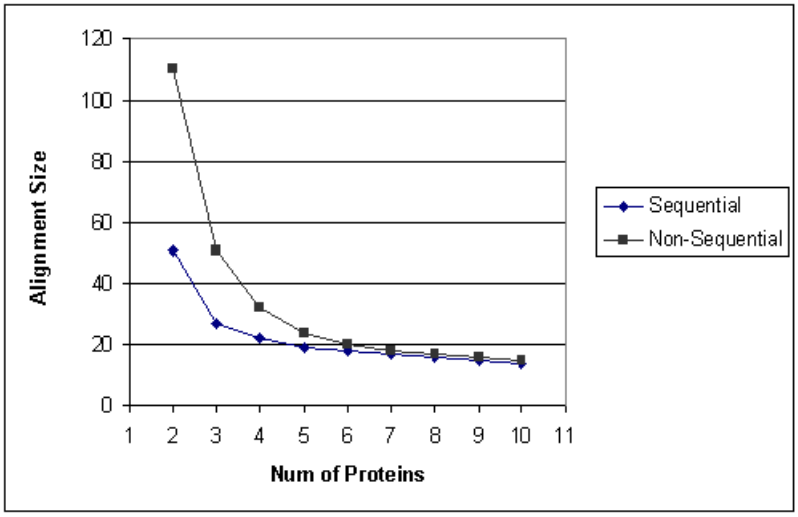
Clearly, larger structures will likely produce larger alignments. Therefore, in order to give a more accurate estimation we apply dependence on the structure size. Given some multiple alignment size and the minimal structure size, s_{min}, of the aligned molecules, we estimate its significance only from distributions of multiple alignments with minimal molecule size within 20% of s_{min}. This criterion is used below in the Results Section.

3.4 Results

Below we provide, along with known results, new multiple protein structural alignments. We present applications of the **MultiProt** method for (1) a non-sequential structural similarity, (2) subset and partial alignments, (3) identification of functional groups of G-proteins, (4) application to the analysis of binding sites and (5) protein-protein interface alignment. Protein structures are taken from the Protein Data Bank[14]. All experiments were performed on a standard PC with Pentium(R) 4 2.00GHz processor with 1024MB internal memory. *Software availability:* **MultiProt** is available for download at `http://bioinfo3d.cs.tau.ac.il/MultiProt/`.



(a)



(b)

Figure 3.6: (a) Distribution of alignment sizes between randomly selected three protein structures. The alignments preserve sequential order. 95% of the alignments have size $\leq$ 27. (b) Two graphs display alignment significance thresholds as a function of the number of input structures. The significance threshold is selected as 5% of the largest size alignments from 10000 randomly picked protein domains. For example, for the randomly picked three structures the chance that a sequential (non-sequential) multiple alignment has the size larger than 27 (57) C_α atoms is 0.05.

The program output consists of 10 (changeable parameter) highest-scoring results for each number of molecules, i.e. if the number of input molecules is 15, then there are sets of results for 2, 3, .. 15 aligned molecules. Each result lists (1) 3-D rigid transformation for each aligned molecule, (2) matrix of aligned amino acids, (3) RMSD of the multiple alignment (mRMSD), calculated as described above. We nickname the largest structural (bio) core to be *struct-core* (*bio-core*). The computed p-value for all the experimental results presented below is less than 0.05, therefore all reported multiple alignments can be considered as significant.

3.4.1 Comparisons with Other Methods:

Pairwise Alignment Cases.

First, we test our method on "hard to detect" *pairwise* alignments. We repeated the experiment presented in Shindyalov & Bourne[115]. The experiment presents a set of ten protein pairs and pairwise alignments performed by different (pairwise) methods. The results are presented in Table 3.1. Two kinds of **MultiProt** results are given: alignments which preserve protein backbone order and sequence order independent alignments. As can be observed from the table, our pairwise results are very competitive. The maximal running time (pair 1crl:534, 1ede:310) is less than 4 s.

Globins.

The globin family has been extensively studied in the literature[12, 132]. We applied MultiProt on seven globin structures (5mbn, 1ecd, 2hbg, 2lh3, 2lhb, 4hhbA, 4hhbB). We compared our results with those obtained in Wu et al.[132]. The largest geometrical core detected by their method[132] consists of 105 C_α atoms (or "corresponding landmarks" as called in the paper). Our program obtains similar results. The size of the detected common *struct-core* varied between 93 C_α atoms ($\varepsilon = 3\mathring{A}$) to 111 C_α atoms ($\varepsilon = 4\mathring{A}$). The structural similarity is detected primarily between α-helices, while loop regions were left un-aligned. The detected *bio-core* is comparatively small. It ranged from 18 C_α atoms ($\varepsilon = 3\mathring{A}$) to 31 C_α atoms ($\varepsilon = 4\mathring{A}$). The running time

was about 15 seconds.

Comparison with sequence order independent multiple alignment method.

A comparison with the results achieved by the **MUSTA** algorithm[75], illustrates that our method achieved similar alignment results. The test includes the following cases:

- *Serpins* family - 7apiA, 8apiA, 1hleA, 1ovaA, 2achA, 9apiA, 1psi, 1atu, 1kct, 1athA, 1attA, 1antl, 2antl.
- *Serine Proteinase*: *Subtilases* family - 1cseE, 1sbnE, 1pekE, 3prkE, 3tecE.
- *Calcium-binding*: *EF hand-like* superfamily. The protein 4cpv is from the *parvalbu-min* family; 2scpA, 2sas, 1top, and 1scmB from the *calmodulin-like* family; and 3icb from the *Cal-binding D9K* family.
- *TIM-Barrels*: The proteins are taken from the 7 different superfamilies. See details in Table 3.2.
- *Helix-Bundle* The proteins are taken from the 6 different superfamilies. Details are given in Table 3.2.

In all cases the size of the geometric core is at least the same (see Table 3.3 (a)). In addition, our method produced high-scoring partial alignments and runs significantly faster (on the same computer).

Comparison with multiple alignments taken from the HOMSTRAD data base.

The HOMSTRAD[89] data base contains multiple alignments of homologous protein families. An average sequence identity (sID) in the protein families varies between 8 and 94 percent. We performed a comparison of 6 families, calcium-binding protein (sID 56%), subtilase (sID 52%), serine proteinase inhibitor (sID 34%), reductases (sID 22%), TPR domain (sID 17%) and plant virus coat protein (sID 13%).

Molecule 1 (size)	Molecule 2 (size)	VAST S_{al}/rms	Dali S_{al}/rms	CE S_{al}/rms	GH S_{al}/rms	**MultiProt**$_1$ S_{al}/rms	**MultiProt**$_2$ S_{al}/rms
1fxi:A(96)	1ubq(76)	48/2.1	-	-	51/1.6	44/1.7	50/1.8
1ten(89)	3hhr:B(195)	78/1.6	86/1.9	87/1.9	81/1.7	81/1.3	82/1.3
3hla:B(99)	2rhe(114)	-	63/2.5	85/3.5	62/1.8	60/1.8	67/1.9
2aza:A(129)	1paz(120)	74/2.2	-	85/2.9	74/1.9	75/2.0	85/2.5
1cew:I(108)	1mol:A(94)	71/1.9	81/2.3	69/1.9	66/1.6	76/1.8	75/1.9
1cid(177)	2rhe(114)	85/2.2	95/3.3	94/2.7	70/1.5	84/1.8	88/1.9
1crl(534)	1ede(310)	-	211/3.4	187/3.2	180/1.9	161/2.3	232/2.4
2sim(381)	1nsb:A(390)	284/3.8	286/3.8	264/3.0	197/2.0	233/2.3	268/2.3
1bge:B(159)	2gmf:A(121)	74/2.5	98/3.5	94/4.1	72/1.8	78/2.5	88/2.2
1tie(166)	4fgf(124)	82/1.7	108/2.0	116/2.9	87/1.7	95/2.1	99/2.3

S_{al} is the number of aligned atoms.

Table 3.1: **Pairwise structural alignment test.** The protein pairs are classified as 'difficult' for structural analysis[41]. The alignments are performed by **VAST**[82], **Dali**[59], **CE**[115], **GH** Geometric Hashing method[11](http://bioinfo3d.cs.ac.il/c_alpha_match/) and **MultiProt**. The information in this table, except for the Geometric Hashing method and **MultiProt** results, is taken from Shindyalov & Bourne[115]. MultiProt$_1$ results do preserve the sequence order, while MultiProt$_2$ are sequence order independent.

TIM-Barrels

Superfamily	PDB code
Triosephosphate isomerase	7timA
Cellulases	1tml
Glycosyltransferases	1btc
Enolase C-terminal domain-like	4enl
Ribulose-phosphate-binding barrel	1pii
Xylose isomerase	6xia
RuBisCO, C-terminal domain	5rubA

Helix-Bundle

Superfamily	PDB code
Protein designs	1flx
Apolipophorin III	1aep
4-helical cytokines	1bgeB,1rcb,3inkC
Apoliporotein	1le2
Cytochromes	256bA,2ccyA,1bbhA
Hemerythrin	2hmzA

Table 3.2: Structural classification of protein sets used for the comparison with the MUSTA method[75].

Proteins	No. of Mols	Average Size	MUSTA S_{al}	MultiProt S_{al}	MultiProt run-time
Serpins 7apiA, 8apiA, 1hleA, 1ovaA, 2achA, 9apiA, 1psi, 1atu, 1kct, 1athA, 1attA, 1antl, 2antl	13	372	163	237	9m04s
Serine Proteinase 1cseE, 1sbnE, 1pekE, 3prkE, 3tecE	5	277	220	227	25s
Calcium-binding 4cpv, 2scpA, 2sas, 1top, 1scmB, 3icb	6	140	31	36	9s
TIM-Barrels 7timA, 1tml, 1btc, 4enl, 1pii, 6xia, 5rubA	7	391	40	44	3m12s
Helix-Bundle 1flx, 1aep, 1bbhA, 1bgeB, 1le2, 1rcb, 256bA, 2ccyA, 2hmzA, 3inkC	10	140	27	27	2m10s

S_{al} is the number of aligned atoms.

Table 3.3: Comparison of MultiProt with the MUSTA[75] algorithm.

Proteins	No. of Mols	Average Size	HOMSTRAD S_{al}	MultiProt S_{al}	MultiProt run-time
Calcium-binding 1rtp, 1pvaA, 5cpv, 1pal, 5pal, 1omd, 1a75A	7	107	101	101	2s
Subtilase 1dbiA, 1thm, 1bh6A, 1a1yE, 1meeA, 1sup, 1gci, 2prk	8	274	217	221	48s
Serine Proteinase Inhibitor 2ach, 1qlpA, 1athA, 1attA, 1hle, 1ovaA, 1a7cA, 1sek	8	376	270	269	2m21s
Reductases 2cnd, 1ndh, 1que, 1qfzA, 1fdr, 1a8p, 1qfjA	7	266	131	121	38s
TPR domain 1a17, 1elwA, 1elrA, 1e96B, 1fchA, 1ihgA	6	153	68	86	6s
Plant virus coat protein 2tbvA, 4sbvA, 1smvA, 1stmA, 1bmv1, 1cwpA, 2stv	7	175			
$\varepsilon = 3\mathring{A}$			33	34	16s
$\varepsilon = 4\mathring{A}$			74	76	21s

S_{al} is the number of aligned atoms.

Table 3.4: Comparison of MultiProt with the HOMSTRAD data base.

To make an adequate comparison we performed the following procedure. From the HOMSTRAD data base, for each of the 6 families we extracted the multiple structural alignments. Then, we computed the alignment size according to the MultiProt scoring method with the default parameters. No optimization on multiple transformations was performed. The computed alignment size represents the scoring of the HOMSTRAD data base. Finally, we run MultiProt on the unaligned structures. As can be observed from Table 3.4 (b) the performance of both methods is very close. Additional, large scale comparison of MultiProt with HOMSTRAD multiple alignments is given in Section 4.3.1.

3.4.2 Applications of MultiProt

Here we demonstrate a variety of biological applications of **MultiProt**. We start with case studies of non-topological multiple alignments, unexpected alignment between alpha-helix and beta-strand, and multiple alignment of *Superhelix*, *Concanavalin* and *Supersandwich* families. Then, we demonstrate a detection of subset and partial alignments, identification of functional groups of G-proteins and an application to the analysis of binding sites and protein-protein interface alignment.

Non-sequential structural similarity

We consider an alignment of a 4-helix bundle. A 4-helix arrangement appears in a large number of proteins. SCOP includes at least 40 folds with a 4-helix bundle. Holm & Sander[59] show an alignment of the Rop protein (1rop) with cytochrome b56 (256b). Both proteins have 4-helix bundle, but the topological arrangement is different, i.e. when the two structures are aligned, at least one helix-pair is aligned in an opposite sequential order. Here we show a multiple structural alignment of 4 proteins (1f4n, 2cbl:A, 1b3q, 1rhg:A) which share a 4-helix bundle (see Figure 3.7). Figure 3.7 (c) shows the direction of the protein sequences according to a structural alignment when all 4 helices are aligned. As one can see the direction is different for the last two helices. Thus, none of the commonly used sequence alignment methods can align simultaneously the 4 α helices. Figure 3.7 (b) shows a multiple structural

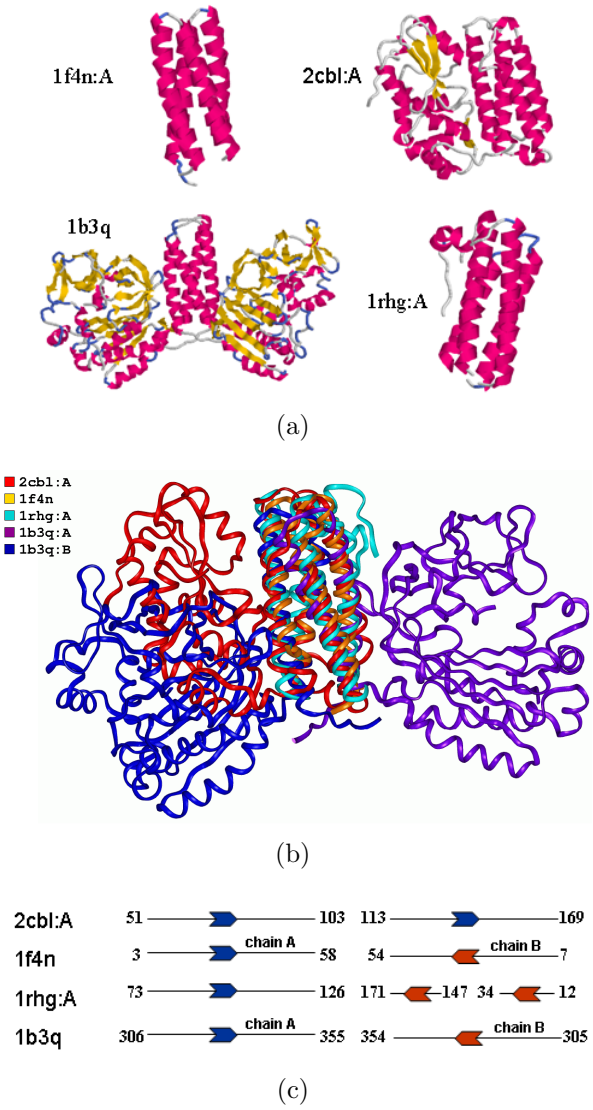


Figure 3.7: 4-helix bundle. (a) There are four structures in our study, 1f4n, 2cbl:A, 1b3q and 1rhg:A. (b) Structural alignment between 1f4n, 2cbl:A, 1b3q, 1rhg:A. 4 -helix bundle is aligned. (c) A multiple alignment of the sequences according to the multiple structural alignment of the 4-helix bundle. Notice, the directions are different. Since proteins 1f4n, 1 b3q have two chains, it leads to an additional difficulty for sequence alignment methods in identifying a 4-helix bundle.

alignment with the 4 helices aligned. See details in Table 3.6.

Alpha-Helix Beta-Strand Alignment

Protein representation by secondary structure elements significantly reduces the problem input size. If an average protein has about 300 amino acids, then the number of secondary structures is only about 10. Such a reduction significantly improves a search against large databases. Alignment methods that utilize a secondary structure representation, in essence, try to align α-helix with α-helix, β-strand with β-strand. However, care should be taken, since in some cases an alignment between an α-helix and β-strand could have a biological meaning.

A well known CYS-CYS bond is conserved for protein fold stability[87]. We considered 51 proteins taken from "C1 set domains (antibody constant domain-like)" family (according to SCOP). A CYS-CYS bond was multiply aligned between all 51 proteins. While in 50 proteins the CYS-CYS bond appears between two β-strands, in one protein it is created between α-helix and β-strand[iv]. For details see Figure 3.8.

Superhelix

In this experiment we compare 5 proteins (1lxa, 1qq0, 1xat, 2tdt, 1fwy(A:252-328)) from the *Superfamily: Trimeric LpxA-like enzymes*. Each protein is taken from a different family (for details see Table 3.5). While the first 4 molecules are between 208 and 274 residues long, the last one (1fwy, A:252-328), is a truncated form and has only 77 residues. Our algorithm detected multiple alignment between all 5 molecules with *struct-core* of size 64 with *mRMSD* 0.9 $\AA$. The *bio-core* for these molecules consisted of 17 C_α-atoms. See the alignment in Figure 3.9 (a).

Four molecules (the first four) gave 88 (18) C_α-atoms in the *struct-core* (*bio-core*). Molecules 1lxa, 1qq0 and 2tdt gave 114 (27) C_α-atoms in the *struct-core* (*bio-core*). (There are additional combinations of the first four molecules which are presented in the solutions). Molecules 1lxa.pdb and 1qq0.pdb gave 143 (62) C_α-atoms in the *struct-core* (*bio-core*). For the three molecules there are other molecule pairs that

[iv]This test case has been provided by Hadar Benyamini.

Superhelix

Family	PDB code
UDP N-acetylglucosamine acyltransferase	1lxa
Carbonic anhydrase	1qq0
Xenobiotic acetyltransferase	1xat
Tetrahydrodipicolinate-N-succinlytransferase, THDP-succinlytransferase, DapD	2tdt
N-acetylglucosamine	1fwy
1-phosphate uridyltransferase GlmU, C-terminal domain	A:252-328

Concanavalin A-like lectins/glucanases

Family	PDB code
Legume lectins	2bqpA
beta-Glucanase-like	1gbg
Galectin (animal S-lectin)	2galA
Laminin G-like module	1d2sA
Pentraxin (pentaxin)	1sacA
Clostridium neurotoxins, the second last domain	1a8d:1-247
Vibrio cholerae sialidase, N-terminal and insertion domains	1kit:25-216
Leech intramolecular trans-sialidase, N-terminal domain	2sli:81-276
Endoglucanase/cellulase I catalitic core	6cel
Xylanase/endoglucanase 12	1xnb

Table 3.5: Structural classification of studied proteins by SCOP database.

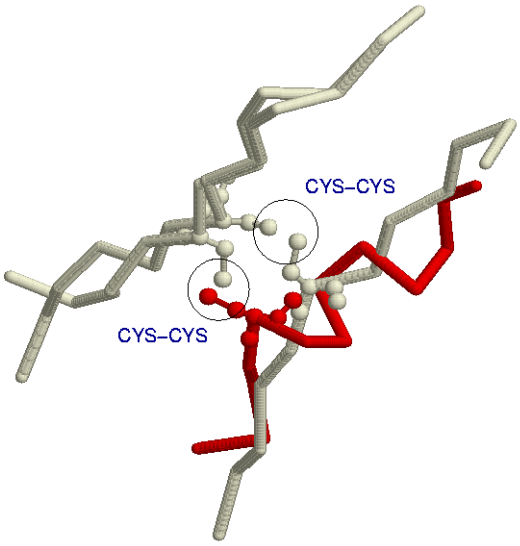


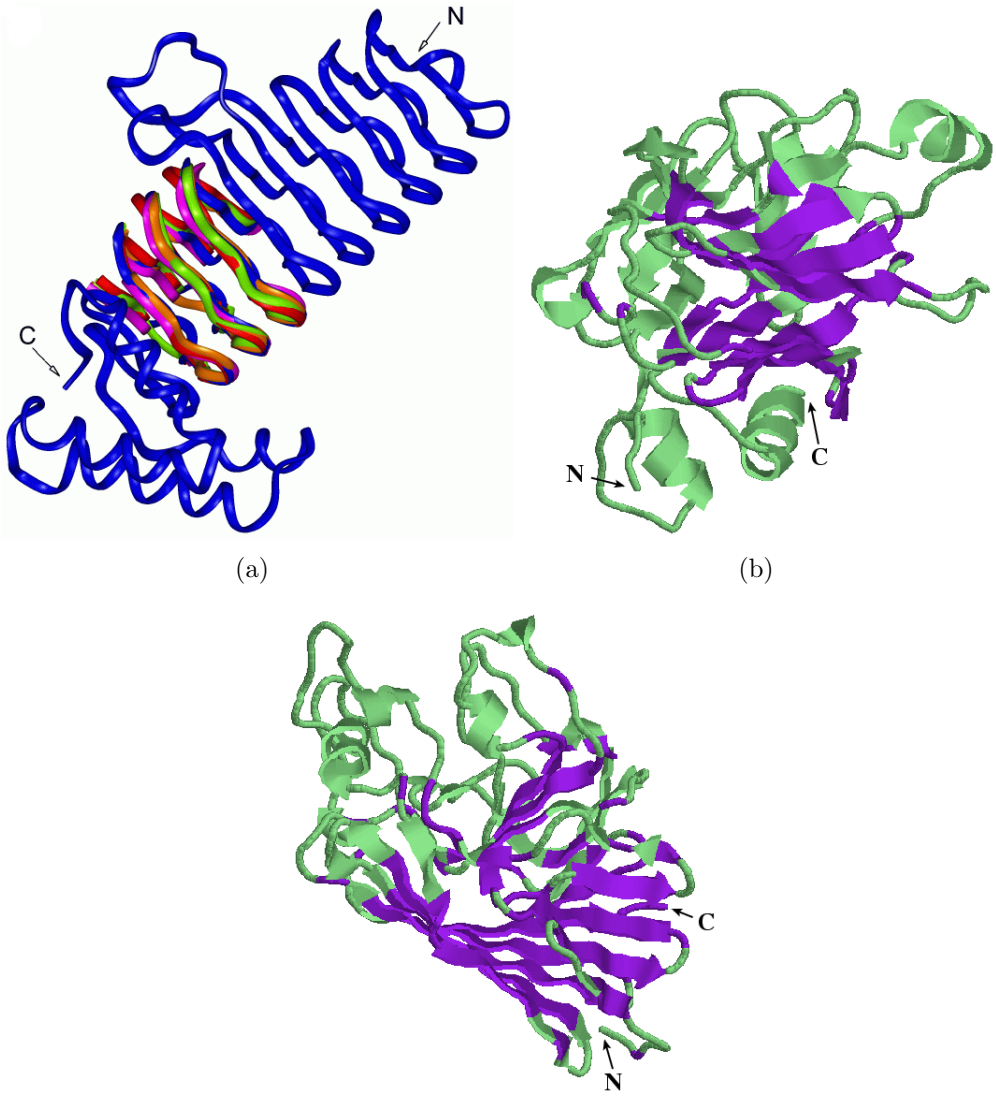
Figure 3.8: The figure shows only two molecules derived from the multiple alignment of 51 proteins taken from "C1 set domains (antibody constant domain-like)" family. While in 1iak:A CYS-CYS bond appears between two β-strands (β-strands are in grey color), whereas in 1r24:A CYS-CYS bond appears between α-helix (in red) and β-strand. This example demonstrates that structural alignment between secondary structures of different types may have a biological meaning.

gave high similarities. The running time was about 14 seconds. For details see Table 3.6.

Concanavalin A-like lectins/glucanases

From the SCOP database we selected the *Concanavalin A-like lectins/glucanases* fold (sandwich; 12-14 strands in 2 sheets; complex topology). This fold contains only one *superfamily* which includes 10 different families. We selected one protein from each family (for details see Table 3.5): 2bqpA, 1gbg, 2galA, 1d2sA, 1sacA, 1a8d:1-247, 1kit:25-216, 2sli:81-276, 6cel, 1xnb.

Aligning all 10 molecules results in a geometric core of size 54. Interestingly, Tetanus Neurotoxin (1a8d:1-247) participated in all alignments containing different numbers of molecules. This protein has 5 (A, B, C, D, E) β-sheets. A, C and D create almost one β-sheet (let us call it S1), and so do B and E (S2). In the alignment of all



(a) (b)

(c)

Figure 3.9: (a) Superhelix. Figure (a) shows the structural core between 5 molecules, (1lxa, 1qq0, 1xat, 2tdt, 1fwy(A:252-328)). The complete backbone of 1lxa is shown in blue. For the other molecules only the common detected core is shown, by assigning a different color to each molecule. (b) *Concanavalin A-like lectins/glucanases.* Structural core of 6 molecules - 1a8d:1-247, 1d2sA, 1gbg, 1sacA, 2galA, 2sli:81-276. The *two-sheet sandwitch* is conserved. The backbone of molecule 1a8d:1-247 is shown completely in light green. The aligned core is in purple color. (c) *Supersandwich.* Multiple structural alignment of proteins 1bgmI:731-1023, 1cb8A:336-599 and 1oacA:301-724. The backbone of 1bgmI:731-1023 is shown in its entirety in light green. The aligned core is in purple color. 3 β-sheets are aligned.

Proteins	No. of Mols	Average Size	S_{al}	run time
4-helix bundle				3s
1f4n, 2cblA, 1b3q, 1rhgA	4	284	75	
1b3q, 1rhgA	2	451	102	
Superhelix				14s
1lxa, 1qq0, 1xat, 2tdt, 1fwyA:252-328	5	205	64	
1lxa, 1qq0, 1xat, 2tdt	4	238	84	
1lxa, 1qq0, 2tdt	3	248	114	
1lxa.pdb, 1qq0.pdb	2	235	143	
Supersandwich				12s
1bgmI:731-1023, 1cb8A:336-599, 1oacA:301-724	3	360	118	
1cb8A:336-599, 1oacA:301-724	2	393	187	
Concanavalin				54s
2bqpA, 1gbg, 2galA, 1d2sA, 1sacA, 1a8d:1-247, 1kit:25-216, 2sli:81-276, 6cel, 1xnb	10	220	54	
1a8d:1-247, 1d2sA, 1gbg, 1sacA, 2galA, 2sli:81-276	6	194	75	
1a8d:1-247, 1gbg, 6cel	3	298	128	
tRNA synthetase				39s
1adjA, 1hc7A, 1qf6A, 1atiA-AntiCodon	4	409	75	
1adjA, 1hc7A, 1qf6A	3	508	176	
G-proteins				29s
1agr, 1tad, 1gfi, 1tx4, 1grn, 1wql	6	370	13	
1agr, 1tad, 1gfi	3	350	199	
PTB domain				8s
1x11,1irs,1shc,1ddm,2nmb,1evh	6	147	66	
1shc,1ddm,2nmb	3	168	111	

S_{al} is the number of aligned atoms.

Table 3.6: Multiple structural alignment results performed by MultiProt.

10 molecules only β-sheet C is aligned well (from 7 β-strands 3 were aligned well and 2 received only small partial alignments) and β-sheet E obtains only a small partial alignment. Investigating multiple alignments containing fewer molecules, we notice that the common core of β-sheet S1 increases and so does that of S2. See Figure 3.9 (b) for a geometric core of 6 molecules - 1a8d:1-247, 1d2sA, 1gbg, 1sacA, 2galA, 2sli:81-276. The size of the *struct-core* is 75 C_α atoms with *mRMSD* 2.0 $\mathring{A}$. The *bio-core* size of these molecules is 9 atoms. The running time was 54 seconds. For details see Table 3.6.

Supersandwich

In this experiment we selected from the SCOP database[92] the *Supersandwich* fold from the *All beta proteins* class. This fold contains three superfamilies. From each superfamily we selected one protein, β-Galactosidase (1bgmI:731-1023), Chondroitinase Ac (1cb8A:336-599) and Copper Amine Oxidase (1oacA:301-724). Our multiple alignment result contains 118 C_α atoms with *mRMSD* 2.21 $\mathring{A}$. β-Galactosidase (1bgmI:731-1023) has 17 strands. Only two strands (914-921,894-901) are not in the alignment. This example demonstrates that our method does not totally depend on the order of the residues in the backbone chain. A number of strands were aligned in the opposite order. Below is part of the alignment (notice the alignment between 1bgmI and 1oacA):

1bgmI: 738 739 740 741 746 747 748 749 750
1cb8A: 444 442 340 341 348 342 350 351 339
1oacA: 327 325 324 323 332 331 330 329 328

See Figure 3.9 (c) for the aligned core. The *bio-core* of this multiple alignment contains 23 atoms, which is a subset of the largest detected *struct-core*. The running time was about 12 seconds. For details see Table 3.6.

Detection of Subset Alignment

In this experiment, in order to show the ability of our method to detect subset alignments, we included in the input set 18 proteins. 5 proteins from the *Superhelix* experiment (1lxa, 1qq0, 1xat, 2tdt, 1fwy(A:252-328)), 3 proteins from the *Supersandwich* experiment (1bgmI:731-1023, 1cb8A:336-599, 1oacA:301-724) and 10 proteins from the *Concanavalin A-like lectins/glucanases* experiment (2bqpA, 1gbg, 2galA, 1d2sA, 1sacA, 1a8d:1-247, 1kit:25-216, 2sli:81-276, 6cel, 1xnb).

It took only 8 minutes for the program to compare these 18 molecules. The multiple structural alignments for 3 molecules contained an alignment of the *Supersandwich* family. The results for 5 molecules contained the alignment of the *Superhelix* family. The results for 6 molecules contained the alignment of 6 proteins from *Concanavalin A-like lectins/glucanases* family (1a8d:1-247, 1d2sA, 1gbg, 1sacA, 2galA, 2sli:81-276), shown in Figure 3.9 (b). Therefore, MultiProt detected the correct subset alignments. Since both families *Supersandwich* and *Concanavalin A-like lectins/glucanases* contain a number of β-sheets, 10-13 molecules contained the mixed alignments of proteins from these two families.

Detection of Partial and Subset Alignments

Here we demonstrate the ability of MultiProt to detect partial and subsets multiple alignments. We consider five multi-domain molecules. Some domains are structurally similar. Our task is to identify these structurally similar domains.

The five proteins included in this study are Histidyl-tRNA Synthetase (1adj:A), Prolyl-tRNA Synthetase (1hc7:A), Threonyl-tRNA Synthetase (1qf6:A), Asparagine Synthetase (12as:A) and Anticodon Binding Domain From Nuclear Receptor Coactivator 5 (1v95:A). Our goal in studying such a set of proteins is to identify the two common domains (see Figure 3.10), *Class II aminoacyl-tRNA synthetase (aaRS)-like, catalyic domain* and *Anticodon-binding domain of Class II aaRS* (the classification is according to SCOP[92]). For simplicity we call these domains A and B.

The multiple alignment for all five structures resulted in a common structural

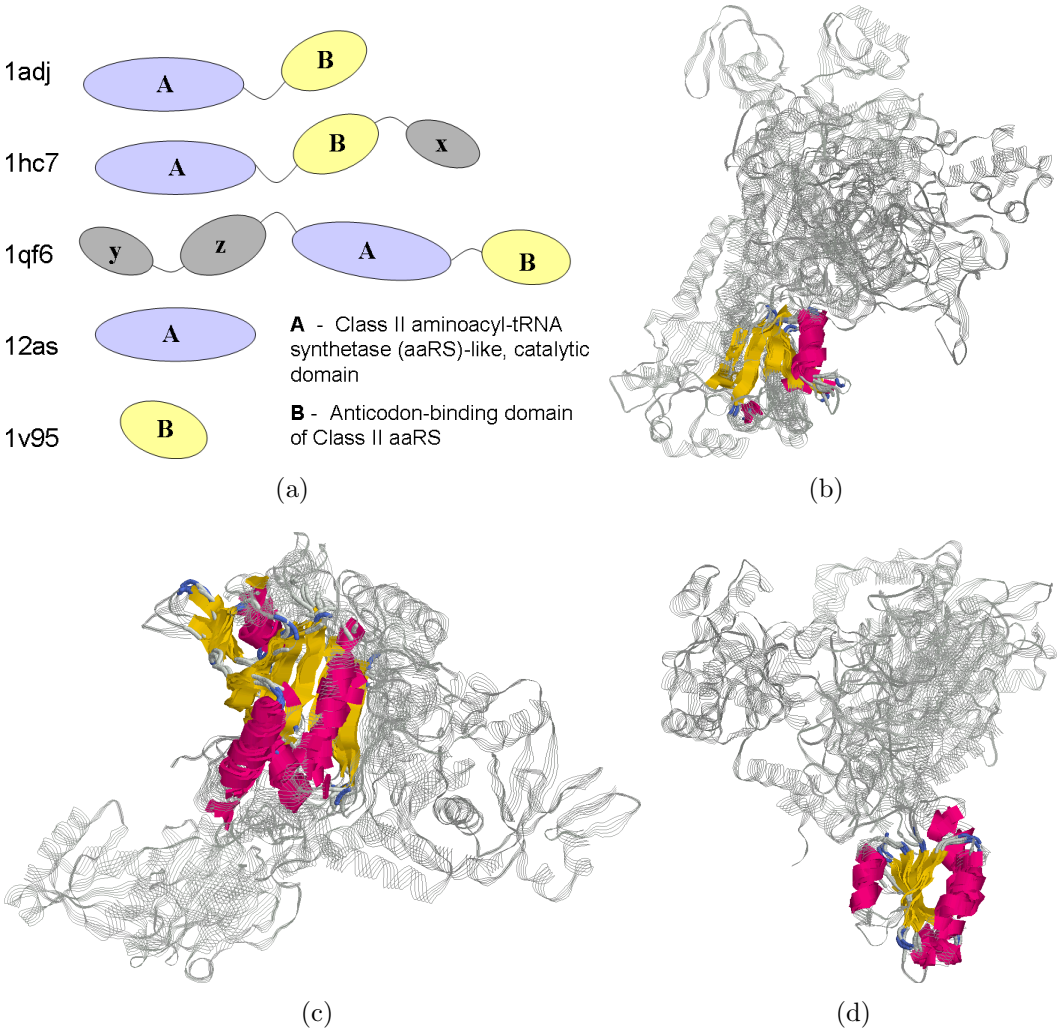


Figure 3.10: Partial and Subset Alignments. (a) Simplified schematic view of protein domains of 1adj:A, 1hc7:A, 1qf6:A, 12as:A, 1v95:A. (b) All five proteins are aligned. The common core is 39 amino acids (p-value < 0.001). (c) Domain A is aligned between the first four proteins. The common core has 125 amino acids, this is highest ranked solution for four structures. (d) Domain B is aligned between 1adj:A, 1hc7:A, 1qf6:A and 1v95:A. The common core has 76 amino acids, this is the second ranked solution with molecule id composition different from the larger alignment of domain A.

core of size 39 amino acids, consisting mainly of beta-sheet and alpha helix. Despite the fact that these two domains are differently classified there is some *partial* non-random (p-value < 0.001) structural similarity. The solutions containing four structures revealed two high scoring multiple alignments with different protein composition. These multiple alignments are alignments of the first domain (1adj:A, 1hc7:A, 1qf6:A, 12as:A) and of the second domain (1adj:A, 1hc7:A, 1qf6:A, 1v95:A). Therefore in this example, MultiProt successfully carries out the task of subset and partial multiple alignment. The running time is 2 minutes.

It is worth noting that multiple alignment of the first three proteins (1adj, 1hc7 and 1qf6) does not align domains A and B at the same time. This is due to a hinge motion between domain A and B. Thus, there is no 3D transformation that simultaneously aligns both domains. Consequently, since domain B is significantly smaller than A, it is found ranked at the 20th place. Therefore, in such case a multiple *flexible* structural alignment would produce a more meaningful result by aligning A and B at the same time.

G proteins

In this experiment we performed a multiple alignment of 6 G-proteins[73]. A binding site of these proteins has two conserved amino acids Gln and Arg. In three proteins 1agr, 1tad and 1gfi the conserved amino acids are located on the same chain while in the 1tx4, 1grn and 1wql these amino acids are on different chains. Therefore, no sequence analysis method can detect these structurally conserved functional groups. Thus, to detect this structurally conserved pattern, a proper protein representation should be selected. The distance between C_α atoms of the conserved Arg of the first group (1agr, 1tad and 1gfi) and the second group (1tx4, 1grn and 1wql) is about 10 Å. Therefore, a method that aligns C_α atoms will not be able to recognize the functionally conserved Arg, since a distance threshold of 10 Å is not realistic.

For this experiment we represented the proteins by side-chain specific points. For Phe, Tyr, His and Pro a geometric center of the ring was selected. Trp, Val, Ile, Thr were represented by a side chain geometric center; Ala, Ser - by C_β atoms; Gly - by a pseudo C_β atom; and Cys, Met - by a sulfur atom. For Asp, Glu, Lys,

Asn, Gln, Arg the last side chain carbon atom was chosen. Leu was represented by a geometric center of the last 3 carbon atoms. Using such a representation, **MultiProt** successfully detected conserved Gln and Arg (Figure 3.11), as well as other 11 amino acids. The selected side-chain specific representation was encouraged by this example. We are currently looking for other cases where such a representation would detect a correct structural alignment of conserved functional residues. Interestingly, multiple alignment using C_α atoms produces the same 3D superposition, but as explained above, the functionally conserved arginines are not 'detected' to be aligned. This problem could be resolved by applying some post-analysis method. However, using a priori the correct representation is more advantageous, since the chance for false-positive solutions is lower.

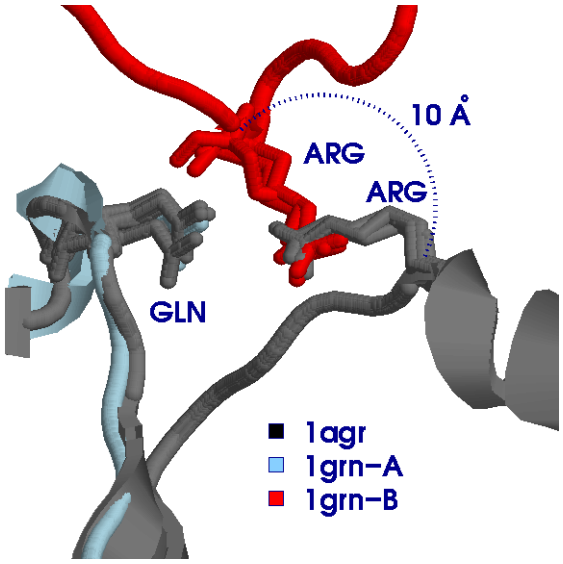


Figure 3.11: The binding site of G-proteins has two functionally conserved amino acids Gln and Arg. In proteins 1agr, 1tad and 1gfi the conserved amino acids are located on the same chain while in 1tx4, 1grn and 1wql these amino acids are on different chains. For illustration, 1agr is colored dark-gray and 2 chains of 1grn are colored light-blue and red. The distance between the C_α atoms of the conserved Arg is about 10Å, therefore a structural representation with C_α atoms is not appropriate. Using the amino acid representation described in text, **MultiProt** detected correctly the multiple alignment. The functional Gln and Arg were among the 13 structurally aligned residues.

PTB-domain

The PTB (phosphotyrosine-binding domains) are particulary interesting. Recent data indicate that these modular domains form part of a superfamily of recognition domains. They can interact with different targets on different parts of their surfaces. While they were originally believed to bind only phosphorylated targets, they are now known to bind also non-phosphorylated targets, some of which lack Tyr altogether. This diverse binding makes them a good case study. The PTB example demonstrates MultiProt's ability to handle protein binding sites. MultiProt is able to compare all binding sites of available PTB domains, to obtain a consensus, common binding site core.

We selected 6 proteins (X11 PTB domain 1x11, Irs-1 PTB domain complexed with a Il-4 receptor phosphopeptide 1irs, Shc PTB domain complexed with a trka receptor phosphopeptide shc, Numb PTB domain complexed with a nak peptide 1ddm, Dnumb PTB domain complexed with a phosphotyrosine peptide 2nmb and Evh1 domain in complex with acta peptide 1evh). These molecules were taken from a study by Forman-Kay & Pawson[42]. The first five molecules are from the PTB family, while 1evh is from the EVH family. The detected common structural core contains 66 amino acids. For details see Figure 3.12.

Interface Alignment

[v]

Protein-protein interactions occur on the surface of a protein and are governed by physical-chemical forces. Thus, inspection of protein-protein interfaces should provide clues to the associations between different protein chains. Toward the ultimate goal of understanding how proteins interact, a good starting point is inspection of the interfaces structurally. A protein-protein interface consists of residues that interact with each other across the binding interface. At least two chains are involved. To be able to investigate the structural features of interfaces, we use our algorithm which does not take into account the linear sequence information. Because interfaces are

[v]These results are contributed by Ozlem Keskin okeskin@ku.edu.tr

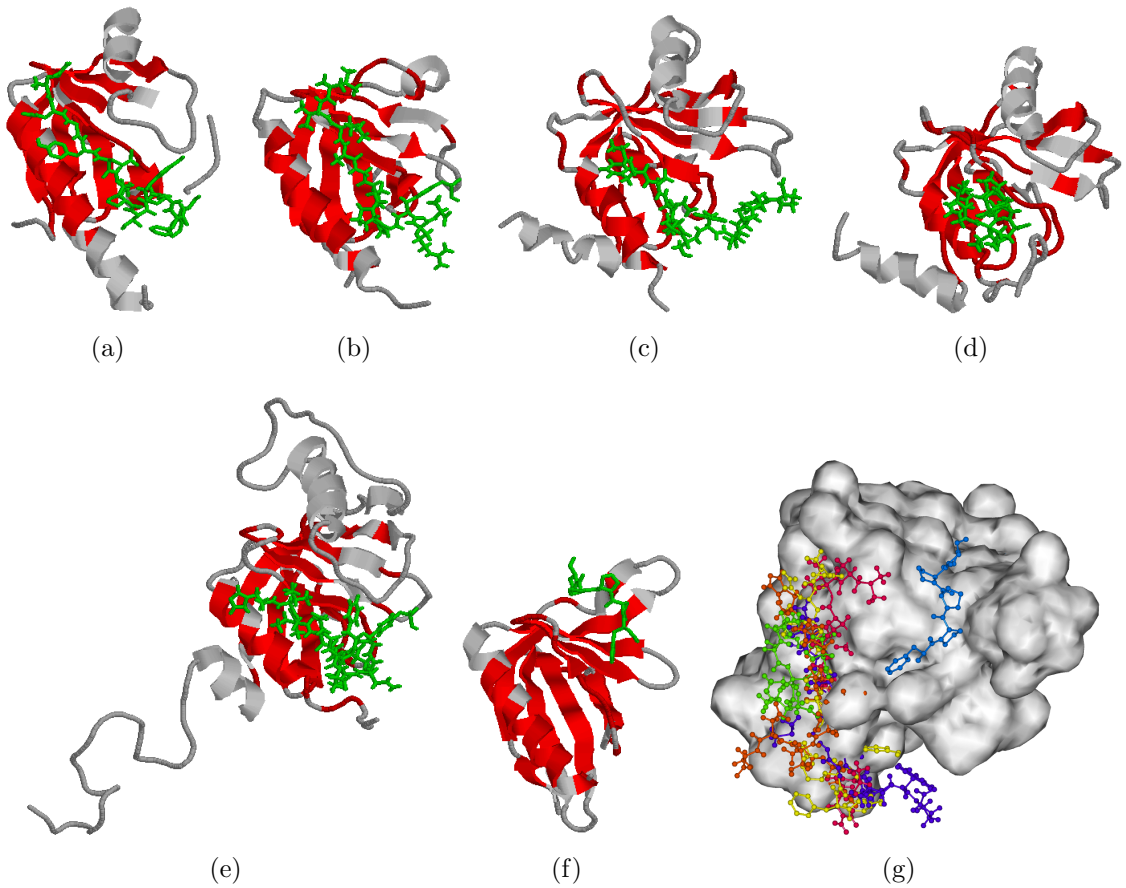


Figure 3.12: (a-f) Proteins (1x11,1irs,1shc,1ddm,2nmb,1evh) are displayed in orientation according to the multiple structural alignment by MultiProt. The common structural core is in red, whereas in gray are the structurally mis-aligned parts. The first five molecules are from the PTB family, while 1evh is from EVH. (g) All peptides/ligands are superimposed according to the same multiple alignment as (a-f). While the first five ligands bind to almost the same surface region, the ligand on the right is taken from 1evh. Comparing structures (a-e) with (f) one can see that the binding site of 1evh is different from others.

composed of at least two chains, and most of the time only discontinuous segments from each chain are involved in the binding, a structural alignment method which is independent of the order of the residues on the chains is essential. There is another very important point one should be careful in aligning interfaces: chain identities should be taken into consideration. Let us say we have two interfaces: Interface1 and Interface2. The first of these two interfaces is composed of chains ChainA and ChainB, and the latter ChainC and ChainD. If a method aligns the interfaces, all the segments from Chain A should be aligned with segments from ChainC (or ChainD) but not a combination of segments from both ChainC and D. **MultiProt** has the capability of aligning different interfaces simultaneously taking this criterion into account. With **MultiProt**, one can also detect sequentially conserved residues at a specific position on the alignments. This is an invaluable tool for finding the conserved residues hot spots at the interfaces.

Here, we apply our algorithm to clusters of protein-protein interfaces that are pre-filtered so that they are known to share common motifs at their interfaces. Only the interfaces are compared. The remainder of the chains are not considered. Figure 3.13 (a-d) shows alignment of four different families of interfaces. Table 3.7 gives the results of the interface alignment. These results have been obtained from O. Keskin et al. (Full details will be given elsewhere; Keskin, Tsai, Wolfson and Nussinov, unpublished).

Figure 3.13 (a) is an example of the Glutathione S-transferases. The green part in the figure shows the results of the structural alignments of 10 interfaces. The yellow and the gray parts are the ribbon diagrams of the two complete chains of Glutathione S-transferases 1c72 (Chains A and B). There are four α-helices and two β-strands in the interface that are being alined by **MultiProt**.

In Figure 3.13 (b), the interfaces of 6 serpins are displayed. The red and the cyan residues are the conserved residues of these 6 families. Note that only the conformations of the side chains of antichymotrypsin (1as4) are shown in the figure. Cyan depicts the conserved residues in Chain A and in red are the conserved residues in Chain B.

In Figure 3.13 (c), the gray region displays the aligned five interfaces of *DNA*

Family Name	PDB codes	No. of Mols	S_{al}	S_{bio}	SCOP classification (Superfamily)
Transferases	10gsAB, 1axdAB, 1c72AB, 1f2eAB, 1gwcBC, 1jlvAB, 1pd212, 1b48AB, 1gnwAB, 1ljrAB	10	67	17	Glutathione S-transferases, C-terminal domain
Ribonuclease, DNA binding proteins	1a2pBC, 1axcAC, 1b77AB, 1b77AC 1axcAE	5	18	0	DNA clamp
Serine proteases	1antLI, 1as4AB, 1d5sAB, 1hleAB, 1paiAB, 1c8oAB, 1jjoCE	7	67	20	Serpins
Transferase S	1c2yAE, 1c41AB, 1ejbAB, 1hqkAB, 1rvv12, 1rvvZ1, 1c41AE, 1hqkAE	8	41	3	Lumazine synthase

S_{al} is the number of aligned amino acids, S_{bio} is the size of the bio-core.

Table 3.7: **Protein interface families.** For the multiple alignment only the interface part was used, not the complete structure.

binding proteins, again the green and red chains are the ribbon diagrams of A and B chains of the DNA clamp 1b77.

Figure 3.13 (d) displays the alignment of 8 interfaces for the Lumazine synthase family. The magenta and the cyan regions show the ribbon diagram of the chains A and E of the Lumazine Synthase 1c2y. Gray depicts the interface. All the aligned interfaces have at most 3.5 A rmsd amongst them. These examples are important, indicating that MultiProt can successfully align interfaces of different sizes and different numbers simultaneously and very efficiently. The interface examples presented here would not be aligned structurally with the other available structural alignment programs since the remainder of the chains would determine the alignments.

3.5 Conclusions

Here we have presented a powerful tool for a simultaneous alignment of multiple protein structures. The advantage of MultiProt over previous methods is a combination of (i) simultaneous structure superposition (no side effects of pairwise alignment methods), (ii) solutions are detected for any number of molecules (separation between more similar structures and outliers), (iii) proteins can consist of several chains, and (iv) the final alignments can optionally preserve the sequence order or be sequence order independent. That is, MultiProt has the ability to detect non-topological similarities. Consequently, if there are at least three consecutive residues in the match, (v) MultiProt can be applied to multiply align binding sites. The case studies presented here demonstrate these abilities. Despite the complex task, MultiProt is extremely efficient and is suitable for simultaneous comparison of up to tens of proteins.

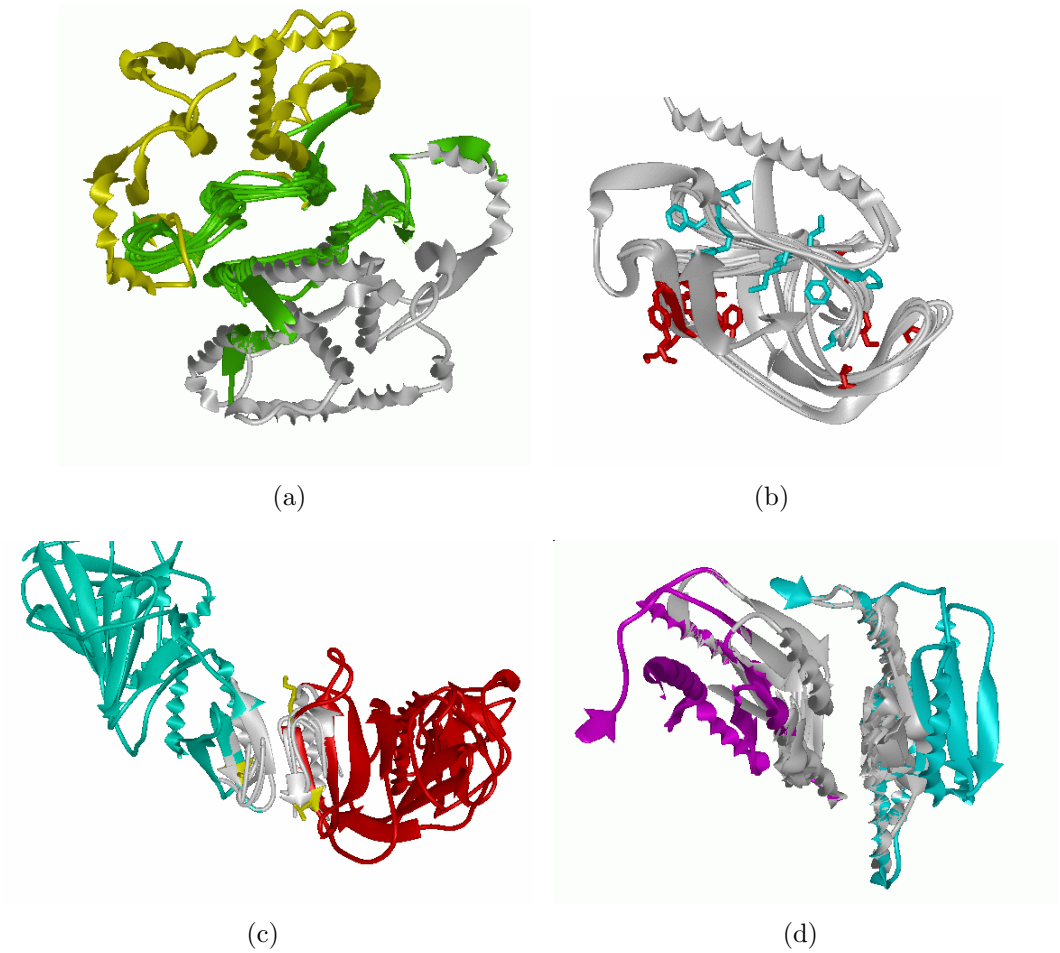


Figure 3.13: (a-d) The structural alignments of four different protein-protein interface families. In each figure, in addition to the aligned interfaces, the ribbon diagrams of the two chains which the interfaces belong to are displayed. These two chains are for the representatives of the family. See discussion in the text.

3.6 Appendix

3.6.1 Pairwise Correspondence

Given two point sets in 3-D, A and B, and a threshold ε, find two equal size subsets A' and B' ($A' \subset A$, $B' \subset B$) satisfying the following criterion: there exist a one-to-one correspondence, f, between A' and B', such that $\forall a \in A'$, $||a - f(a)|| \leq \varepsilon$. The task is to maximize the size of A' and B'. This problem can be solved by detecting maximal matching in a bipartite graph.

Let us construct the bipartite graph $G = (A \cup B, E)$. A graph edge $e = (u, v)$, $u \in A$ and $v \in B$, is created if and only if the distance between these points is less than ε. Now, the task is to select the maximal number of edges that satisfy the "matching" criterion - "A matching in a graph G is a subset of the edges of G such that no two share an end-point". This set of edges would give us a one-to-one correspondence of maximal size. This is exactly the problem of finding "maximal cardinality matching in the bipartite graph" [84], which can be solved with time complexity $O(\sqrt{n} \cdot m)$, where $n = |A| + |B|$ and $m = |E|$. In practice (for the case of protein molecules), a straightforward greedy method has been proven to work adequately well. It works as follows. An edge $e = (u, v)$ is added to the matching if and only if nodes u and v are not already matched. In other words the edge e should not violate the already created matching in the graph. The complexity of this greedy algorithm is only linear in the number of the graph edges $O(m)$. We show in the next paragraph that $m \in O(n)$.

A naive approach of building the "correspondence" graph $G = (A \cup B, E)$ takes $\Theta(n^2)$ steps, since every pair of nodes, one node from each part, is considered in order to decide, whether there is an edge between the nodes. To resolve this we have to find for each query point $u \in A$ all the points $v \in B$ within a ball of radius ε from u. Taking into consideration the geometrical constraints on atomic centers of protein molecules, this task can be accomplished in almost linear time $O(n)$.

We place one molecule, say A, on a three dimensional geometric grid, with grid size $max(\varepsilon, 3)$. Then, for each center of atom v of molecule B, which is just a point in 3-D space, we access the geometric grid and extract all atoms of A within radius ε from atom v. If ε is small enough, and it should be small since otherwise there

is no meaning for alignment of two atoms, then we extract from the grid a constant number of atoms. If $\varepsilon = 3\mathring{A}$, then in the ball of radius ε there are at most 2 or 3 C_α atoms. Therefore, the degree of each vertex in the graph is at most 3 (using the notation defined above, $dens(\varepsilon)$ is bounded by a small constant). It follows that the number of edges is linear in the number of nodes. Thus, the experimental complexity of building the graph G is (|Putting A onto a grid| + $|B|\cdot$ |Number of atoms within radius ε|) $\in O(n)$.

To summarize, the complexity of solving the correspondence task using a greedy approach is linear in the number of atoms ($O(n)$), while using an exact algorithm of *"Maximal Cardinality Matching in the Bipartite Graph"* is $O(n^{3/2})$ (since $m \in O(n)$).

3.6.2 Multiple Correspondence

Our aim is to give a practical solution to the *3D-ε-K-partite-pivot* matching. As we have discussed above in Section 2.4.2, this optimization problem is NP-Hard, therefore, in practice we adopted two heuristic strategies.

The first method aims to combine optimal pairwise solutions. For each $i = 2...K$ solve *Pairwise Correspondence* for S_1 and S_i, as described above. The resulted correspondence-lists might contain different points from the set S_1. We are interested only in those that appear in all correspondence-lists. Thus we create a list of K-tuples, where each K-tuple contains - (1) a point p from S_1, which appeared in all correspondence-lists, (2) $K-1$ points which are corresponding to p from *Pairwise Correspondences*. Assuming that the maximal point density, $dens(\varepsilon)$, is bounded by a small constant, the time complexity is $O(K \cdot n^{1.5})$.

The second method is a greedy approximation. Iteratively select an ε-K-tuple-pivot until no selection is possible. The method guarantees to find a matching of size at least $OPT/min(dens(2 \cdot \varepsilon), K)$. The running time is linear in the total number of points, $O(K \cdot n)$.

In practice we tried to apply both methods, but in most cases the difference between the results was minor. While two options are available in MultiProt, the second method is used by default.

Chapter 4

Structure-Derived Multiple Sequence Alignment

4.1 Introduction

High protein sequence identity (> 30%) usually implies structural similarity and similar protein function. However, the opposite does not always hold. Examples of Isomerases, Cytokines, Myoglobins, Immunoglobulins, Transferases and other protein families show that proteins may share a similar 3D fold but nevertheless have less than 25% sequence identity. Yet, despite low sequence similarity, some specific amino acids are evolutionarily conserved for function. This presents a major challenge: how to accurately explore multiple protein sequence and structure information?

Methods for multiple sequence alignment have become almost everyday tools for computational biologists, for example Psi-BLAST[6] or ClustalW[58]. Structural alignment of pairs of proteins is also acquiring a similar status. On-line resources are available for high quality pairwise alignments: CE[115], DALI[29], Geometric Hashing (GH)[11], VAST[82] and others[38]. Obviously, much more information can be derived from multiple structure alignment. Recognition of a structural core common to a set of protein structures has many applications such as in the studies of protein evolution and classification, analysis of similar functional binding sites and homology modeling[3, 47]. A number of methods have been developed to perform

multiple alignment of protein structures[107, 30, 89].

Given a protein molecule, one may ask: what is the evolutionary conservation of each of its amino acids? A natural assumption is that highly conserved positions may indicate conservation of protein function. Several methods have been developed to compute amino acid conservation (or mutation rate) from phylogenetic information derived from protein sequences [78, 46]. Can we obtain additional information from protein structures? An accurate multiple alignment is one of the most critical steps toward the answer to this question. Most of the available multiple structure alignment methods aim to maximize the number of spatially aligned C_α atoms. The result is the superimposed protein structures. In addition these methods may compute a set of simultaneously spatially aligned residues. These are analogous to columns without gaps in sequence alignment. Consider the following scenario, where we are given three proteins $a_1a_2a_3$, $b_1b_2b_3$, $c_1c_2c_3$ and a multiple structure alignment between them. Assume that the following groups of amino acids have been found to be close in 3D space: (a_1, b_1), (a_2, b_2, c_1), (a_3, b_3, c_2) and (a_3, b_3, c_3). There are several combinations of spatially consisted structural alignments between these three proteins. For example the following three combinations[i]:

$$\text{I.}\ \begin{array}{cccc} a_1 & a_2 & a_3 & \text{-} \\ b_1 & b_2 & b_3 & \text{-} \\ \text{-} & c_1 & c_2 & c_3 \end{array} \qquad \text{II.}\ \begin{array}{cccc} a_1 & a_2 & \text{-} & a_3 \\ b_1 & b_2 & b_3 & \text{-} \\ \text{-} & c_1 & c_2 & c_3 \end{array} \qquad \text{III.}\ \begin{array}{cccc} a_1 & a_2 & \text{-} & a_3 \\ b_1 & b_2 & \text{-} & b_3 \\ \text{-} & c_1 & c_2 & c_3 \end{array}$$

Which alignment is better? All three satisfy the geometrical constraints. However, a structural superposition does not uniquely define a sequence alignment. Obviously, in this case we would prefer an alignment that places more similar amino acid types in the same column. Therefore, we face an optimization problem which is similar to the multiple sequence alignment problem but has additional geometrical constraints. To the best of our knowledge, only two methods, SSAPM [124] and CE-MC[50] (CE-MC is readily accessible on the Internet), have been devised to compute a global multiple structural alignment (i.e. to align all protein amino acids, like global sequence alignment, while minimizing the number of gaps). However, both methods do not

[i]In this work we will assume that all sequence and structural alignments are according to protein sequence order, i.e. alignments preserve the protein topological order.

perform optimization according to amino acid types, i.e. they consider only 3D information. The CE-MC method requires that in a multiple alignment column only some ratio of amino acids have to be close in 3D space, others may be distant. This may result in discrepancies in the alignment as demonstrated in Figure 4.4 of the Experimental Results section. Other available multiple structure alignment methods [135, 108, 30] only report columns without gaps, e.g. the second and the third column from combination (I). Obviously, non-gap regions provide only a limited information and for general applications a complete multiple alignment is required. Recently, a new method 3DCoffee[97] incorporated pairwise structural alignment information into the construction of multiple sequence alignments. However, only structurally aligned residue pairs (according to some *pairwise* structural alignment method) are used to increase the weight for that particular pairs during the construction of a sequence alignment. That is, the approach is not able to optimally solve the situation described above. COMPARER[103] represents a protein structure as a sequence of amino acid features (other element types such as secondary structure elements, motifs etc. can also be encoded). The features include amino acid biochemical properties as well as structural properties such as local main-chain dihedral angles, side-chain accessibility, side-chain orientation etc. In addition, relations between a pair of amino acids from the same protein structure are also encoded. These include hydrogen bonds and hydrophobic interactions. The method uses a combination of simulated annealing and dynamic programming to solve the alignment problem between two proteins. The multiple alignment is solved iteratively according to a dendrogram tree. While COMPARER can produce a global multiple sequence alignment it does not aim to find a good global structural superposition. Its encoded structural properties are of local nature including pairwise relations. Similarity of related pairs does not necessary imply congruence of the global 3D shapes.

For a better analysis of protein families there is a need for a multiple alignment that optimizes both sequence and structure information. Since protein structure is generally more conserved than sequence, we propose to perform an optimization of the multiple alignment, first, according to structure and then according to amino acid types with combination of 3D information. Namely, we propose the following scheme:

Given a set of protein structures, first, perform a multiple structural alignment (in this work we used the MultiProt[107, 108] method). Second, based on the multiple structural alignment, perform a multiple sequence alignment optimizing our newly defined scoring function. The scoring function is a likelihood of amino acid a to substitute b assuming that a (b) is located in secondary structure X (Y) and the 3D distance, according to a multiple structural alignment, between C_α atoms of a and b is d. To solve the multiple alignment we apply an iterative profile-profile alignment procedure. Our method is implemented in a program named STACCATO (STructural-sequence Alignment, Correspondence and Conservation Analysis TOol). To summarize, STACCATO solves the following tasks: (1) Given multiple structures compute the multiple sequence alignment optimizing a sequence-structure scoring function (defined below). The resulted alignment can be used as a profile for alignment of a set of protein sequences. (2) Compute the sequence-structure conservation for a column of the multiple alignment.

In the Results section we compare STACCATO with other available methods. Our tests include comparison with the HOMSTRAD[89] benchmark of multiple structure based sequence alignments. We argue that our approach produces more accurate alignments. We present some applications of STACCATO, which include analysis of loop motion in Tyrosine Kinase and improving the accuracy of protein-protein docking methods.

4.2 Methods

Our goal is to devise a scoring function that reflects the likelihood of matching two amino acids a and b given a structural alignment. We will call it a *3D substitution*. The likelihood ratio (LR) of such an event can be defined as:

$$LR(a,b) = \frac{P(a,b)}{P'(a,b)} \frac{P_{3D}(d(a,b))}{P'_{3D}(d(a,b))},$$

where $P(a,b)/P'(a,b)$ is a commonly used likelihood ratio of amino acid substitutions, $P(a,b)$ is the probability for a to substitute b and $P'(a,b)$ is the randomly expected probability (our default values are taken from the Blosum62 matrix[57]), $d(a,b)$ is the 3D distance between the C_α atoms in the structural alignment, $P_{3D}(d(a,b))$ is the observed probability of distance $d(a,b)$ in a set of structural alignments of closely related proteins and $P'_{3D}(d(a,b))$ is the randomly expected probability of 3D distances. We computed the P_{3D} and P'_{3D} values in the following way. We selected a representative set of 5674 protein structures from the SCOP database[92] which have less than 40% of pairwise sequence identity (this data set is provided by ASTRAL [20]). This resulted in 1033 protein domains (according to the SCOP classification) each containing at least 2 structures. Therefore, in order to compute the distribution of P_{3D} values (observed probabilities of related structures) we performed structural alignments for each protein pair from the same SCOP protein domain. For structural alignment we applied the MultiProt method[107, 108]. Each pairwise structural alignment defines a set of observed distances between aligned amino acids. However, here we face exactly the same problem of correct alignment definition, since the observed distances are influenced by the definition of the alignment. To overcome this problem, first, we required the structural alignment to be sequential (i.e. we are not interested in non-topological alignments) and global. Second, to overcome the ambiguity of alignment we selected a minimal 3D distance from a window of length 5 around a structurally matched amino acid. This set of distances defined the distribution of P_{3D}. The number of all pairwise alignments in this data set is 16,919 and the number of distances used to compute probabilities is 2,858,443. To compute randomly expected distances, P'_{3D}, we used the same data set and the same technique, but performed alignments between structures from different protein domains.

Practically, we computed distance distributions for the interval [0,10] with unit size bins. Since the actual distance range is larger, we performed the following scaling:

$$D(d) = \begin{cases} 9 & if \quad d > 22.62\mathring{A}, \\ \dfrac{1}{f + \dfrac{(1-f)}{d}} & otherwise, \end{cases}$$

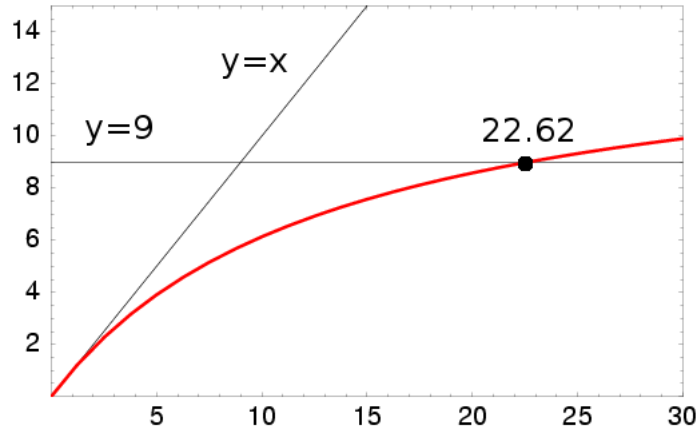


Figure 4.1: Function used (colored red) to map 3D distances (x-axis) of aligned amino acids. Distances larger than 22.62Å are mapped to 9.

The value of f is operationally set to 0.07. This function was selected to reflect the relevant distance range for the alignment. It behaves almost linearly for small values less than 5Å (this is arguably the most sensitive range), and gradually decreases for large values (basically, alignment at 25Å distance is not much different from alignment at 40Å). See the function graph in Figure 4.1. The computed distance distribution values, $2log_2(P_{3D}(d)/P'_{3D}(d))$, are presented in Table 4.1 (a).

Secondary Structure Information. Knowledge of secondary structure type can improve the correctness of alignment[125, 123]. A score of aligning two amino acids from the same kind of secondary structure should be higher than a score of amino acids coming from different secondary structure types even when both pairs are placed at the same distance in a structural alignment. Therefore, to stress the significance of secondary structure information we extend the definition of $LR(a, b)$ to:

$$LR\text{-}SSE(a,b) = \frac{P(a,b)}{P'(a,b)} \frac{P_{3D}(d(a,b)|SSE(a),SSE(b))}{P'_{3D}(d(a,b)|SSE(a),SSE(b))},$$

where $SSE(a)$ is the secondary structure type of a. To compute conditional probabilities P_{3D} and P'_{3D} we repeated the same steps as described above and the resulted values, $2log_2(P_{3D}(d|s_1, s_2)/P'_{3D}(d|s_1, s_2))$, are presented in Table 4.1 (b). In

(a)

Distance	[0,1)	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,8)	[8,9)	[9,∞)
Score	6.54	3.98	2.64	1.48	-0.85	-2.18	-3.07	-3.50	-3.83	-3.85

(b)

Distance	[0,1)	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,8)	[8,9)	[9,∞)
H H	5.23	4.23	3.82	2.62	-0.19	-1.93	-3.06	-3.37	-3.28	-3.13
H S	-5.01	-5.45	-4.90	-5.23	-5.79	-6.26	-6.40	-6.17	-6.09	-5.94
H U	2.36	1.07	0.92	0.15	-1.17	-2.26	-3.15	-3.58	-3.98	-4.23
S S	9.09	5.80	3.99	3.11	-2.07	-3.67	-4.33	-4.61	-4.97	-4.11
S U	4.73	2.37	1.63	0.81	-2.28	-3.54	-4.26	-4.56	-4.56	-4.72
U U	8.66	5.26	3.97	3.04	1.19	-0.30	-1.58	-2.42	-3.21	-3.63

Table 4.1: **Distance Scores.** H - helix, S - strand, U - undefined

all the experimental results we used only SSE conditional probabilities.

Finally, the score for the *3D substitution* is defined as log-odds:

$$Score_{SM}(a,b) = 2log_2(\frac{P(a,b)}{P'(a,b)}),$$
$$Score_{3D}(a,b) = 2log_2(\frac{P_{3D}(d(a,b)|SSE(a),SSE(b))}{P'_{3D}(d(a,b)|SSE(a),SSE(b))}),$$

$$Score(a,b) = 2log_2(\frac{P_{3D}(d(a,b)|SSE(a),SSE(b))}{P'_{3D}(d(a,b)|SSE(a),SSE(b))}) = Score_{SM}(a,b) + Score_{3D}(a,b)$$

4.2.1 The Optimization Method

The input to our optimization method is a set of protein structures superimposed by some multiple structure alignment method. In our work we used MultiProt[107, 108] that performs a simultaneous structure superposition and does not have the drawbacks of the methods which are based on pairwise structural alignments. The proposed method, STACCATO, is based on progressive alignment, or profile-profile alignment. This strategy has been applied in many multiple alignment methods. First, each protein is initialized as a singleton profile. Then, the most similar pair of profiles is detected and merged. This progressive alignment is repeated until only one profile, which includes all the sequences, is left. Most sequence alignment methods,

like ClustalW[58], apply progressive alignment steps according to a dendrogram tree computed at the beginning of the procedure. A dendrogram tree defines similarity between sequences and consequently it defines similarity between created profiles. Here, we recompute similarity between sequences/profiles each time a new profile is created. Such a method is slower, but is more robust. Each time, a new profile is created it defines more precisely the protein family characteristics, therefore an evolutionary distance between the new profile and other sequences/profiles can be calculated with higher precision.

Iterative Profile – Profile Alignment :

$Input : S_1, ..., S_K$

$P = \{Profile(S_i)\}$

$do\ until\ |P| == 1$

$select\ P_i, P_j \in P,\ s.t.\ max_{i \neq j = 1...|P|} Score(Profile(P_i \cup P_j))$

$P = P \cup \{Profile(P_i \cup P_j)\} \quad (join\ two\ profiles)$

$P = P \setminus \{P_i, P_j\}$

end

The step of joining two profiles, $Profile(P_i \cup P_j)$, and computing its score, $Score(Profile(P_i \cup P_j))$, is done by a standard dynamic programing[120]. We need to define a score of joining column c_i, from the first profile, with column c_j, from the second profile. We join c_i with c_j by applying the sum-of-pairs scheme:

$$Score(c_i, c_j) = \Sigma_{x,y \in c_i \cup c_j, x \neq y}\ Score_{SM}(x, y) + Score_{3D}(x, y)$$

$Score_{SM}(x, y)$ is taken from an amino acid substitution matrix. Our default matrix is Blosum62. $Score_{3D}$ is as defined above. The columns may include gaps, $Score_{3D}(x, -) = Score_{3D}(-, y) = Score_{3D}(-, -) = 0$. For Blosum62 we define

$Score_{SM}(x, -) = Score_{SM}(-, x) = -4$ and $Score_{SM}(-, -) = 0$[ii].

Assume that the number of sequences in the first and second profile is n and m respectively. Let us define a column c_-^n which contains exactly n gaps. Note that aligning column c_i with a gap means aligning it with a column of m gaps. Analogously, aligning column c_j with a gap means aligning it with n gaps. Thus, the score of introducing a gap into an alignment equals to:

$$Score(c_i, -) = Score(c_i, c_-^m) \quad \text{and} \quad Score(-, c_j) = Score(c_-^n, c_j),$$

i.e. the gap penalty is also calculated by the sum-of-pairs scheme. The default gap opening penalty is set to -7.

In case we are given a set of protein structures, S^1, and a set of sequences, S^2 (for which structural information is not available), we compute an alignment of the two sets in the following way. First, we compute a multiple structure based alignment of S^1 as described above. Then, the same iterative scheme is applied on the profile of S^1 and singleton profiles initialized from each sequence of S^2. In the second iterative stage the structural information ($Score_{3D}$) is not used. Thus, the final alignment is based on the first structure set S^1.

Distance Constrained Multiple Alignment. Our method allows to produce structurally constrained alignments. We can require that all pair-wise distances between the C_α atoms of amino acids from the same column are less than ε (user defined parameters). To adopt this requirement we only need to update the scoring function as follows:

[ii]There is no yet a commonly agreed standard to measure a gap against a gap for the sum of pairs scoring method. For example, consider three columns c_1, c_2 and c_3. Half of column c_1 are characters 'a', and half are 'b'. Half of column c_2 are 'a' and half are gaps. One third of column c_3 are 'a', one third are 'b' and one third are 'c'. Assume that $Score_{SM}(a, a) = Score_{SM}(b, b) = Score_{SM}(c, c) = k$ and $Score_{SM}(a, b) = Score_{SM}(a, c) = Score_{SM}(b, c) = Score_{SM}(a, -) = -k$. We want the following property: $Score(c_3) < Score(c_2) < Score(c_1)$. If selecting $Score_{SM}(-, -) = -k$ then for ten (or more) proteins in the multiple alignment $Score(c_3) > Score(c_2)$. This would be an undesired property. This is one of the reasons why we chose $Score_{SM}(-, -) = 0$.

$$Score^*(c_i, c_j) = \begin{cases} Score(c_i, c_j) & if \;\; \forall x, y \in c_i \cup c_j \qquad d(x, y) \leq \varepsilon \\ -\infty & otherwise, \end{cases}$$

The constrained multiple alignment allows identification and clustering of structurally similar regions. We show an application in the Experimental Results section.

4.2.2 Structure-Sequence Conservation Score

For each column c of multiple alignment we compute a conservation score, which combines the sequence information, i.e. amino acid types and the structural information, i.e. amino acid superposition in 3D. For reasons of practical convenience to display an alignment along with a conservation score for each column, we scaled the conservation score into the range $[0, 9]$. In the program output the score values are rounded to the nearest integer. Such representation allows a quick visual estimation of alignment regions whether they are conserved (low values) or not. See examples in the Experimental Results section.

Computation of sequence conservation (amino acid type conservation), $Cons_{seq}(c)$, has been extensively addressed[126]. Here, we adopt a weighted sum-of-pairs scheme:

$$Cons_{seq}(c) = \Sigma_i^N \Sigma_{j>i}^N \; w_i \; w_j \; Score^*_{SM}(c(i), c(j)) \; / \; W,$$

where N is the number of sequences in the alignment and $c(i)$ is the amino acid in column c of sequence i. A modified scoring matrix $Score^*_{SM}$ is defined as:

$$Score^*_{SM}(a, b) = \begin{cases} Score_{SM}(a, b) & if \;\; a \neq b, \\ \Sigma_{i=1}^{20} \; Score_{SM}(i, i)/20 & if \;\; a = b, \end{cases}$$

i.e. only the matrix diagonal is changed (for discussion see [126]). To overcome an over-weight of similar sequences their contribution is balanced by an appropriate weighting scheme. The weight of sequence i and the normalized weight W are defined as:

$$w_i = \Sigma_{j \neq i}^N \; d(i, j)/(N - 1), \qquad\qquad W = \Sigma_i^N \Sigma_{j>i}^N \; w_i w_j \;,$$

where $d(i,j)$ is the distance between sequence S_i and S_j, which is defined as:

$$d(i,j) = 1 - PercentIdentity(S_i, S_j)/100,$$

i.e. if the sequences are identical then $d(i,j) = 0$, while as sequence identity diminishes $d(i,j)$ approaches one. Therefore, using the Blosum62 matrix, the sequence conservation score $Cons_{seq}$ ranges from -4 to 5.75. In order to obtain a $[0,9]$ scale, with low values indicating a higher conservation, a linear transformation is applied:

$$Cons_{seq} = 9 * (1 - (Cons_{seq} + 4)/9.75).$$

The structural conservation $Cons_{str}$ is defined as $D(rmsd(c))$, where $rmsd(c)$ is the RMSD of the C_α atoms from column c (gaps are not counted) and D is the function defined above in the beginning of the section. The $Cons_{str}(c)$ range is $[0,9]$ (if all C_α atoms are in the same 3D position, then it is equal to 0).

The final sequence-structure conservation score is defined as a combination of structural and sequence scores:

$$Cons(c) \;=\; w * Cons_{seq}(c) + (1-w) * Cons_{str}, \qquad Cons(c) \in [0,9]$$

where w is set to 0.5. Below, in the Experimental Results section, residues with a score $Cons(c)$ less than 5 have been defined as 'conserved'.

4.3 Experimental Results

The output of the STACCATO program is a multiple alignment in ClustalX or PIR formats. In addition, for each column the following scores are optionally displayed: $Cons_{str}$ - structural score, $Cons_{seq}(c)$ - sequence score, $Cons(c)$- combined score and $|Cons_{str} - Cons_{seq}|$ - the absolute difference between the structural and sequence scores. In addition, for each input protein file the amino acid temperature factor

field can be set to the corresponding conservation score. This allows convenient 3D visualization in color of the amino acid conservation scores (see Figure 4.5).

4.3.1 Comparison against HOMSTRAD data base of benchmark multiple alignments.

HOMSTRAD (the first version) is a database of multiple alignments of 1032 homologous protein families with available 3D structures[89]. Here we want to compare the quality of multiple alignments produced by STACCATO against the HOMSTRAD database. To perform the most objective comparison we need to know either (1) the ultimate scoring function (yet generally unknown) or (2) the correct alignment (which will require to know, for instance, the exact phylogenetic tree of the protein family). Therefore, due to the lack of the ultimate measure, we selected two kinds of scores. First, *Seq* score, is a normalized sum-of-pairs score according to the BLOSUM62 matrix, i.e. a commonly used sequence similarity measure. Second, we measure how well the structures are fit according to a multiple sequence alignment. We aim for an intuitively simple and informative definition. We define the *Str* score as a normalized number of columns in a multiple alignment which contain at least half non-gap positions and their 3D root mean square deviation is less than 3Å. We conjecture that if an alignment improves *both* kind of scores, than it is closer to the optimum. However, it is not clear which alignment is better if one score is increased while the other is decreased. Notice, that *Str* scores is not used explicitly in the scoring function of the STACCATO optimization method, while the *Seq* score is only part of the optimization score.

Table 4.2 shows average improvement of STACCATO over HOMSTRAD applying *Seq* and *Str* scores. Different gap opening penalties were used to show robustness of the results. For example, for a scoring function based on conditional SSE distributions, and gap opening penalty -7 STACCATO gave the following results. In total, there are 1032 multiple alignments. The number of alignments where STACCATO improved either *Seq* score or *Str* score is 1000. The number of alignments where STACCATO improved at least one score while the other score was at least as good as HOMSTRAD

```
(a)  1gnw   136> VLDV-YEARLKEFKYLAGETFTLTDLHHIPAIQYLLGTP
     1gwc   129> VLEGALRECSKGGGFFGGDGVGLVDVALGGVLSWMKVTE
     1g7o   138> DLRALDKLIVKPNA-VNGE-LSEDDIQLFPLLRNLTLVA
                  *         *       *       *
                 2221222333555422223222333222222333322333

(b)  1gnw     EAVVAEEEAKLA-KVLDVYEARLKEFKYLAGETFTLT--DLHH
     1gwc     LRECSKGGGFFGGDGVGLVDVALGGVLSWMKVTEALS--GDKI
     1g7o     FADLLAHSDGLIKNISDDLRALDKLIVKPNAVNGELSEDDIQL
                                                *
```

Figure 4.2: (a) A multiple sequence alignment as produced by STACCATO. $Cons_{str}$ score is displayed. Most of the region is aligned within 2-3Å. (b) A multiple alignment produced by ClustalW[58] on the same set of sequences. Notice the discrepancies in the alignment (some are marked with boxes).

score is 587. The number of alignments where both scores were the same is 4. The number of alignments where STACCATO degraded both *Seq* score and *Str* score is only 16.

According to Table 4.2 a preferable gap opening penalty is in the range of [-7,-10]. Outside this range either the number of gap openings is increased, or the *Seq* and *Str* scores are decreased.

Notice that the use of conditional SSE distributions decreases the *Str* score since well matched SSE's do not reward the *Str* score. Despite that, STACCATO still improves over the HOMSTRAD alignments. Therefore, STACCATO alignments are arguably more accurate.

4.3.2 Low Sequence Identity with High Structural Similarity.

Here, we analyze proteins from the Glutathione S-Transferase family (GST): 1gnwA:86-211 (Class phi GST from Arabidopsis thaliana), 1g7oA:76-215 (Glutaredoxin 2 from Escherichia coli) and 1gwcA:87-224 (Class tau GST from Triticum tauschii l.). Only the C-terminal domain is considered.

Despite the fact that all three proteins are from the GST family, they have less than 15% of pairwise sequence identity, which is extremely low. Therefore, it is not

surprising that sequence alignment methods may produce an inaccurate alignment. However, in this case, an accurate alignment can be obtained from the structural information, since all GST proteins share high structural similarity. Figure 4.2 shows a structure based multiple sequence alignment produced by STACCATO and discrepancies in the multiple sequence alignment produced by ClustalW[58]. Therefore, this simple example demonstrates that it is essential to use structural information (when protein structures are available) and our proposed scheme is adequate for this task.

4.3.3 Glutathione S-Transferase active site residues

Zhang et.al.[135] showed that the standard sequence alignment methods can not align correctly the GST active site residues. Their method, SAPS, using multiple structure alignment information does produce a correct alignment. Figure 4.3 demonstrates that STACCATO succeeds to achieve the correct multiple alignment as well. The SAPS method aligns only non-gap fragments, therefore the advantage of STACCATO is in its ability to produce a complete multiple alignment including gaps.

4.3.4 Loop Movement in Tyrosine Kinase.

Tyrosine kinase is a large family of evolutionarily conserved enzymes which are critical in cellular signaling pathways[62]. Here, we consider four protein structures, two in the active (1ir3A, 1cdkA) and two in the inactive state (1irk, 1iep). Proteins 1ir3A, 1irk are insulin receptors from human, 1cdkA is a cAMP-dependent protein kinase catalytic subunit from pig, and 1iep is Abelsone tyrosine kinase from mouse. In this case study we focus on the activation loop. A well conserved DFG motif, which is involved in Mg-ATP binding, is located at the beginning of the activation loop. During activation the loop must undergo a substantial conformational change. Some amino acids in the loop change their position by as much as 31Å (see Figure 4.4 (a)). A question which can be asked is what kind of analysis should be performed in order to detect and distinguish between the two states, and which residues participate in this rearrangement?

The multiple sequence alignment of these four proteins does not distinguish between the active and inactive forms. In Figure 4.4 (b), the DFG motif as well as the whole activation loop are multiply aligned. However, the structurally constrained multiple alignment does distinguish between the two states. In Figure 4.4 (c-d) it can be seen that the loop forms two separate clusters, of the active and inactive states. Only the aspartic residue from the DFG motif is spatially conserved (within 5Å distance threshold).

4.3.5 Applications to Improve Protein-Protein Docking.

Given the structures of two molecules, whether two proteins or a protein and a small molecule, the task of a docking method is to computationally predict their possible interaction. A result of a docking program is a 3D complex between the input molecules. There are two major stages in docking methods: a 3D search for potential complexes and a quantitative estimation of the quality of each sampled solution. Quantitative estimation is done by some scoring function, which usually considers geometrical and biochemical properties of a predicted complex. The bottleneck of the current docking methods is the quality of the scoring function. In most cases a near native ("correct") solution is detected during the 3D search. However, hundreds of false-positive solutions are ranked higher than near native ones. One of the possible ways to reduce the number of false-positives is to reduce the combinatorial space during the 3D search, i.e. to reduce the number of potential solutions. Knowledge of the protein binding site significantly reduces the number of false-positives, but usually such information is not available. Yet, evolutionary conservation of amino acids may indicate potential binding sites[iii]. Given a list of conserved amino acids this information can be used in the following way. During a 3D search, only those solutions are considered that place at least one conserved residue on the molecular surface into the protein-protein interface, i.e. at least one conserved residue should participate in binding.

To illustrate the above mentioned application, we consider the prediction of the complex between Methylamine Dehydrogenase (2bbkLH) and Amicyanin (1aan) from

[iii] Antibody proteins have an opposite property, their highly variable binding site allows adaptation for binding specific antigens. We consider antibody-antigen binding as an exception from the rule.

Paracoccus denitrificans. First, we run the PatchDock[33] algorithm without any information about conserved residues. The first near native solution (RMSD with correct solution less than 3Å) was ranked 159, i.e. there were 158 false-positives (the native complex of these proteins 2mta is used as a reference). Second, we used the ConSurf[46] method to compute the conserved residues of the Amicyanin protein (1aan). The method computes an evolutionary conservation (or alternatively a mutation rate) based on sequences alone. For the conservation list, we selected amino acids having a rank higher or equal to 6 (from 1-9 range). It contained 54 amino acids. With this list as input to PatchDock, a near native solution was ranked 133'rd. However, not all amino acids conserved in sequence are conserved in structure. Third, we applied the STACCATO method on the Plastocyanin/azurin-like family, to which Amicyanin (1aan) belongs. According to SCOP[92] this family contains 8 proteins. Each protein has several PDB representatives belonging to different species. We arbitrarily selected one representative for each protein (1aan, 1plc, 1paz, 2cbp, 1azcA, 1qhqA, 1e30A, 1jer). According to STACCATO, there are 24 conserved amino acids in 1aan. Conserved amino acids were defined as those for which the $Cons(c)$ score is less than 5. This list results in ranking a near native solution in the 62 place, which is a considerable improvement over both previous results. See Figure 4.5 for additional details.

4.4 Conclusions

Here we have presented an optimization method unifying sequence and structural information. When protein structures are available, a multiple sequence alignment consistent with a multiple structural alignment overcomes the problems inherent to sequence alignment methods. The results presented here show a high potential for the STACCATO method. On the benchmark of 1032 protein families STACCATO produces multiple alignments as good (and arguably slightly better) as multiple alignment from the HOMSTRAD data base.

Structures contain considerable information. Combining multiple sequence alignment with a multiple structure alignment should yield improved results particularly

in cases where the alignment is more difficult, due to different sizes, lower sequence identity, presence of variable loops, etc. The caveat is however, the existence of the respective structures in the Protein Data Bank. Yet, as the number of these increases very rapidly, a method such as the STACCATO should be increasingly applicable.

Gap Opening Penalty	SSE Seq %	SSE Str %	no SSE Seq %	no SSE Str %	SSE #Gaps	no SSE #Gaps
-3	+9.5 (+9.4)	+0.4 (+8.3)	+9.3 (+9.2)	+1.4(+9.2)	+24	+23
-7	+7.5 (+7.2)	+0.3 (+8.2)	+7.2(+7.1)	+1.2(+9.0)	-7	-7
-10	+6.4 (+6.1)	-0.04 (+8.0)	+6.3(+6.0)	+0.5(+8.6)	-16	-16
-13	+5.4(+5.1)	-0.4(+7.6)	+5.4(+5.0)	+0.1(+8.2)	-20	-21
-15	+4.8(+4.6)	-0.7(+7.3)	+4.8(+4.6)	-0.1(+7.9)	-23	-23
-30	+1.3(+1)	-3(+5.4)	+1.4(+1.1)	-3.1(+5.3)	-33	-33

Table 4.2: **Comparison against the HOMSTRAD data base.** Two experiments have been conducted. In the first one STACCATO has been applied on the multiple structure alignments as found in HOMSTRAD. In the second experiment, STACCATO has been applied on the multiple structure alignments computed by the MultiProt method[107, 108]. The numeric values represent the difference in scores of STACCATO and HOMSTRAD alignments measured in percents relative to the HOMSTRAD score (positive values mean an improvement over the HOMSTRAD alignments). The results of the second experiment are presented in the parentheses. SSE/no SSE - the scoring function used by STACCATO according to the distribution of Table 4.1 (b) / 4.1 (a). The last two columns represent a relative difference in the number of gap openings (negative values mean less gaps are opened in the STACCATO alignments).

```
1gnwA    2> -GIKVFGHPASIATRRVLIALHEKNLDFELVHVELKD-------GEHKKEPFLSRNPFGQVPAFEDG-DLKLFESRAITQYIAHRYENQ
1aOfA    1> --MKLFY-KPGACSLASHITLRESGKDFTLVSVDLMK----KRLENGDDYFAV--NPKGQVPALLLDDGTLLTEGVAIMQYLADSVPDR
2gstA    1> -PMILGYWNVRGLTHPIRLLLEYTDSSYEEKRYAMGDAPDYDRSQWLNEKFKLGLDFPN-LPYLIDG-SRKITQSNAIMRYLARKH---
2gsq     1> PKYTLHYFPLMGRAELCRFVLAAHGEEFTDRVVEMAD--------WPNLKATM---YSNAMPVLDID-GTKMSQSMCIARHLAREF---
                                *                                  *                      *    *
            453324354454434433241444343124443335444446677675555546665545313244344432322420322214445 56
```

Figure 4.3: Multiple alignment of Glutathione S-Transferase proteins as produced by STACCATO. The numeric row corresponds to the combined sequence-structure conservation score, $Cons(c)$, of the aligned columns. Correctly aligned active site residues are marked in blue boxes. These are structurally stable regions. The correct alignment of these regions was demonstrated by the SAPS method [135]. Red box region includes some additional active site residues which were not reported by SAPS. As it can be seen from the one-digit $Cons(c)$ score this region is more variable.

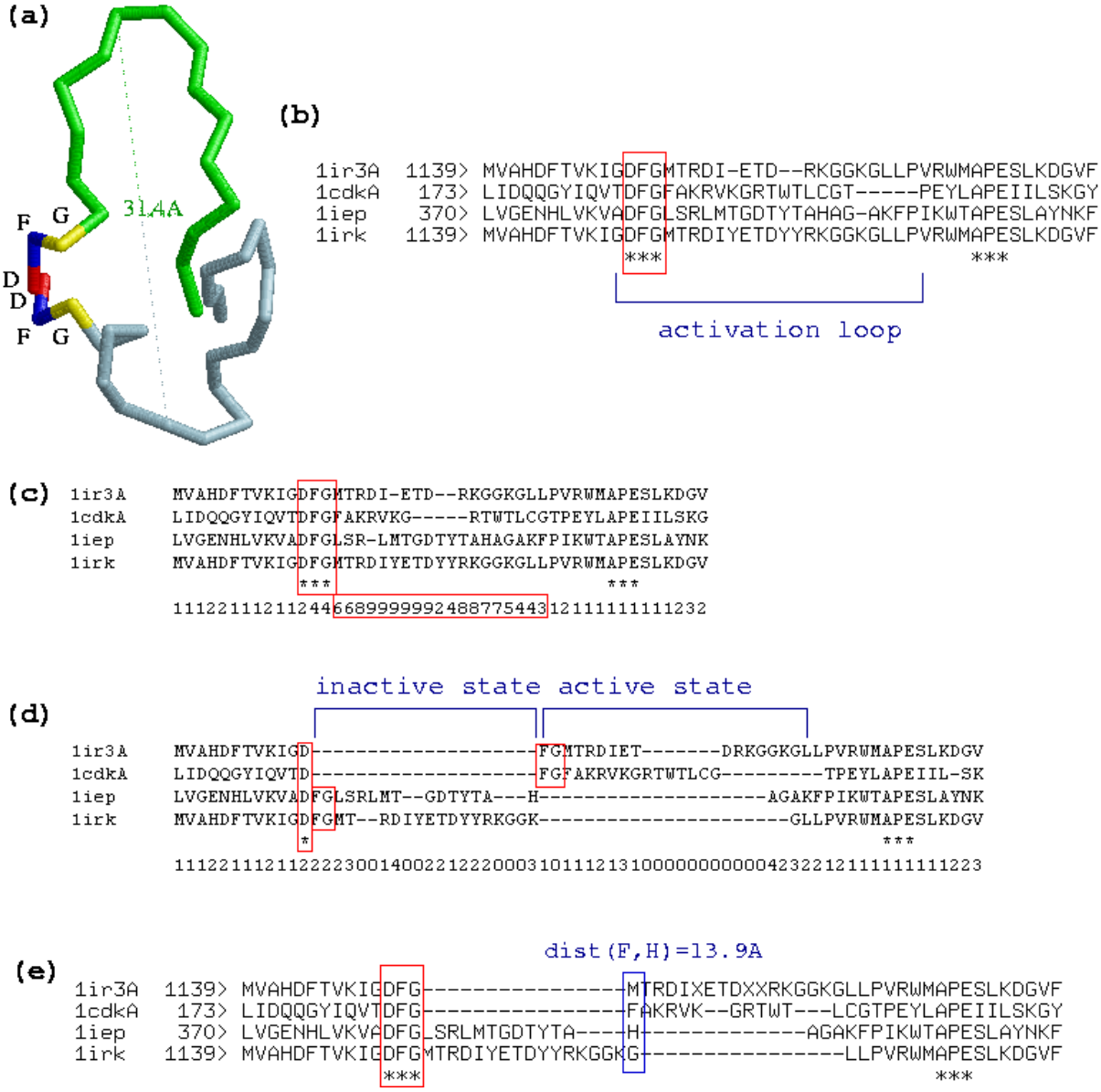


Figure 4.4: (a) Tyrosine kinase activation loop in the active and the inactive state. Structural changes may be as large as 31Å. The DFG motif is colored red-blue-yellow. (b) Multiple sequence alignment by ClustalW[58]. Only the activation loop and some region around it is shown. Obviously, the sequence alignment cannot distinguish between the two structural states. (c) Multiple alignment produced by STACCATO. Investigating the structural conservation score, $Cons_{str}$, reveals a high structural variability of the activation loop region. (d) Applying the distance constrained multiple alignment with the threshold value of 5Å clearly reveals two clustered regions for active and inactive states. For these structural constraints only the aspartic acid from the DFG motif is aligned. (e) Alignment produced by the CE-MC[50] method. While it correctly aligns the DFG motif and identifies the two states of the activation loop, one column (colored blue) incorrectly aligns residues from the beginning of the loop with residues from the end.

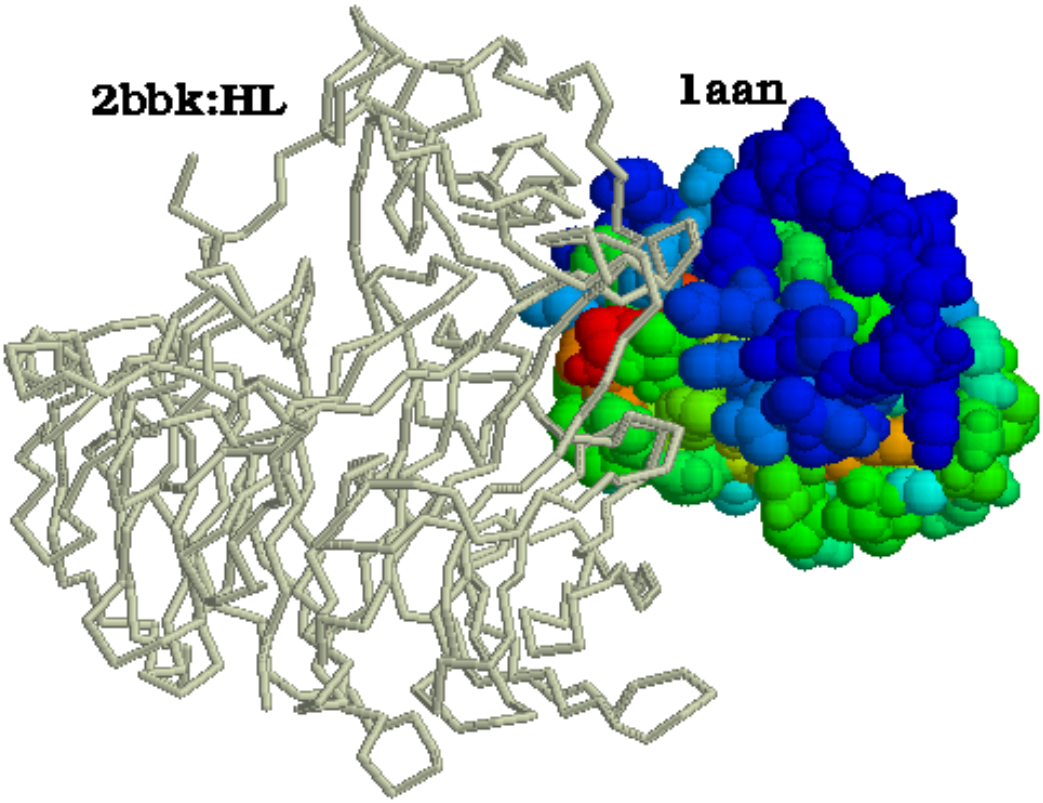


Figure 4.5: Prediction of a protein-protein complex. The complex of Methylamine Dehydrogenase, 2bbkLH (backbone), with Amicyanin, 1aan (surface), is shown. The surface of 1aan is colored according to the conservation score as calculated by the STACCATO method, red - the most conserved, blue - the least conserved. Notice that in the binding site of 1aan there are two highly conserved residues, PRO-94 and HIS-95. The list of conserved residues provided to the docking program helped to raise the rank of a nearly native solution from 159 to 62.

Chapter 5

Recognition of Common Binding Patterns

5.1 Introduction

Binding sites with similar physico-chemical and geometrical properties may perform similar functions and bind similar binding partners. Such binding sites may be created by evolutionarily unrelated proteins that share no overall sequence or fold similarities. Their recognition has become especially acute with the growing number of protein structures determined by the Structural Genomics project. Multiple alignment of binding sites that are known to have similar binding partners allows recognition of the physico-chemical and geometrical patterns that are responsible for the binding. These patterns may help to understand and predict molecular recognition. Moreover, multiple alignment of binding sites allows analysis of the dissimilarities of the binding sites which are important for the specificity of drug leads.

Sequence patterns have been widely used for comparison and annotation of protein binding sites[39]. Several methods search for patterns of residues that are conserved in their 3D positions and amino acid identities[128, 101, 8]. However, there are numerous examples of functionally similar binding sites that cannot be detected by PROSITE-like sequence patterns[39] or by amino acid identity based structural patterns[91, 28]. Figure 5.1 gives an example of four proteins that bind the same

molecule, estradiol, however, these proteins don't have similar overall sequence or 3D structural fold. Multiple structure alignment of functionally similar binding sites might help to reveal a common binding pattern. Several methods have been developed for protein multiple structural alignment[102, 124, 76, 108, 31]. To overcome the alignment complexity of large protein structures these methods apply a variety of heuristics as well as some assumptions on properties of protein backbone, e.g. sequentiality of some backbone fragments. However, similar binding site patterns may appear in proteins with different overall folds. In addition, such patterns may be relatively small and can be easily missed when applying heuristic approaches used for protein backbone alignment. Methods for recognition of a pharmacophore common to a set small ligands[77, 32] share some methodological aspects with problems of protein backbone or binding site alignment. However, most developed methods for common pharmacophore detection are optimized for the specific problem definition, e.g. assume a tree-like ligand topology or, in order to overcome a large number of different ligand conformations, apply randomized techniques[40]. Consequently, the methods for protein backbone alignment or common ligand pharmacophore detection are generally not suitable for recognition of common patterns of protein binding sites.

Several works have analyzed complexes of different proteins with the same ligand. The superimposition between the binding sites has been obtained by alignment of their common ligands[74, 28]. This approach has several limitations. First, it can analyze only protein structures with exactly the same partner ligands. Second, the same ligand can bind in alternative modes even to the same protein binding site[28]. Therefore, alignment according to ligands may fail to recognize the binding pattern.

Computational methods have been developed for direct alignment of protein binding sites. From the algorithmic standpoint, this involves solving a problem of spatial labeled/unlabeled pattern detection. Most of the methods apply clique detection algorithms. Recently, several methods have been proposed that recognize similarity of functional sites in the absence of the overall sequence or fold similarity [128, 101, 122, 104, 66, 70, 71, 117]. However, all of these methods perform a comparison of only two molecules or a comparison of a predefined structural motif against a protein structure during a database search[21]. Pairwise alignments may contain a

large number of features that are not necessarily required for the binding. Multiple alignments of binding sites with the same function may help to recognize the smallest set of features, a *consensus*, that is essential to achieve the desired biological effect. This may improve sensitivity in the database searches of remotely related binding sites. Although it is possible to combine the results of various pairwise comparisons, high scoring pairwise solutions do not necessarily lead to a high scoring solution for a set of molecules[2].

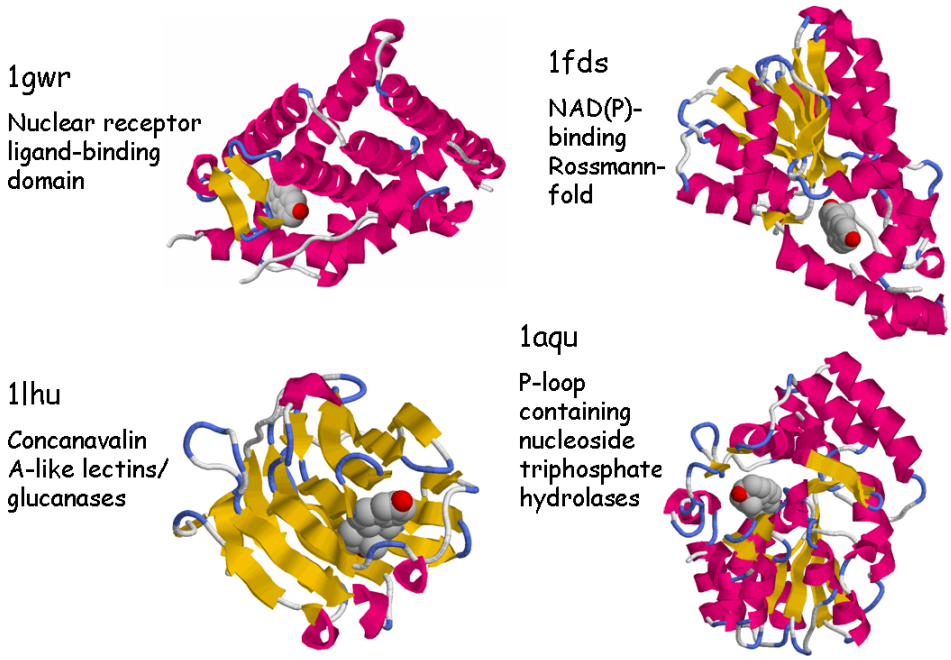


Figure 5.1: Example of four proteins, with different overall function, different amino acid sequence and different overall 3D fold, yet they bind the same molecule, *estradiol* (depicted in spacefill).

The review of computational methods to solve the geometrical problem of the largest common pattern detection (LCP problem) is given in Chapter 2. Here, we briefly state the main results. For two structures the LCP problem is polynomial, though unpractical for implementation. The multiple LCP, mLCP, is NP-Hard even for three structures. Practical approximation algorithms apply alignment technique to generate a polynomial number of Euclidean transformations. Given a set of transformations the problem is reduced to computation of the matched points, which we

denoted as the mSim problem. For K structures and for the bottleneck metric, we defined it as the *3D-ε-K-partite* matching problem. As we have shown above, the *3D-ε-K-partite* matching problem is NP-Hard even for three structures.

Despite the theoretical difficulty, we present an efficient, practical, method, MultiBind, for identification of common protein binding patterns by solving the multiple structure alignment problem. The problem we aim to solve is NP-Hard, therefore our goal is to find a trade-off between practical efficiency and theoretical bounds of solution accuracy, while validating the biological correctness of the results. Here, we solve the multiple alignment problem defined above as pmLCP (Problem (7)) with modifications introduced to adopt the protein physico-chemical properties. We represent a protein binding site as a set of 3D points that are assigned a set of physico-chemical and geometrical properties important for protein-ligand interactions. The implementation of our method includes three major computational steps. The first one is a generation of 3D transformations that *align* the molecular structures. Here, we apply the time efficient Geometric Hashing method[131, 130]. The advantage of this method is that it enables to avoid processing of points that cannot be matched under any transformation. In other words, its time complexity is proportional to the number of potentially matched points included in the defined set of transformations. The second step is a search for a combination of 3D transformations that gives the highest scoring common 3D core. For this step we provide an algorithm that guarantees to find the optimal solution by applying an efficient filtering procedure, which practically overcomes the exponential number of multiple combinations. The final step is a computation of matching between points under multiple transformations, namely *3D-ε-K-partite* matching. Here, we give a fast approximate solution with factor $min(d, K)$, where d is the maximal point density in a sphere of radius 2ε. The overall scheme guarantees to approximate the ε-congruence as well as the cardinality of multiple alignment. We apply MultiBind to some well studied biological examples such as estradiol, ATP/ANP and transition state analogues binding sites. Our computational results agree with the available biological data.

5.2 The MultiBind Algorithm

5.2.1 Input Representation: Physico-Chemical Properties

Selection of the proper representation is crucial for the biochemical significance of the recognized patterns. Given the atomic coordinates of a protein structure, we follow Schmitt et al.[104] and for each amino acid we group atoms with similar physico-chemical properties to functional groups. These are localized by 3D points in space, denoted as pseudocenters. Each pseudocenter represents one of the following properties important for protein-ligand interactions: *hydrogen-bond donor(DON), hydrogen-bond acceptor(ACC), mixed donor/acceptor(DAC), hydrophobic aliphatic(ALI) and aromatic(PI) contacts*. Each amino acid is represented by a set of such pseudocenters (both its backbone and side-chain atoms are considered) (for examples see Figure 5.2). We construct the smooth molecular surface as implemented by Connolly[24] and retain only pseudocenters that represent at least one surface exposed atom. When considering binding sites, we refer only to the surface regions that are within 4Å from the binding partner.

In practice, a comparison of the spatial locations of the retained pseudocenters is not sufficient for the accurate prediction of protein-ligand interactions. Thus, we are interested in the maximal number of matching pseudocenters that are most similar in all the physico-chemical and geometrical aspects. For each pair of pseudocenters, p and q, we define a scoring function *PC-Score(p,q)* which measures the similarity of the properties important for the specific type of interaction in which they can participate. For example, we consider the similarity of partial charges for pseudocenters that can form hydrogen bonds, whereas for aromatic properties we compare the directions of normal vectors to the aromatic moieties. The similarity score between two binding sites, S_1 and S_2 is denoted by *PC-Score*(S_1, S_2) and is defined as a sum of the matched pseudocenters scores. The exact definitions and default parameters are detailed in Appendix A. Therefore, practically, we extend the original pmLCP problem (Problem (7)) to a *weighted pmLCP* problem that we define as[i]:

[i]In our implementation we consider only the pmLCP problem since the 3D-ε-K-partite matching of the mLCP problem introduces additional complications even for greedy approaches, see Section

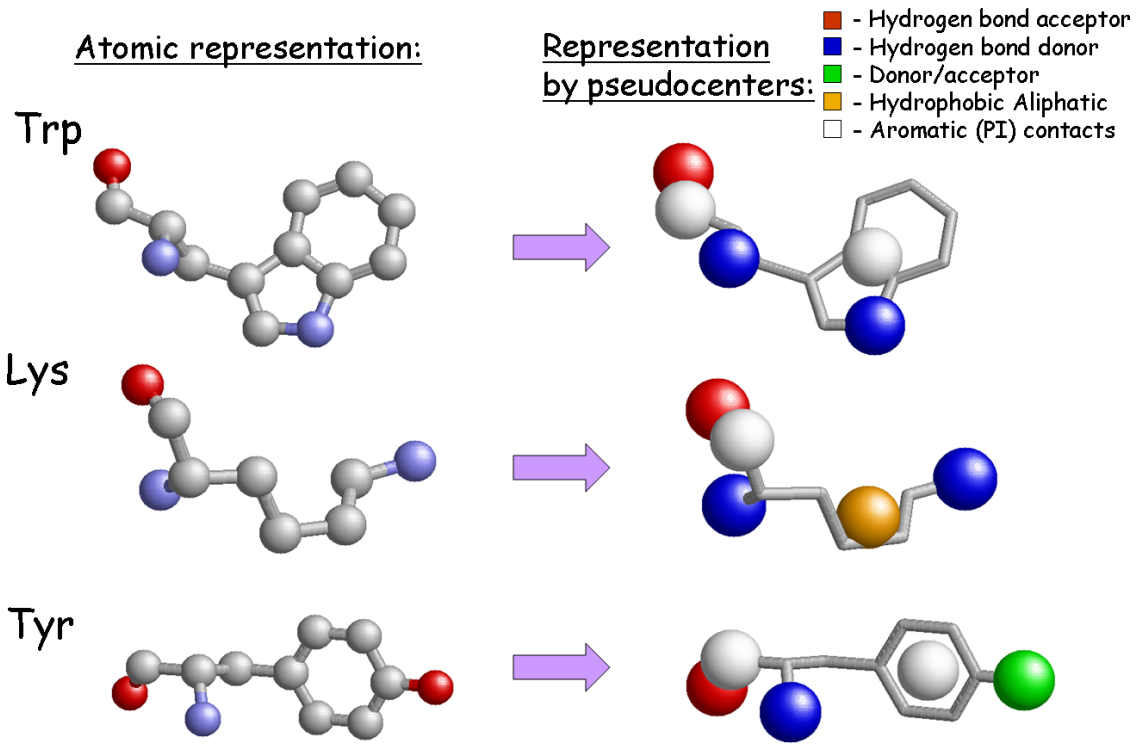


Figure 5.2: Pseudocenter representation[104]. Atoms with similar physico-chemical properties are grouped into functional groups, or *pseudocenters*. Examples of pseudocenter representation are depicted for amino acids Lryptophan, Lysine and Tyrosine.

Problem 9. (Max-Min Weighted *pmLCP* Problem) Given $\varepsilon > 0$, a scoring function PC-Score, a pivot set S_1 and $K-1$ point sets S_i, $i = 2, ..., K$ find transformations $\{T_i\}$, $i = 2, ..., K$, and equal sized sets $\{S'_i \subseteq S_i\}$, $i = 1, ..., K$, such that $d(S'_1, T_i(S'_i)) \leq \varepsilon$, $i = 2, ..., K$, and min_iPC-Score$(S'_1, T_i(S'_i))$ is maximal.

5.2.2 The Pattern Matching Algorithm

The algorithm consists of three major computational steps: (1) generation of 3D transformations and potential points for matching; (2) combinatorial search for a combination of 3D transformations that gives the highest scoring common 3D core (*Traversal stage*); and (3) computation of *3D-ε-K-partite-pivot* matching. The short description of the algorithm flow is given in Figure 5.3.

2.3.

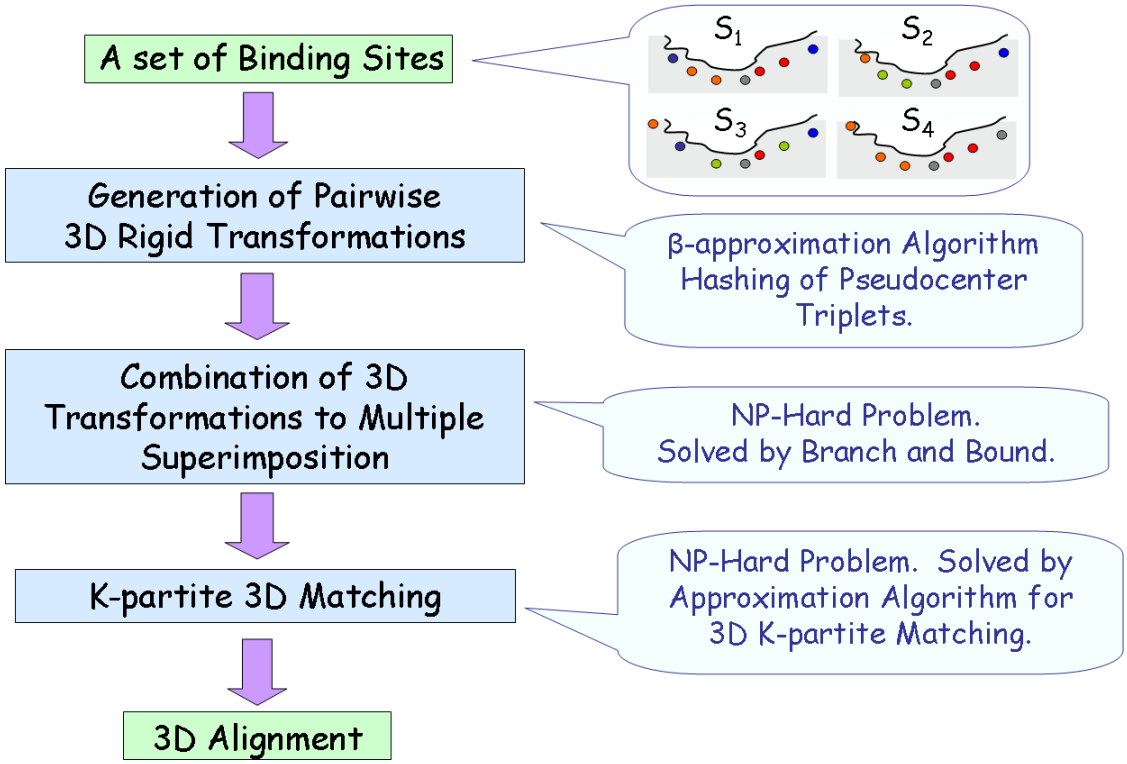


Figure 5.3: MultiBind algorithm main three stages.

Preprocessing Stage

In our approach we follow the efficient strategy of the Geometric Hashing method[131, 130]. The Geometric Hashing method consists of two stages, *preprocessing* and *recognition*. At the *preprocessing* stage each triplet of pseudocenters, (a, b, c), from each molecule except the pivot is considered as a local reference frame $r = LRF(a, b, c)$ (definition of local reference frame is given in Section 2.2.1). The coordinates of the other pseudocenters are calculated with respect to the local reference frame r. This information is stored in a *Geometric Hash Table*. The key to the hash table is (x^r, y^r, z^r, p), where (x^r, y^r, z^r) are pseudocenter coordinates with respect to the local reference frame r, and p is the physico-chemical property of the pseudocenter. Only pseudocenters with the same property can be matched [ii]. The data stored in the

[ii]Pseudocenters that can function both as hydrogen bond donors and acceptors are encoded twice, once as donors and once as acceptors.

hash table includes the key itself and the identifiers of the molecule and the reference frame.

Recognition Stage

In the *recognition* stage the same process as in the *preprocessing* stage is repeated for the pivot molecule. However, instead of storing data in the hash table, all entries close to the key within radius ε and with the same physico-chemical property are retrieved. For each reference frame r of the pivot structure a voting table is created. It counts the number of matched pseudocenters for each reference frame stored in the hash table. For simplicity, we explain the method for the pure geometrical case, i.e. for the pmLCP problem. If a reference frame r' from structure i received v votes that means the following. Define a 3D transformation $T_{r,r'}$ that superimposes the triplets of points r' on r according to the *Alignment Rule*. Applying $T_{r,r'}$ on S_i will result in v point pairs from the pivot and i'th structure that are within ε distance. Thus, the size of a maximal matching between S_{pivot} and $T_{r,r'}(S_i)$ is less than v. Therefore, if $v < M^*$, where M^* is the size of the largest multiple solution found so far (initially $M^* = 0$), there is no need to consider transformation $T_{r,r'}$ for the next steps of the algorithm. For each survived transformation T we store the list of matched points, $\{(p, q) : p \in S_{pivot}, q \in S_i, |p - T(q)| \leq \varepsilon\}$.

Traversal stage

For each reference frame of the pivot structure we create a combinatorial bucket that contains transformations that received a high number of votes. Namely, a combinatorial bucket for the reference frame r is defined as $CB_r = \{T^2, T^3, ..., T^K\}$, where T^i is a set of transformations for structure i that received $v > M^*$ votes. A multiple alignment is a combination of $K - 1$ transformations, $(T_{i_2}, T_{i_3}, ..., T_{i_K})$. The number of all possible combinations equals to $|T^2| \cdot |T^3| \cdot ... \cdot |T^K|$, which is exponential in K. However, we have implemented a branch-and-bound traversal method which in practice is very efficient. First we provide some definitions. Given a transformation vector of the first t structures, $T = (T_{i_2}, ..., T_{i_t})$, create a *3D-$\varepsilon$-t-partite-pivot* graph,

$G(T) = G(S_1, T_{i_2}(S_2), ..., T_{i_t}(S_t))$. Define single sides of the graph $G(T)$, $G(T)[j] = \{p_j : p_j \in S_j,\ \exists p_1 \in S_1\ (p_1, p_j) \in G(T)\ and\ \forall k \leq t\ \exists p_k \in S_k\ (p_1, p_k) \in G(T)\}$. Let $M(G(T))$ be a maximal *3D-ε-t-partite-pivot* matching of the graph $G(T)$. Obviously, $M(G(T)) \leq M(G(S_{pivot}, T_{i_j}(S_j))) \leq |G(T)[j]|$.

Given a combinatorial bucket $CB = \{T^2, T^3, ..., T^K\}$ we iteratively traverse it in the following manner. Assume that we have created a vector $T = (T_{i_2}, T_{i_3}, ..., T_{i_t})$, $T_{i_j} \in T^j$. We try to extend it with a transformation $T_{i_{t+1}} \in T^{t+1}$, $T^* = (T_{i_2}, T_{i_3}, ...$ $,T_{i_t}, T_{i_{t+1}})$. Clearly, $|G(T^*)[j]| \leq |G(T)[j]|$, $j = 2, ..., t$. Therefore, if for some index j holds $|G(T^*)[j]| \leq M^*$, then we can disregard the vector T^* and start to build another combination of transformations (a simplified schematic diagram is given in Figure 5.4). Essentially, we continue with the vector T and try to add another transformation from T^{t+1}, and so on. The number of traversals may be exponential, however in practice $M(G(T))$ drops very quickly below M^* as the algorithm advances in iterations in the *recognition* stage [iii]. Still, the theoretical bound is $O(n^3 n^{3(K-1)})$.

3D-ε-K-partite-pivot Matching

During the traversal stage, once we reach the last bucket we have a uniquely defined 3D-ε-K-partite-pivot graph. The next step is to solve the matching problem. As we have shown above this problem is NP-Hard. We apply a greedy method, which iterates over pivot points and selects K-tuples, from non-selected points. This method gives a K approximation to the largest matching since at each greedy selection of K-tuples it may violate at most K nodes that may belong to the optimal matching[iv]. In the context of molecular structures for small ε (around 3Å) the maximal node degree is bound by a small constant. Therefore the time complexity of the greedy method is $O(Kn)$.

[iii] In the second example from the *Results* section, the total number of combinations for all combinatorial buckets is about $1.3 \cdot 10^{11}$, which shows the exponential nature of the problem. The filtering procedure leaves only 246310 combinations of multiple alignments. Most filtering is done already at the third structure ($t = 3$).

[iv] The best known approximation algorithm for hyper-graphs gives K/2 ratio[63]

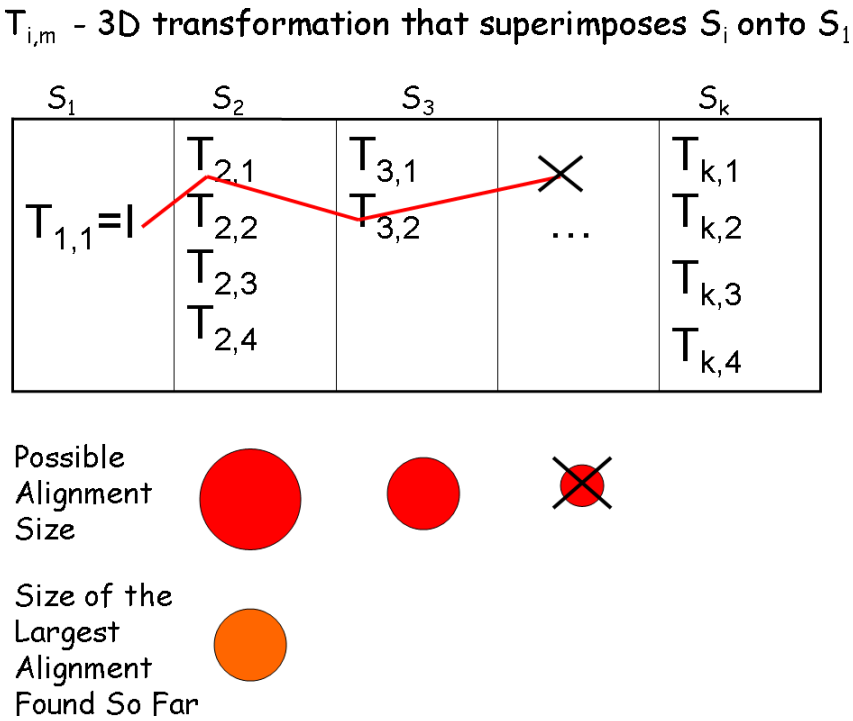


Figure 5.4: Traversal stage. The bucket i contains pairwise transformations that superimpose molecule S_i onto S_1. Each time a transformation from the next bucket is added to the multiple alignment, a possible common core is reduced. If its upper bound size becomes less then the size of the largest solution found so far, then this multiple combination is discarded.

Experimental Properties of ε-K-partite Graphs

The running times the algorithm depend on the maximal node degree of experimental ε-K-partite graphs. Specifically, it depends on the maximal node degree of any sub 2-partite graph. Therefore, it is enough to estimate node degree properties from pairwise alignments. We conducted 150 pairwise binding site alignments to estimate the node degree properties (the number of graphs in each pairwise alignment is $O(n^6)$).

In the case of $\varepsilon = 3\mathring{A}$, the maximal node degree of the constructed graphs is 7, while average degree is only 0.28 with standard deviation 0.65. Therefore, for any practical purpose the size of the graphs is linear in the number of points.

As discussed above, the approximation ratio of the greedy algorithm for the 3D-ε-K-partite-pivot matching is K. At each greedy selection of a K-tuple it may violate at

most K nodes that may belong to the optimal matching. However, the approximation ratio depends on the point distribution density. If for each input point a sphere of radius 2ε centered at the point contains at most d points (from the same point set), then any greedily selected K-tuple may violate at most d K-tuples from the optimal matching. Therefore, the approximation ratio of the greedy algorithm is $min(d, K)$.

In the case of $\varepsilon = 3\mathring{A}$, we computed point distribution densities in spheres of radius $6\mathring{A}$ centered at each input point. Notice, that the counted points should have the same physico-chemical properties as the point at the sphere center. We considered 250 different binding sites. The maximal density is 12 points, while average density is only 4.2 with standard deviation 2.9. Therefore, in practice the worst case approximation ratio is 12 and it is independent of the number of structures.

Theorem 5.1. *MultiBind algorithm is an [$\delta = min(d, K)$]-[$\gamma = 7\varepsilon$-additive]- approximation*[v] *for* **Problem 7** *and has time complexity $O(n^{3K}nK)$.*

In practice, when solving the *Max-Min Weighted pmLCP* we introduce the following modifications. First, we define M^* to be the highest physico-chemical score of the multiple solution found so far. Given equally sized sets $(S_1, ..., S_t)$ the physico-chemical score M is defined as in Problem 4 by $M = min_j PC\text{-}Score(S_1, S_j)$, $j = 2...t$. When traversing the combinatorial buckets, instead of looking at the cardinality of the side j, $|G(T^*)[j]|$, we estimate the upper bound of $PC\text{-}Score(S_1, T_j(S_j))$ as $PC\text{-}Score(G(T^*)[j])= \sum_{q \in T_j(S_j)} max_p PC\text{-}Score(p, q)$. Therefore, we disregard vectors T^* for which $PC\text{-}Score(G(T^*)[j]) \leq M^*$ for some j. We retain a user defined number of high scoring solutions, which are then evaluated by an additional *Overall Surface Scoring*[117] function which compares the corresponding surfaces of the binding sites.

5.3 Biological Results

Below we present examples of application of MultiBind for recognition of patterns required for binding of different ligands. In each of the presented examples, we describe

[v] It is possible to reduce the 7ε-additive approximation to any accuracy γ, $0 < \gamma$, by applying a discretization technique of the transformational space[56, 44]. However, the payoff is increasing the time complexity factor proportional to $(\frac{\varepsilon}{\gamma})^5$ (see Section 2.2.1).

the details of a single solution that received the highest score. Additional examples of application of MultiBind are presented in [86, 13]. The running times are measured on a standard PC, Intel(R) Pentium(R) IV 2.60GHz CPU with 2GB RAM. The default distance threshold for the ε-congruence is 3.0Å .

5.3.1 ATP/ANP Binding Sites of Protein Kinases.

To validate the performance of the method on a well studied example we have selected a set of ATP/ANP binding sites extracted from 5 different protein kinases: cAMP-dependent PK (1cdk), Cyclin-dependent PK, CDK2 (1hck), Glycogen phosphorylase kinase (1phk), c-Src tyrosine kinase (2src), Casein kinase-1, CK1 (1csn). We applied MultiBind to perform a multiple alignment of the corresponding ATP/ANP binding sites. These were recognized to share 13 pseudocenters, 4 of which are created by amino acids with the same identity (see Figure 5.5 and Table 5.1). The RMSD between the adenine moieties (which are not a part of the input and are used for verification only) under these transformations is less than 1.2Å . This indicates that the multiple superposition is computed correctly. The average size of the binding sites is 69 pseudocenters, and the running time is 62 minutes. It must be noted that since these proteins share similar overall folds, the 3D superposition problem of the binding sites can be solved by multiple backbone alignment methods[108, 31]. However, these methods do not give solution to the *3D-ε-K-partite* matching problem of physico-chemical features (since these are not-ordered on the protein surface). Below we present two examples for which both the superimposition and the matching problems can not be solved by standard protein backbone alignment methods.

5.3.2 Transition State Analogue Binding Sites.

We have selected five binding sites complexes with endo-oxabicyclic transition state analogues (TSA/BAR). The binding sites were extracted from proteins of three different SCOP[92] folds: (1) Chorismate mutase II (1ecm, 4csm, 3csm); (2) Bacillus chorismate mutase-like (2cht); (3) Immunoglobulin-like beta-sandwich (1fig). Figure 5.6 depicts these five structures.

Figure 5.7 presents 8 functional groups that were recognized by MultiBind to be shared by all the binding sites. Table 5.2 presents the exact details of the pattern and calculated K-partite matching. Two of the compared proteins (1ecm and 4csm) were previously aligned by Schmitt et al[104]. Most of the pseudocenters recognized by MultiBind are indeed a subset of those obtained by their pairwise alignment method (except for two donors contributed by 1ecm:Arg28). However, 10 of the functional groups common to a pair of chorismate mutases according to their study, were not recognized to be common to the five structures compared by MultiBind. Alignment of multiple structures with different folds helps to identify the minimal set of features required for the binding of endo-oxabicyclic transition state analogues. The average size of a binding site is 30, and the running time is 8 minutes.

5.3.3 Estradiol Binding Sites.

Estradiol molecules are known to bind to protein receptors with different overall sequences and folds. The dataset of this study (Figure) was comprised of the binding sites of 7 proteins from 4 different SCOP[92] folds: (1) Nuclear receptor ligand-binding domain (3ert, 1a52, 1err, 1gwr); (2) NAD(P)-binding Rossmann-fold (1fds); (3) Concanavalin A-like lectins/glucanases (1lhu); (4) P-loop containing nucleoside triphosphate hydrolases (1aqu). Two of these structures were crystallized with Raloxifen (1err) and 4-hydroxytamoxifen (3ert), which are different from estradiol. In spite of the conformational changes required to accommodate these ligands, MultiBind has recognized 6 functional groups shared by all the binding sites (see Figure 5.9). One of them is a conserved Phenylalanine (1lhu:Phe67) with an aromatic property shared by all the binding sites. The mean binding site size is 39 pseudocenters and the running time is 16 minutes.

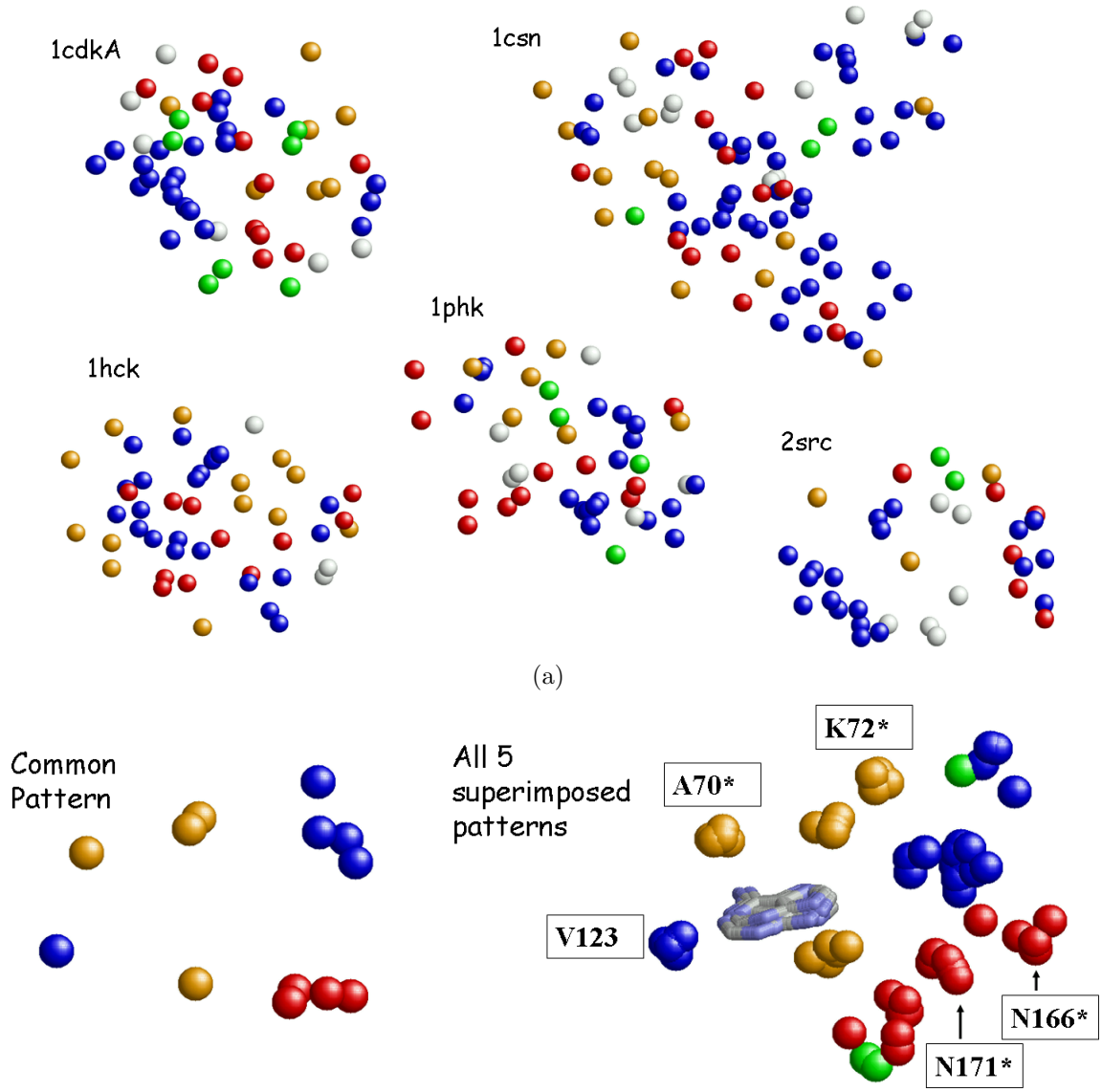


Figure 5.5: ATP/ANP binding sites of protein kinases. (a) Five ATP/ANP binding site represented by pseudocenters. (b) Common pattern as detected by MultiBind. The labeling is according to 1cdkA. The pseudocenters that are common to identical amino acids are marked with asterisk. The ligand molecules, adenine moieties, are presented for verification purpose only and are not a part of the input to MultiBind.

Mol 1 (1cdk)			Mol 2 (1csn)			Mol 3 (1hck)			Mol 4 (1phk)			Mol 5 (2src)		
Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA
A.51	T	DON	.20	E	DON	.12	E	DON	.27	R	DON	.275	Q	DON
A.51	T	PII	.20	E	PII	.12	E	PII	.27	R	PII	.275	Q	PII
A.51	T	ACC	.20	E	ACC	.12	E	ACC	.27	R	ACC	.275	Q	ACC
A.55	G	DON	.24	G	DON	.15	Y	DON	.31	S	DAC	.279	G	DON
A.57	V	ALI	.26	I	ALI	.18	V	ALI	.33	V	ALI	.281	V	ALI
A.70	A	ALI	.39	A	ALI	.31	A	ALI	.46	A	ALI	.293	A	ALI
A.72	K	ALI	.41	K	ALI	.33	K	ALI	.48	K	ALI	.295	K	ALI
A.123	V	DON	.88	L	DON	.83	L	DON	.106	M	DON	.341	M	DON
A.127	E	ACC	.91	S	DAC	.86	D	ACC	.110	E	ACC	.345	S	DAC
A.166	D	ACC	.131	D	ACC	.127	D	ACC	.167	D	ACC	.386	D	ACC
A.170	E	ACC	.135	D	ACC	.131	Q	ACC	.153	E	ACC	.390	A	ACC
A.171	N	ACC	.136	N	ACC	.132	N	ACC	.154	N	ACC	.391	N	ACC
A.183	T	ALI	.153	V	ALI	.144	A	ALI	.166	T	ALI	.403	A	ALI

Table 5.1: The details of the common pattern calculated for the ATP/ANP binding sites. Each three columns correspond to a different molecule. For each molecule, the first column provides the chain identifier and the residue number. The second column, contains the residue name (one letter amino acid code). The third column gives an abbreviation of the physico-chemical property (see Section (5), Input Representation).

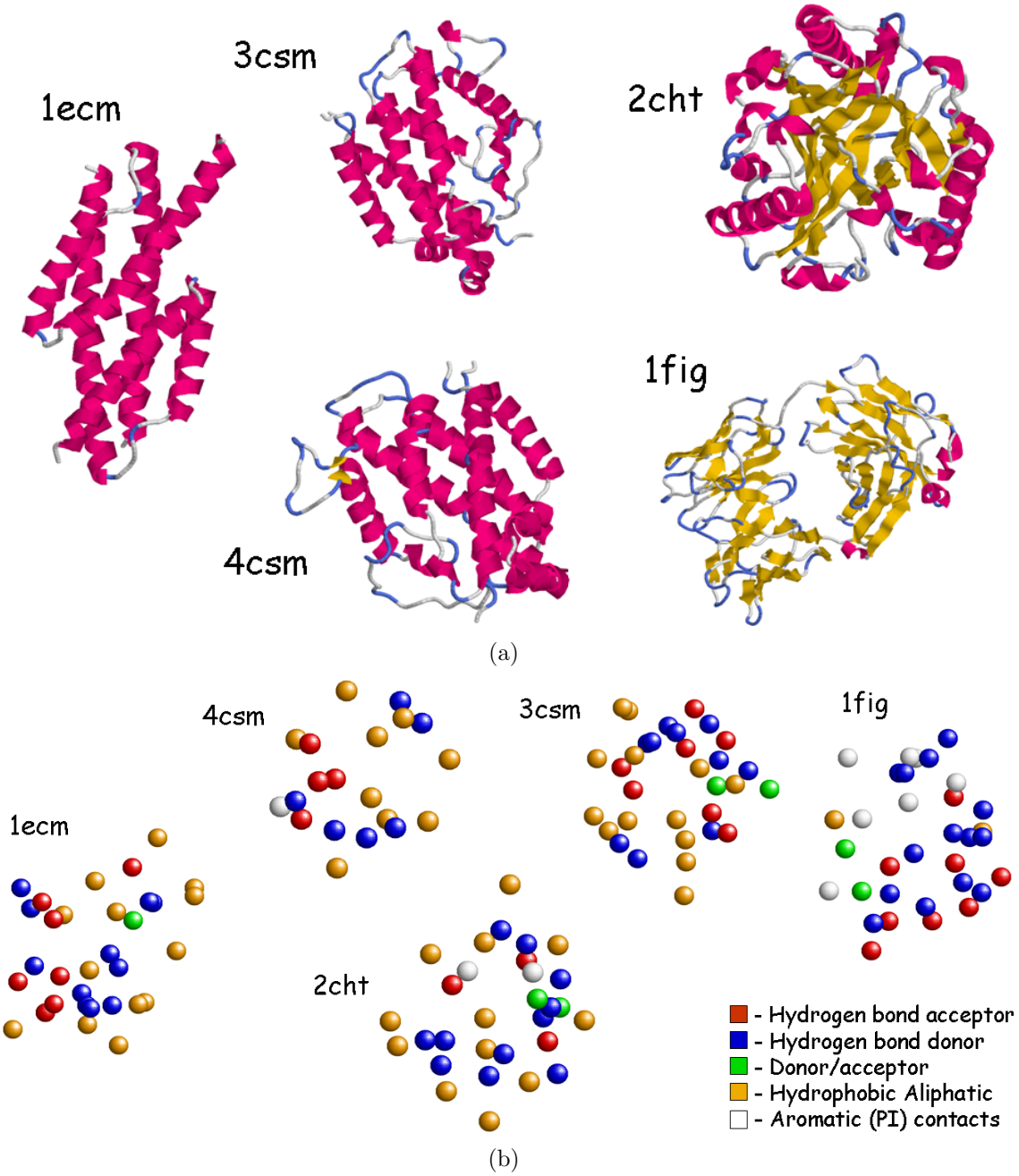


Figure 5.6: Transition state analogue binding proteins. (a) Five different proteins, from three different SCOP folds, bind endo-oxabicyclic transition state analogues (TSA/BAR). (a) Binding sites represented by pseudocenters.

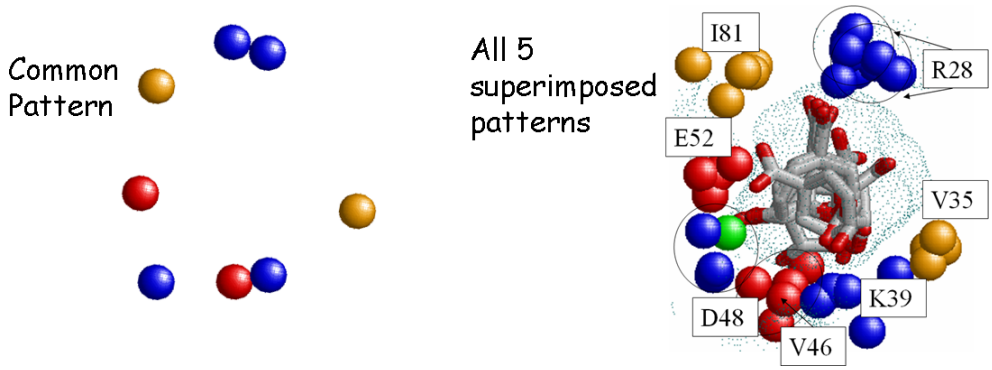


Figure 5.7: Multiple alignment of five endo-oxabicyclic transition state analogue binding sites as detected by MultiBind. The labeling and the surface (depicted in dots) is according to 1ecm. The ligand molecules are presented for verification purpose only and are not a part of the input to MultiBind.

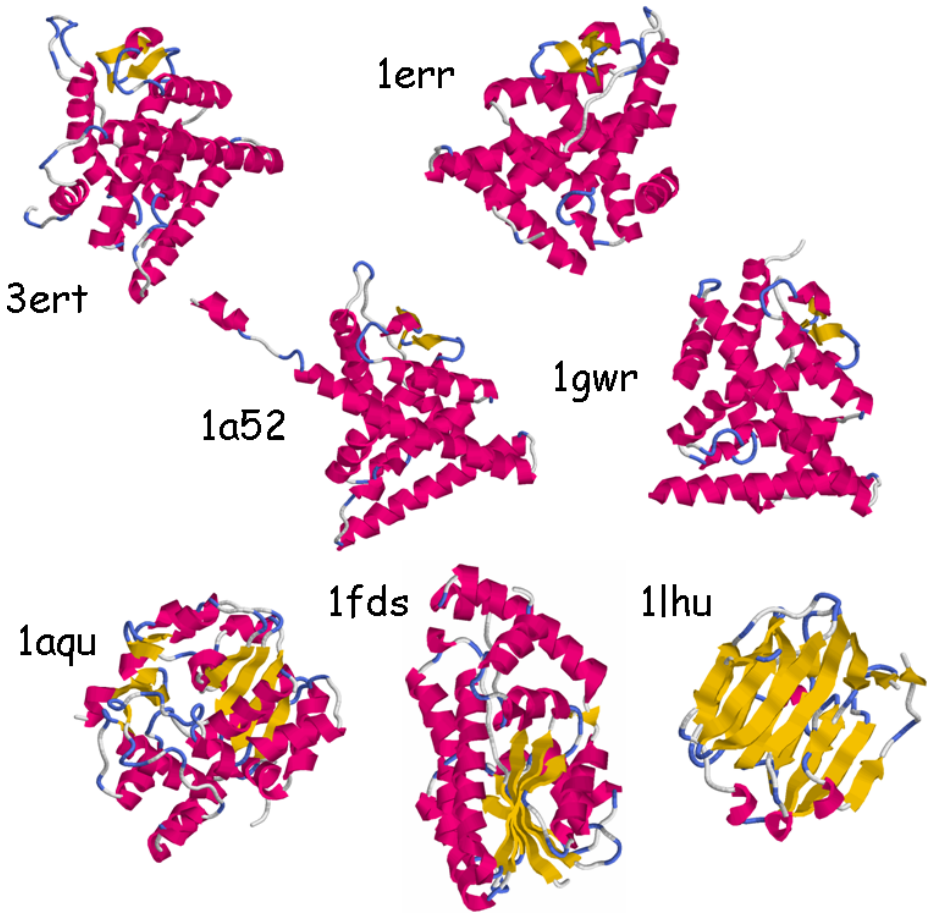


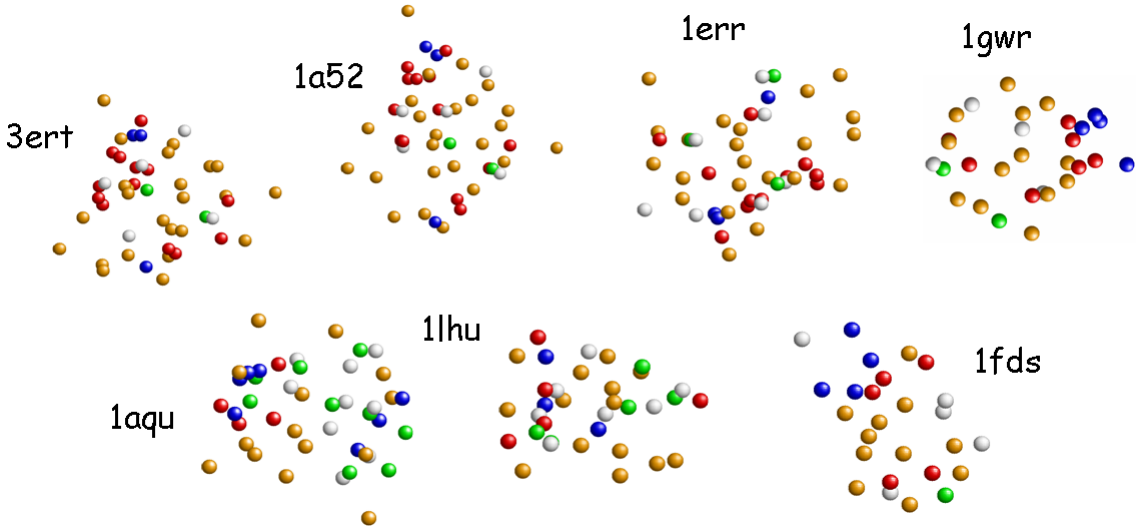
Figure 5.8: Estradiol binding proteins.

Mol 1 (1ecm)			Mol 2 (4csm)			Mol 3 (3csm)			Mol 4 (1fig)			Mol 5 (2cht)		
Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA
A.28	R	DON	B.157	R	DON	A.157	R	DON	H.100	G	DON	B.116	R	DON
A.28	R	DON	B.157	R	DON	A.157	R	DON	H.100	Y	DON	B.116	R	DON
A.35	V	ALI	B.164	V	ALI	A.16	R	ALI	H.95	R	ALI	B.115	L	ALI
A.39	K	DON	B.168	K	DON	A.16	R	DON	H.33	N	DON	B.90	R	DON
A.46	V	ACC	B.192	I	ACC	A.192	I	ACC	H.33	N	ACC	B.78	E	ACC
A.48	D	DON	B.194	N	DON	A.194	N	DON	H.50	N	DON	A.74	T	DAC
A.52	E	ACC	B.198	E	ACC	A.198	E	ACC	H.35	N	ACC	A.73	V	ACC
A.81	I	ALI	B.239	I	ALI	A.239	I	ALI	L.96	L	ALI	A.73	V	ALI

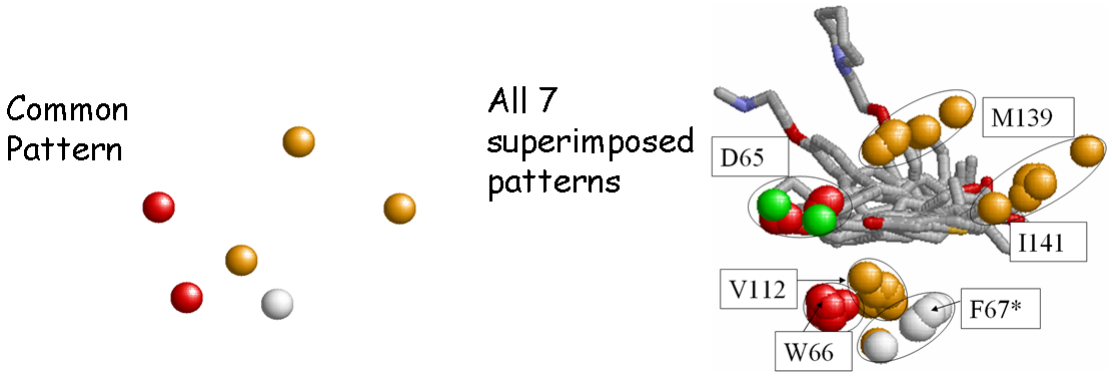
Table 5.2: The details of the common pattern calculated for the transition state analogue binding sites. For the table description see Table 5.1.

Mol 1 (3ertA)			Mol 2 (1a52A)			Mol 3 (1fds)			Mol 4 (1errA)			Mol 5 (1gwrA)			Mol 6 (1lhuA)			Mol 7 (1aquA)		
Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA	Ch.Id	AA	PHA
A.343	M	ALI	A.343	M	ALI	.96	L	ALI	A.343	M	ALI	A.343	M	ALI	A.141	I	ALI	A.243	M	ALI
A.349	L	ALI	A.349	L	ALI	.149	L	ALI	A.349	L	ALI	A.349	L	ALI	A.107	M	ALI	A.75	I	ALI
A.387	L	ACC	A.386	I	ACC	.186	G	ACC	A.387	L	ACC	A.387	L	ACC	A.65	D	ACC	A.51	T	DAC
A.387	L	ALI	A.387	L	ALI	.187	P	ALI	A.387	L	ALI	A.387	L	ALI	A.134	K	ALI	A.106	K	ALI
A.394	R	DON	A.394	R	DON	.142	S	DAC	A.394	R	DON	A.394	R	DON	A.82	N	DON	A.52	T	DAC
A.404	F	PII	A.404	F	PII	.155	Y	PII	A.404	F	PII	A.404	F	PII	A.67	F	PII	A.255	F	PII

Table 5.3: The details of the common pattern calculated for the estradiol binding sites. For the table description see Table 5.1.



(a)



(b)

Figure 5.9: Estradiol Binding Sites. (a) Binding sites represented by pseudocenters. (b) Multiple alignment of eight estradiol binding sites, the labeling is according to 1lhu.

5.3.4 Evaluation of the Recognized Pattens

There are four approaches which we have used to evaluate the quality of the obtained patterns. First, as discussed above, we have compared them to patterns which were previously described in the literature for alignment between pairs of molecules. Second, the recognized common patterns were compared with the results of the LPC

program[121], which computes a protein-ligand interactions of the input protein-ligand complexes. In all the patterns detected by MultiBind, the common core is a subset of interactions computed by LPC.

Third, we have superimposed the binding sites which are complexed with the same ligand molecules and compared the obtained pattern with that of MultiBind. Fourth, for each pattern, we have calculated the frequency of its occurrence in PDB and analyzed the influence of the common pattern size on the algorithm running times.

Specifically, in order to compare the patterns recognized by MultiBind with the results obtained by superimposition of ligand molecules, we performed such an alignment for cases where the proteins are complexed with the same binding partners. In this case, the transformations were calculated by the superimposition between the ligand molecules (i.e. no transformational search is performed by MultiBind), while the rest of the MultiBind stages remained the same. As expected, in the first example of ATP/ANP binding sites of protein kinases the results of MultiBind were similar to those obtained by the superimposition of the adenine moieties. However, in the last two examples alignment by ligands failed to recognize any significant pattern (less than 3 pseudocenters), while MultiBind identified patterns of size 8 and 6 pseudocenters.

Next, we calculate the frequency of occurrence of patterns recognized by MultiBind on the surfaces of proteins from the PDB[14]. For each of the above mentioned examples we searched the patterns recognized by MultiBind on the complete surfaces of proteins in the non-redundant PDB representation by the ASTRAL dataset[19] (release 1.65, less than 40% sequence identity). Specifically, for each example we searched the recognized structural pattern represented by pseudocenters of the *pivot* molecule. Each pattern recognized on the surface of some protein was scored according to the *PC-Score* function of MultiBind described in the Appendix. We count the number of times that a pattern with a score higher than the score of the multiple alignment was observed and calculate its ratio to the total number of searched proteins[vi]. The obtained ratio represents the frequency of occurrence of the recognized pattern in the

[vi]The score of the multiple alignment is defined according to Problem 4 as min_iPC-Score$(S_1, T_i(S_i))$, where S_1 is the pivot and S_i is the outlier.

ASTRAL dataset. The lower the ratio the more significant and rare is the pattern. It must be noted, that using the score of the outlier for our measurements provides the highest ratio, i.e. using non-outliers will increase the rarity and the significance of the pattern. The ratio obtained for the transition state analogues binding pattern is 0.006, which means that its frequency of occurrence in the ASTRAL dataset is 0.6%.

The ATP/ANP binding pattern of protein kinases was found in 0.2% of the ASTRAL dataset. Half of the proteins that were found to contain this pattern are protein kinases, which are the top ranking solutions. The specificity of this pattern can be explained by the fact that it was constructed from a set of binding sites all of which are extracted from protein kinases (as opposed to other examples where the pattern was constructed by multiple alignment of proteins with totally different overall sequences and folds). Thus, the obtained pattern may contain features which are important for the structure or activity of this family and are not necessarily required for the binding of ATP/ANP molecules.

A different picture was observed when the ASTRAL dataset was searched with the estradiol binding pattern obtained by the alignment of 7 proteins from 4 different folds. The size of the pattern is 6 pseudocenters and it was found in 10% of the ASTRAL dataset. Next, we performed multiple alignment with MultiBind, without the outlier (PDB:1aqu), which is the binding site most different from the rest of the aligned binding sites. Alignment of 6 binding sites from 3 different folds revealed a pattern of 7 features. The frequency of occurrence of this pattern in the ASTRAL dataset is 0.6%. This shows that addition of the seventh molecule of Estrogen sulfotransferase (PDB:1aqu) to the multiple alignment with MultiBind, significantly reduces the rarity and the specificity of the common pattern. Furthermore, we have removed one additional molecule (PDB: 1fds) and aligned between 5 molecules of 3 different folds. This time, the recognized common pattern was of size 9 and its frequency of occurrence was 0.2%.

To summarize, patterns recognized by multiple alignment with MultiBind can be used to search for other proteins that have similar binding patterns and biological functions. The results of such searches depend on the number and diversity of structures used in the multiple alignments. When the structures are very diverse the

common patterns are smaller and their frequency of occurrence in PDB is higher. Searches with these patterns will lead to a high number of false positive solutions but may reveal unexpected similarities to structurally different proteins. On the other hand, when the structures are similar or when the number of aligned structures is small, the obtained pattern is larger and less frequent. Searches with these patterns will usually lead to less false positives but may fail to recognize novel similarities to members of different protein families. For example, the results of pairwise alignments usually provide patterns that are not found in other proteins except the compared pair and their homologues.

5.3.5 Algorithm Performance

For each of the above described biological examples we evaluate the practical influence of the number of molecules (K) on the algorithm running times. In each example we gradually reduce the number of aligned molecules by excluding the outlier (the molecule most different from the pivot). The rest of the parameters were left unchanged. As we have shown in Section (5) the theoretical complexity of MultiBind is $O(n^{3K}nK)$. However, as can be seen in Figure 5.10 the practical running times are significantly better.

To estimate the influence of the common pattern size on the algorithm running time, we used an additional example of adenine binding sites. Here, we took the dataset used in our first example of ATP/ANP binding patterns, but have made two modifications. First, we have limited the compared binding sites to regions in the vicinity (4Å) of the adenine moieties. This has significantly reduced the size of the binding sites, and, as can be seen in Figure 5.10, lead to significantly lower running times. However, when we added two additional adenine binding sites (PDB codes: 1mjh and 1e8x) and aligned between 7 molecules the running times jumped significantly. This is explained by the fact that the size of the common pattern became small (was reduced from 8 to 5 pseudocenters) and the bound stage of our branch-and-bound algorithm became inefficient. Our method is more effective when the common pattern is larger. However, its running times may become exponential

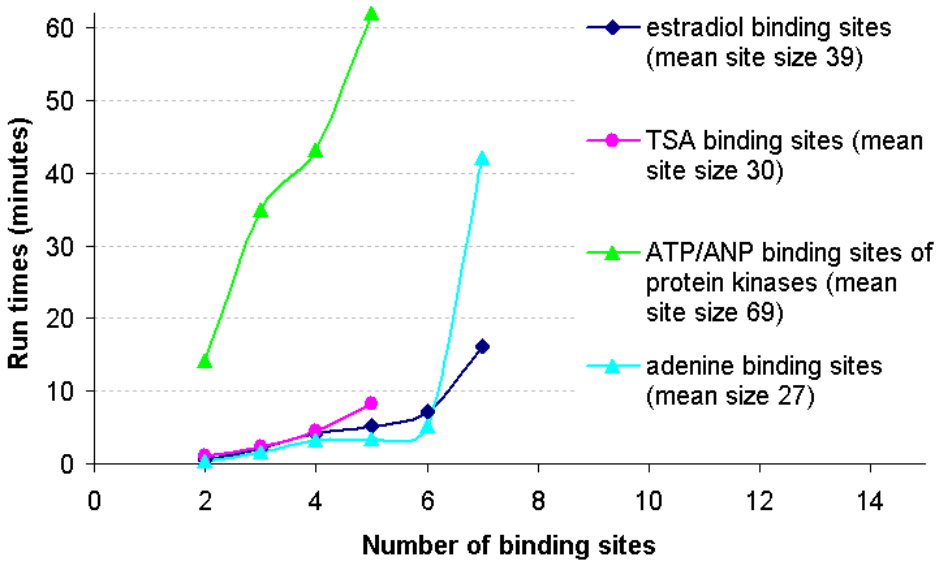


Figure 5.10: MultiBind running times as function of the number of the alignment molecules. The size of the binding sites used in the alignments are detailed in the legends of the series and are measured by the number of pseudocenters.

as the common pattern drops to 3-4 pseudocenters, though the significance of such a pattern drops as well.

An additional parameter, which has a significant impact on the practical running times, is the distance threshold, ε, for the maximal allowed distance between the matched pseudocenters. In all of the described examples we have used a rather permissive default threshold of $\varepsilon = 3$.

5.4 Conclusions

We have presented a novel computational method, MultiBind, for recognition of physico-chemical binding patterns. The presented method is practically efficient for multiple alignment of protein binding sites and guarantees to detect an approximate solution for the case of pure geometrical problem. Despite the computational hardness of the general structural alignment problem, we have presented an efficient filtering procedure which in our applications practically overcomes the exponential number of

multiple combinations.

MultiBind was applied to several biological targets, such as the binding sites of estradiol, ATP/ANP and transition state analogues. MultiBind is the first method that performs multiple alignment of binding sites in the absence of overall sequence, fold or binding partner similarity. To the best of our knowledge, the presented results can not be obtained by any other existing computational method. We hope that it will be a useful tool in prediction of molecular recognition and in identification of *consensus binding patterns*. These common patterns computed from known binding sites, can help to locate and specify binding sites on protein surfaces with unknown function.

However, from the biological standpoint the method has several limitations. First, there is no explicit treatment of protein flexibility which is introduced only through a set of thresholds to allow variability in locations. Second, due to the hardness of the problem the method is practically limited to point sets of size about 100. Third, scoring functions are known to be one of the major problems in all types of *in silico* predictions. The scoring function of MultiBind suffers from the same limitations[117].

5.5 Multiple Alignment of Protein-Protein Interfaces

Above, we studied the problem of protein binding site similarity without considering the information about the binding partner. Clearly, when structural data on a protein-protein complex with both partners is available, then much more information about the binding properties can be gained. With the increasing significance for study of protein association and dissociation, the number of high resolution structures of protein-protein complexes grows rapidly in the Protein Data Bank. This, motivates use to study the computational problem of *protein-protein interface* (PPI) alignment.

Given a complex of two protein structures, a PPI is defined by a pair of corresponding binding sites. In addition, two interacting binding sites introduce a set

of interactions defined by non-covalent bonds. Comparison and analysis of physico-chemical properties of PPI families may assist in recognition of certain binding configurations that are responsible for the formation and stability of protein-protein complexes[80, 85]. This, may further assist in developing more specific drugs that target the conserved non-covalent interactions.

Sequential information is not sufficient to annotate protein binding sites and PPIs[128]. The MultiProt method (Chapter 3) was used by Keskin et al.[69] to classify all known PPIs according to their C_α patterns. However, backbone atoms may not be enough to capture the binding properties, thus, the complete residues at the atomic level must be considered[104]. The methods developed for the binding site comparison[128, 104, 70, 117, 101, 16, 66], in general, are not applicable for the PPI comparison, since such methods optimize the similarity of a single binding partner without considering the second partner and the common interactions present in the PPIs. Recently, a method for alignment between a pair of PPIs has been developed[116, 86].

Here we propose and discuss several formulations of the the Largest Common Interface problem between $K \geq 2$ PPI's. Our main motivation is that a feature common to a number of proteins is (probably) functionally more significant than a similar feature found only between a pair of proteins. We present a novel method, MAPPIS, for PPI multiple structural alignment and detection of common interaction patterns shared by the set of PPIs. The computational approach is similar to the MultiBind algorithm (Chapter 5). The implemented method is practically efficient, it takes only minutes on a standard PC.

5.5.1 PPIs Representation and Comparison

Given a complex of two protein structures, a PPI is defined by a pair of corresponding binding sites and by non-covalent bonds created between the binding sites. Specifically, for the binding site representation we use the same model of pseudocenters as described in Section 5.2.1.

Putative Interactions. An *interaction* is defined by a pair of close enough pseudocenters, one from each side of the interface, possessing *complementary* physico-chemical properties. Specifically, hydrogen bond donors are complementary to acceptors, while hydrophobic aliphatic and aromatic properties interact with similar ones. The interaction distance thresholds are 3.9Å [83] for hydrogen bonds and 8Å for interactions between other functional groups. We will denote these interactions as *putative* since in practice, defining the exact non-covalent interactions is not straightforward[83]. Because of this problem, for some pseudocenters, our definition may result in a larger number of non-covalent interactions than the maximal number of theoretically possible (from the perspective of chemistry) interactions. For each pseudocenter we define its maximal number of interactions in which it can be involved, we call this number the pseudocenter interaction *degree*. For the final solution of a common PPI pattern we restrict the number of interactions per pseudocenter to its interaction *degree*. We will address this issue both in the problem formulation and implementation.

Similarity. We define two pseudocenters, a and a', to be similar, $a \sim a'$, if $|a - a'| \leq \varepsilon$ and if they have the same physico-chemical properties, except for mixed donor/acceptor that can be matched to both donor and acceptor. Two interactions $i = (a, b)$ and $i' = (a', b')$ are considered similar, $(a, b) \sim (a', b')$, if the corresponding pseudocenters (a, a') and (b, b') are similar and $|a-b| \leq \varepsilon_i$ and if $|a'-b'| \leq \varepsilon_i$. Below we describe several computational approaches using a general similarity scoring function and propose our own in the Appendix 5.6.2.

Biological Motivation. Given a set of PPIs our goal is to find a set of 3D transformations (superpositions) that maximize the similarity of their physico-chemical and geometrical properties as well as the number of interactions between them. We will refer to a set of features (pseudocenters or interactions) shared by all the compared interfaces as *conserved*. Thus, the goal of an alignment algorithm is to maximize the number of the conserved features. Since there is no consensus regarding the assessment of similarity between PPIs we suggest several alternative criteria for this task.

We define two, biologically relevant, goals:

1. **Maximize the similarity of pseudocenters.**

2. **Maximize the similarity of interactions.**

In general, patterns of conserved interactions are biologically more significant than patterns of pseudocenters conserved in a single binding site[99]. Moreover, the second goal seems to be more suitable for drug design, where the interest is to prevent the formation of protein-protein interactions. However, consideration of the interactions alone is not always sufficient, and there are cases in which the optimal solution may be missed. For example, such a situation may occur, when we align between interfaces created by the same protein with different binding partners. Since the same protein may create slightly different interactions with different binding partners the algorithms that consider only the interactions between the proteins will not necessarily provide a perfect superimposition of the backbones of the same protein. It is not clear which of the two goals is more important. Moreover, there are additional considerations that may be important for the biological analysis:

Optional Requirements:

3. Recognize features conserved in one binding site that are not in direct contact with the binding partner (e.g co-factor binding sites).

4. Recognize correlated mutations that may lead to similar interactions created by different properties (e.g. swapped hydrogen bonds in which donors and acceptors are replaced).

5. Tolerance to flexibility, water mediated interactions and crystallographic artifacts.

It must be noted that it is impossible to achieve all the goals and requirements simultaneously. For example, the similarity optimization according to (1) does not necessary optimize the goal (2) and vice versa. Recognition of correlated mutations (4)

implies tolerance to the differences between the pseudocenters of the aligned binding sites, which immediately contradicts the first goal. Therefore, below, we describe algorithms to achieve each of the proposed goals separately and present the MAPPIS method which attempts to achieve both.

5.5.2 The Largest Common Interface Problem

In this work we extend the LCP problem to the Largest Common Interface problem. The input is K PPIs $\{(A_i, B_i)\}_{i=1}^{K}$. We assume that correspondence between the interface sides (interacting protein chains) is given, i.e. we do not seek for an alignment between A_i and B_j[vii].

Detection of LCP involves two interrelated problems, a transformation search and a matching problem. As we described above, the approximation methods with reasonable time complexity, treat these two sub-problems sequentially. A polynomial size set of 3D transformations is generated by some alignment technique and then for each transformation the similarity measure is calculated. Therefore, in the sequel, for the sake of clear description, we explain the LCP problem for the protein interface alignment assuming that the 3D transformation is given, i.e. there is only the optimization problem of 3D matching between PPIs. As above, we denote this problem $pmSim$. The final approximation ratio depends on the alignment technique and the matching technique.

We denote by $IN^K = \{(a_i, b_i)\}_{i=1}^{K}$ a set of similar interactions and by $PC[A]^K = \{a_i\}_{i=1}^{K}, a_i \in A_i$ ($PC[B]^K = \{b_i\}_{i=1}^{K}, b_i \in B_i$) a set of similar pseudo-centers between K molecules. Here, as in the case of multiple binding site alignment (the **pmSim** problem), we calculate a multiple similarity with respect to a *pivot* interface, i.e. $IN^K = \{(a_i, b_i)\}_{i=1}^{K}$ iff $\forall i = 2...K$ $(a_1, b_1) \sim (a_i, b_i)$, $PC^K = \{p_i\}_{i=1}^{K}$ iff $\forall i = 2...K$ $(p_1) \sim (p_i)$. In the general case we are given similarity scoring functions, $S_{IN}(IN^K[i]) = S_{IN}((a_1, b_1), (a_i, b_i))$ and $S_{PC}(PC^K[i]) = (p_1, p_i)$. We define

[vii]This correspondence can be obtained from the biological data. Otherwise, it can be estimated by running twice the pairwise alignment between (A_1, B_1) - (A_i, B_i) and (A_1, B_1) - (B_i, A_i), for each $i \neq 1$.

the scoring function of multiple alignment to be the minimum[viii] of the scores between the pivot PPI and the rest of PPI's: $S^K = min_{i=2...K} S_i$, where S_i is either $\sum_t S_{IN}(IN^K[i]_t)$ or $\sum_p S_{PC}(PC^K[A][i]_p) + \sum_l S_{PC}(PC^K[B][i]_l)$ or a sum of them. Below we discuss these combinations. Our goal is to maximize S^K.

We extend the original *δ-(γ-additive)-approximation* definition (Section 2.2.1) to *(δ_{pc},δ_{in},γ_{pc}, γ_{in})-approximation*, where δ_{pc} (δ_{in}) is the approximation factor of scoring function S_{PC} (S_{IN}), γ_{pc} is the approximation of the original maximal allowed distance between the matched pseudo-centers (additive approximation to ε) and γ_{in} is the additive approximation to ε_i (the interaction threshold).

Approach I: Similarity of Pseudocenters.

Sum of binding site similarities[116], i.e. $S_i = \sum S_{PC}(PC[A]^K[i]) + \sum S_{PC}(PC[B]^K[i])$. This problem is equivalent to the original *pmSim* problem. Therefore for two structures it is solvable by applying the weighted bipartite matching for each side separately. For the case of several interfaces the problem is NP-Hard (Chapter 2). A (δ_{pc},γ_{pc})-approximation algorithm in this case is the straightforward extension of the MultiBind method (Section 5.2), to a simultaneous alignment of both sides of an interface.

Advantages: Optimizes goal (1) and fulfills optional requirements (3) and (5). In the case of two interfaces the optimal solution can be found in polynomial time.

Disadvantages: Does not consider directly goal (2) and does not fulfill optional requirement (4).

Approach II: Similarity of Interactions.

There are two approaches to solve this problem. In the first approach, we maximize the number of similar interactions, among all the potential interactions that can be created, i.e. $S_i = \sum S_{IN}(IN^K[i])$. Computationally this problem is similar to the first approach, and is solvable by bipartite matching for 2 interfaces (the nodes are interaction pairs) and is NP-Hard for more than two. Notice, that some pseudocenters

[viii] In this definition the similarity score is measured by the distance of the outlier from the pivot.

may participate in several interactions. This approach requires an assumption that the interactions are well defined, i.e. there is no problem of putative interactions.

The second option is to maximize the number of similar *disjoint* interacting pairs, i.e. $S_i = \sum_j S_{IN}(IN^K[i]_j)$, s.t. $\forall t \neq j \; IN^K[i]_t \cap IN^K[i]_j = \emptyset$. For two interfaces this problem is similar to the 3D 4-partite matching problem. Two interfaces define four partitions (A_1, B_1, A_2, B_2). There are interaction edges between (A_1, B_1) and (A_2, B_2), and there are edges between similar type pseudocenters, i.e. between (A_1, A_2) and (B_1, B_2). There are no edges between partitions (A_1, B_2) and (A_2, B_1). The goal of the optimization is to find disjoint 4-tuples that maximize the scoring function. Since these 4-tuples are actually 4-node circles (there are no edges between (A_1, B_2) and (A_2, B_1)), this problem may appear more simple than the 3D 4-partite matching. However, the ε-3-partite matching can be trivially reduced to the two interface matching, by splitting each vertex of the first partition into two vertices and connecting them by an edge. Therefore, maximizing the number of similar *disjoint* interacting pairs is NP-Hard even for K=2. Here, we propose two simple approximation algorithms to solve the matching problem for 2 interfaces. The first algorithm, in case $S_{IN} = constant$, is ($\delta_{in} = 1$,$\gamma_{pc} = 0$,$\gamma_{in} = 2\varepsilon$)-approximation. Consider the network flow depicted in Figure 5.11 (a). Edges between the nodes of the partitions (A_1, A_2) and (B_2, B_1) are created iff the nodes are similar and the distance between them is less than ε. Edges between (A_2, B_2) are created iff the nodes are chemically complementary and the distance between them is less than ε_i. All edges have one unit capacity. The maximum flow capacity of each node of A_2 and B_2 is set to its corresponding interaction *degree*. Therefore, the largest flow in this network will give the optimal solution with ε_i approximated by $\varepsilon_i + 2\varepsilon$.

Using a similar greedy approach as in Section5.2.2 we can obtain a (δ_{in}= $min(2K, 2d)$,$\gamma_{pc} = 0$,$\gamma_{in} = 0$)-approximation algorithm, where d is the maximal point density in a sphere of radius 2ε. The δ_{in} factor is increased twice, comparing to the point set matching, since each greedy selection of an interaction edge can intersect with two edges from an optimal matching.

Advantages: Optimizes goal (2) and fulfills (4). Approximate solution for two interfaces.

Disadvantages: Does not treat (1), (3) and (5).

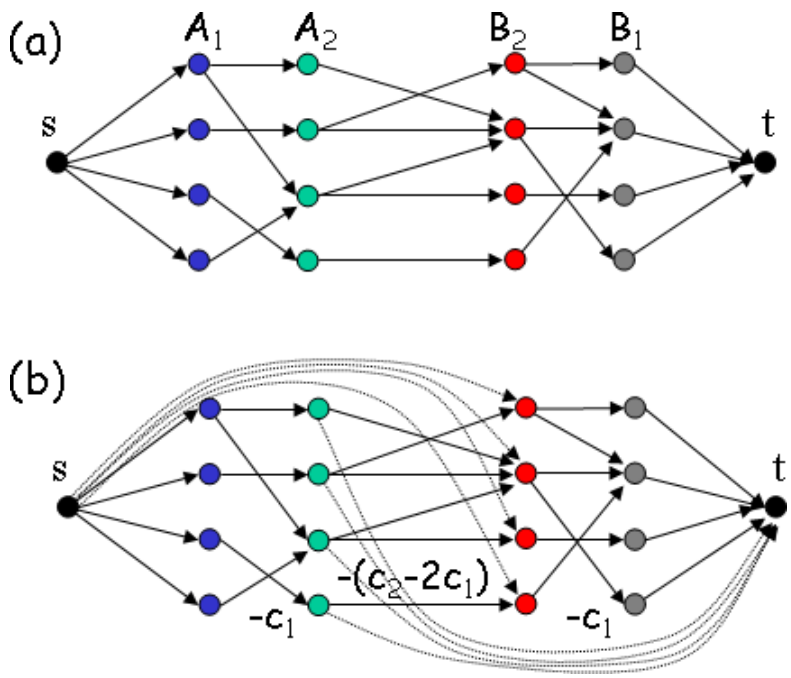


Figure 5.11: Flow networks: (a) Approach-II-b, (b) Approach-III.

Approach III: Similarity of Interactions and PseudoCenters.

$S_i = \sum_t S_{IN}(IN^K[i]_t) + \sum_p S_{PC}(PC^K[A][i]_p) + \sum_l S_{PC}(PC^K[B][i]_l)$. The problem is NP-Hard even for two interfaces since it is an extension of Approach II-b. As above, for $K = 2$, we propose two polynomial time approximation algorithms. The first algorithm, for the case $S_{PC} = c_1$, $S_{IN} = c_2$, where c_1, c_2 are integer constants, is a ($\delta_{in} = \delta_{pc} = 1$,$\gamma_{pc} = 0$,$\gamma_{in} = 2\varepsilon_i$)-approximation. Consider the network depicted in Figure 5.11 (b). The structure of the network from Approach II-b is extended as follows. We add edges $\{(s, b) : \forall b \in B_2\}$ and $\{(a, t) : \forall a \in A_2\}$. We assign cost $-c_1$ for each edge between (A_1, A_2) and for each edge between (B_2, B_1). For edges between (A_2, B_2) we assign cost $-(c_2 - 2c_1)$. The rest of the edges have zero cost. All the edges have one unit capacity. Therefore the min-cost-max-flow in this network will maximize the above scoring function (the same method is also applicable in case S_{PC} is an integer function and $S_{IN}((a, b), (a`, b`)) = c_2 + S_{PC}(a, a') + S_{PC}(b, b')$). It

is not clear how to extend this method to K PPI's with less than $\gamma_{pc}, \gamma_{in} \in O(K\varepsilon)$ approximation. Such approximation, $O(K\varepsilon)$, will produce biologically meaningless results. Therefore the greedy approach represents a better choice. We apply two stage greedy selection. First, we greedily select a matching of interaction edges, then, for each interface side separately, we greedily select a matching of pseudocenters. This results in $(\delta_{in} = min(2K, 2d), \delta_{pc} = min(K, d), \gamma_{pc} = \gamma_{in} = 0)$-approximation, where d is $dens(2\varepsilon)$.

Additional advantage of this greedy approach is that it allows us to overcome the problem of putative interactions, since the interaction *degree* is easily incorporated into iterative greedy selection. The approximation factors still hold.

Advantages: Optimizes aim (1) and (2) and fulfills (3) and (5). Approximate solution for two interfaces.

Disadvantages: Does not treat (4).

5.5.3 MAPPIS Algorithm

In our implementation we solve the problem defined in *Approach III*. We use the model of the **pmLCP** problem where the scoring function is the minimum of the scores between the pivot PPI and the rest of PPI's: $S^K = min_{i=2...K} S_i$. The implemented method is similar to the MultiBind algorithm. First we generate a polynomial number of transformations with approximation error $\varepsilon + \gamma$. Then, we apply the *branch-and-bound* technique to effectively filter out the low scoring multiple alignments. Third, we apply the greedy method of *Approach III* to solve the multiple matching problem. The scoring functions S_{IN} and S_{PC} are defined in the Appendix 5.6.2.

The time complexity depends mainly on the second combinatorial stage. Assume that the maximal depth of the filtering iterations is $K' \leq K$. Therefore, the time complexity $O(n^{3K'} nK\ log(n))$. The major advantages of our technique are (1) polynomial time approximation algorithm for two PPI's, (2) polynomial space for any K, (3) in practice the method quickly detects a high scoring solution and the exponential number of iterations is avoided (as the input structures are more similar the *bound* filter is more effective, thus $K' << K$). The practical running times are low

as reported in the Results section.

Results

Here we present examples of application of MAPPIS for recognition of physico-chemical properties and interactions shared by different groups of interfaces (following Approach III). We have analyzed five different families of PPIs.

Serine proteases include three Trypsin-like serine proteases (4sgb, 1ppf, 1acb) and three Subtilisin-like (1cse, 2sic, 1oyv).

Two PPI clusters, **Cluster 99** (1l3bAD, 1l0oAB, 1b99AD, 1e7pAD, 1gttBC, 1iunAB) and **Cluster 673** (TNF (tumor necrosis factor) family: 1i9rAB,1jh5AB, 1d0gAB, 1a8mAB), were taken from the PPI data set composed by Mintz et al.[86].

G proteins with GAPs is a set PIIs of G proteins which are regulated by GTPase activating proteins (GAPs) includes GAPs from two different folds, GTPase activation domain, type p50 RhoGAP (1tx4, 1ow3, 1am4, 1grn, 2ngr) and Four-helical up-and-down bundle (PDB: 1he1, 1g4u).

G proteins with GEFs is a set PIIs of G proteins which are regulated by Guanine nucleotide Exchange Factors (GEFs) includes GEFs from two different SCOP folds, DBL homology domain (1lb1, 1foe, 1kz7, 1ki1) and GEF domain of SopE toxin (1gzs)[34].

The results produced by MAPPIS are given in Table 5.4 and in Figure 5.12. Only a solution with the highest score is considered. The running time as a function of PPI number is discussed in Figure 5.13. The running times are measured on a standard PC, Intel(R) Pentium(R) IV 2.60GHz CPU with 2GB RAM.

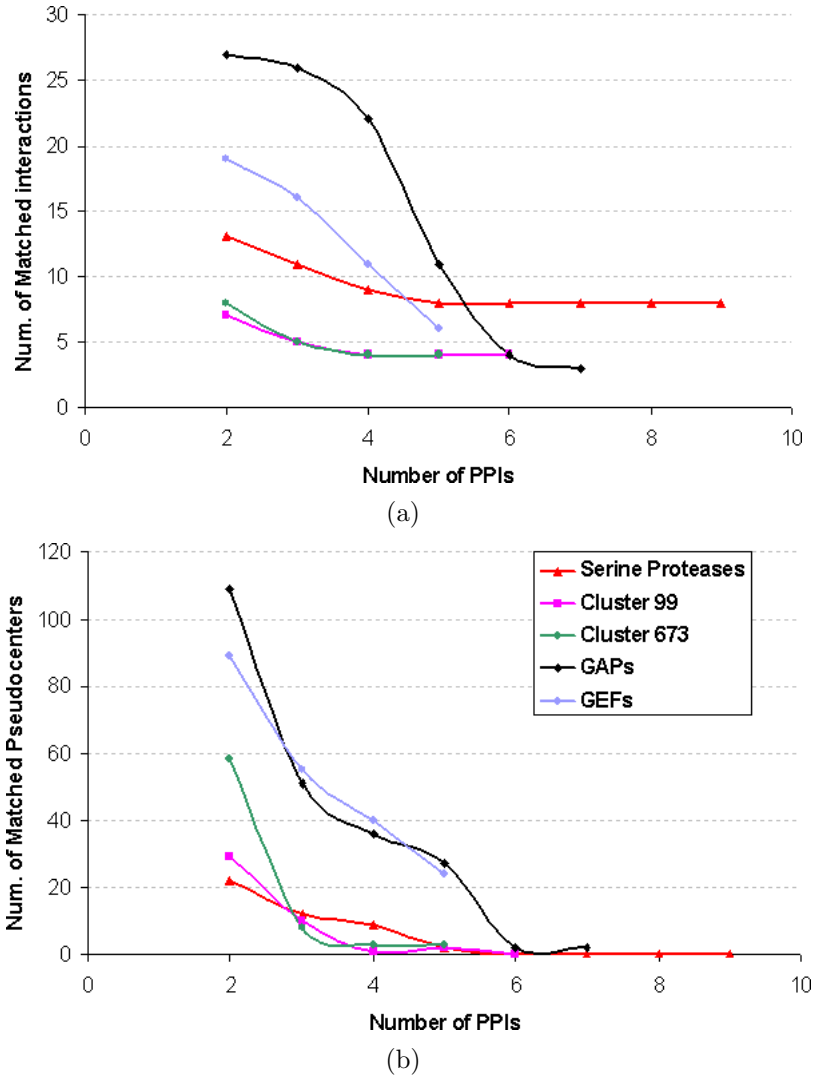


Figure 5.12: (a) Number of matched interactions (see legend (b)). (b) Number of matched pseudocenters on both interface sides (does not include the matched interactions).

Case study	Num. of PPIs	Mean PPI size	Num. of common interactions:pseudocenters	Run time (sec.)
Serine Proteases		120		
4sgbEI,1oyvBI,1cseEI	3		11 : 12	31
4sgbEI,1oyvBI,1cseEI,2sicEI	4		9 : 9	71
4sgbEI,1oyvBI,1cseEI,2sicEI,1acbEI	5		8: 2	84
4sgbEI,1oyvBI,1cseEI,2sicEI,1acbEI,1ppfEI	6		8: 0	200
Cluster 99[86]		120		
1l0oAB,1b99A,1e7pAD	3		5 : 10	10
1l0oAB,1b99A,1e7pAD,1gttBC	4		4 : 1	12
1l0oAB,1b99A,1e7pAD,1gttBC,1iunAB	5		4 : 2	35
1l0oAB,1b99A,1e7pAD,1gttBC,1iunAB,1l3bAD	6		4 : 0	44
Cluster 673[86], TNF Family		165		
1a8mAB,1d0gAB,1i9rAB	3		5 : 8	36
1a8mAB,1d0gAB,1i9rAB,1iqaAB	4		4 : 3	49
1a8mAB,1d0gAB,1i9rAB,1iqaAB,1jh5AB	5		4 : 3	62
1a8mAB,1d0gAB,1i9rAB,1iqaAB,1jh5AB,1iqaAB	6		4 : 3	80
G-proteins with GAPs		177		
1tx4,1ow3, 1am4, 1grn, 2ngr,1he1, 1g4u	7		3:2	155
G-proteins with GEFs		124		
1lb1, 1foe, 1kz7,1ki1, 1gzs	5		6 : 24	54

Table 5.4: **Performance evaluation.**

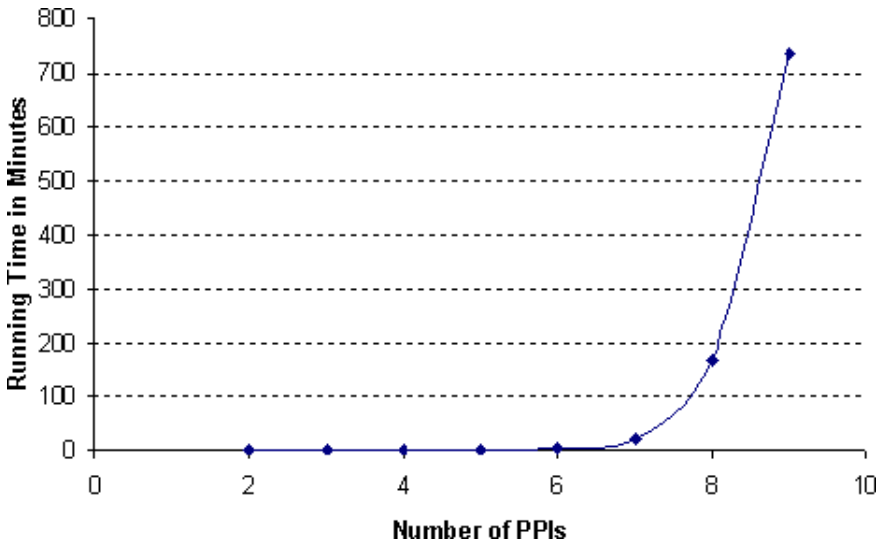


Figure 5.13: Running time on PPIs of Serine Proteases. When the number of common features becomes too small the branch-and-bound method becomes less effective, thus, the running time begins to behave exponentially in the number of PPIs.

5.5.4 Summary and Conclusions

Here we presented a novel computational method, MAPPIS, for recognition of patterns of interactions shared by a set of protein-protein interfaces (PPIs). We discussed several approaches to the problem with the related computational aspects. We presented the MAPPIS method with application to several biological case studies. The results of MAPPIS on examples which were previously analyzed by other sources, are consistent with the available biological data.

To the best of our knowledge, MAPPIS is the first method that performs multiple structural alignment of PPIs in the absence of overall sequence or fold homology. It allows recognition and analysis of the conserved binding organizations that are shared by different protein families. These are important for drug discovery as well as functional annotations. However, there is still a lack of biological understanding in assessment of similarity between PPIs. We hope that this work will stimulate additional research in the field. Additional issue, that has to be addressed in future research, is the estimation of the biological significance of the recognized patterns.

Specifically, it is unclear what is the minimal number, combination, and type of interactions that are responsible for the similarity in binding and function. Consequently, there is a difficulty in the discrimination between randomly obtained patterns and a small set of essential interactions that are required for the desired biological effect. Another issue is the extent of the flexibility and variability of interactions that should be incorporated in the algorithm. Finally, the scoring function, is known to be one of the major problems in all types of *in silico* predictions, like protein folding, docking, pharmacophore detection etc.[53].

5.6 Appendices

5.6.1 Appendix: MultiBind Physico-Chemical Scoring

Let p and q be the two matched pseudocenters.

• $dist(p, q)$ - the distance between p and q after the superimposition.

• $max\_dist(p, q)$ - maximal allowed distance between a pair of pseudocenters, by default defined by ε =3.0Å.

• $chem(p)$ - the physico-chemical property of the point p. There are three types of properties: Hydrogen Bonding (HB), Aliphatic Hydrophobic (ALI) and Aromatic (PII).

• $charge(p)$ - the partial atomic charge of the atom p, which can form hydrogen bonds. $charge(p, q) = |charge(p) - charge(q)|$ - measures the similarity of charges.

• $shape(p)$ - the average curvature of the surface region created by p. Calculated as an average of the solid angle shape functions[25] with spheres of radius 4,5,6 and 7Å . The sphere centers are located at projection point of p to the surface. $shape(p, q) = |shape(p) - shape(q)|$.

• $n_S(p)$ - normal vector at projection point of p to the surface, $n_S(p, q) = n_S(p) \cdot n_S(q)$.

• $n_{PII}(p)$ - for aromatic pseudocenters denotes the normal to the plane of the aromatic ring. $n_{PII}(p, q) = n_{PII}(p) \cdot n_{PII}(q)$.

• $v_{ALI}(p, q)$ - the overlap of the hydrophobic group spheres of p and q, approximated by the difference between sum of radiuses and the distance between the centers.

Each pair of matched pseudocenters is assigned a score according to the similarity of the properties important for the specific type of interaction:

$$PC\_Score\ (p,q) = \begin{cases} 0, & dist(p,q) > max\_dist(p,q)\ or\ chem(p) \neq chem(q) \\ 0, & shape(p,q) > 0.2\ or\ n_S(p,q) > 0.2 \\ (max\_dist(p,q) - dist(p,q))/(1 + charge(p,q)) & chem(p) = HB \\ (max\_dist(p,q) - dist(p,q))/(1 + shape(p,q) + n_{PII}(p,q)) & chem(p) = PII \\ (max\_dist(p,q) - dist(p,q) + v_{ALI}(p,q))/(2 + 20 \cdot shape(p,q)) & chem(p) = ALI \end{cases}$$

5.6.2 Appendix: MAPPIS Physico-Chemical Scoring

Similarity between two superimposed interactions $i = (a, b)$ and $i' = (a', b')$ is measured by:

$$S(i, i') = I\_Score(i) + I\_Score(i') + PC\_Score(a, a') + PC\_Score(b, b')$$

Where the similarity between two superimposed interactions is defined by:

$$I\_Score\ (i) = \begin{cases} 0, & dist(i) > max\_dist(i) \\ (max\_dist(i) - dist(i))/(1 + charge\_comp(i)) & chem(i) = HB \\ (max\_dist(i) - dist(i))/(1 + shape\_comp(i)) & chem(i) = ALI \\ (max\_dist(i) - dist(i))/(1 + shape\_comp(i) + n_{PII}(i)) & chem(i) = PII \end{cases}$$

The $PC\_Score(a, b)$ is defined in the above Appendix.

• $dist(i = (a, b))$ - the distance between interacting pseudocenters a and b.

• $chem(i)$ - the physico-chemical property of the interaction i. There are three types of properties: Hydrogen Bonding (HB), Aliphatic Hydrophobic (ALI) and Aromatic

(PII).

• $max\_dist(i)$ - the maximal distance allowed for the specific type of interaction. The default thresholds are ε_i =3.9Å for hydrogen bonds[83] and ε_i =8.0Å for hydrophobic aliphatic and aromatic interactions.

• $charge\_comp(i) = |charge(a) + charge(b)|$ - measures the complementarity of charges.

• $shape\_comp(i = (a, b)) = |1 - shape(a) - shape(b)|$ - measures the complementarity of shapes which sums to one.

• $n_{PII}(i = (a, b)) = n_{PII}(a) \cdot n_{PII}(b)$ - represents the angle between two interacting aromatic ring.

Bibliography

[1] T. Akutsu. Protein structure alignment using dynamic programming and iterative improvement. *IEICE Trans. Information and Systems*, E79-D:1629–1636, 1996.

[2] T. Akutsu and M. M. Halldorson. On the approximation of largest common subtrees and largest common point sets. *Theoretical Computer Science*, 233:33–50, 2000.

[3] T. Akutsu and K.L. Sim. Protein threading based on multiple protein structure alignment. In *Genome Informatics (GIW'99)*, pages 23–29, 1999.

[4] V. Alesker, R. Nussinov, and H.J. Wolfson. Detection of non-topological motifs in protein structures. *Protein Engineering*, 9:1103–1119, 1996.

[5] Nickolai N. Alexandrov and Daniel Fischer. Analysis of topological and non-topological structural similarities in the PDB: New examples with old structures. *Proteins: Structure, Function and Bioinformatics*, 25:354–365, 1996.

[6] S.F. Altschul, T.L. Madden, A.A. Schffer, J. Zhand 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, 1997.

[7] Christoph Ambuhl, Samarjit Chakraborty, and Bernd Gartner. Computing largest common point sets under approximate congruence. In *Proceedings of the 8th Annual European Symposium on Algorithms*, pages 52–63. Springer-Verlag, 2000.

[8] P. J. Artymiuk, A. R. Poirrette, H. M. Grindley, D. W. Rice, and P. Willett. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *J. Mol. Biol.*, 243:327–344, 1994.

[9] P.J. Artymiuk, D.W. Rice, E.M. Mitchell, and P. Willett. Structural resemblance between the families of bacterial signal- transduction proteins and of G proteins revealed by graph theoretical techniques. *Protein Engineering*, 4:39–43, 1990.

[10] A.S. Aytuna, A. Gursoy, and O. Keskin. Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces. *Bioinformatics*, 21(12):2850–2855, 2005.

[11] O. Bachar, D. Fischer, R. Nussinov, and H.J. Wolfson. A Computer Vision based technique for 3-D Sequence Independent Structural Comparison. *Protein Engineering*, 6:279–288, 1993.

[12] D. Bashford, C. Chothia, and A.M. Lesk. Determinants of a protein fold. Unique features of the globin amino acid sequences. *J Mol Biol.*, 196:199–216, 1987.

[13] H. Benyamini, H.J. Wolfson, A. Shulman-Peleg, B. Belgorodsky, L. Fadeev, and M. Gozin. Interaction of C_60 -Fullerene and Carboxyfullerene with Proteins: Docking and Binding Site Alignment. *Bioconjugate Chemistry*, 2006. `http://dx.doi.org/10.1021/bc050299g`.

[14] 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, 2000.

[15] P. Berman and T. Fujito. the approximation properties of independent set problem in degree 3 graphs, 1995.

[16] T. Binkowski, L. Adamian, and J. Liang. Inferring functional relationship of proteins from local sequence and spatial surface patterns. *J. Mol. Biol.*, 232:505–526, 2003.

[17] Orhan Camoglu, Tamer Kahveci, and Ambuj K. Singh. Psi: indexing protein structures for fast similarity search. *Bioinformatics*, 19 Suppl. 1:i81–i83, 2003.

[18] Samarjit Chakraborty and Somenath Biswas. Approximation algorithms for 3-d commom substructure identification in drug and protein molecules. In *Proc. 6th Int. Workshop on Algorithms and Data Structures*, pages 253–264, Vancouver, Canada, 1999. Springer-Verlang.

[19] J. M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and SE. Brenner. The astral compendium in 2004. *Nucleic Acids Res.*, 32:189–192, 2004.

[20] J.M. Chandonia, G. Hon, N.S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S.E. Brenner. The astral compendium in 2004. *Nucleic Acids Res.*, 32:D189–D192, 2004.

[21] B. Y. Chen, V. Y. Fofanov, D. M. Kristensen, M. Kimmel, O. Lichtarge, and L. E. Kavraki. Algorithms for structural comparison and statistical analysis of 3d protein motifs. In *In Pacific Symposium on Biocomputing*, pages 334–345, Hawaii, USA, 2005. World Scientific.

[22] J.L. Chung, W. Wang, and P.E. Bourne. Exploiting sequence and structure homologs to identify protein-protein binding sites. *Proteins: Structure, Function and Bioinformatics*, Epub ahead of print., 2005.

[23] Matteo Comin, Concettina Guerra, and Giuseppe Zanotti. Proust: A comparison method of three-dimensional structures of proteins using indexing techniques. *Journal of Computational Biology*, 11(6):1061–1072, 2004.

[24] M. L. Connolly. Analytical molecular surface calculation. *J. Appl. Cryst.*, 16:548–558, 1983.

[25] M. L. Connolly. Measurement of protein surfaces shape by solid angles. *J. Mol. Graph.*, 4:3–6, 1986.

[26] M. Dawande, P. Keskinocak, J.M. Swaminathan, and S. Tayur. On bipartite and multipartite clique problems. *J. Algorithms*, 41(2):388–403, 2001.

[27] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry - Algorithms and Applications*. Springer-Verlag, 2000.

[28] K. A. Denessiouk, V. Rantanen, and M.S. Johnson. Adenine Recognition: A motif present in ATP-,CoA-,NAD-,NADP-, and FAD-dependent proteins. *Proteins: Structure, Function and Bioinformatics*, 44:282–291, 2001.

[29] S. Dietmann, J. Park, C. Notredame, A. Heger, M. Lappe, and L. Holm. A fully automatic evolutionary classification of protein folds: dali domain dictionary version 3. *Nucleic Acids Res*, 29(1):55–57, 2001. http://www.embl-ebi.ac.uk/dali/.

[30] O. Dror, H. Benyamini, R. Nussinov, and H. Wolfson. Multiple structural alignment by secondary structures: – algorithm and applications. *Protein Science*, 12:2492–2507, 2003.

[31] O. Dror, H. Benyamini, R. Nussinov, and H.J. Wolfson. MASS: multiple structural alignment by secondary structures. *Bioinformatics*, 19 Suppl. 1:i95–i104, 2003.

[32] O. Dror, A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson. Predicting molecular interactions in silico: I. A guide to pharmacophore identification and its applications for drug design. *Curr. Med. Chem.*, 11:71–90, 2004.

[33] D. Duhovny, R. Nussinov, and H.J. Wolfson. Efficient unbound docking of rigid molecules. In R. Guigo and D. Gusfield, editors, *Workshop on Algorithms in Bioinformatics*, pages 185–200, Rome, Italy, 2002. Springer Verlag. Lecture Notes in Computer Science 2452.

[34] R. Dvorsky and M. R. Ahmadian. Always look on the bright site of Rho: structural implications for a conserved intermolecular interface. *EMBO Rep.*, 5:1130–6, 2004.

[35] M.E. Dyer and A.M. Frieze. Planar 3DM is NP-complete. *J. Algorithms*, 7:174–184, 1986.

[36] R.C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, 32(5):1792–1797, 2004.

[37] Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.

[38] I. Eidhammer, I. Jonassen, and WR. Taylor. Structure Comparison and Structure Patterns. *J Comput Biol.*, 7:685–716, 2000.

[39] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann, and A. Bairoch. The PROSITE database, its status in 2002. *Nucleic Acids Res.*, 30:235–238, 2002.

[40] P. W. Finn, L. E. Kavraki, J.-C. Latombe, R. Motwani, C. Shelton, S. Venkatasubramanian, and A. Yao. Rapid: randomized pharmacophore identification for drug design. *Comput. Geom. Theory Appl.*, 10(4):263–272, 1998.

[41] D. Fischer, A. Elofsson, D. Rice, and D. Eisenberg. Assessing the performance of fold recognition methods by means of a comprehensive benchmark. In L. Hunter and T. Klein, editors, *In Proc. Pacific Symposium on Biocomputing*, Singapore, 1996. World Scientific Press.

[42] J.D. Forman-Kay and T. Pawson. Diversity in protein recognition by ptb domains. *Curr Opin Struct Biol*, 9:690–695, 1999.

[43] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman, San Francisco, 1979.

[44] Martin Gavrilov, Piotr Indyk, Rajeev Motwani, and Suresh Venkatasubramanian. Combinatorial and experimental methods for approximate point pattern matching. *Algorithmica*, 38:59–90, 2004.

[45] M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In *in Proceedings of the Fourth International Conference on Intelligent Systems in Molecular Biology*, pages 59–67, Heidleberg, Germany, 1996. Menlo Park, CA, AAAI press.

[46] F. Glaser, T. Pupko, I. Paz, D. Bechor, E. Martz, and N. Ben-Tal. Consurf: a server for the identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics*, 19(1):163–164, 2003. `http://consurf.tau.ac.il`.

[47] S. Goldsmith-Fischman and B. Honig. Structural genomics: computational methods for structure analysis. *Protein Science*, 12(9):1813–1821, 2003.

[48] Michael T. Goodrich, Joseph S. B. Mitchell, and Mark W. Orletsky. Practical methods for approximate geometric pattern matching under rigid motions: (preliminary version). In *Proceedings of the tenth annual symposium on Computational geometry*, pages 103–112. ACM Press, 1994.

[49] H.M. Grindley, P.J Artymiuk, D.W Rice, and P. Willett. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.*, 229:707–721, 1993.

[50] C. Guda, E.D. Scheeff, P.E. Bourne, and I.N. Shindyalov. A new algorithm for the alignment of multiple protein structures using Monte Carlo optimization. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 275–286, 2001. online available on `http://cemc.sdsc.edu`.

[51] C. Guerra, S. Lonardi, and G. Zanotti. Analysis of secondary structure elements of proteins using indexing techniques. In *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)*, pages 812–823, 2002.

[52] D. Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 529–562. CRC Press LLC, Boca Raton, FL, 2004.

[53] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function and Bioinformatics*, 47:409–443, 2002.

[54] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *FOCS '96: Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, page 627, Washington, DC, USA, 1996. IEEE Computer Society.

[55] E. Hazan, S. Safra, and O. Schwartz. On the Complexity of Approximating k-Dimensional Matching. In *Approximation, Randomization, and Combinatorial Optimization*, volume 2764 of *LNCS*, pages 83–97. Springer, 2003.

[56] Paul J. Heffernan and Stefan Schirra. Approximate decision algorithms for point set congruence. *Comput. Geom. Theory Appl.*, 4(3):137–156, 1994.

[57] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89(22):10915–9, 1992.

[58] D. Higgins, J. Thompson, and T. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22:4673–4680, 1994.

[59] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, 1993.

[60] L. Holm and C. Sander. 3-D lookup: Fast protein structure database searches at 90% reliability. In Christopher J. Rawlings, Dominic A. Clark, Russ B. Altman, Lawrence Hunter, Thomas Lengauer, and Shoshana J. Wodak, editors, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 179–187, Menlo Park, California, 1995. The AAAI press.

[61] J.E. Hopcroft and R.M. Karp. A n 2.5 Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[62] S.R. Hubbard and J. H. Till. Protein tyrosine kinase structure and function. *Annual Review of Biochemistry*, 69:373–398, 2000.

[63] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discret. Math.*, 2(1):68–72, 1989.

[64] D.P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[65] I.Koch, T.Lengauer, and E.Wanke. An algorithm for finding maximal common subtopologies in a set of proteins. *Journal of Computational Biology*, 3(2):289–306, 1996.

[66] M. Jambon, A. Imberty, G. Deleage, and C. Geourjon. A new bioinformatic approach to detect common 3d sites in protein structures. *Proteins: Structure, Function and Bioinformatics*, 52:137–145, 2003.

[67] I. Jonassen, I. Eidhammer, D. Conklin, and W.R. Taylor. Structure motif discovery and mining the pdb. *Bioinformatics*, 18(2):362–367, 2002.

[68] Viggo Kann. Maximum bounded 3-dimensional matching is max snp-complete. *Inf. Process. Lett.*, 37(1):27–35, 1991.

[69] A. Keskin, C. H. Tsai, H. J. Wolfson, and R. Nussinov. A new, structurally non-reduntant, diverse dataset of protein-protein interfaces and its implications. *Prot. Sci.*, 13(4):1043–55, 2004.

[70] K. Kinoshita and H. Nakamura. Identification of protein biochemical functions by similarity search using the molecular surface database ef-site. *Protein Science*, 12:1589–1595, 2003.

[71] K. Kinoshita and H. Nakamura. Identification of the ligand binding sites on the molecular surface of proteins. *Protein Science*, 14:711–718, 2005.

[72] R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. *Proc. Natl. Acad. Sci. USA*, 101(33):12201–12206, 2004.

[73] M. Kosloff and Z. Selinger. Substrate assisted catalysis - application to g proteins. *Trends Biochem Sci.*, 26:161–166, 2001.

[74] Y. Y. Kuttner, V. Sobolev, A. Raskind, and M. Edelman. A consensus-binding structure for adenine at the atomic level permits searching for the ligand site in a wide spectrum of adenine-containing complexes. *Proteins: Structure, Function and Bioinformatics*, 52:400–411, 2003.

[75] N. Leibowitz, Z.Y. Fligelman, R. Nussinov, and H.J. Wolfson. Automated multiple structure alignment and detection of a common substructural motif. *Proteins*, 43:235–45, 2001.

[76] N. Leibowitz, R. Nussinov, and H.J. Wolfson. MUSTA-a general, efficient, automated method for multiple structure alignment and detection of common motifs: application to proteins. *J Comput Biol.*, 8:93–121, 2001.

[77] C. Lemmen and T. Lengauer. Computational methods for the structural alignment of molecules. *J. of Computer-Aided Mol. Design*, 14:215–232, 2000.

[78] O. Lichtarge, H.R. Bourne, and F.E. Cohen. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.*, 257(2):342–358, 1996.

[79] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[80] L. Lo Conte, C. Chothia, and J. Janin. The atomic structure of protein-protein recognition sites. *J. Mol. Biol.*, 285:2177–2198, 1999.

[81] B. Ma, T. Elkayam, H.J. Wolfson, and R. Nussinov. Protein-protein interactions: Structurally conserved residues distinguish between binding sites and exposed protein surfaces. *Proc. Natl. Acad. Sci. USA*, 100(10):5772–5777, 2003.

[82] T. Madej, J.F. Gibrat, and S.H. Bryant. Threading a database of protein cores. *Proteins*, 23:356–369, 1995. Online available on `http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml`.

[83] I. K. McDonald and J. M. Thornton. Satisfying hydrogen bonding potential in proteins. *J. Mol. Biol.*, 238:777–793, 1994.

[84] K. Mehlhorn. *The LEDA Platform of Combinatorial and Geometric Computing.* Cambridge University Press, 1999.

[85] J. Mintseris and Z. Weng. Atomic contact vectors in protein-protein recognition. *Proteins: Structure, Function and Bioinformatics*, 53:629–639, 2003.

[86] S. Mintz, A. Shulman-Peleg, H. J. Wolfson, and R. Nussinov. Generation and analysis of a protein-protein interface dataset with similar chemical and spatial patterns of interactions. *Proteins: Structure, Function and Bioinformatics*, 61:6–20, 2005.

[87] L.A. Mirny and E.I. Shakhnovich. Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function. *JMB*, 291(1):177–96, 1999.

[88] E.M. Mitchel, P.J. Artymiuk, D.W. Rice, and P. Willet. Use of Techniques Derived from Graph Theory to Compare Secondary Structure Motifs in Proteins. *J. Mol. Biol.*, 212:151–166, 1989.

[89] K. Mizuguchi, C.M. Deane, T.L. Blundell, and J.P. Overington. Homstrad: a database of protein structure alignments for homologous families. *Protein Science*, 7:2469–2471, 1998.

[90] K. Mizuguchi and N. Go. Comparison of spatial arrangements of secondary structural elements in proteins. *Protein Eng.*, 8:353–362, 1995.

[91] S. L. Moodie, J. B. O. Mitchell, and J. M. Thornton. Protein recognition of adenylate: An example of a fuzzy recognition template. *J. Mol. Biol.*, 263:486–500, 1996.

[92] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540, 1995.

[93] R. Nussinov and H.J. Wolfson. Efficient Detection of three-dimensional strcutral motifs in biological macromolecules by computer vision techniques. *Proc. Natl. Acad. Sci. USA*, 88:10495–10499, 1991.

[94] M.E. Ochagavia and Wodak S. Progressive combinatorial algorithm for multiple structural alignments: application to distantly related proteins. *PROT*, 55(2):436–454, 2004.

[95] C.A. Orengo. CORA-topological fingerprints for protein structural families. *Protein Science*, 8(4):699–715, 1999.

[96] C.A. Orengo, A.D. Michie, S. Jones, M.B. Swindells, and Thornton J.M. CATH - a Hierarchic Classification of Protein Domain Structure. *Structure*, 5(8):1093–1108, 1997.

[97] O. O'Sullivan, K. Suhre, C. Abergel, D.G. Higgins, and C. Notredame. 3dcoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol.*, 340(2):385–95, 2004.

[98] R. Peeters. The maximum edge biclique problem is NP-complete. *Tilburg University Department of Econometrics, Research Memorandum series*, 789, 2000.

[99] D. Reichmann, O. Rahat, S. Albeck, R. Meged, O. Dym, and G. Schreiber. The modular architecture of protein-protein binding interfaces. *Proc. Natl. Acad. Sci. USA*, 102:57–62, 2005.

[100] S.D. Rufino and T.L. Blundell. Structure-based identification and clustering of protein families and superfamilies. *J. of Computer-Aided Mol. Design*, 8:5–27, 1994.

[101] R.B. Russell. Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. *J. Mol. Biol.*, 279(5):1211–1227, 1998.

[102] R.B. Russell and G.J. Barton. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins: Structure, Function and Bioinformatics*, 14:309–323, 1992.

[103] A. Sali and T.L. Blundell. Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *JMB*, 212:403–428, 1990.

[104] S. Schmitt, D. Kuhn, and G. Klebe. A new method to detect related function among proteins independent of sequence or fold homology. *J. Mol. Biol.*, 323:387–406, 2002.

[105] M. Shatsky, O. Dror, D. Schneidman-Duhovny, R. Nussinov, and H.J. Wolfson. BioInfo3D: A suite of tools for structural bioinformatics. *Nucleic Acids Res.*, 32:503–7, 2004.

[106] M. Shatsky, Z.Y. Fligelman, R. Nussinov, and H.J. Wolfson. Alignment of Flexible Protein Structures. In *8th International Conference on Intelligent Systems for Molecular Biology*, pages 329–343. The AAAI press, 2000.

[107] M. Shatsky, R. Nussinov, and H.J. Wolfson. MultiProt - a multiple protein structural alignment algorithm. In R. Guigo and D. Gusfield, editors, *Workshop on Algorithms in Bioinformatics*, pages 235–250, Rome, Italy, 2002. Springer Verlag. Lecture Notes in Computer Science 2452.

[108] M. Shatsky, R. Nussinov, and H.J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins: Structure, Function, and Genetics*, 56(1):143–156, 2004.

[109] M. Shatsky, R. Nussinov, and H.J. Wolfson. *Protein Structure Prediction: Methods and Protocols 2nd ed.*, chapter Algorithms for Multiple Protein Structure Alignment and Structure-Derived Multiple Sequence Alignment. Humana, 2006. (to appear).

[110] M. Shatsky, R. Nussinov, and H.J. T Wolfson. Optimization of Multiple Sequence Alignment Based on Multiple Structure Alignment. *Proteins: Structure, Function and Bioinformatics*, 62(1):209–217, 2006.

[111] M. Shatsky, A. Shulman-Peleg, R. Nussinov, and H.J. Wolfson. Recognition of binding patterns common to a set of protein structures. In S. Miyano, editor, *RECOMB 2005, Cambridge MA*, volume 3500, pages 440–455. LNCS, 2005.

[112] M. Shatsky, A. Shulman-Peleg, R. Nussinov, and H.J. Wolfson. The multiple common point set problem and its application to molecule binding pattern detection. *Journal of Computational Biology*, 13(2):407–428, 2006.

[113] M. Shatsky, H.J. Wolfson, and R. Nussinov. Flexible protein alignment and hinge detection. *Proteins: Structure, Function, and Genetics*, 48:242–256, 2002.

[114] Edward S.C. Shih and Ming-Jing Hwang. Protein structure comparison by probability-based matching of secondary structure elements. *Bioinformatics*, 19:735–741, 2003.

[115] I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorical extension (ce) of the optimal path. *Protein Engineering*, 11(9):739–747, 1998. online available on `http://cl.sdsc.edu/ce.html`.

[116] A. Shulman-Peleg, S. Mintz, R. Nussinov, and H.J. Wolfson. Protein-protein interfaces: Recognition of similar spatial and chemical organizations. In I. Jonassen and J. Kim, editors, *Workshop on Algorithms in Bioinformatics*, pages 194–205. Springer Verlag, 2004. LNCS, 3240.

[117] A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson. Recognition of functional sites in protein structures. *J. Mol. Biol.*, 339(3):607–633, 2004. `http://bioinfo3d.cs.tau.ac.il/SiteEngine/`.

[118] A. Shulman-Peleg, M. Shatsky, , R. Nussinov, and H.J. Wolfson. MAPPIS: Multiple 3D Alignment of Protein-Protein Interfaces. In *Complife, Konstanz, Germany, September 25-27*. LNCS, 2005.

[119] A. P. Singh and D. L. Brutlag. Hierarchical protein structure alignment using both secondary structure and atomic representations. In *Fifth Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'97)*, pages 284–293, 1997.

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

[121] V Sobolev, A Sorokine, J Prilusky, EE Abola, and M Edelman. Automated analysis of interatomic contacts in proteins. *Bioinformatics*, 15(4):327–332, 1999.

[122] R. V. Spriggs, P. J. Artymiuk, and P. Willett. Searching for patterns of amino acids in 3d protein structures. *J. Chem. Inf. Comput. Sci.*, 43:412–421, 2003.

[123] C.L. Tang, L. Xie, I.Y. Koh, S. Posy, E. Alexov, and B. Honig. On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. *JMB*, 334(5):1043–62, 2003.

[124] W. R. Taylor, T.P. Flores, and C.A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1858–1870, 1994.

[125] J.D. Thompson, F. Plewniak, and O. Poch. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–88, 1999.

[126] WS Valdar. Scoring residue conservation. *Proteins: Structure, Function and Bioinformatics*, 48(2):227–240, 2002.

[127] G. Vriend and C. Sander. Detection of Common Three-Dimensional Substructures in Proteins. *Proteins*, 11:52–58, 1991.

[128] A. C. Wallace, R. A. Laskowski, and J. M. Thornton. Derivation of 3D coordinate templates for searching structural databases: application to Ser-His-Asp catalytic triads in the serine proteinases and lipases. *Protein Science*, 5:1001–1013, 1996.

[129] H. J. Wolfson, M. Shatsky, D. Schneidman-Duhovny, O. Dror, A. Shulman-Peleg, B. Ma, and R. Nussinov. From structure to function: Methods and applications. *Curr. Prot. and Pep. Sci.*, 6:171–83, 2005.

[130] Haim J. Wolfson and Isidore Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 04(4):10–21, 1997.

[131] H.J. Wolfson. Model-Based Object Recognition by Geometric Hashing. In *Proceeding of the* 1st *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 526–536. Springer-Verlang, 1990.

[132] T.D. Wu, S.C. Schmidler, and T. Hastie. Regression analysis of multiple protein structures. *J Comput Biol*, 5:585–595, 1998.

[133] A. S. Yang and B. Honig. An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structure alignment and quantitative measure for protein structural distance. *JMB*, 301(3):665–678, 2000.

[134] Xin Yuan and Christopher Bystroff. Non-sequential structure-based alignments reveal topology-independent core packing arrangements in proteins. *Bioinformatics*, 21(7):1010–1019, 2005.

[135] Z. Zhang, M. Lindstam, J. Unge, C. Peterson, and G. Lu. Potential for dramatic improvement in sequence alignment against structures of remote homologous proteins by extracting structural information from multiple structure alignment. *J. Mol. Biol.*, 332(1):127–42, 2003.