Graph Alignments: A New Concept to Detect Conserved Regions in Protein Active Sites

Nils Weskamp*‡, Eyke Hüllermeier*†, Daniel Kuhn‡ and Gerhard Klebe‡

Abstract: We introduce the novel concept of graph alignment, a generalization of graph isomorphism that is motivated by the commonly used multiple sequence alignments. Graph alignments and graph isomorphisms are equivalent in the case of pairwise comparisons, but if many graphs should be analyzed simultaneously, graph alignments are more robust against noise and perturbations. Thus, graph alignments can detect conserved patterns in large sets of related graphs in a more reliable way.

We study simple heuristics for the efficient calculation of approximate graph alignments and apply our approach to the structural analysis and functional classification of the active sites of proteins. Our approach is able to automatically detect even weakly conserved structural patterns in protein structures. Among other applications, these motifs can be used to classify proteins according to their function.

1 Introduction

Multiple sequence alignment is an established method for the detection of conserved motifs in genes and proteins nowadays [SM97]. Different databases contain multiple alignments and conserved motifs of known sequence families ([BBC$^+$02],[SBC$^+$02]). The use of position specific scoring matrices allows for an automated search of such conserved regions in sequence data [AMS$^+$97] and thus for the classification of sequences into families and classes.

Until now, in the field of graph-based methods for protein structure analysis (i.e., methods that model the properties of a protein structure as a graph and that are independent of the inherent linear ordering of the residues in the protein backbone), comparable concepts have not been examined intensively [KLW96, APG$^+$94, SNW02]. These methods often focus on exact graph matching techniques and thus rely on the concept of graph isomorphism to detect common (i.e., conserved) regions. This is mainly due to the fact that exact matching is often sufficient for pairwise comparisons and already challenging in its complexity. But as the number of available protein structures grows exponentially, multiple examples are accessible for most of the protein families nowadays. Thus, the simultaneous comparison of multiple protein structures becomes an interesting opportunity as it allows one to detect slight structural variations due to mutations and also conformational flexibility upon ligand

[*] Department of Mathematics and Computer Science, Philipps-University Marburg, Hans-Meerwein-Straße, 35032 Marburg, Germany

[†] Corresponding author: eyke@mathematik.uni-marburg.de

[‡] Institute of Pharmaceutical Chemistry, Philipps-University Marburg, Marbacher Weg 6, 35032 Marburg, Germany

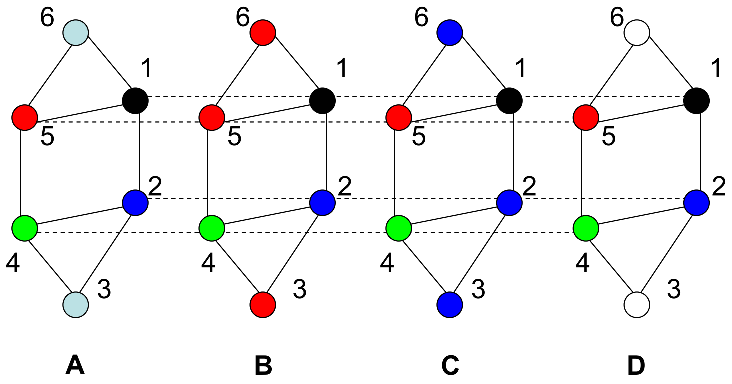

Abbildung 1: A (multiple) isomorphism among the labeled graphs A, B, C, D is indicated by the dashed lines. All four graphs contain the identical subgraph with the nodes $1, 2, 4, 5$ while nodes 3 and 6 have different labels in all graphs.

binding. For the case of multiple comparisons, some authors suggest a generic extension of the exact matching approaches [KLW96]. Yet, as we will show in this paper, the concept of graph isomorphism can be too stringent for the analysis of large assemblies of diverse structures as it is sensitive to outliers and noise. We therefore introduce the more robust concept of a graph alignment that relies on approximate graph matching techniques and thus allows one to detect weakly conserved motifs.

Graph alignments can be seen as a generalization of (sub-)graph isomorphisms as each graph isomorphism is also a graph alignment. Yet graph alignments introduce an additional degree of freedom as they allow one to introduce new, in the following called dummy (or: gap), nodes into a graph whenever appropriate. Figure 1 shows an exemplary set of four graphs that contain an identical (i.e., isomorphic) subgraph of size 4. This “conserved” subgraph may be detected by well-established standard techniques [Le72, BK73]. Figure 2 shows instead a perturbed version of the graph collection as it is more likely to occur in real-life datasets. In this paper, we consider the deletion of nodes or the changing of node-labels and edge-weights as possible perturbations. Standard approaches are highly susceptible to these kinds of changes. For example, in Figure 1, all nodes with the number 5 correspond to each other. Yet in Figure 2, the node labeled number 5 is missing in graph A. Therefore, also the correspondence among the remaining three nodes is “lost” as no matching node exists in graph A.

On the other hand, as each of the four centers $1, 2, 4, 5$ is significantly conserved across the series (i.e., in three out of four graphs), one would appreciate to detect the (partially) conserved pattern defined by these nodes. By introducing dummy nodes (shown as boxes in dark gray), it is possible to detect this conserved pattern. Therefore, the concepts proposed in this paper may be useful to improve the robustness against errors and noise contained inherently in protein structure data.

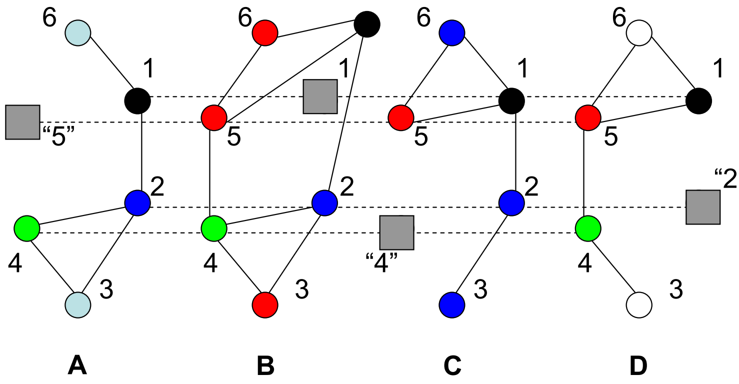

Abbildung 2: A perturbed version of the graphs in Figure 1. No non-trivial isomorphism exists among these graphs as each of the "conserved" nodes is missing (or is strongly shifted to another position which leads to changed edge weights) in one of the graphs. By inserting additional "dummy-nodes" (shown as boxes in dark gray) it is still possible to identify the (partially) conserved pattern.

This paper is organized as follows: First, we give a formal definition of graph alignments (section 2) and describe some algorithms for their construction (section 3). Additionally, we show that graph alignments may be used to screen large datasets of protein structures for members of a particular functional family (section 4).

2 Graph Alignments

Usually, a (multiple) sub-graph isomorphism across a set of node-labeled and edge-weighted graphs $\{G_1(V_1, E_1), \ldots, G_n(V_n, E_n)\}$ is defined as a set $T \subseteq V_1 \times \cdots \times V_n$ such that

1. For all $t = (v_1, \ldots, v_n), t' = (v'_1, \ldots, v'_n) \in T$ with $t \neq t'$: $v_i \neq v'_i$ for $i = 1, \ldots, n$, i.e. each node of each graph is contained in at most one tuple such that the mapping defined by the graph isomorphism is unambiguous.

2. For $t = (v_1, \ldots, v_n) \in T$: $l(v_1) = \cdots = l(v_n)$, i.e. all nodes that are mapped onto each other share the same label.

3. For all $t = (v_1, \ldots, v_n), t' = (v'_1, \ldots, v'_n) \in T$ with $t \neq t'$: If $(v_i, v'_i) \in E_i$ for an $i = 1, \ldots, n$, then $(v_j, v'_j) \in E_j$ for all $j = 1, \ldots, n$ and $\left|w(v_i, v'_i) - w(v_j, v'_j)\right| \leq \epsilon$ for an $\epsilon \in \mathbb{R}$, i.e. if two nodes are connected in one of the graphs, then the associated nodes in the other graphs must also be connected by an edge with a similar weight.

This definition implies that a node may be included in the isomorphism T only if it is contained in *each* of the graphs $G_1, \ldots, G_n$. For large values of n, the presence of slight perturbations thus leads to very small or even empty isomorphisms. A simple consideration shows the relevance of this problem: Assume that p is the (small) probability that a particular conserved center is perturbed in one particular graph, n is the number of conserved nodes and t is the number of graphs. Then — assuming independence of the perturbations — the probability that the complete conserved subgraph will be discovered is $(1-p)^{n \cdot t}$. For $p = 0.01$ and $n = t = 10$, one obtains a probability of 0.37 (0.13 for $t = 20$, 0.05 for $t = 30$). More generally, the expected size of the actually conserved subgraph is only $(1-p)^t \cdot 100$% of the complete subgraph.

Thus, even if the graphs share a significant amount of similarity and contain highly conserved nodes, it is likely that for each conserved node, at least one graph G_i exists such that the node is missing in this particular graph. For example, in Figure 2, each of the four nodes $1, 2, 4, 5$ is "conserved" in three of the four graphs, but as it is also missing in the remaining graph, it is not part of an isomorphism of the four graphs. Considering only a subset of the four graphs does not solve this problem as in this case at most three of the centers could be identified as "conserved".

To overcome this problem, we suggest the concept of a graph alignment: The central idea is to allow the insertion of additional nodes — named "dummy nodes" — into the analyzed graphs. These dummy nodes play a role similar to that of gaps in sequence alignments. If a conserved node is "missing" in a particular graph for what reason ever, a dummy node can be introduced to allow for the construction of a valid mapping of the conserved nodes. (See the dummy nodes shown as gray boxes in Figure 2.)

Formally, a graph alignment[1] is defined as a set $T \subseteq V_1 \cup \{\square\} \times \cdots \times V_n \cup \{\square\}$ (where $\square$ represents a dummy node), such that

1. For all $t = (v_1, \ldots, v_n), t' = (v'_1, \ldots, v'_n) \in T$ with $t \neq t'$: If $v_i \neq \square$, then $v_i \neq v'_i$ for $i = 1, \ldots, n$, i.e. each non-dummy node of each graph is contained in at most one tuple such that the mapping defined by the graph alignment is unambiguous.

2. For $t = (v_1, \ldots, v_n) \in T$: $l(v_1) = \cdots = l(v_n)$ for all $v_i \neq \square$, i.e. all non-dummy nodes that are mapped onto each other share the same label.

3. For all $t = (v_1, \ldots, v_n), t' = (v'_1, \ldots, v'_n) \in T$ with $t \neq t'$: If $v_i, v'_i \neq \square$ and $(v_i, v'_i) \in V_i$ for an $i = 1, \ldots, n$, then $(v_j, v'_j) \in E_j$ for all $j = 1, \ldots, n$ with $v_j, v'_j \neq \square$ and $\left|w(v_i, v'_i) - w(v_j, v'_j)\right| \leq \epsilon$ for an $\epsilon \in \mathbb{R}$, i.e. if two non-dummy nodes are connected in one of the graphs, then the associated non-dummy nodes in the other graphs should also be connected by an edge with a similar weight.

We call the nodes v_i contained in a tuple $t = (v_1, \ldots, v_n) \in T$ aligned.

The additional degree of freedom introduced by this extension has a number of consequences. First, it is now always possible to align two graphs such that all of their nodes are included in the alignment. Thus, it is no longer required to distinguish between graph

[1] The term "Graph Alignment" was introduced independently by Berg and Lässig [BL03]. The authors understand by "Graph Alignment" a related, yet different concept that does, e.g., not allow for the introduction of dummy nodes.

alignments and subgraph alignments. Second, the additional freedom can lead to ambiguities as many different alignments possibly exist for a given set of graphs. Therefore, it is necessary to introduce a scoring system to separate useful from inadequate alignments. Intuitively, dummy nodes should only be inserted if no other valid match partner can be found. Therefore, the insertion of dummy nodes should be penalized. A very simple (exemplary) scoring system is defined by

$$s(T) = \sum_{t \in T} \sum_{t=(t_1,\ldots,t_n)} \begin{cases} -\lambda & t_i = \square \\ 0 & t_i \neq \square \end{cases} \tag{1}$$

for a $\lambda \in \mathbb{R}_+$. This scoring system simply counts the number of dummy nodes that have been inserted into the alignment. Depending on the application area, it is of course possible to apply more sophisticated scoring schemes, which may, e.g., assign different penalties λ to different (types of) nodes. Additionally, one could also weaken point 2 of the requirements in the definition above and allow for mismatches (i.e., an alignment of nodes with different labels), which would then have to be penalized in the scoring system.

In principle, it is possible to extend the standard methods for the detection of graph isomorphisms based on clique-detection in association graphs [Le72, BK73]. These require the scoring system to be additive and monotonic in the size of T. Yet, as the maximum clique problem is computationally very complex [DP99], such an approach is not feasible for more than just a few graphs: One would have to construct an association graph that contains all potential mappings among the different graph nodes. I.e., this graph would contain as nodes all tuples $t \in V_1 \cup \{\square\} \times \cdots \times V_n \cup \{\square\}$ that fulfill point 2 of the definition above. Apparently, the size of such a structure grows very fast with the number and size of the input graphs and easily reaches impractical values.

In the next section, we therefore suggest two simple heuristic algorithms for the efficient calculation of graph alignments. These are — as the idea for graph alignments itself — stimulated by concepts from multiple sequence analysis.

3 Algorithms

In the case of $n = 2$, i.e. for pairwise comparisons, the difference between graph isomorphisms and graph alignments is small. In particular, a generic extension of (maximal) pairwise subgraph isomorphisms to pairwise alignments exists that is optimal with respect to additive scoring systems such as (1). This extension simply assigns all nodes of both graphs that are not part of the isomorphism to a dummy node in the other graph.

Therefore, the calculation of optimal pairwise graph alignments easily reduces to the calculation of pairwise maximal subgraph isomorphisms. For this problem, the standard approaches for the detection of graph isomorphisms are usually applicable in practice [BK73, Le72]. For the graphs that we consider in the experiments section, such a pairwise isomorphism can be calculated in less than a second on a standard Linux computer, if some heuristics are used [SKK02]. To optimize the runtime, we first use an exhaustive search strategy to calculate a small “seed” isomorphism, which is then extended with a greedy strategy.

Thus, an efficient way has to be developed to calculate multiple graph alignments from pairwise graph alignments. In sequence analysis, star alignments and tree alignments are frequently used schemes for calculating multiple sequence alignments [SM97]. We therefore propose to use similar schemes for the calculation of multiple graph alignments. We plan to investigate more sophisticated methods (like, e.g. algorithms based on continuation methods [GR96] or techniques for the construction of large isomorphisms from small local matches) in the future.

Let $G = \{G_1, \ldots, G_n\}$ be a set of graphs and let $T_1(G_1, G_2), \ldots, T_{n-1}(G_1, G_n)$ be a set of pairwise alignments of G_1 to $G_2, \ldots, G_n$. Then, Algorithm 1 can be used to iteratively merge two alignments in a star-like manner. The algorithm uses the nodes of the "central" graph G_1 as pivot elements to join the two alignments. If a pivot element is not included in one of the alignments, the respective positions are filled with dummy nodes. The resulting alignment is not necessarily optimal with respect to the scoring system, but it can be calculated very efficiently. In the next section, we show that alignments calculated by Algorithm 1 are often sufficiently reliable in practice.

The quality of star alignments strongly depends on the initially selected central graph. The latter can either be selected manually (e.g., if one of the graphs is known to be particularly representative or characteristic for the dataset) or automatically on the basis of the used scoring system (e.g., (1)). For this purpose, an alignment can be calculated for different central graphs and then the alignment with the highest overall score is selected. In our experiments, we simply tested all graphs of the dataset as centers.

Alternatively, it is sometimes better to use a tree alignment scheme to produce the graph alignment. The algorithm used to perform this calculation is conceptually similar to algorithm 1, but it does not use a fixed central graph. Instead, it merges each alignment with its nearest neighbor (i.e., the alignment with the highest number of overlapping pivot elements). Tree alignments often have a higher relevance than star alignments and are better suited for rather heterogeneous sets of graphs. Yet, for the graphs used in our experiments, we found the difference among star and tree alignments to be relatively small. The examination of other, more sophisticated algorithms for the efficient calculation of robust graph alignments is one direction of our future research.

4 Experiments

Our practical experiments are based on the Cavbase extension of the Relibase system [SKK02, HBGK03]. In Cavbase, active sites of protein structures are described by graphs in a first-order approximation. The geometrical structure and the physico-chemical properties of a binding pocket are represented by pre-defined pseudocenters — points in the three dimensional space that represent the center of a particular property. Two centers are connected by an edge in the graph representation, if their Euclidean distance is below $12.0\,\text{Å}$ and each edge is labeled with the respective distance. The edges of the graph thus represent geometrical constraints among certain properties. The obtained graphs have a size of approximately 85 nodes on the average, whereas graphs with hundreds of nodes are frequent and graphs with thousands of nodes do exist. The graphs are also quite dense as

Algorithm 1 Algorithm for the star-like merging of graph alignments using a fixed reference graph G_i as center.

```
Input: Alignments $T_1$,$T_2$, both
       containing $G_i(V_i, E_i)$
For each $t = (t.1, \ldots, t.n) \in T_2$:
 If $\exists t' = (t'.1, \ldots, t'.m) \in T_1$
    s.t. $t.i = t'.i$:
  Replace $t' \in T_1$ by
    $t'' = (t'.1, \ldots, t'.m, t.1, \ldots, t.n)$
 Else:
  Insert $t'' = (\square, \ldots, \square, t.1, \ldots, t.n) \in T_1$
For each $t' = (t'.1, \ldots, t'.n) \in T_1$ :
 If $\nexists t = (t.1, \ldots, t.n) \in T_2$
    s.t. $t.i = t'.i$:
  Insert
    $t'' = (t'.1, \ldots, t'.n, \square, \ldots, \square) \in T_1$
Return $T_1$
Output: Merged Alignment from $T_1$ and $T_2$
```

approximately 20 percent of all pairs of nodes are connected by an edge on the average. As proteins exhibit a certain amount of flexibility and experimental structure determination is error-prone or mutations could create additional structural variation, one cannot expect to find identical occurrences of a conserved pattern. Instead, one has to search also for patterns that occur only partially in the different structures to characterize a particular protein family.

To examine the performance of our approach, we examined a diverse set of 10 enzyme families (cf. Table 1). We took a number of representative binding pockets from each of these families and used algorithm 1 to calculate a multiple graph alignment. Using the generated alignments, it was possible to calculate a "consensus pocket" that contains information about all pseudocenters conserved in many of the considered pockets[2]. For each such center c, an average position and the spatial variance $var(c)$ across the data set are stored (cf. Figure 3). (These values are determined from a rigid superimposition of the pockets). Additionally, the degree of conservation $con(c)$ is stored for each of the centers. Thus, the consensus pocket may be regarded as an analog to position specific scoring matrices in the sequence domain.

Consensus pockets may not only serve as a guideline to analyze structural conservation and variation with respect to one particular binding pocket (and could therefore be of interest for structure-based drug design), they are also useful as search templates to screen large datasets of protein structures or to classify proteins of unknown function. As the consensus pocket contains information about the expected position and the allowed deviation for each of the conserved centers along with information about the importance of the centers, it is possible to search for occurrences of a particular pattern in protein structures. If a compatible pattern is found in a protein structure, its similarity to the respective consensus pocket is calculated. For each conserved center c that is detected (as center c') in the protein structure, a score from the interval $[0, 1]$ is added to the overall similarity score. This score

[2] In this particular case, we used a threshold of 25%.

is defined as

$$score_c(c') = con(c) \cdot fit_c(c') \tag{2}$$

where

$$fit_c(c') = 1 - 2 \cdot (\Phi_{0,\sqrt{var(c)}}(dist(c, c')) - 0.5), \tag{3}$$

and $dist(c, c')$ denotes the spatial distance of the centers c and c'. $\Phi_{\mu,\sigma}$ denotes the cumulative distribution function of the normal distribution with mean μ and standard deviation σ. fit thus reaches a value of 1.0, if c' is located exactly at the position of c. The value of fit decreases if the distance among c and c' grows, whereas the slope of the decrease depends on the observed variation in the consensus pocket, $var(c)$.

We used the consensus pockets derived for the 10 enzyme families to screen a large and diverse set of 9352 binding pockets from 6044 PDB entries (i.e., representing roughly one quarter of the PDB). As expected, binding pockets corresponding to enzymes of the respective family lie on the best-scored ranks. For example, among the 65 binding pockets from carbonic anhydrases contained in the dataset, 63 were found on the first ranks — and these were the only hits scoring with a significant (i.e., $\leq 10^{-5}$) e-value[3]; 15 of the binding pockets were used to calculate the initial graph alignment. The results for the remaining families are summarized in Table 1.

In general, using the automatically derived consensus pockets, it is possible to identify most of the members of a protein family and of related families. Without question, the relevance of the generated consensus pockets strongly depends on the structures selected to calculate the initial alignment. Thus, it might be possible to improve the results using carefully selected and unbiased subsets of the representative structures.

5 Conclusions and Outlook

We introduced the new concept of graph alignment, which is a generalization of graph isomorphism and conceptually related to (multiple) sequence alignment. If multiple graphs have to be compared simultaneously, graph alignments could lead to better results in the presence of noise as they are more robust against structural perturbations. In particular, they allow the detection of partially conserved patterns that occur with slight variations in many of the analyzed graphs.

We applied our new concept to automatically identify conserved patterns in the binding pockets of enzymes from different families. The resulting patterns were used to screen a large dataset of binding pockets. They were relevant enough to identify most of the structures of the respective families. Using our approach, it should therefore also be possible to perform an automatic functional annotation of newly determined protein structures in the context of high-throughput crystallography. A major advantage of our approach is the fact that the consensus pockets can be calculated fully-automated. Other approaches for template-based search in protein structure databases (e.g., [WBT97, Ha03, Ru98]) are often restricted to search templates of a fixed size and cannot be used to derive the templates

[3] This e-value has been estimated from the empirical distribution of the similarity scores (i.e., from the comparison of one consensus pocket against the dataset of 9352 pockets). These scores follow approximately a normal distribution.

EC-Number	Family Name	Cons. Centers	Entries	Used	Hits	Overlap	True Hits
1.01.01.0001	Alcohol dehydrogenase	103	82	67	134	65	72
2.01.01.0045	Thymidylate synthase	91	108	101	105	96	98
2.05.01.0018	Glutathione transferase	196	77	47	30	29	30
2.06.01.0001	Asp. aminotransferase	53	92	83	80	62	64
2.07.01.0112	Protein-tyrosine kinase	61	43	24	50	19	19
2.07.07.0049	Reverse transcriptase	65	100	55	125	55	55
3.04.21.0004	Trypsin	58	115	110	243	109	109
3.04.23.0016	HIV protase	66	129	111	125	111	113
3.05.02.0006	Beta-lactamase	96	52	40	61	33	33
4.02.01.0001	Carbonate dehydratase	42	65	15	63	15	63

Tabelle 1: The 10 enzyme families considered in our application example. The fourth column shows the number of cavities contributing to establish the consensus pocket of the respective family, the fifth column gives the number of cavities that were actually contributing to calculate the final alignment. Hits are entries from the dataset that achieved a significant (i.e., $\leq 10^{-5}$) e-value. The "Overlap"-column contains the number of entries that were used to calculate the alignment and scored significantly. "True Hits" belong to the respective enzyme family according to their classification. "Hits" that are not "True Hits" typically belong to closely related enzyme families or are lacking a correct annotation. "Cons. Centers" shows the number of conserved centers that have been included in the consensus pocket.

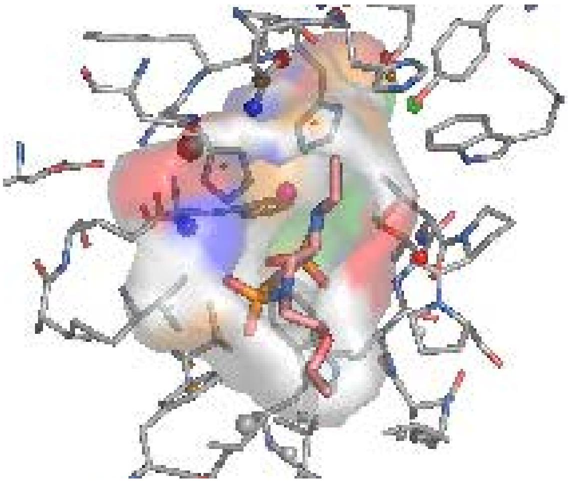

Abbildung 3: The consensus pocket for the family of carbonic anhydrases. The amino acids and the solvent accessible surface from one structural example (PDB-code 1a42) are shown only for orientation. The spheres represent the mean position of the conserved pseudocenters whereas small spheres represent areas of high structural conservation, while large spheres represent centers with higher spatial variation across the analyzed data set.

automatically from a large set of reference structures. In the future, we plan to examine more elaborate algorithms and scoring systems for the calculation of optimal graph alignments. The general concept of graph alignments might also be applicable to a number of different problems and application areas.

Literatur

[AMS$^+$97] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and J., L. D.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*. 25(17):3389–3402. Sep 1997.

[APG$^+$94] Artymiuk, P. J., Poirette, A. R., Grindley, H. M., Rice, D. W., and Willet, P.: 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.

[BBC$^+$02] Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S. R., Griffiths-Jones, S., Howe, K. L., Marshall, M., and Sonnhammer, E. L. L.: The Pfam Protein Families Database. *Nucleic Acids Research*. 30.175):276–280. 2002.

[BK73] Bron, C. and Kerbosch, J.: Algorithm 457: Finding All Cliques of an Undirected Graph. *Communications of the ACM*. 16(9):575–577. September 1973.

[BL03] Berg, J. and Lässig, M.: Local graph alignment and motif search in biological networks. Technical Report COND-MAT/0308251. ArXiv. 2003. Available from `http://xxx.lanl.gov/abs/cond-mat/0205589`.

[DP99] Du, D.-Z. and Pardalos, P. M. (Eds.): *Handbook of Combinatorial Optimization*. chapter The Maximum Clique Problem. Kluwer Academic Publishers. 1999.

[GR96] Gold, S. and Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 18(4):377–388. 1996.

[Ha03] Hamelryck, T.: Efficient identification of side-chain patterns using a multidimensional index tree. *Proteins*. 51(1):96–108. 2003.

[HBGK03] Hendlich, M., Bergner, A., Günther, J., and Klebe, G.: Relibase: Design and Development of a Database for Comprehensive Analysis of Protein-Ligand Interactions. *J. Mol. Biol.* 326:607–620. 2003.

[KLW96] Koch, I., Lengauer, T., and Wanke, E.: An Algorithm for Finding Maximal Common Subtopologies in a Set of Protein Structures. *Journal of Computational Biology*. 3(2):289–206. 1996.

[Le72] Levi, G.: A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*. 9:341–352. 1972.

[Ru98] Russell, R. B.: Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. *J. Mol. Biol.* 279(5):1211–1227. 1998.

[SBC$^+$02] Servant, F., Bru, C., Carrère, S., Courcelle, E., Gouzy, J., Peyruc, D., and Kahn, D.: Prodom: Automated clustering of homologous domains. *Briefings in Bioinformatics*. 3(3):246–251. 2002.

[SKK02] Schmitt, S., Kuhn, D., and Klebe, G.: A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.* 323(2):387–406. 2002.

[SM97] Setubal, J. C. and Meidanis, J.: *Introduction to Computational Molecular Biology*. International Thomson Publishing. 1997.

[SNW02] Shatsky, M., Nussinov, R., and Wolfson, H. J.: MultiProt - A Multiple Protein Structural Alignment Algorithm. In: Guigó, R. and Gusfield, D. (Eds.), *WABI 2002*. Number 2452 in LNCS. pp. 235–250. Springer-Verlag. 2002.

[WBT97] Wallace, A. C., Borkakoti, N., and Thornton, J. M.: TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases - Application to enzyme active sites. *Protein Science*. 6(11):2308–2323. 1997.