

# Efficient Discovery of Common Substructures in Macromolecules \*

Srinivasan Parthasarathy and Matt Coatney  
Computer and Information Science, Ohio State University  
{srini,coatney}@cis.ohio-state.edu

## Abstract

*Biological macromolecules play a fundamental role in disease; therefore, they are of great interest to fields such as pharmacology and chemical genomics. Yet due to macromolecules' complexity, development of effective techniques for elucidating structure-function macromolecular relationships has been ill explored. Previous techniques have either focused on sequence analysis, which only approximates structure-function relationships, or on small coordinate datasets, which does not scale to large datasets or handle noise. We present a novel scalable approach to efficiently discover macromolecule substructures based on three-dimensional coordinate data, without domain-specific knowledge. The approach combines structure-based frequent pattern discovery with search space reduction and coordinate noise handling. We analyze computational performance compared to traditional approaches, validate that our approach can discover meaningful substructures in noisy macromolecule data by automated discovery of primary and secondary protein structures, and show that our technique is superior to sequence-based approaches at determining structural, and thus functional, similarity between proteins.*

## 1. Introduction

In recent years, data mining techniques have been extended beyond traditional domains such as business and marketing to apply to the physical science arena. Many of the hallmark characteristics of data mining, including detection of important patterns, handling of large search spaces, and coping with noisy datasets, are well-suited for the unique challenges presented by chemical domains. The substructure discovery sub-domain is a classic example of the need for powerful, efficient tools to discover unknown patterns. The motivation is that the structure of a molecule determines its biochemical function. This

---

\*This work was partially supported by an Ameritech Faculty Fellowship.

structure-function relationship is critical to understanding the mechanisms of drug activity, toxicity, and disease.

However, substructure mining is more difficult for macromolecules than for small chemicals. Macromolecules have large numbers of atoms, large interesting substructures, and a high amount of three-dimensional coordinate noise due to limitations of current coordinate retrieval techniques. Macromolecule size can produce an enormous search space; macromolecules can have thousands of atoms, each analogous to an object in a spatial database. Even smaller combinations of these atoms can lead to a computationally infeasible search space. For instance, a macromolecule containing 5000 atoms contains 12.5 million 2-atomsets and 20 billion 3-atomsets. The problem is exacerbated by the fact that interesting patterns in macromolecules often contain at least five atoms and perhaps twenty or more. This is because macromolecules are polymers, consisting of repeated similar subcomponents. Thus, interesting patterns are typically particular conformations and sequences of these subcomponents.

This paper presents a scalable method that combines traditional frequent pattern techniques with search space and noise-handling optimizations to efficiently identify *internal* substructures in *large* coordinate datasets. One of the key advantages of our approach is its relative independence from detailed chemical domain knowledge. While this algorithm was designed specifically to address issues related to macromolecules, it is capable of efficiently identifying substructures in any spatial coordinate data, without the need for in-depth scientific understanding of the dataset's properties. The search space is dramatically reduced by discarding atom pairs outside a specified range, an optimization that is possible because atom-atom interaction is inversely related to distance. This and other search space reductions allow efficient construction of larger substructures. Furthermore, we employ an approximate counting mechanism and distance binning to minimize coordinate data noise.

Experimental results on protein data show that this approach is capable of efficiently identifying the peptide backbone and secondary structures of several proteins. This serves as a validation that our technique can indeed find

meaningful substructures without protein-specific chemical knowledge. Finally, we show that the approach is more robust than traditional sequence-based approaches at determining structural similarity between proteins, including high structural similarity of Ribonuclease A from two evolutionarily diverse species.

The ideas presented in this paper form the first part of an exact structure- function macromolecular data mining process. Frequent substructures identified by this approach can then be fed into various data mining techniques, including classification [17], clustering [8], and sequence mining [16] algorithms to produce structure-derived macromolecule functional classes. This not only dramatically reduces the time to develop such classes, the technique also improves the quality of the classes, since they are formed directly from the three-dimensional structure and thus adhere to the structure-function principle.

## 2. Related Work

Many of the techniques recently developed for efficient data mining of frequent patterns are applicable to substructure discovery in molecules. The Apriori algorithm presented in [1] dramatically reduces the search space of a database through use of anti-monotone frequency constraints. The algorithm combines smaller patterns to produce only those candidate larger patterns that can possibly be frequent. Recent research has examined spatial data for such domains as cartography, network topography, and geographical information systems [12]. Sequence analysis and episodes have been analyzed as well by [2] [3] [14].

Application of data mining to discover frequent substructure patterns in three-dimensional graphs is not novel; however, previous research has primarily targeted small molecules and has not addressed issues of scalability and minimization of noise effects. The work of [19] detects substructures of three-dimensional graphs, but the algorithm does not consider atom type, which due to steric and electrostatic behavior is critical to the quality of discovered substructures. [5] more explicitly targets the chemical domain by considering atom and bond types as well as background biological information. However, this approach appears to have scalability issues for even small toxicological compounds let alone for macromolecules. [6] and [18] present more general and scalable approaches to substructure discovery, including using compression and interest- ingness heuristics as well as domain knowledge bias.

Recently, macromolecules such as proteins and nucleic acids have received increased attention in data mining [11] [15] [20]. Most macromolecular analysis has focused primarily on sequence analysis, which at best only approximates structure-function relationships. For example, [9] analyzes a small group of PDB protein data for substructure

patterns through sequences of proximal amino acids. This technique does consider some meta-structure information, but it still does not fully model structural aspects of the proteins. [10] presents a powerful method for obtaining useful meta-structural information when obtaining coordinate data from crystallography is infeasible. This approach complements the work presented here.

We extend the ideas presented in [13], providing a formal description of the algorithm, several performance optimizations, and quantified experimental results for the algorithm and optimizations. This paper also extends the analysis of secondary structures to include a  $\beta$ -sheet case study and includes using substructure fingerprints for finding similarity between proteins.

## 3. Algorithm

### 3.1. Substructure Representation

Molecules and their substructures are often described as three-dimensional coordinate graphs, where atoms are nodes and chemical bonds are edges. Examples include the MDL MOL format and the Protein Database PDB format. Substructures are subgraphs of overall coordinate graphs, normalized to account for varying orientations in space. Two substructures are considered equal if, after an arbitrary number of spatial translations on one substructure, both substructures are described by the same graph.

The type of connection between two atoms, known as the chemical bond type (single, double, or triple), can be ignored without loss of information, since the bond type can be inferred from the types of the two atoms and the distance between them. We consider all proximal atom relationships rather than just connections. This allows for detection of unconnected spatial interactions.

We define a new concept, the *atomset*, that captures all information of a three-dimensional graph in a form that facilitates quick comparison without the need for coordinate translation. We store three-dimensional information between a pair of atoms,  $A_i$  and  $A_j$ , in a *mining bond*. The mining bond  $M(A_i A_j)$  is a 3-tuple of the form

$$M(A_i A_j) = \{A_i type, A_j type, distance(A_i A_j)\}$$

A  $k$ -atomset  $X$ , which is a substructure containing  $k$  connected atoms, is then defined as a tuple of the form

$$X = \{\mathbf{S}_X, A_1, A_2, \dots, A_k\},$$

where  $A_i$  is the  $i^{th}$  atom and  $\mathbf{S}_X$  is the set of mining bonds describing the atomset. By defining atom pair combinations with mining bonds, the three-dimensional graph is completely represented in a redundant form, such that two atomsets  $X$  and  $Y$  are considered to be the same chemical

substructure if  $S_X = S_Y$ . While stereochemistry is not explicitly handled by this representation, it can be implicitly handled by appending a chirality label (e.g. L or R) to the atom type, such that different stereoisomers produce different, non-equivalent atomsets.

In a naive approach,  $|S_X|$  is equal to  $\binom{k}{2}$ , which represents all possible atomset permutations. We next introduce the concept of range pruning, which allows us to not only dramatically reduce the search space of the algorithm but also significantly reduce the number of mining bonds needed to describe a given atomset.

### 3.1.1 Range Pruning

An exhaustive analysis of possible atom combinations, even at lower levels, is computationally infeasible for large graphs such as macromolecules. Application of chemical domain knowledge through range pruning affords a great reduction in search space and allows us to fully describe the three-dimensional representation of a substructure with fewer mining bonds. Range is a user-specified constraint defined as the maximum allowable Euclidean distance between two atoms for them to be considered an atom pair.

The optimization relies on the fact that the associated energy between any two atoms in a molecule is inversely related to the inter-atom distance. Beyond a certain range, atom-atom interaction is negligible and the two atoms can be considered independent of one another. Thus, we can limit the space of atom pairs for candidate 2-atomset generation, ignoring atom combinations outside of the range.

The range of interaction differs depending on the atom types involved. For bio-molecules, the predominant atom by far is carbon, so we ensure that the range sufficiently describes possible carbon-carbon interactions. The typical carbon-carbon single bond has a bond length of 1.54 Å, with double and triple bonds having shorter length. We have found that a range of 4.5 Å, roughly three times a carbon-carbon single bond, is sufficient for encompassing all possible carbon-carbon interactions [13] and hydrogen bonding.

While we enforce the range restriction on initial atom pairs (2-atomsets), we relax this restriction when building up larger substructures such that two atoms may be included in an atomset even if they do not satisfy the range constraint, so long as they share common atoms that do meet the range constraint. For instance, if we have two atomsets  $X$  and  $Y$  with atoms  $(A_1, A_2, A_3)$  and  $(A_2, A_3, A_4)$  respectively, we allow the creation of a candidate atomset  $Z$  with atoms  $(A_1, A_2, A_3, A_4)$  even if  $\text{distance}(A_1, A_4) > \text{range}$ .

Without such a relaxation, higher range values would be needed to detect large substructures; this in turn would limit range pruning’s ability to reduce the search space, which is a key optimization of this algorithm. Also, such a relaxation

allows us to reduce the number of mining bonds needed to fully describe the atomset, since atoms not directly joined by mining bonds are still indirectly associated through mining bonds with their shared atoms. This significantly improves both computational performance and memory efficiency.

## 3.2. Generating Incremental Atomsets

Our approach to generating incremental  $k$ -atomsets from two  $(k-1)$ -atomsets is similar in concept to frequent pattern discovery algorithms. However, our approach is distinctly different from traditional techniques in that removal of internal elements can cause the resulting substructures to become infrequent, based on the range constraint. We describe here a range-based anti-monotone frequency restriction for the chemical domain.

**Definition 1** *Extremal atoms for a given  $k$ -atom structure are the two or more atoms that are furthest away from each other within the structure.*

**Definition 2** *Extremal  $(k-1)$ -atom substructures for a given  $k$ -atom substructure are those substructures that contain at least one of the extremal atoms.*

By definitions 1 and 2, each structure has at least two extremal substructures. These definitions lead to the statement of Lemma 1, which will help prune the number of potentially frequent patterns (candidate atomsets) that will need to be evaluated.

**Lemma 1** *Any frequently occurring  $k$ -atom structure has at least two  $(k-1)$ -atom substructures that are frequent and that satisfy the input range criteria ( $R$ )*

**Corollary 1** *For any frequently occurring  $k$ -atom structure all of its extremal substructures are frequent and satisfy the input range criteria ( $R$ )*

### 3.2.1 Proof Sketch

It is trivial to show that all substructures of a given frequent substructure must be frequent. However not all of these substructures will satisfy the range criteria. A simple example that shows why this is so involves a line of points each separated by a distance corresponding to the range. If one takes out any of the points except the two end points, the resulting substructure will not satisfy the range criteria, because eliminating such a point breaks the linkage. The rest of the proof is based on the fact that eliminating either of the extremal points cannot possibly break the linkage.

Now assume we are given the set  $S$  of frequently occurring  $(k-1)$ -atomsets. By the above lemma and its corollary,

1. define set  $C$  ; Incremented candidate atomsets
2. **for** all  $atomset_i$  in frequent k-atomsets
3. **for** all  $atomset_j$  in frequent k-atomsets:  $j > i$
4. Set  $D = (atom_i \notin atoms_j \forall atoms_i) \cup (atom_j \notin atoms_i \forall atoms_j)$  ; Find atoms different
5. if  $|D| = 2$  and if  $\forall atom_x \in atomset_i \neq D_i$ ,  
 $distance(atom_x, D_j) \leq distance(D_i, D_j)$  ; if two extremal atoms are different
6.  $candidateAtomset_{ij} = \text{new atomset } (S_i \cup \{M(A_i D_j) \text{ for } A_i \in atomset_i$   
 $| distance(A_i, D_j) < range\}, A_1, \dots, A_k, D_j)$
7.  $C = C \cup \{candidateAtomset_{ij}\}$  ; add to set of incremented atomsets

Figure 1. Candidate atomset generation algorithm

the set  $C$  of potentially frequent k-atomsets is limited to those candidates whose extremal substructures are in  $S$ . In essence, if the extremal substructures are not in  $S$  then the potential candidate can never be frequent.

### 3.3. Algorithm Details

Candidate generation and pruning of (k+1)-atomsets from frequent k-atomsets requires special consideration. In a standard frequent pattern discovery approach, (k+1)-sets are constructed from k-sets without regard to the source from which the sets originated. Infrequent (k+1)-sets are then pruned by re-scanning the source and counting the number of occurrences for each (k+1)-set. This approach in a molecular context is computationally very expensive, due to a vast number of possible structural permutations. Rather, candidate generation and pruning are performed using only the available atomsets and do not re-scan the entire molecule.

(k+1)-atomsets are formed using only frequent k-atomsets; this approach is possible since atomset frequency is anti-monotone. Rather than simply generating incremented substructure *patterns*, which would then be used to query the molecule for frequency, *all* instances of (k+1)-atomsets are generated by combining the atoms of frequent k-atomsets. This is detailed in Figure 1.

Pruning is then accomplished simply by counting the number of atomsets that define, based on their mining bond sets, the same substructure, keeping those whose count is above a user-specified minimum support. This approach is faster than traditional approaches both because counting occurs without the need to go back to the entire molecule and because counting is done through pattern-pattern instead of pattern-dataset matching, which results in far fewer comparisons. Standard pruning is straightforward and can be accomplished simply by hashing atomsets into bins of substructures based on the set of mining bonds. More advanced approaches to pruning are needed when handling noisy data; this will be discussed in Section 4.

The complete algorithm is shown in Figure 2. This algo-

1. Prune infrequent atoms (1-atomsets)
2. Generate candidate 2-atomsets from frequent atoms
3. Prune infrequent 2-atomsets
4.  $k = 3$
5. **while** ( $|\text{frequent k-atomsets}| > 0$ )
6. Generate candidate k-atomsets from frequent (k-1)-atomsets
7. Prune infrequent k-atomsets
8.  $k = k + 1$

Figure 2. Substructure discovery algorithm

gorithm is general enough to handle substructure analysis for any type of molecule, and with additional optimizations is suitable for analyzing macromolecules.

## 4. Optimizations

**Distance Binning and Resolution for Noise Handling:** Macromolecules are difficult to isolate, crystallize, and analyze; even with impressive advances in the field, the resulting data is still relatively noisy. One approach for handling this noise is discretization, a common data mining technique [7]. We discretize the raw Euclidean distance between two atoms by binning; a resolution value is chosen that divides the distance into equiwidth bins, represented efficiently as bits in the mining bond. Binning of the data not only simplifies calculations (thus improving performance), but it also handles *minor* fluctuations in distance. Initial studies on a PDB protein dataset suggest that algorithm resolutions between  $0.04 \text{ \AA}$  and  $0.10 \text{ \AA}$  best minimize noise effects while maintaining meaningful atom-atom relationships. Studies have also found that there is a direct relationship between crystallography resolutions and algorithm resolutions suitable for detecting meaningful substructures.

**Recursive Fuzzy Hashing for Noise Handling:** Due to the high level of noise inherent in current macromolecule structure deduction techniques, strict matching of patterns,

```

1.  $m = 1$  ; Examine first mining bond
2.  $C = \text{candidateAtomsets}$  ;
/*Start by examining all candidate atomsets*/
3. for all  $\text{atomset}_i$  in  $C$ 
4. hash( $M_m, M_m + 1, M_m - 1$  in  $\text{atomset}_i$ )
5. for all  $H_i$  in hash_bins
6. remove  $H_i$  if  $|H_i| < \text{minSupport}$ 
7.  $m = m + 1$ 
8. while  $m \leq |M|$ 
9. for all  $H_i$  as  $C$ 
10. goto step 3
11. frequentAtomsets =  $H_1 \cup H_2 \dots H_n$ 

```

**Figure 3. Recursive fuzzy hash pruning**

even with binning, leads to poor results. What is needed is a pruning mechanism that relaxes the strict matching criteria such that two atomsets  $X$  and  $Y$  are considered to be the same chemical substructure if  $S_X \approx S_Y$ . One such approach is recursive fuzzy hashing (RFH), in which atomsets are analyzed one mining bond at a time and atomsets are hashed (using the current mining bond's integer value) both to the exact and neighboring locations.

This is significantly more computationally expensive than the standard approach; however, it is essential for effectively minimizing noise effects to allow for identification of larger substructures. This will be quantified in the next section; we present the technique in Figure 3.

RFH produces a top-down hash tree (see [4] for a thorough description of hash trees), with the root node representing the set of candidate atomsets for the particular  $k$ . We recursively split the nodes into child hash bins, based on the atomsets' mining bond for a particular tree level. We hash both to the exact hash location and  $+/- 1$  resolution unit. Hash bins whose atomset count is less than a user-specified minimum support are pruned from further consideration. Once the tree is constructed, a new set of pruned atomsets is generated from the remaining leaf nodes.

**Depth First Pruning for Performance** Hash trees produced by RFH are only as deep as the number of mining bonds used to represent the atomsets. The tree is significantly wider, however, due both to the large numbers (thousands or tens of thousands) of atomsets and the use of fuzzy-hashing, which can triple the width of the tree.

Breadth-first tree analysis must generate all hash bins for a particular depth before being able to prune away infrequent bins and free up memory. This can be quite memory intensive and does not scale well to deeper depths, where the tree width explodes. Depth-first analysis, on the other hand, only analyzes one branch of the tree at a time, which is a fraction of the tree's width. This results in a much smaller memory footprint.

As an example, let us consider a hash tree generated for hemoglobin. For a breadth-first analysis at level 7, the memory footprint includes 15000 bins, each including a group of atomset references. A depth-first analysis considers only 80 bins at any given time, requiring  $\frac{1}{2}\%$  of the memory. In larger substructure analyses, this ratio will be even more pronounced. Clearly the depth-first approach is more memory efficient. Without further optimizations the bread-first approach leads to an explosion of the memory space at moderate ( $k = 6$ ) levels, while depth-first analysis maintains a small memory footprint even at higher ( $k \geq 9$ ) levels.

**Dynamic Duplicate Screening for Performance:** The RFH approach is not without its drawbacks. Its primary issue is redundant recursive calculations caused by overlapping hash bins. A naive approach to RFH prunes all duplicate substructure bins once the hash tree has been fully formed. However, this leads to a tremendous amount of redundant work.

We instead use a novel approach, dynamic duplicate screening (DDS), that handles duplicates *during* the run. At each level of the hashing algorithm, the set of substructure bins are analyzed, and duplicates are discarded from further consideration. While the analysis itself is expensive, significant time is saved in avoiding redundant calculations. In addition to speed improvements, this technique also has the benefit of significantly reducing the memory footprint needed for the hash tree by decreasing its width.

**Analyzing Polymer Backbones for Performance:** Often, we are primarily interested only in the global conformation and super-structures of macromolecules. When this is the case, we can further reduce the search space of a given macromolecule by only considering the polymer's backbone. Such an approach has the advantage of reducing the number of atoms and candidate atomsets through a pre-processing step. We applied this approach to the protein domain to analyze backbone conformations, and the results were promising. As an example, the backbone-only approach identified the same peptide substructures of lysozyme as the full-blown search and ran *five times faster*.

## 5. Experimental Results

All experiments were conducted under the Java 1.4 runtime environment on a 4 CPU Sun 420R workgroup server and were allocated 1 GB of memory. The program utilizes multi-threading to take advantage of an SMP architecture.

### 5.1. Performance

#### 5.1.1 $k + k$ vs. $k + 2$ Candidate Generation

The initial algorithm presented in [13] utilized  $k + 2$  candidate generation, in which  $k$ -atomsets were combined with 2-atomsets to produce  $(k+1)$ -atomsets. This approach was taken prior to full development of the range pruning theory

**Table 1. Effect of range on search space**

Range ( $\text{\AA}$ )	2-atomsets	3-atomsets	Exec Time (s)
4.5	9K	12K	4
6.0	20K	147K	25
9.0	54K	2.3M	536
$\infty$	500K	N/A	N/A

presented in 3.2. With completion of this theory, we are now able to take full advantage of range pruning through  $k + k$  candidate generation.  $k + k$  combines  $k$ -atomsets with themselves for a more restricted superset of the frequent incremented  $(k+1)$ -atomsets than  $k + 2$ .

We compared the two approaches on the protein lysozyme. In terms of search space, the  $k + k$  approach generates 13% of 4-atomsets produced by  $k + 2$  and only 1% of 5-atomsets. The result is a 3-fold improvement in performance at  $k = 4$  and a 4-fold improvement at  $k = 5$ . The benefit of using  $k + k$  generation increases with larger substructure size; this is due to the larger number of possible combinations in the  $k + 2$  approach and the increased ability of  $k + k$  to restrict atomset permutations. For the remaining experiments, we utilize only  $k + k$  candidate generation.

### 5.1.2 Range Pruning

Range pruning, as mentioned before, has a dramatic impact on reduction of search space. Table 1 demonstrates this for the protein lysozyme for several different ranges. As expected, search space, and thus run time, decreases as the range becomes more restrictive. So long as the range incorporates all atom-atom interactions of interest, there is no loss of domain-relevant information. Traditional substructure discovery approaches do not consider range pruning and thus fail to scale for macromolecules.

### 5.1.3 Dynamic Duplicate Screening

The benefits of the DDS optimization are quite pronounced, particularly at higher levels, due to its ability to handle the exponential growth of redundant calculations. This is illustrated in Table 2. Clearly, DDS maintains a manageable number of duplicates even at higher levels. On the other hand, the standard approach, which screens duplicates only at the end of the run, suffers from an explosive exponential growth rate of duplicate calculations, as evidenced by over 2.5 million atomsets in duplicate substructures during level 5 of the run. For this same level, DDS produces only 19 thousand atomsets, a mere 1% of the standard approach.

At higher levels, the impressive gains of DDS become evident. For instance, a hemoglobin run to identify  $\alpha$ -helices using 9-atomsets took 90 minutes without DDS. *With the optimization, the run took only 5 minutes, resulting in an eighteen-fold increase.*

**Table 2. Effect of dynamic duplicate screening on redundant calculations**

Level	DDS	Standard
	Duplicate Atomsets	Duplicate Atomsets
2	18K	19K
3	64K	81K
4	49K	312K
5	19K	2.5M

### 5.1.4 Performance of Combined Optimizations

We combined all optimizations and compared the performance with traditional structure discovery techniques. The optimized run used  $k + k$  candidate generation and depth-first RFH pruning with DDS. The standard run used  $k + 2$  candidate generation with standard breadth-first RFH. The run analyzed the first subunit of hemoglobin (PDB ID: 1BZ0) with minimum support of 70, resolution of 0.06, and range of 4.5. Note that the runs located the same substructures; they merely differed in their approach to generating candidates and locating these substructures.

Table 3 shows the results for optimized and standard runs. The combined optimization approach is capable of efficiently detecting large substructures while maintaining a small memory footprint. On the other hand, the standard run exhibits an explosive search space; this leads to poor performance and memory use.

## 5.2. Substructure Discovery in Proteins

### 5.2.1 Discovery of the Peptide Backbone

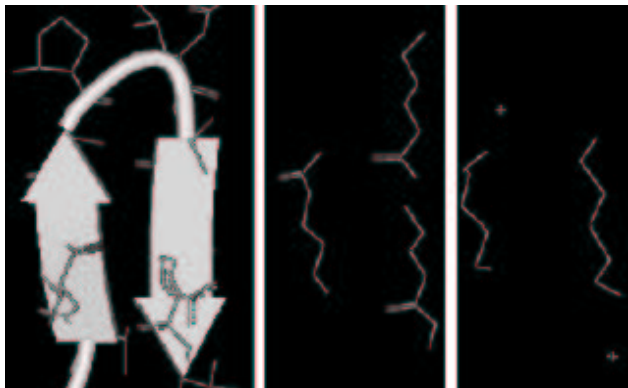
We now turn to identifying relevant substructures in protein macromolecules as validation of our technique. We begin by analyzing the 128-amino-acid protein lysozyme (PDB ID: 193L) for 5-atomset peptide substructures. We configured the algorithm with a minimum support of 100 and range of 4.5  $\text{\AA}$ . A resolution of 0.06  $\text{\AA}$  was settled on for RFH after attempting a series of values.

We also analyzed lysozyme without using RFH, and the strength of RFH for handling noise is evident; RFH was able to find 128 5-atomsets that describe the same peptide substructure. Thus, RFH is capable of fully defining lysozyme’s peptide backbone. Even with significant optimizations to resolution, the standard approach found at most 125 atomsets. Above  $k = 6$ , the standard approach could not reliably detect any substructures. We therefore rely on RFH to identify interesting protein substructures.

The 5-atomsets found in lysozyme represent one substructure pattern, that of a peptide. This includes the backbone oxygen and carbon, the  $\alpha$ -carbon of residue  $i$  and the backbone nitrogen and  $\alpha$ -carbon of residue  $i + 1$ . Combin-

**Table 3. Comparison of combined optimizations and standard algorithm**

Level	Combined Optimizations			Standard		
	Exec Time (s)	Memory (MB)	Atomsets	Exec Time (s)	Memory (MB)	Atomsets
2	2	8	9030	2	9	9030
3	24	221	89717	38	257	92524
4	98	152	68770	370	454	335677
5	121	153	10281	1100	1000	183141
6	165	295	1485	Out Of Memory		
7	183	297	147	Out Of Memory		

**Figure 4.  $\beta$ -sheet (left) and reconstructed 8-atom strands (center, right) of Antibody 21D8**

ing the atomsets, we can reconstruct the peptide backbone of lysozyme in its entirety.

### 5.2.2 Discovery of $\beta$ -Sheets

We now turn to the discovery of protein secondary structures. We previously discovered  $\alpha$ -helices in hemoglobin, the details of which may be found in [13]. We next set out to discover  $\beta$ -sheets. We examined Antibody 21D8 (PDB ID: 1C5C), a two-chain  $\beta$ -sheet rich protein. We considered the first 73 residues of the first chain; this portion contains 3 sheets composed of 11  $\beta$ -strands.

Since  $\beta$ -sheet structures are formed primarily from peptide-peptide interactions, we employed the backbone optimization for pre-processing the protein. This reduced the search space by a factor of 2. A larger range of  $6.5 \text{ \AA}$  was chosen to accommodate for the linear nature of these strands. We set the minimum support to 10 in an attempt to capture all 11 strands of interest. Lastly, we used a higher resolution,  $0.1 \text{ \AA}$  due to poor coordinate resolution.

The algorithm located five classes of frequent 8-atomsets; each of these classes describe different portions of the same substructure. This substructure consists of three linear, connected peptides. These results are validated by biochemical data showing that  $\beta$ -strands have a linear struc-

ture and consist of at least 3 amino acids.

Figure 4 shows the first two  $\beta$ -strands of Antibody 21D8 along with two different atomset representations. The results when compared against the original antibody are impressive; between the five classes of atomsets, all  $\beta$ -strands can be fully reconstructed. Furthermore, a smaller portion (15%) of atomsets describe the ends of *beta*-strands and are distinct from central *beta*-strand atomsets. This shows the power of the algorithm for detecting subtle yet important differences in three-dimensional structure.

### 5.2.3 Structural Similarity of Proteins

Last, we demonstrate how structural features, represented by substructure fingerprints, provide better insight into structure-function relationships than traditional sequence analysis. A substructure fingerprint is a vector representation of a set of interesting substructures. Elements contained in that molecule are marked either with a 1 for a bit vector or with the occurrence count for a frequency vector. Elements not in the molecule are marked with a 0.

We consider the protein Ribonuclease A from two disparate species: bovine (PDB ID: 1JVT) and rat (PDB ID: 1RRA), as well as a similar protein from a related protein kinase class (PDB ID: 1BDY) and a significantly different protein, the  $\rho$  transcription terminator (PDB ID: 1A8V). The coordinate sets all have comparable resolution (between  $2.0$  and  $2.5 \text{ \AA}$ ) and chain lengths (between 120 and 125 residues).

We ran our algorithm using the following parameters:  $0.12 \text{ \AA}$  resolution,  $6.5 \text{ \AA}$  range, minimum support of 20, and the backbone optimization. From the run, we collected all substructure motifs with five or more atoms and generated bit vector substructure fingerprints. We then analyzed fingerprints and sequences using the common Tanimoto similarity coefficient, defined as

$$Tanimoto(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

The traditional sequence-based approach using the primary amino acid sequence gave only moderate similar-

ity between the two Ribonuclease A proteins (Tanimoto coefficient of 0.50); this is a sign of evolutionary divergence and demonstrates the main limitation of sequence analysis. However, when comparing the Ribonuclease A substructure-based bit vector fingerprints, we obtain a Tanimoto similarity coefficient of 0.75, which is a significant improvement over the sequence-based approach. This is to be expected, since functional structures are evolutionarily conserved despite innocuous sequence mutations.

Furthermore, sequence analysis of the two Ribonuclease A proteins with the kinase (functionally related) and transcription factor (functionally unrelated) proteins give Tanimoto coefficients of only 0.03, suggesting no relationship. Yet our approach detects substantial structural similarity between Ribonuclease A and the related kinase (coefficient 0.35-0.45), which is consistent with the similar function of the proteins' classes. Furthermore, our approach detects little structural similarity (coefficient of 0.17) between the Ribonuclease A and transcription factor, a result consistent with the disparity in function.

While preliminary, these results support the commonly held belief that a structure-based technique such as ours is capable of more robust macromolecule classification than traditional sequence-based approaches.

## 6. Conclusion

This paper presents a novel approach and several key optimizations for mining frequent substructures in complex, noisy spatial data such as macromolecules. The approach differs from previous substructure mining techniques in that it locates frequent substructures *within a single large molecule* and is designed specifically to address scalability and noise issues chronic to the macromolecule domain. Furthermore, it operates on exact structures instead of sequences or meta-structural information. Through a series of experiments, the algorithm is validated both for good performance when compared to standard techniques and for good frequent substructure identification as evidenced by its ability to detect meaningful substructures in proteins as well as common structural features between similar proteins from different species.

Performance is always an issue when dealing with such large molecules, and as such research is in progress for further optimizing analysis based on both domain knowledge and computer science principles. One domain-centered approach for proteins is consideration of peptide  $\phi$  and  $\psi$  angles in reduction of search space. More efficient approximate pruning, such as use of a three-dimensional sliding box, and graph compression techniques using substructures [6] are under consideration.

With a framework in place for efficient analysis of substructures in macromolecules, we are now able to conduct further research in this area in a timely manner. Future work

includes the analysis of higher-order substructures in an attempt to discover novel secondary structures and the development of functionally significant classification models for proteins based on discovered substructures. In addition, we hope to extend this approach to other types of biologically significant macromolecules such as DNA and the various forms of RNA. Eventually, this algorithm will be combined with other data mining techniques to provide a robust structural analysis framework for spatial datasets.

## References

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB Conference*, 1994.
- [2] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, 1995.
- [3] R. Agrawal and R. Srikant. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, 1996.
- [4] E. G. Coffman and J. Eve. File structures using hashing functions. *Comm. Assoc. Comp. Mach.*, 1970.
- [5] L. Dehaspe, H. Toivonen, and R. King. Finding frequent substructures in chemical compounds. In *KDD*, 1998.
- [6] S. Djoko *et al.* Analyzing the benefits of domain knowledge in substructure discovery. In *KDD*, 1995.
- [7] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *ICML*, 1995.
- [8] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [9] I. Jonassen *et al.* Structure motif discovery and mining the pdb. In *German Conference on Bioinformatics*, 2000.
- [10] J. Kim *et al.* Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. In *Bioinformatics* 2002.
- [11] R. King *et al.* Genome scale prediction of protein functional class from sequence using data mining. In *KDD*, 2000.
- [12] K. Koperski, J. Han, and N. Stefanovic. An efficient two-step method for classification of spatial data. In *Proceedings of the Intl. Symposium on Spatial Data Handling*, 1998.
- [13] H. Li and S. Parthasarathy. Automatically deriving multi-level protein structures through data mining. In *HiPC Workshop on Bioinformatics and Computational Biology*, 2001.
- [14] H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. In *KDD*, 1996.
- [15] W. Pan, J. Lin, and C. Le. Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, 2002.
- [16] S. Parthasarathy *et al.* Incremental and interactive sequence mining. In *ACM CIKM*, 1999.
- [17] J. Quinlan. Induction of decision trees. *Machine Learning*, 5(1):71-100, 1996.
- [18] L. D. Raedt and S. Kramer. The level-wise version space algorithm and its application to molecular fragment finding. In *IJCAI*, 2001.
- [19] X. Wang and *et. al.* Automated discovery of active motifs in three dimensional molecules. In *KDD*, 1997.
- [20] X. Zheng and T. Chan. Chemical genomics: A systematic approach in biological research and drug discovery. *Current Issues in Molecular Biology*, 2002.