Efficient Algorithms for Range Queries in Protein Sequence Analysis

N. Madhusudhanan* Prosenjit Gupta*† Abhijit Mitra‡

Abstract

A protein chain consists of a sequence of amino acids. Each amino acid may occur several times in the chain. Associated with each amino acid are various categorical attributes like charged, polar, hydrophobic etc. Protein sequences are often subjected to standard sequence analysis. Often statistical criteria are used for evaluation of various protein sequence attributes. Several queries arise during the statistical analysis of such a protein sequence which involves a segment of the protein. We model some of these problems as generalized range searching problems in computational geometry and provide efficient data structures and algorithms for solving them.

1 Introduction

Proteins are long sequences of different amino acid residues, connected through what are known as peptide bonds. Each of the common 20 different amino acids is respectively associated with unique side chains which may be classified in terms of several non-orthogonal, often hierarchical, physical and chemical categories such as size, polarity, charge, hydrophobicity etc. Apart from specific physicochemical categories, other abstract statistical attributes such as frequency of overall or context related occurrence or mutation probability etc are also used to categorize protein sequences [11]. Analysis of protein sequences is important to biologists in a variety of contexts including determination of structure and function, phylogeny etc. It is almost universally held that sequence of amino acids in a protein determines its structure which in turn determines its function. It is also observed that nature is abound with conserved structural motifs comprising of sequences which are not identical but have similarities in the statistical distribution of amino acid attributes. In protein secondary structure prediction methods, sequence ranges are examined [13]. Information regarding the statistical distribution of individual amino acids or classes of amino acids in the whole or in different parts of proteins is thus vital for biologists [6].

1.1 Contributions

For reasons discussed above, protein sequences are often subjected to standard sequence analysis and often statistical criteria are used for evaluation of various protein sequence properties [6]. In statistics, *categorical* (or *nominal*) variables represent types of data which may be divided into groups. Let us assume that a protein $\mathcal{P}$ consists of a chain of amino acids $\mathcal{P} = (a_1, a_2, \ldots, a_n)$. Each amino acid may occur several times in the chain. Associated with each amino acid are various categorical attributes like charged, polar, hydrophobic etc. A segment of a protein is a sequence $(a_i, a_{i+1}, \ldots, a_j)$ for $1 \leq i \leq j \leq n$. Several queries may be of interest in the analysis of such a protein, which involves categorical queries about a given segment. In the rest of this paper, we consider such queries and provide efficient solutions for them using techniques of generalized intersection searching in computational geometry. Work involving other computational geometry approaches to answering queries in large chain molecules may be found in [2, 3, 14, 15]. However those papers mostly dealt with structural properties.

2 Generalized Intersection Searching

In a class of problems called *generalized intersection searching problems*, a set of geometric objects S comes aggregated in groups. We associate with each group an unique color and assume that all objects of that group have the same color. In the more common repetitive mode variant, the problem of interest is to preprocess S into a data structure such that given a query object q, the colors (i.e. groups and not their individual members) of the groups that intersect q can be reported or counted efficiently. Here, we say that q intersects a group if and only if it intersects at least one object in the group. In the single-shot variant, the problem of interest is to find the groups (colors) which intersect. Generalized intersection searching problems arise in many contexts where the number of groups and the size of each group are non-constant. Note that a generalized problem reduces to the standard one when each

*Algorithms and Computation Theory Laboratory, International Institute of Information Technology, Gachibowli, Hyderabad, Andhra Pradesh 500 019, India. Email: {madhusudhanan@students.iiit.net,pgupta@iiit.net}.

†Corresponding author. Supported in part by grant SR/S3/EECE/22/2004, Department of Science and Technology, Govt. of India.

‡Bioinformatics Research Center, International Institute of Information Technology, Gachibowli, Hyderabad, Andhra Pradesh 500 019, India. Email: abi_chem@iiit.net

color class has cardinality one.

Generalized intersection problems were first studied by Janardan and Lopez in [10]. A survey in this area may be found in [9]. In this paper, we apply solutions for generalized intersection problems to some problems in protein sequence analysis.

3 Querying on sequence of amino acids

3.1 Reporting distinct amino acids in a range

Consider a set of P of n points $[1..n]$ which represents the protein chain. Let $A(i)$ denote the amino acid associated with the the i-th position in the chain. We wish to preprocess P into a data structure such that given a query interval $q = [a, b]$, the distinct amino acids in q can be reported efficiently.

If we associate with each amino acid an unique color, the the above problem is an instance of the 1-dimensional generalized range searching problem which can be solved in $O(n)$ space with $O(\log n+i)$ query time [10, 8] where i is the output size. However considering that we are range searching on the 1-d grid, using the solution of [1], this can be solved in $O(n \log n)$ space and $O(\log \log n + i)$ time. We can do better, taking advantage of the fact that our sequence is really $[1..n]$. First we use the transformation technique of [8] to transform the problem to a standard three-sided query on the 2-d grid $\mathbb{Z}^2$. For each point p of color c, let $pred(p)$ be its predecessor in the ordering. We map point p to the point $p' = (p, pred(p))$ in $\mathbb{Z}^2$ and associate with it the color c. The query interval $[a, b]$ is mapped to the grounded rectangle $[a, b] \times (-\infty, a)$. Using the solution of [4], we conclude:

Theorem 1 *A protein chain P can be preprocessed into a data structure of size $O(n)$ such that given a query range $q = [a, b]$, the distinct amino acids in P which occur in the range q, can be reported in time $O(i)$, where i is the output size.*

3.2 Reporting count of each distinct amino acid in a range

The 1-dimensional generalized static type-2 counting problem was considered in [8]: preprocess a set of colored points on the x-axis such that for each color in a given query interval $q = [a, b]$, the number of points of that color in q can be reported efficiently. For this problem, a $O(n \log n)$ space $O(\log n+i)$ query time solution, where i is the output size was given in [8]. This space bound was improved to $O(n)$ in [5] with $O(\log n + i)$ query time.

In this section, we consider the following problem: we wish to preprocess a protein chain P into a data structure such that for each distinct amino acid in a given query range $q = [a, b]$, we can report the number of occurrences of that amino acid in the range q efficiently. Clearly this is an instance of a generalized 1-dimensional type-2 counting problem.

The solution in [5] uses two priority search trees [12] $PST1$ and $PST2$. $PST1$ is built to answer three-sided queries of the form $q' = [a, b] \times (-\infty, a)$ and $PST2$ to answer queries of the form $q'' = [a, b] \times (b, +\infty)$. We can replace the two priority search trees $PST1$ and $PST2$ in the solution of [5] with $TSQ1$ and $TSQ2$ respectively which are instances of the data structure of [4] for answering three-sided queries. The rest of the solution remains the same as in [5].

Theorem 2 *A protein chain P can be preprocessed into a data structure of size $O(n)$ such that for each distinct amino acid in a given query range $q = [a, b]$, the number of occurrences of that amino acid in the range q, can be reported in time $O(i)$, where i is the output size.*

4 Querying on amino acid properties

A set of amino acids $\mathcal{A}$ may be grouped according to properties (*hydrophobic*, *polar* etc.) into various subsets. The subsets may have mutually non-empty intersections. Certain subsets may be contained in others. For e.g. the set of *charged* amino acids is properly contained in the set of polar amino acids. Also the set of charged amino acids contains *positive* and *negative* ones.

In this section, we consider queries involving the various properties of the amino acids in a protein chain rather than the amino acids themselves. Typical questions that can be answered using these algorithms include queries related to *compositional analysis*. For instance a distribution of the properties in a segment of the sequence will give an idea of whether the segment is particularly rich or poor in certain property types.

We are given a protein chain $P = (a_1, a_2, \ldots, a_n)$. Each a_k is an amino acid chosen from the set $\mathcal{A}$. If amino acid a_j has r_j properties, we can create a new sequence P' from P by substituting each instance of a_j with r_j points labelled with the respective properties, each of which we can associate with an unique color. Then P' is a sequence of $N = \sum_{j=1}^{n} r_j$ colored points, $N \geq n$.

Let us define $N_0 = 0$, $N_j = \sum_{t=1}^{j} r_t$. Given a range query $q = [a, b]$ on the sequence P, let $a' = (N_{a-1} + 1)$ and $b' = N_b$. P' is a sequence of colored points $[1..N]$ and $q' = [a', b']$ is a query interval in $[1..N]$. To find the distinct properties of the amino acids in a range $[a, b] \subseteq [1..n]$, we can preprocess the sequence P' using the data structure of Theorem 1 and query with $q' = [a', b']$.

Theorem 3 *A protein chain P of size n with a total of $N \geq n$ properties can be preprocessed into a data structure of size $O(N)$ such that given a query range*

$q = [a, b] \subseteq [1..n]$, the distinct properties of the amino acids in P which occur in the range q, can be reported in time $O(i)$, where i is the output size.

When we create the sequence P' from P by substituting amino acid instances with points labelled with the respective properties, each of the r_j points substituting for amino acid A_j gets labelled with a different property (color). Thus given a query range $q = [a, b]$, for each distinct property (color) c of the amino acids in the range q in sequence P, the number of amino acids in the range q with property c is exactly equal to the number of occurrences of color c in the range $[a', b']$ in sequence P'.

Theorem 4 *A protein chain P of size n with a total of $N \geq n$ properties can be preprocessed into a data structure of size $O(N)$ such that given a query range $q = [a, b] \subseteq [1..n]$, for each distinct property c of the amino acids in range $q = [a, b]$, the number of amino acids in the range q having property c, can be reported in time $O(i)$, where i is the output size.*

5 Handling a hierarchy of properties

Sets of amino acids classified according to properties may be contained within others. For e.g. the set of charged amino acids is a proper subset of the set of polar amino acids. Such containment relationships may be represented hierarchically in the form of a tree or a dag (directed acyclic graph) in general. Range queries on the protein sequence as defined in Section 4 will output all distinct properties ignoring the hierarchical relationships between them. However if there is an amino acid with property "charged" and another with property "polar" in the query range, we may simply want to see the property "polar" in the output to the query. Solutions based on the data structures of Section 4 will not be output-senstive in that case. Clearly we need a different technique. In this section, we propose output-sensitive solutions to the protein sequence query problems in the presence of a property hierarchy mentioned above.

Given a set of amino acids $\mathcal{A}$, we define a property DAG (directed acyclic graph) $\mathcal{G}(\mathcal{A}) = (V, E)$ as follows: V is a set of properties of the amino acids in $\mathcal{A}$. Let $u, v \in V$ be amino acid properties. Then there is a directed edge from u to v if the set of amino acids with property u is properly contained in the set of amino acids with property v. We also create a dummy *universal* property U to which all nodes in the dag point. We are given a protein chain $P = (a_1, a_2, \ldots, a_n)$. For a query range $q = [a, b]$, we define a property u to be *maximal* with respect to q if there are no amino acids in the range $q = [a, b]$ in the protein sequence with property v such that there is a directed edge from u to v in the property dag $\mathcal{G}(\mathcal{A})$.

We would like to preprocess a protein chain P such that given a query interval $q = [a, b]$, the distinct properties of the amino acids in the range q which are maximal with respect to q can be reported efficiently. In preprocessing, we create a sequence P' as in Section 4. We also prepend and append to P' a point each to the left and right of the sequence colored with the universal property U. With each point in $p \in P'$, we associate the following:

- $c(p)$: The color or property associated with p. This also represents a vertex in the property dag.
- $pred(p)$: The predecessor of p in the sequence P' which is of the same color as p. Thus $pred(p)$ is a point with the same color as p, to the left of p and the rightmost amongst all such points. If p is the leftmost point of color c, then we set $pred(p) = -\infty$.
- $left(p)$: Amongst all points to the left of p, the rightmost point t such that there is a directed edge from $c(p)$ to $c(t)$ in the property dag $\mathcal{G}(\mathcal{A})$.
- $right(p)$: Amongst all points to the right of p, the leftmost point t such that there is a directed edge from $c(p)$ to $c(t)$ in the property dag $\mathcal{G}(\mathcal{A})$.

P' is a sequence $[1..N]$. We map each point $p \in P'$ to a point $s = (p, pred(p), left(p), right(p))$ in $[1..N]^4$. We give s the same color as p. Given a query interval $q = [a, b] \subseteq [1..n]$ we transform it into a query interval $q' = [a', b'] \subseteq [1..N]$ as before. Given q', we would like to report $c(p)$ for all colors p such that:

$$a' \leq p \leq b'$$

$$-\infty \leq pred(p) < a'$$

$$-\infty \leq left(p) < a'$$

$$b' < right(p) \leq +\infty$$

We transform q' to a 4-dimensional hyper-rectangle q'' in $\mathbb{Z}^4$ defined as follows: $q'' = [a', b'] \times (-\infty, a') \times (-\infty, a') \times (b', +\infty)$.

The following theorem, extended from a similar result for the 1-d colored range searching problem in [8], is crucial to obtaining an output-sensitive algorithm:

Theorem 5 *There is a maximal point of color c in q' if and only if there is a point of color c in q''. Moreover if there is a point of color c in q'', then this point is unique.*

Theorem 5 above helps us in transforming the problem of reporting distinct properties in a range in presence of property hierarchies to a simple range searching problem in $[1..N]^4$ which can be solved in $O(N \log^3 N)$ space with $O(\log^3 N + i)$ query time using range trees.

The space bound can be improved to $O(N \log^2 N)$ by using a priority search tree for grounded range queries and adding two range restrictions.

Theorem 6 *A protein chain P of size n with a total of $N \geq n$ properties can be preprocessed into a data structure of size $O(N \log^2 N)$ such that given a query range $q = [a, b] \subseteq [1..n]$, the distinct properties of the amino acids in P which occur in the range q, and are maximal with respect to q can be reported in time $O(\log^3 N + i)$, where i is the output size.*

We can transform the problem in a different way. Given $q' = [a', b']$, we would like to report distinct colors of points p such that

$$a' \leq p \leq b'$$
$$-\infty \leq left(p) < a'$$
$$b' < right(p) \leq +\infty$$

If we transform point p to the point $s = (p, left(p), right(p))$ in $[1..N]^3$ and the range q' to the hyper-rectangle $q'' = [a', b'] \times (-\infty, a') \times (b', +\infty)$, the result is an instance of the generalized 3-dimensional range searching problem. This can be solved in $O(N \log^4 N)$ space and $O(\log^2 N + i)$ time [8]. Alternatively, since our points and hyper-rectangle endpoints are all in $[1..N]^4$, we can use the results of [1] to get an $O(N^{1+\epsilon})$ space, $O(\log \log N + i)$ query time. Note that there may be multiple points of the same maximal property (color) c that satisfies the range restrictions on p, $left(p)$ and $right(p)$ shown above, but the generalized solution for the 3-dimensional problem ensures that the query time is output-sensitive.

Theorem 7 *A protein chain P of size n with a total of $N \geq n$ properties can be preprocessed into a data structure of size $O(N \log^4 N)$ (respectively $O(N^{1+\epsilon})$) such that given a query range $q = [a, b] \subseteq [1..n]$, the distinct properties of the amino acids in P which occur in the range q, and are maximal with respect to q can be reported in time $O(\log^2 N + i)$ (respectively $O(\log \log N + i)$), where i is the output size.*

6 Conclusions and further research

We have given a unified framework for solving some problems which are of relevance in statistical analysis of protein sequences. Existing techniques can be used to count [5] or report only in colors which have *significant presence* [7]. If instead we wish to compute a weighted sum to indicate some score of interest, maximum count amongst all colors in the range, maximum run for each color in a range etc., existing solutions can be suitably modified to solve these as well. We have implemented the algorithms in Sections 3 and 4 to build a tool for sequence analysis and plan to add additional queries.

References

[1] P.K. Agarwal, S. Govindarajan, and S. Muthukrishnan. Range Searching in Categorical Data: Colored Range Searching on Grid, *Proceedings 10th European Symposium on Algorithms*, Lecture Notes in Computer Science, Vol. 2461, Springer-Verlag, 2002, 323–334.

[2] G. Aloupis, E.D. Demaine, V. Dujmovic, J. Erickson, S. Langerman, H. Meijer, I. Streinu, J. O'Rourke, M. Overmars, M. Soss, and G.T. Toussaint. Flat-state connectivity of linkages under dihedral motions, *Proceedings, ISAAC*, LNCS Vol. 2518, 2002, 369–380.

[3] G. Aloupis, E. Demaine, H. Meijer, J. O'Rourke, I. Streinu, and G.T. Toussaint. On flat-state connectivity of chains with fixed acute angles, *Proceedings, CCCG*, 2002, 27–30.

[4] S. Alstrup, G. Brodal, and T. Rauhe. New data structures for orthogonal range searching. *Proceedings IEEE Symposium on Foundations of Computer Science*, 2000, 198–207.

[5] P. Bozanis, N. Kitsios, C. Makris, and A. Tsakalidis. New upper bounds for generalized intersection searching problems. *Proceedings 22nd ICALP*, Lecture Notes in Computer Science, Vol. 944, Springer-Verlag, Berlin, 1995, 464–475.

[6] V. Brendel, P. Bucher, I.R. Nourbakhsh, B.E. Blaisdell, and S. Karlin. Methods and algorithms for statistical analysis of protein sequences. *Proceedings, National Academy of Sciences*, Biochemistry, 89, 1992, 2002–2006.

[7] M. de Berg, and H.J. Haverkort. Significant-presence range queries in categorical data, *Proceedings, WADS 2003, Springer Verlag Lecture Notes in Computer Science*, Vol. 2748, 2003, 462–473.

[8] P. Gupta, R. Janardan, and M. Smid. Further results on generalized intersection searching problems: counting, reporting, and dynamization, *Journal of Algorithms*, 19, 1995, 282–317.

[9] P. Gupta, R. Janardan, and M. Smid. Computational Geometry: Generalized Intersection Searching *Handbook of Data Structures*, Sartaj Sahni and Dinesh Mehta eds., CRC Press, 2004.

[10] R. Janardan, and M. Lopez. Generalized intersection searching problems. *International Journal on Computational Geometry & Applications*, 3, 1993, 39–69.

[11] A.M. Lesk. *Introduction to protein architecture: the structural biology of proteins*, Oxford University Press, 2000.

[12] E.M. McCreight. Priority search trees, *SIAM Journal of Computing*, 14(2), 1985, 257–276.

[13] D.W. Mount. Bioinformatics sequence and genome analysis, *Cold Spring Harbor Laboratory Press*, 2001.

[14] M. Soss, J. Erickson and M. Overmars. Preprocessing chains for fast dihedral rotations is hard or even impossible, *Computational Geometry*, 26(3), 2003, 235–246.

[15] M. Soss, and G.T. Toussaint. Geometric and computational aspects of polymer reconfiguration, *Journal of Mathematical Chemistry*, 27(4), 2000, 303–318.