Proceedings of the Intelligent Systems for Molecular Biology (ISMB-93), AAAI Press, Menlo Park, CA

Computationally Efficient Cluster Representation in Molecular Sequence Megaclassification

David J. States*, Nomi L. Harris†, and Lawrence Hunter‡
*Institute for Biomedical Computing, Washington University, St. Louis, MO 63110
†Dept. of Pharmaceutical Chemistry, UCSF, San Francisco, CA 94143
and the ‡Lister Hill Center, National Library of Medicine, Bethesda, MD 20892
states@ibc.wustl.edu nomi@cgl.ucsf.edu hunter@nlm.nih.gov

Abstract

Molecular sequence megaclassification is a technique for automated protein sequence analysis and annotation. Implementation of the method has been limited by the need to store and randomly access a database of all the sequence pair similarities. More than 80,000 protein sequences are now present in the public databases, and the pair similarity data table for the full protein sequence database requires over 1 gigabyte of storage. In this paper we present a computationally efficient representation of groups based on a graph theory approach where sequence clusters are described by a minimal spanning tree of highest scoring similarity pairs. This representation allows a classification of N proteins to be stored in order(N) memory. The use of this minimal spanning tree representation simplifies analysis of groups, the description of group characteristics and the manual correction of artifacts resulting from false hits. The new tree representation also introduces new possibilities for artifact generation in sequence classification. Methods for detecting and removing these artifacts are discussed.

Introduction

Megaclassification of protein sequences is a useful tool for molecular sequence analysis (Hunter, Harris, and States, 1992; Harris, Hunter and States, 1992). Megaclassification involves automatically dividing a large sequence database into a collection of groups of related subsequences. These classes describe the database well with few ambiguously assigned sequence segments and clear distinctions between sequence clusters. Each group of protein subsequences may be associated with a particular function in the cell, and thus the classification can be used to predict the possible function of a novel protein.

The implementation of massive classification is computationally demanding. Although algorithmic speed is important, the main practical limitation is space complexity. We developed a massive classification algorithm, called HHS (Hunter, Harris, States, 1992), that can be used to classify very large sequence databases. HHS assembles sequence groups by using a sequence-comparison tool called BLAST (Altschul et al 1990), which generates pairwise similarity information for all pairs of sequences in the database. As the groups are assembled, the pair similarity database must be available for random access. This pair database requires over 500 megabytes of storage for the current sequence collections and grows with the square of the number of sequences. To make massive classification a feasible calculation, the pair information must reside in RAM; the five order of magnitude time penalty required to access magnetic disks is prohibitively slow. These space requirements are the main impediment to work in this area, so we sought to develop alternative algorithms for massive classification with reduced memory requirements.

A second issue that arises in the practical use of the HHS algorithm is its susceptibility to overaggregation due to false positive similarity judgments. Our algorithm does an approximate transitive closure on the similarity judgments, and a single false positive is enough to merge two unrelated groups of sequences. We take a variety of steps in the clustering algorithm to avoid this problem, including the use of sequence filters that eliminate repetitive and low-entropy sequences, such as XNU (States and Claverie, 1993) and seg (Wootton and Federhen, 1992). The use of these filters dramatically reduces the number of high scoring false positive alignments generated in the course of a sequence database self-comparison. However, these filters do not completely eliminate false positives. The problem is compounded by the fact that false positives often occur in sets. If a high scoring alignment is seen between two members of biologically unrelated sequence classes, sequence correlations within the classes often imply that many high scoring alignments will be observed between closely related members of the two classes. That means that increasing the strictness of the similarity measure (e.g., increasing the number of similar sequences required for two groups to be merged) does not solve the problem. Although testing of the method on synthetic data shows that this problem occurs in fewer than 1% of groups (Hunter, Harris & States, 1992), current databases produce many thousand groups, and overaggregation does occur.

Because the number of overaggregated groups can be expected to be relatively low (a few dozen out of thousands), it is plausible to identify incorrectly merged groups manually. However, this has proven to be a difficult task because of the size and complexity of the individual classes. The overaggregated groups are going to be the largest ones, and these can include several thousand sequences and millions of similarity pairs. We sought a method of representing these large groups that would clarify the sequence relationships within them and that would allow manual reviewers to more readily identify and eliminate false positive hits and falsely merged sequence classes.

A third related problem is that of how to build a total order over the members of each group. In contrast with many classification tasks, the classes or groups formed by our program don't have obvious definitions: each group is a set of protein subsequences that have been found to resemble each other. The similarity relationships within groups are often complex and are not guaranteed to be entirely self-consistent. Each sequence in a given class resembles some other sequence in the class; that is how they ended up together, but this may not be sufficient to generate a complete order of all the sequence segments in a class. In particular, the process of hit assembly prior to clustering allows the possibility of cyclic graph formation during the clustering phase of the HHS algorithm (Hunter, Harris, and States, 1992). If an unambiguous ordering could be generated, this ordering could be used to align all sequences in a group with each other, and we could fill in a consensus frequency matrix that shows the frequency of each amino acid at each position along the set of sequences. If desired, this could be used to represent the class as a single consensus sequence by taking the most common amino acid at each position.

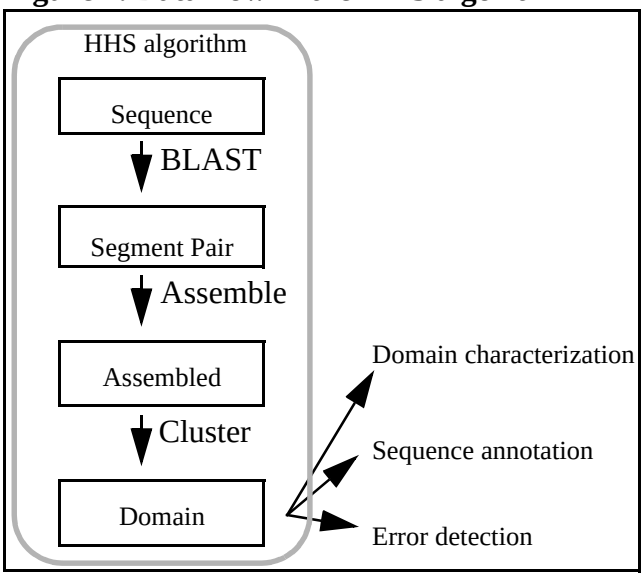
To address these multiple issues, we have developed an alternative classification algorithm which uses a minimal spanning tree of similarity relationships to build sequence classes. This approach dramatically reduces the random access memory requirements needed to implement the classification. In addition, the minimal spanning tree provides a more compact view of sequence relationships within a family that is useful in identifying false hits and removing them from the classification. Finally, it provides a method for unambiguously ordering the sequence segments within a group. In this paper we will describe the minimal spanning tree classification algorithm in greater detail, we will compare classifications generated by this approach with classifications generated storing the full pair similarity set, we will show how this representation can be used to facilitate manual editing of classifications, and we will discuss classification artifacts which arise as a result of using this representation.

Protein Sequence Megaclassification

Although many protein families and functional domains are known, many more have not yet been recognized, and there are errors and disagreements over some of the existing definitions of families and domains. In previous work, we reported on HHS, our algorithm for automatic clustering of large protein sequence databases. Our algorithm was applied to the largest collection of protein sequences that we could assemble, totaling about 17,000,000 amino acids. This classification resulted in the identification of more than 10,000 groups of protein subsequences, including families, domains, and some artifacts.

In this section, we describe the framework we use for classifying these databases, and introduce some of the difficulties involved. Figure 1 shows a data flow representation of the classification process.

Figure 1. Data flow in the HHS algorithm



Database Search and Hit Assembly

Binary similarity judgments are found using BLAST (Altschul et al 1990) to search the molecular sequence database against itself to generate a database of all similar sequence segments. These are assembled into sequence similarity pairs.

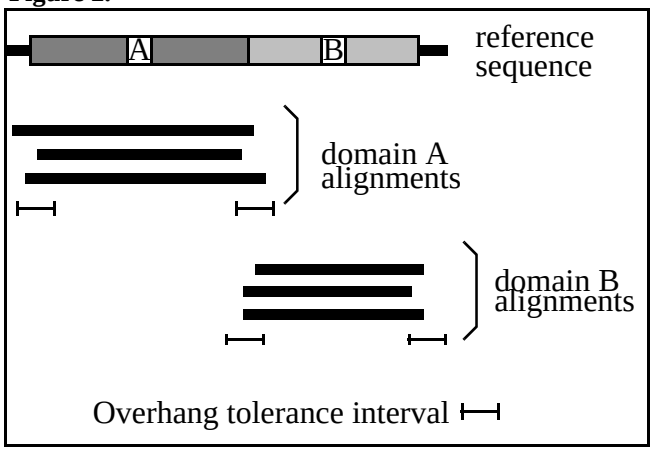
BLAST is a computationally efficient sequence similarity search tool that produces a list of statistically significant ungapped similarity segments for a pair of sequences, called the query and the subject (Altschul et al 1990). We used BLAST to search the molecular sequence database against itself to generate a list of all similar sequence segments. Biologically occurring insertion and deletion mutations may break a single region of similarity into several segments, each of which appears as a separate BLAST hit. The HHS algorithm compensates for this hit fragmentation by assembling together hits that belong to the same region of similarity. Overall, the database search phase of the calculation requires order($N_{sequence}^2$) time, but the database of similarities can be stored and updated incrementally.

Clustering Assembled Hits

After the assembly phase, the BLAST hits have been reduced to a somewhat smaller number of assembled hits. We now want to group these assembled hits into equivalence classes, forming the transitive closure of the pairwise similarity judgments. Hits that should be grouped together may have “ragged ends,” and be of somewhat different lengths.

Hits belong in the same group if they refer to the same region of similarity. In order to be grouped together, two hits should demonstrate significant overlap, but they need not coincide exactly. The non-overlapping portions of the hits are referred to as overhang.

Figure 2.



BLAST hits establish equality relations across proteins; the query and subject portions of a hit are nonrandomly similar. Constructing groups is a matter of building the transitive closure of the similarity judgments provided by BLAST. The ragged ends issue complicates the determination of whether two regions (within a protein) are in fact the same, and, therefore, whether hits that include those two regions should be placed in the same group. Building equivalence classes is then a matter of determining when two hits contain references to the same region. However, there are several complications that make building the transitive closure difficult. BLAST searching is probabilistic and therefore noisy. It can miss regions of similarity, and it can fragment a single region of similarity into multiple hits. Also, BLAST handles approximate matches in the content of the sequences, but it requires exact registration for matching, and its matches have fixed extent. We need to build groups that have approximately matching extents, and where the registration between regions of similarity is not perfect.

HHS address these issues by storing all of the similarity judgments about a sequence segment throughout the clustering calculation. Each new similarity judgement is tested against all of the previously saved similarities to see if any of them are consistent with clustering this new similarity into an existing group. In large groups, much of this similarity data is redundant; since all of the segments in a group are, by definition, related to each other, the ways in which they are related are also similar. The number of similarity judgements that must be saved is proportional to the square of the number of group members. Figure 3. shows that for large groups, the pair similarity dataset can be very large.

Figure 3.

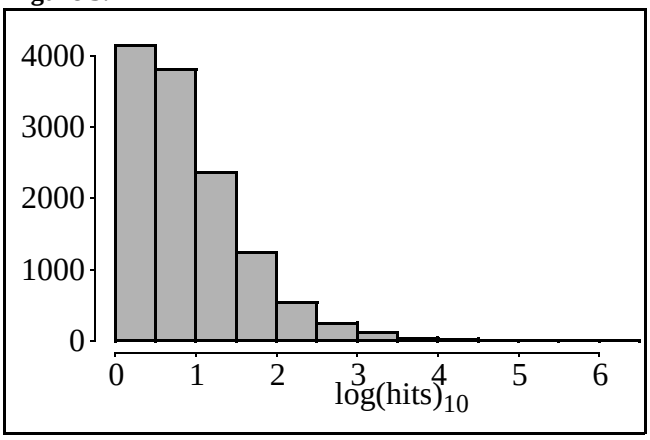
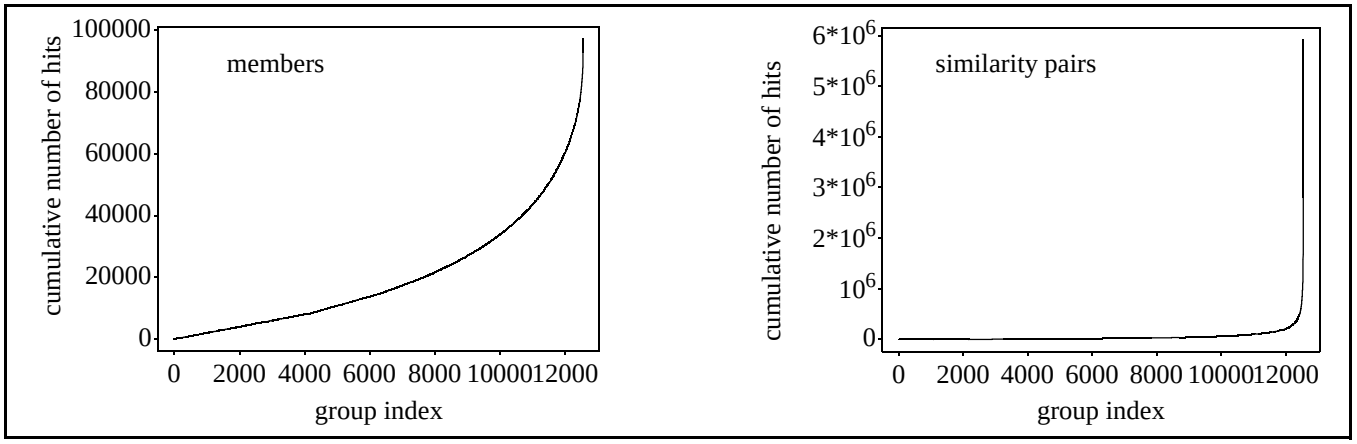


Figure 3. shows the distribution of the number of members per group in classification of the NCBI non-redundant sequence database (NRDB). The X-axis is on a logarithmic scale. The vast majority of groups contain only a few members, while a small number of groups have many members; the largest group contains over 2 million hits.

A small number of very large clusters account for most of the memory required to run the HHS algorithm. As Figure 4. demonstrates, these large groups are inefficient in terms of storage required per sequence.

The figure shows that a few large groups include the vast majority of similarity relationships, and the number of similarity values in these groups is out of proportion to the number of members they contain. This observation led us to modify our clustering method to reduce this redundancy. The modification also has salutary effects on the memory required to cluster and the human comprehensibility of the

Figure 4. Cumulative number of members and similarity pair



results, and provides a mechanism to impose a total order on the members of each group.

Computationally Efficient Class Representation: The HHS/MST Algorithm

It occurred to us that most of the similarity data saved for large groups was redundant, and it was clear that the storage of this excess data was limiting our ability to classify increasingly larger sequence databases. Recall that HHS works by approximate transitive closure: If a to-be-classified sequence is similar to a single member of a group, it is added to that group; if it is similar to members of more than one group, those groups are combined. HHS keeps track of all the similarity relationships between sequence in a group (by definition, there are no similarity relationships outside a group). The hope was that we could reduce this storage requirement by keeping only a subset of the similarity relationships within group, rather than all of them. The most aggressive way to do this is to keep only one similarity relationship for each member of a group.

If we take this aggressive approach, we can throw away all but one of the similarity relationships between that sequence and the other members of its group. Which similarity relationship should be kept? The highest scoring similarity pair for a sequence is an obvious candidate as the relationship to store. There are several reasons for choosing this pair. The sequence pair with the highest similarity score is likely to have diverged least evolutionarily. Since the information content of a sequence alignment declines with evolutionary divergence (Dayhoff et al, 1978; Altschul, 1990), the highest scoring pair is the most informative. Since the information content of the alignment is greatest, the highest scoring pair is likely to give the most accurate estimate for the endpoints of the aligned segments. The highest scoring pair is the similarity pair least likely to miss a region of similarity distal to an insertion or deletion mutation. The number of insertion and deletion mutations in an alignment correlates with the number of substitution mutations; high scoring pairs are likely to have fewer of each. If an insertion or deletion event has occurred in a closely related sequence pair, the distal segments are most likely to be recognizable for the most similar sequence pair.

To recognize a sequence segment as a member of a particular group, the segment must demonstrate similarity to a single member of the group. HHS stores all of the sequence similarity relationships within every class, and thus additional similarity relationships may modify the endpoints of the segment that is assigned to the sequence class. In some cases a new similarity relationship may be consistent with some, but not all, of the similarity hits already in a group. Testing a new hit against only a subset of the similarity data might, therefore, alter the group to which a segment is assigned, but in practice such cases are rare. To test how limiting the amount of similarity data stored might affect classifications, we implemented a classification in which only a single similarity relationship was retained for each new sequence segment.

The memory requirements and computational complexity of the classification algorithm can be analyzed by graph theory. Sequence segments may be considered to be nodes, and similarity relationships may be viewed as edges with the length of an edge being inversely proportional to the similarity score. A sequence class is then a connected graph. Representing the class by storing only the single highest scoring similarity relationship for each new sequence is equivalent to replacing the class relationship graph with a minimal spanning tree. This analogy is valid as long as the reduction to minimal spanning tree representation does not alter the segment endpoints for the sequence segments which are the nodes of the class. In practice, we have found that this condition is usually met. We refer to this algorithm as the minimal spanning tree variant of the HHS algorithm or HHS/MST.

The computational complexity of sequence classification is equivalent to the computational complexity of defining the minimal spanning trees in the forest of graphs defined by the full set of edges. This is a well known problem which has been analyzed in detail. The forest of minimal spanning trees can be generated by sorting the edges by length (computational complexity order(N_{edge} log(N_{edge})), taking them in order and rejecting any edge which generates a cyclic graph. By marking the nodes of each tree, the graph can be tested by cycles in constant time for each additional edge. A new edge will be incorporated into the forest at most once for each node. A new edge may merge two previous trees, and remarking the nodes of the tree will require time proportional to the number of members in either of the two merged groups.

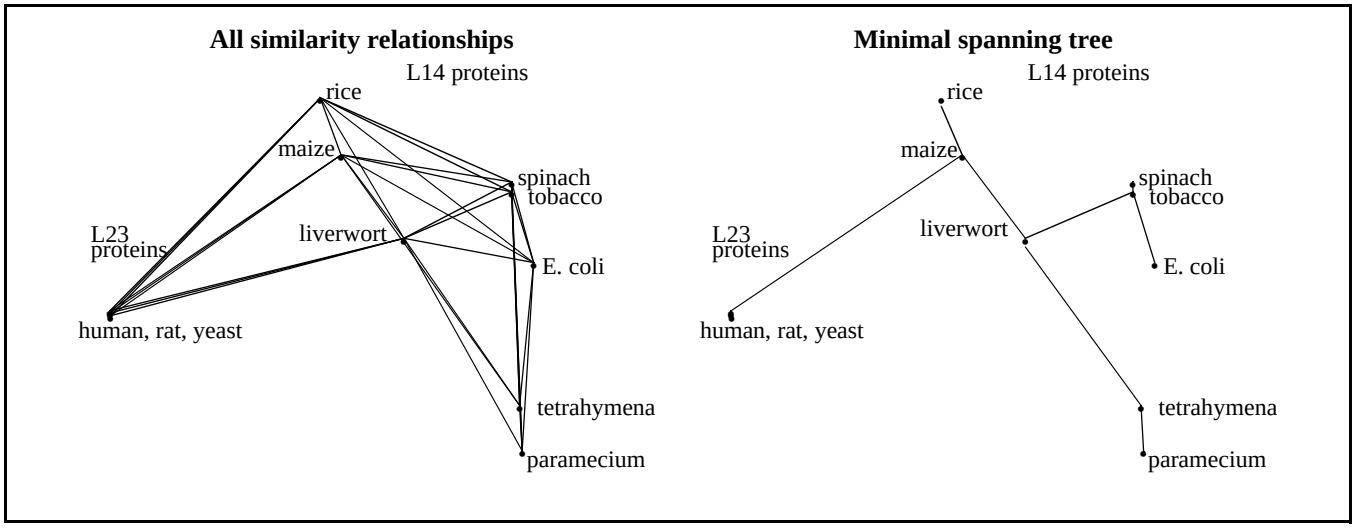
The storage requirements of the minimal spanning tree algorithm are proportional to the number of nodes in the forest. This contrasts with the HHS algorithm, in which the full set of edges is stored. Since the number of edges is proportional to the number of nodes squared, the use of a minimal spanning tree representation results in a dramatic reduction in storage requirements.

Testing the HHS/MST Algorithm

There is a possible problem with this approach. Recall that the extent of similar regions is used to determine whether a new sequence belongs in an existing group. Suppose a group contains a region of sequence A. Suppose further that sequence B has a region that is similar to part of sequence A. Whether sequence B will be added to the group depends on the extent of the overlap (and non-overlap) with A. If we keep only one similarity pair for A, then the extent of A in the group is the one associated with that similarity. If there is a wide range of extents of similarities for sequence A, and sequence B is at a different end of that distribution than the hit that was saved for A, then it is possible that using the HHS/MST method will cause B to fail to be added to the group. This could also cause a pair of groups to fail to be merged together.

We have reason to believe that this is not likely to be practically significant. The highest similarity score gener-

Figure 5. Fully connected and minimal spanning tree representations of a group



ally goes to the longest pair of sequences, so all of the extents will tend to be at the long end of the distribution. We also ran the two methods on the same dataset to compare the differences in classification.

Table 1.

Category	HHS	HHS/MST
families	139	140
mixed	11	10
domains	11	14
Total	161	164

Table 1. compares the results of classifications generated using the HHS and HHS/MST algorithms to classify the sequences in the Brookhaven Protein Data Bank (PDB). For the vast majority of the classes, the members included and the precise endpoints of the domains were exactly the same in the two classifications. There were two exceptions to this. One group that contained both whole protein hits and subprotein hits in the HHS classification was altered in the HHS/MST classification so that the domain hit was eliminated from the group, and this group therefore became a protein family class instead of a mixed domain/family class. In addition, three domain groups were defined in the HHS/MST classification that were not defined in the HHS classification. These additional subgroups resulted from cases where assembled alignments between distantly related members of a group spanned less than the full extent of the domain and were short enough that the endpoint length cutoff used in the classification did not allow these hits to be included in the domain. In these three cases, new groups were created which represent these conserved cores of sequence. These new groups are not simply artifacts of the HHS/MST algorithm because they provide additional information about regional sequence conservation within the parent domains. In this sense, the HHS/MST classification may actually be more informative than the full HHS classification.

The use of a minimal spanning tree representation provides a useful tool for generating subgroup descriptions. This is illustrated in Figure 5. This family contains both L14 and L23 ribosomal proteins. Viewing all of the similarity relationships within the group, it is difficult to distinguish between these two subgroups. When only the hits making up the minimal spanning tree are shown, the tightly clustered L23 subgroup is more apparent. In addition, the phylogenetic relationships of the L14 members are also more easily discerned.

The ability to generate subgroups is also useful in making functional and biological correlates. For example, the tyrosine kinase domains which are found in transmembrane receptors such as the insulin and epidermal growth factor receptors form a distinct subtree in the kinase class. Similarly, the trypsin and elastase subtrees of the serine proteases correlate with substrate preferences.

Using the MST for manual review of the classification

As described above, even when sequences are prefiltered to remove low entropy regions, false positive similarities can generate some overaggregated groups. Trying to manually screen the HHS produced classes and manually repair errors proved infeasible. The large size and complexity of the groups in which false positive hits occurred make it difficult to eliminate them by manual editing. If one false hit was present, there were often other false hits between proteins closely related to those in the initial false hit. Even if a false positive hit can be identified and eliminated, there is no guarantee that all of the false hits have been removed.

In HHS/MST classifications, false hits can easily be identified by searching the path which connected two biologically unrelated proteins in an artifactually merged group. Even for very large groups, only a few dozen edges were typically found. This is illustrated in Table 2 which

Table 2. False positive hit identification in a large group by link tracing

Segment span	Protein
(133 to 341)	MUSNCAMR precursor polypeptide >513435 0 NCA3_MOUSE
(57 to 543)	RATTAG1 axonal glycoprotein
(502 to 610)	RATNCAM14 neural cell adhesion molecule
(1 to 108)	HUMNCAM neural cell adhesion molecule secreted isoform
(485 to 681)	XELNCAMA cell adhesion molecule
(1 to 235)	HUMNCAMA N-CAM >1019770 1 A26883 Neural cell adhesion
(621 to 681)	XELNCAM neural cell adhesion molecule precursor
(82 to 951)	HUMTITINC2 titin
(248 to 940)	A40985 *Projectin - Fruit fly (Drosophila melanogaster) (fragment)
(2515 to 2738)	HUMTITINC3 titin
(251 to 606)	RATMLCK skeletal muscle light chain kinase
(1 to 368)	A05120 Myosin light chain kinase, skeletal muscle
(263 to 608)	RABMLCKA myosin light chain kinase >511296 0 KMLC_RA

shows the hits connecting an immunoglobulin-like neural cell adhesion molecule (NCAM) to a protein kinase domain The table lists a set of segments, each of which was linked by a similarity hit to the segments above and below it in the table. In this example, the hits, or edges, connecting an NCAM to a kinase were traced in the cluster tree. Hits to a set of "titin" proteins were seen to link the NCAMs and the kinases. Titins are large structural proteins (Labeit et al, 1992) containing several regions of low entropy sequence, and XNU was not successful in completely eliminating associated false hits. By manually deleting the hit from HUMTITINC3 to RATMLCK, the kinase domain family was correctly dissociated from the titins and NCAMs. Deleting the hit from XELNCAM to HUMTITINC2 removed the link from the cell adhesion molecules to the titins. The minimal spanning tree representation guaranteed that when a false positive hit was identified and eliminated from the dataset, the falsely merged groups were divided. If they were not, then a cycle would have been present in the graph and the original class representation would not have been a minimal spanning tree.

Finally, using HHS/MST makes it more difficult to detect a certain kind of database artifact that we discovered with HHS. This artifact arises as a result of technical difficulties in cDNA cloning: partial sequences for many proteins were present in the database along with complete sequences for the same proteins. The fragmented nature of these sequences often was not annotated and occasionally was unknown to the contributing author. For HHS classifications, these artifactual groups could be detected using post-classification analysis. The manifestation of the artifact was a pair of two nearly identical groups. In one group, each protein had hits with many other proteins. This was the correct group. In the corresponding artifactual group, one protein (the fragment) had hits with all the other proteins, but because the non-fragment proteins had longer regions of similarity (which are in the true group), none of these other proteins had hits of this size with anything but the fragment. This artifact produced an easily distinguishable star topology in the connectivity graph of the group. In addition, the members of an artifact group, other than the fragment, were all members of a corresponding true family group. In the HHS/MST classifications, automated recognition of these fragment artifacts has proven more difficult because much of the redundancy information used to discriminate between the true and artifactual group has been eliminated.

One of our goals in the use of a minimal spanning tree representation was a significant reduction in the storage requirements for the classification calculation. This was achieved. While classifications of the full NRDB using the HHS algorithm required in excess of 500 MB of RAM memory and required the use of a supercomputer with 512 MB of main memory, classifications using the minimal spanning tree representation could be performed in 60 MB of RAM and can be run easily on available workstations.

Discussion

Scalability of algorithms (Schank, 1991) and the ability to work in large and complex data sets (Almuallim and Dietterich, 1991) are critical issues in machine learning. One of our expressed goals in the sequence megaclassification project has been the application of machine learning and pattern induction techniques to large real world problems. The HHS algorithm was successful in attacking real world problems on datasets of interest to the biological community (Hunter, Harris, and States, 1992), but given the rapid growth in biological sequence data, even the quadratic scaling of memory requirements with dataset size in HHS has proven to be a significant limitation. In addition, some cases of real biological interest have proven to be impossible to analyze on available computing resources. In particu-

lar, although we have been able to classify the current protein sequence databases, much larger databases of nucleic acid sequence are also available; analysis of these datasets using HHS would require several gigabytes of RAM. Much of the progress in computational molecular sequence analysis has resulted from algorithms development. We sought and were successful in deriving an algorithmic solution to the limitation of the HHS approach. The use of the HHS/MST approach will allow nucleic acid sequence datasets and datasets containing both protein and protein coding nucleic acid sequences to be analyzed jointly.

The ability to work with combined protein and nucleic acid sequence databases is of particular importance in dealing with the classification artifacts created by the presence of fragmentary sequences in the database. It may be possible to recognize partial mRNA sequences, but there is no definitive way to recognize partial sequences by protein sequence classification alone. For example, the relationship of the src kinase domain to the kinase domain of the insulin receptor is entirely analogous to the relationship of a partial protein to its full parental sequence, but the proteins in the src/insulin receptor examples are full length sequences and the true relationship is an example of composite protein structure. Messenger RNAs (mRNAs) typically contain a number of distinctive features at their 5' end including ribosome binding sites and initiator codons. If these are absent, it is likely that the mRNA is a partial sequence. By jointly classifying protein and nucleic acid coding regions, such partial sequences can be recognized by criteria which are independent of the classification.

The reduced storage requirements of the HHS/MST algorithm will also be important in keeping pace with the rapidly expanding databases of molecular sequence. Sequence databases have been doubling in size every two years. While computing speed has been able to match this rate of growth to date, the corresponding pair similarity datasets quadruple every two years. The cost of RAM has fallen dramatically in recent years, but it has not fallen fast enough to accommodate the projected space requirements of a quadratic scaling calculation.

Improved ability to manually review and edit groups is an interesting benefit of the HHS/MST representation. In a sense, the requirement of a more compact representation forces a higher level view of the problem. Using the minimal spanning tree representation made it easier to find false positive hits and to manually edit and correct classifications.

The higher level view of the classification generated by HHS/MST also elucidates important relationships between sequences within a group. As the L14/L23 ribosomal protein example illustrated, there may be significant substructure within a group. Reducing group representation to a minimal number of strong similarity relationships highlights this extra level of structure.

Calculation of consensus sequences or sequence profile descriptions for groups is also facilitated by the spanning tree description of groups. When the full segmental pair similarity list is used, ambiguous ordering or alignment relationships were often generated by cycles in the similarity graph describing a group. Since the tree representation contains no cycles, these ambiguities are eliminated. Furthermore, the use of the spanning trees based on the highest scoring similarities optimizes the likelihood that the ordering and alignments defined for the group will be correct.

The tree representation implicit in the HHS/MST problem maps closely to the hierarchic organization of protein domains generated by the evolutionary process of gene duplication and mutation (Patterson, 1988; Felsenstein, 1988; Doolittle, 1992). The HHS/MST algorithm does not retain any notion of a parent-sibling relationship, and all of the nodes in the HHS/MST tree are currently extant proteins. Nevertheless, there is some similarity between the highest scoring links selected by HHS/MST and a true evolutionary tree. Homologs of closely related species are typically found near each other in HHS/MST trees, and the longer branches of HHS/MST trees frequently correspond with ancient divergence events between orthologous proteins. The HHS/MST classification algorithm may be a valuable tool in the exploration of the relationship between protein sequence, structure, and function.

Bibliography

1 Almuallim, H., & Dietterich, T. (1991) Learning with many irrelevant features. In Ninth National Conference on Artificial Intelligence, 547-552. Anaheim, CA: AAAI Press.

2 Altschul SF (1991). Amino acid substitution matrices from an information theoretic perspective. J Mol Biol 219: 555-65.

3 Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). A Basic Local Alignment Search Tool. Journal of Molecular Biology 215:403-410.

4 Dayhoff, M. O., Schwartz, R. M., & Orcutt, B. C. (1978). A Model of Evolutionary Change in Proteins. In Atlas of Protein Sequences and Structure.

5 Doolittle, R. (1992). Reconstructing History with Amino Acid Sequences. Prot. Sci., 1, 191-200.

6 Felsenstein, J. (1988). Phylogenies from molecular sequences: inference and reliability. Annu Rev Genet 22: 521-65.

7 Harris, N., Hunter, L., & States, D. (1992). Megaclassification: Discovering Motifs in Massive Datastreams. In Tenth National Conference on Artificial Intelligence (AAAI), pp. 837-842. San Jose, CA: AAAI Press.

8 Harris, N., States, D. & Hunter, L. (1993). ClassX: A Browsing Tool for Protein Sequence Megaclassifcations. In Proceedings of the Twenty-Sixth Hawaii International

Conference on System Sciences, pp 554-563. Los Alamitos, CA: IEEE Computer Society Press.

9 Hunter, L., Harris, N., & States, D. (1992). Efficient Classification of Massive, Unsegmented Datastreams. In International Machine Learning Workshop, pp. 224-232, Eds. D. Sleeman & P. Edwards, Morgan Kaufman, San Mateo, CA.

10 Labeit, S., Gautel, M., Lakey, A., Trinick, J. (1992) Towards a molecular understanding of titin. EMBO J 11: 1711-6.

11 Patterson, C. (1988) Homology in classical and molecular biology. Mol Biol Evol 5: 603-25

12 Schank, R. C. (1991). Where's the AI? AI Magazine 12(4):38-49.

13 States, D. J. and Claverie, J.M. (1993) Computational Chemistry, in press.

14 Wootton, J. C. & Federhen, S. (1993). Statistics of Local Complexity in Amino Acid Sequences and Sequence Databases. Computers in Chemistry (in the press).

Proceedings of the Intelligent Systems for Molecular Biology (ISMB-93), AAAI Press, Menlo Park, CA

Computationally Efficient Cluster Representation in Molecular Sequence Megaclassification

David J. States*, Nomi L. Harris†, and Lawrence Hunter‡
*Institute for Biomedical Computing, Washington University, St. Louis, MO 63110
†Dept. of Pharmaceutical Chemistry, UCSF, San Francisco, CA 94143
and the ‡Lister Hill Center, National Library of Medicine, Bethesda, MD 20892
states@ibc.wustl.edu nomi@cgl.ucsf.edu hunter@nlm.nih.gov

Abstract

Molecular sequence megaclassification is a technique for automated protein sequence analysis and annotation. Implementation of the method has been limited by the need to store and randomly access a database of all the sequence pair similarities. More than 80,000 protein sequences are now present in the public databases, and the pair similarity data table for the full protein sequence database requires over 1 gigabyte of storage. In this paper we present a computationally efficient representation of groups based on a graph theory approach where sequence clusters are described by a minimal spanning tree of highest scoring similarity pairs. This representation allows a classification of N proteins to be stored in order(N) memory. The use of this minimal spanning tree representation simplifies analysis of groups, the description of group characteristics and the manual correction of artifacts resulting from false hits. The new tree representation also introduces new possibilities for artifact generation in sequence classification. Methods for detecting and removing these artifacts are discussed.

Introduction

Megaclassification of protein sequences is a useful tool for molecular sequence analysis (Hunter, Harris, and States, 1992; Harris, Hunter and States, 1992). Megaclassification involves automatically dividing a large sequence database into a collection of groups of related subsequences. These classes describe the database well with few ambiguously assigned sequence segments and clear distinctions between sequence clusters. Each group of protein subsequences may be associated with a particular function in the cell, and thus the classification can be used to predict the possible function of a novel protein.

The implementation of massive classification is computationally demanding. Although algorithmic speed is important, the main practical limitation is space complexity. We developed a massive classification algorithm, called HHS (Hunter, Harris, States, 1992), that can be used to classify very large sequence databases. HHS assembles sequence groups by using a sequence-comparison tool called BLAST (Altschul et al 1990), which generates pairwise similarity information for all pairs of sequences in the database. As the groups are assembled, the pair similarity database must be available for random access. This pair database requires over 500 megabytes of storage for the current sequence collections and grows with the square of the number of sequences. To make massive classification a feasible calculation, the pair information must reside in RAM; the five order of magnitude time penalty required to access magnetic disks is prohibitively slow. These space requirements are the main impediment to work in this area, so we sought to develop alternative algorithms for massive classification with reduced memory requirements.

A second issue that arises in the practical use of the HHS algorithm is its susceptibility to overaggregation due to false positive similarity judgments. Our algorithm does an approximate transitive closure on the similarity judgments, and a single false positive is enough to merge two unrelated groups of sequences. We take a variety of steps in the clustering algorithm to avoid this problem, including the use of sequence filters that eliminate repetitive and low-entropy sequences, such as XNU (States and Claverie, 1993) and seg (Wootton and Federhen, 1992). The use of these filters dramatically reduces the number of high scoring false positive alignments generated in the course of a sequence database self-comparison. However, these filters do not completely eliminate false positives. The problem is compounded by the fact that false positives often occur in sets. If a high scoring alignment is seen between two members of biologically unrelated sequence classes, sequence correlations within the classes often imply that many high scoring alignments will be observed between closely related members of the two classes. That means that increasing the strictness of the similarity measure (e.g., increasing the number of similar sequences required for two groups to be merged) does not solve the problem. Although testing of the method on synthetic data shows that this problem occurs in fewer than 1% of groups (Hunter, Harris & States, 1992), current databases produce many thousand groups, and overaggregation does occur.

Because the number of overaggregated groups can be expected to be relatively low (a few dozen out of thousands), it is plausible to identify incorrectly merged groups manually. However, this has proven to be a difficult task because of the size and complexity of the individual classes. The overaggregated groups are going to be the largest ones, and these can include several thousand sequences and millions of similarity pairs. We sought a method of representing these large groups that would clarify the sequence relationships within them and that would allow manual reviewers to more readily identify and eliminate false positive hits and falsely merged sequence classes.

A third related problem is that of how to build a total order over the members of each group. In contrast with many classification tasks, the classes or groups formed by our program don't have obvious definitions: each group is a set of protein subsequences that have been found to resemble each other. The similarity relationships within groups are often complex and are not guaranteed to be entirely self-consistent. Each sequence in a given class resembles some other sequence in the class; that is how they ended up together, but this may not be sufficient to generate a complete order of all the sequence segments in a class. In particular, the process of hit assembly prior to clustering allows the possibility of cyclic graph formation during the clustering phase of the HHS algorithm (Hunter, Harris, and States, 1992). If an unambiguous ordering could be generated, this ordering could be used to align all sequences in a group with each other, and we could fill in a consensus frequency matrix that shows the frequency of each amino acid at each position along the set of sequences. If desired, this could be used to represent the class as a single consensus sequence by taking the most common amino acid at each position.

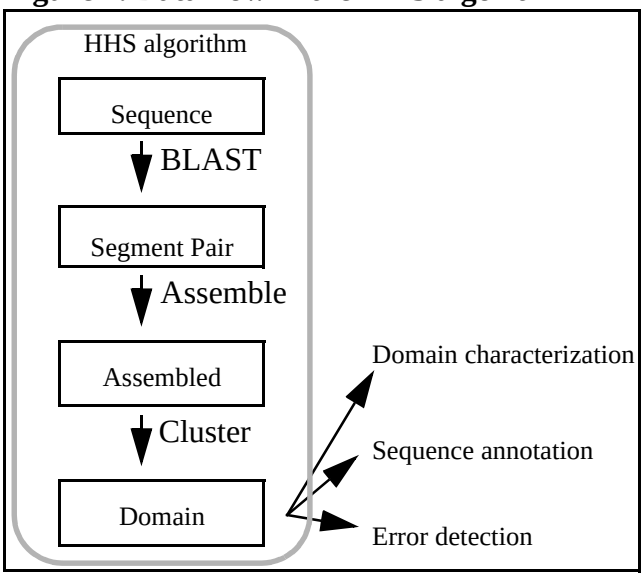
To address these multiple issues, we have developed an alternative classification algorithm which uses a minimal spanning tree of similarity relationships to build sequence classes. This approach dramatically reduces the random access memory requirements needed to implement the classification. In addition, the minimal spanning tree provides a more compact view of sequence relationships within a family that is useful in identifying false hits and removing them from the classification. Finally, it provides a method for unambiguously ordering the sequence segments within a group. In this paper we will describe the minimal spanning tree classification algorithm in greater detail, we will compare classifications generated by this approach with classifications generated storing the full pair similarity set, we will show how this representation can be used to facilitate manual editing of classifications, and we will discuss classification artifacts which arise as a result of using this representation.

Protein Sequence Megaclassification

Although many protein families and functional domains are known, many more have not yet been recognized, and there are errors and disagreements over some of the existing definitions of families and domains. In previous work, we reported on HHS, our algorithm for automatic clustering of large protein sequence databases. Our algorithm was applied to the largest collection of protein sequences that we could assemble, totaling about 17,000,000 amino acids. This classification resulted in the identification of more than 10,000 groups of protein subsequences, including families, domains, and some artifacts.

In this section, we describe the framework we use for classifying these databases, and introduce some of the difficulties involved. Figure 1 shows a data flow representation of the classification process.

Figure 1. Data flow in the HHS algorithm



Database Search and Hit Assembly

Binary similarity judgments are found using BLAST (Altschul et al 1990) to search the molecular sequence database against itself to generate a database of all similar sequence segments. These are assembled into sequence similarity pairs.

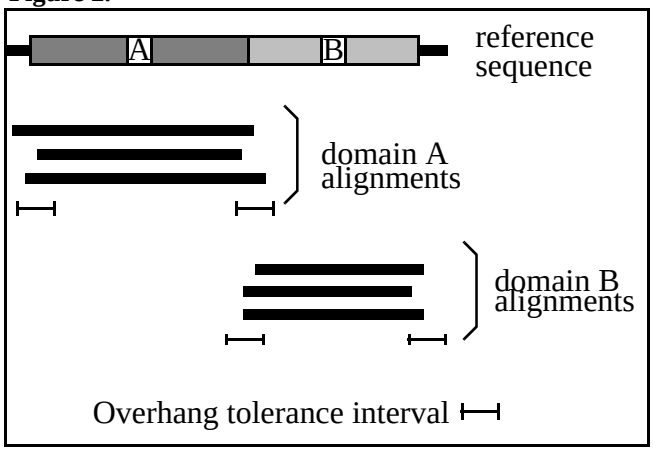
BLAST is a computationally efficient sequence similarity search tool that produces a list of statistically significant ungapped similarity segments for a pair of sequences, called the query and the subject (Altschul et al 1990). We used BLAST to search the molecular sequence database against itself to generate a list of all similar sequence segments. Biologically occurring insertion and deletion mutations may break a single region of similarity into several segments, each of which appears as a separate BLAST hit. The HHS algorithm compensates for this hit fragmentation by assembling together hits that belong to the same region of similarity. Overall, the database search phase of the calculation requires order($N_{sequence}^2$) time, but the database of similarities can be stored and updated incrementally.

Clustering Assembled Hits

After the assembly phase, the BLAST hits have been reduced to a somewhat smaller number of assembled hits. We now want to group these assembled hits into equivalence classes, forming the transitive closure of the pairwise similarity judgments. Hits that should be grouped together may have “ragged ends,” and be of somewhat different lengths.

Hits belong in the same group if they refer to the same region of similarity. In order to be grouped together, two hits should demonstrate significant overlap, but they need not coincide exactly. The non-overlapping portions of the hits are referred to as overhang.

Figure 2.



BLAST hits establish equality relations across proteins; the query and subject portions of a hit are nonrandomly similar. Constructing groups is a matter of building the transitive closure of the similarity judgments provided by BLAST. The ragged ends issue complicates the determination of whether two regions (within a protein) are in fact the same, and, therefore, whether hits that include those two regions should be placed in the same group. Building equivalence classes is then a matter of determining when two hits contain references to the same region. However, there are several complications that make building the transitive closure difficult. BLAST searching is probabilistic and therefore noisy. It can miss regions of similarity, and it can fragment a single region of similarity into multiple hits. Also, BLAST handles approximate matches in the content of the sequences, but it requires exact registration for matching, and its matches have fixed extent. We need to build groups that have approximately matching extents, and where the registration between regions of similarity is not perfect.

HHS address these issues by storing all of the similarity judgments about a sequence segment throughout the clustering calculation. Each new similarity judgement is tested against all of the previously saved similarities to see if any of them are consistent with clustering this new similarity into an existing group. In large groups, much of this similarity data is redundant; since all of the segments in a group are, by definition, related to each other, the ways in which they are related are also similar. The number of similarity judgements that must be saved is proportional to the square of the number of group members. Figure 3. shows that for large groups, the pair similarity dataset can be very large.

Figure 3.

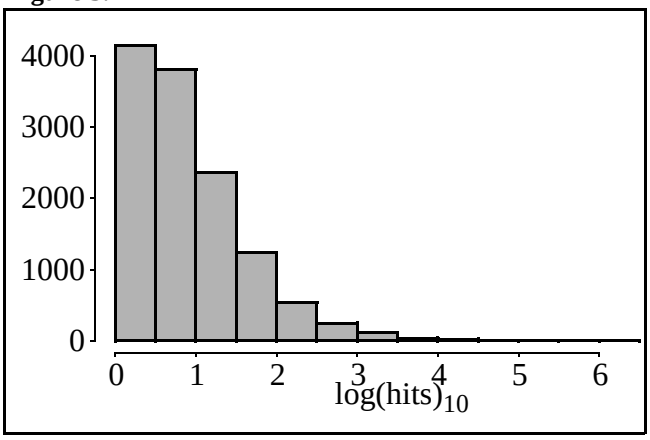
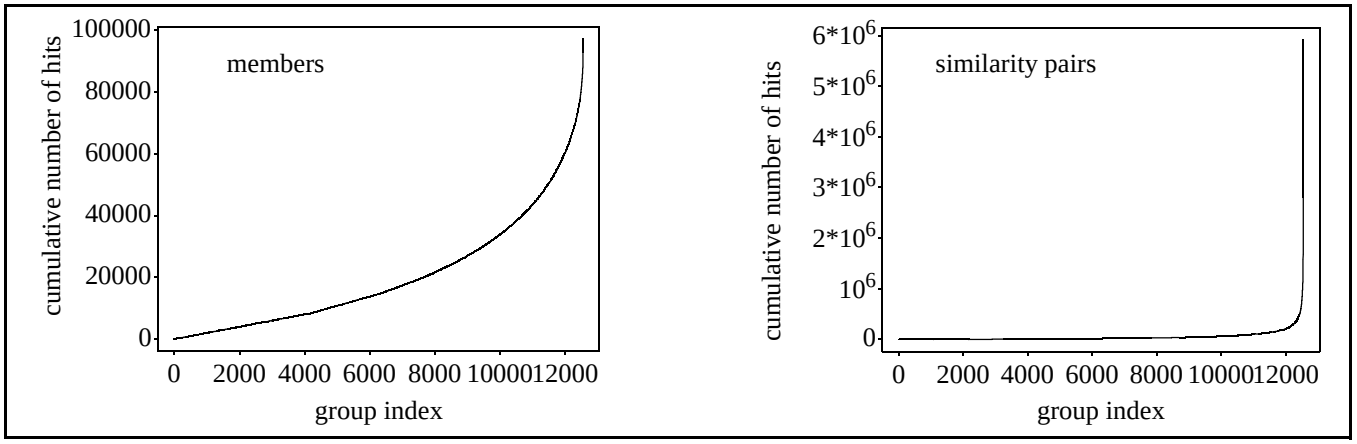


Figure 3. shows the distribution of the number of members per group in classification of the NCBI non-redundant sequence database (NRDB). The X-axis is on a logarithmic scale. The vast majority of groups contain only a few members, while a small number of groups have many members; the largest group contains over 2 million hits.

A small number of very large clusters account for most of the memory required to run the HHS algorithm. As Figure 4. demonstrates, these large groups are inefficient in terms of storage required per sequence.

The figure shows that a few large groups include the vast majority of similarity relationships, and the number of similarity values in these groups is out of proportion to the number of members they contain. This observation led us to modify our clustering method to reduce this redundancy. The modification also has salutary effects on the memory required to cluster and the human comprehensibility of the

Figure 4. Cumulative number of members and similarity pair



results, and provides a mechanism to impose a total order on the members of each group.

Computationally Efficient Class Representation: The HHS/MST Algorithm

It occurred to us that most of the similarity data saved for large groups was redundant, and it was clear that the storage of this excess data was limiting our ability to classify increasingly larger sequence databases. Recall that HHS works by approximate transitive closure: If a to-be-classified sequence is similar to a single member of a group, it is added to that group; if it is similar to members of more than one group, those groups are combined. HHS keeps track of all the similarity relationships between sequence in a group (by definition, there are no similarity relationships outside a group). The hope was that we could reduce this storage requirement by keeping only a subset of the similarity relationships within group, rather than all of them. The most aggressive way to do this is to keep only one similarity relationship for each member of a group.

If we take this aggressive approach, we can throw away all but one of the similarity relationships between that sequence and the other members of its group. Which similarity relationship should be kept? The highest scoring similarity pair for a sequence is an obvious candidate as the relationship to store. There are several reasons for choosing this pair. The sequence pair with the highest similarity score is likely to have diverged least evolutionarily. Since the information content of a sequence alignment declines with evolutionary divergence (Dayhoff et al, 1978; Altschul, 1990), the highest scoring pair is the most informative. Since the information content of the alignment is greatest, the highest scoring pair is likely to give the most accurate estimate for the endpoints of the aligned segments. The highest scoring pair is the similarity pair least likely to miss a region of similarity distal to an insertion or deletion mutation. The number of insertion and deletion mutations in an alignment correlates with the number of substitution mutations; high scoring pairs are likely to have fewer of each. If an insertion or deletion event has occurred in a closely related sequence pair, the distal segments are most likely to be recognizable for the most similar sequence pair.

To recognize a sequence segment as a member of a particular group, the segment must demonstrate similarity to a single member of the group. HHS stores all of the sequence similarity relationships within every class, and thus additional similarity relationships may modify the endpoints of the segment that is assigned to the sequence class. In some cases a new similarity relationship may be consistent with some, but not all, of the similarity hits already in a group. Testing a new hit against only a subset of the similarity data might, therefore, alter the group to which a segment is assigned, but in practice such cases are rare. To test how limiting the amount of similarity data stored might affect classifications, we implemented a classification in which only a single similarity relationship was retained for each new sequence segment.

The memory requirements and computational complexity of the classification algorithm can be analyzed by graph theory. Sequence segments may be considered to be nodes, and similarity relationships may be viewed as edges with the length of an edge being inversely proportional to the similarity score. A sequence class is then a connected graph. Representing the class by storing only the single highest scoring similarity relationship for each new sequence is equivalent to replacing the class relationship graph with a minimal spanning tree. This analogy is valid as long as the reduction to minimal spanning tree representation does not alter the segment endpoints for the sequence segments which are the nodes of the class. In practice, we have found that this condition is usually met. We refer to this algorithm as the minimal spanning tree variant of the HHS algorithm or HHS/MST.

The computational complexity of sequence classification is equivalent to the computational complexity of defining the minimal spanning trees in the forest of graphs defined by the full set of edges. This is a well known problem which has been analyzed in detail. The forest of minimal spanning trees can be generated by sorting the edges by length (computational complexity order(N_{edge} log(N_{edge})), taking them in order and rejecting any edge which generates a cyclic graph. By marking the nodes of each tree, the graph can be tested by cycles in constant time for each additional edge. A new edge will be incorporated into the forest at most once for each node. A new edge may merge two previous trees, and remarking the nodes of the tree will require time proportional to the number of members in either of the two merged groups.

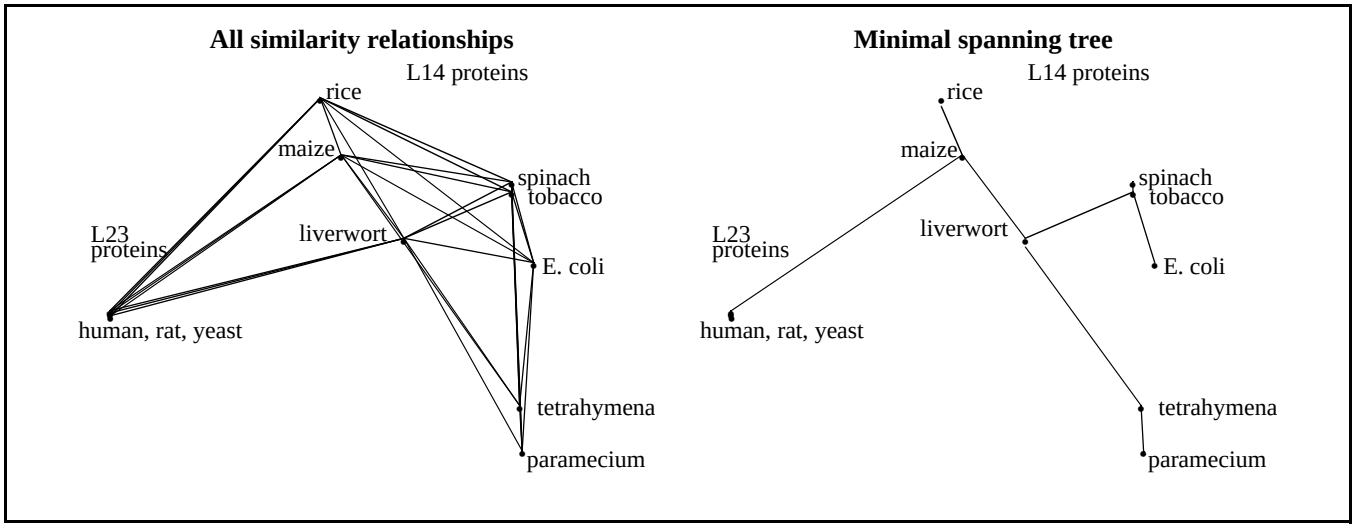
The storage requirements of the minimal spanning tree algorithm are proportional to the number of nodes in the forest. This contrasts with the HHS algorithm, in which the full set of edges is stored. Since the number of edges is proportional to the number of nodes squared, the use of a minimal spanning tree representation results in a dramatic reduction in storage requirements.

Testing the HHS/MST Algorithm

There is a possible problem with this approach. Recall that the extent of similar regions is used to determine whether a new sequence belongs in an existing group. Suppose a group contains a region of sequence A. Suppose further that sequence B has a region that is similar to part of sequence A. Whether sequence B will be added to the group depends on the extent of the overlap (and non-overlap) with A. If we keep only one similarity pair for A, then the extent of A in the group is the one associated with that similarity. If there is a wide range of extents of similarities for sequence A, and sequence B is at a different end of that distribution than the hit that was saved for A, then it is possible that using the HHS/MST method will cause B to fail to be added to the group. This could also cause a pair of groups to fail to be merged together.

We have reason to believe that this is not likely to be practically significant. The highest similarity score gener-

Figure 5. Fully connected and minimal spanning tree representations of a group



ally goes to the longest pair of sequences, so all of the extents will tend to be at the long end of the distribution. We also ran the two methods on the same dataset to compare the differences in classification.

Table 1.

Category	HHS	HHS/MST
families	139	140
mixed	11	10
domains	11	14
Total	161	164

Table 1. compares the results of classifications generated using the HHS and HHS/MST algorithms to classify the sequences in the Brookhaven Protein Data Bank (PDB). For the vast majority of the classes, the members included and the precise endpoints of the domains were exactly the same in the two classifications. There were two exceptions to this. One group that contained both whole protein hits and subprotein hits in the HHS classification was altered in the HHS/MST classification so that the domain hit was eliminated from the group, and this group therefore became a protein family class instead of a mixed domain/family class. In addition, three domain groups were defined in the HHS/MST classification that were not defined in the HHS classification. These additional subgroups resulted from cases where assembled alignments between distantly related members of a group spanned less than the full extent of the domain and were short enough that the endpoint length cutoff used in the classification did not allow these hits to be included in the domain. In these three cases, new groups were created which represent these conserved cores of sequence. These new groups are not simply artifacts of the HHS/MST algorithm because they provide additional information about regional sequence conservation within the parent domains. In this sense, the HHS/MST classification may actually be more informative than the full HHS classification.

The use of a minimal spanning tree representation provides a useful tool for generating subgroup descriptions. This is illustrated in Figure 5. This family contains both L14 and L23 ribosomal proteins. Viewing all of the similarity relationships within the group, it is difficult to distinguish between these two subgroups. When only the hits making up the minimal spanning tree are shown, the tightly clustered L23 subgroup is more apparent. In addition, the phylogenetic relationships of the L14 members are also more easily discerned.

The ability to generate subgroups is also useful in making functional and biological correlates. For example, the tyrosine kinase domains which are found in transmembrane receptors such as the insulin and epidermal growth factor receptors form a distinct subtree in the kinase class. Similarly, the trypsin and elastase subtrees of the serine proteases correlate with substrate preferences.

Using the MST for manual review of the classification

As described above, even when sequences are prefiltered to remove low entropy regions, false positive similarities can generate some overaggregated groups. Trying to manually screen the HHS produced classes and manually repair errors proved infeasible. The large size and complexity of the groups in which false positive hits occurred make it difficult to eliminate them by manual editing. If one false hit was present, there were often other false hits between proteins closely related to those in the initial false hit. Even if a false positive hit can be identified and eliminated, there is no guarantee that all of the false hits have been removed.

In HHS/MST classifications, false hits can easily be identified by searching the path which connected two biologically unrelated proteins in an artifactually merged group. Even for very large groups, only a few dozen edges were typically found. This is illustrated in Table 2 which

Table 2. False positive hit identification in a large group by link tracing

Segment span	Protein
(133 to 341)	MUSNCAMR precursor polypeptide >513435 0 NCA3_MOUSE
(57 to 543)	RATTAG1 axonal glycoprotein
(502 to 610)	RATNCAM14 neural cell adhesion molecule
(1 to 108)	HUMNCAM neural cell adhesion molecule secreted isoform
(485 to 681)	XELNCAMA cell adhesion molecule
(1 to 235)	HUMNCAMA N-CAM >1019770 1 A26883 Neural cell adhesion
(621 to 681)	XELNCAM neural cell adhesion molecule precursor
(82 to 951)	HUMTITINC2 titin
(248 to 940)	A40985 *Projectin - Fruit fly (Drosophila melanogaster) (fragment)
(2515 to 2738)	HUMTITINC3 titin
(251 to 606)	RATMLCK skeletal muscle light chain kinase
(1 to 368)	A05120 Myosin light chain kinase, skeletal muscle
(263 to 608)	RABMLCKA myosin light chain kinase >511296 0 KMLC_RA

shows the hits connecting an immunoglobulin-like neural cell adhesion molecule (NCAM) to a protein kinase domain The table lists a set of segments, each of which was linked by a similarity hit to the segments above and below it in the table. In this example, the hits, or edges, connecting an NCAM to a kinase were traced in the cluster tree. Hits to a set of "titin" proteins were seen to link the NCAMs and the kinases. Titins are large structural proteins (Labeit et al, 1992) containing several regions of low entropy sequence, and XNU was not successful in completely eliminating associated false hits. By manually deleting the hit from HUMTITINC3 to RATMLCK, the kinase domain family was correctly dissociated from the titins and NCAMs. Deleting the hit from XELNCAM to HUMTITINC2 removed the link from the cell adhesion molecules to the titins. The minimal spanning tree representation guaranteed that when a false positive hit was identified and eliminated from the dataset, the falsely merged groups were divided. If they were not, then a cycle would have been present in the graph and the original class representation would not have been a minimal spanning tree.

Finally, using HHS/MST makes it more difficult to detect a certain kind of database artifact that we discovered with HHS. This artifact arises as a result of technical difficulties in cDNA cloning: partial sequences for many proteins were present in the database along with complete sequences for the same proteins. The fragmented nature of these sequences often was not annotated and occasionally was unknown to the contributing author. For HHS classifications, these artifactual groups could be detected using post-classification analysis. The manifestation of the artifact was a pair of two nearly identical groups. In one group, each protein had hits with many other proteins. This was the correct group. In the corresponding artifactual group, one protein (the fragment) had hits with all the other proteins, but because the non-fragment proteins had longer regions of similarity (which are in the true group), none of these other proteins had hits of this size with anything but the fragment. This artifact produced an easily distinguishable star topology in the connectivity graph of the group. In addition, the members of an artifact group, other than the fragment, were all members of a corresponding true family group. In the HHS/MST classifications, automated recognition of these fragment artifacts has proven more difficult because much of the redundancy information used to discriminate between the true and artifactual group has been eliminated.

One of our goals in the use of a minimal spanning tree representation was a significant reduction in the storage requirements for the classification calculation. This was achieved. While classifications of the full NRDB using the HHS algorithm required in excess of 500 MB of RAM memory and required the use of a supercomputer with 512 MB of main memory, classifications using the minimal spanning tree representation could be performed in 60 MB of RAM and can be run easily on available workstations.

Discussion

Scalability of algorithms (Schank, 1991) and the ability to work in large and complex data sets (Almuallim and Dietterich, 1991) are critical issues in machine learning. One of our expressed goals in the sequence megaclassification project has been the application of machine learning and pattern induction techniques to large real world problems. The HHS algorithm was successful in attacking real world problems on datasets of interest to the biological community (Hunter, Harris, and States, 1992), but given the rapid growth in biological sequence data, even the quadratic scaling of memory requirements with dataset size in HHS has proven to be a significant limitation. In addition, some cases of real biological interest have proven to be impossible to analyze on available computing resources. In particu-

lar, although we have been able to classify the current protein sequence databases, much larger databases of nucleic acid sequence are also available; analysis of these datasets using HHS would require several gigabytes of RAM. Much of the progress in computational molecular sequence analysis has resulted from algorithms development. We sought and were successful in deriving an algorithmic solution to the limitation of the HHS approach. The use of the HHS/MST approach will allow nucleic acid sequence datasets and datasets containing both protein and protein coding nucleic acid sequences to be analyzed jointly.

The ability to work with combined protein and nucleic acid sequence databases is of particular importance in dealing with the classification artifacts created by the presence of fragmentary sequences in the database. It may be possible to recognize partial mRNA sequences, but there is no definitive way to recognize partial sequences by protein sequence classification alone. For example, the relationship of the src kinase domain to the kinase domain of the insulin receptor is entirely analogous to the relationship of a partial protein to its full parental sequence, but the proteins in the src/insulin receptor examples are full length sequences and the true relationship is an example of composite protein structure. Messenger RNAs (mRNAs) typically contain a number of distinctive features at their 5' end including ribosome binding sites and initiator codons. If these are absent, it is likely that the mRNA is a partial sequence. By jointly classifying protein and nucleic acid coding regions, such partial sequences can be recognized by criteria which are independent of the classification.

The reduced storage requirements of the HHS/MST algorithm will also be important in keeping pace with the rapidly expanding databases of molecular sequence. Sequence databases have been doubling in size every two years. While computing speed has been able to match this rate of growth to date, the corresponding pair similarity datasets quadruple every two years. The cost of RAM has fallen dramatically in recent years, but it has not fallen fast enough to accommodate the projected space requirements of a quadratic scaling calculation.

Improved ability to manually review and edit groups is an interesting benefit of the HHS/MST representation. In a sense, the requirement of a more compact representation forces a higher level view of the problem. Using the minimal spanning tree representation made it easier to find false positive hits and to manually edit and correct classifications.

The higher level view of the classification generated by HHS/MST also elucidates important relationships between sequences within a group. As the L14/L23 ribosomal protein example illustrated, there may be significant substructure within a group. Reducing group representation to a minimal number of strong similarity relationships highlights this extra level of structure.

Calculation of consensus sequences or sequence profile descriptions for groups is also facilitated by the spanning tree description of groups. When the full segmental pair similarity list is used, ambiguous ordering or alignment relationships were often generated by cycles in the similarity graph describing a group. Since the tree representation contains no cycles, these ambiguities are eliminated. Furthermore, the use of the spanning trees based on the highest scoring similarities optimizes the likelihood that the ordering and alignments defined for the group will be correct.

The tree representation implicit in the HHS/MST problem maps closely to the hierarchic organization of protein domains generated by the evolutionary process of gene duplication and mutation (Patterson, 1988; Felsenstein, 1988; Doolittle, 1992). The HHS/MST algorithm does not retain any notion of a parent-sibling relationship, and all of the nodes in the HHS/MST tree are currently extant proteins. Nevertheless, there is some similarity between the highest scoring links selected by HHS/MST and a true evolutionary tree. Homologs of closely related species are typically found near each other in HHS/MST trees, and the longer branches of HHS/MST trees frequently correspond with ancient divergence events between orthologous proteins. The HHS/MST classification algorithm may be a valuable tool in the exploration of the relationship between protein sequence, structure, and function.

Bibliography

1 Almuallim, H., & Dietterich, T. (1991) Learning with many irrelevant features. In Ninth National Conference on Artificial Intelligence, 547-552. Anaheim, CA: AAAI Press.

2 Altschul SF (1991). Amino acid substitution matrices from an information theoretic perspective. J Mol Biol 219: 555-65.

3 Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). A Basic Local Alignment Search Tool. Journal of Molecular Biology 215:403-410.

4 Dayhoff, M. O., Schwartz, R. M., & Orcutt, B. C. (1978). A Model of Evolutionary Change in Proteins. In Atlas of Protein Sequences and Structure.

5 Doolittle, R. (1992). Reconstructing History with Amino Acid Sequences. Prot. Sci., 1, 191-200.

6 Felsenstein, J. (1988). Phylogenies from molecular sequences: inference and reliability. Annu Rev Genet 22: 521-65.

7 Harris, N., Hunter, L., & States, D. (1992). Megaclassification: Discovering Motifs in Massive Datastreams. In Tenth National Conference on Artificial Intelligence (AAAI), pp. 837-842. San Jose, CA: AAAI Press.

8 Harris, N., States, D. & Hunter, L. (1993). ClassX: A Browsing Tool for Protein Sequence Megaclassifcations. In Proceedings of the Twenty-Sixth Hawaii International

Conference on System Sciences, pp 554-563. Los Alamitos, CA: IEEE Computer Society Press.

9 Hunter, L., Harris, N., & States, D. (1992). Efficient Classification of Massive, Unsegmented Datastreams. In International Machine Learning Workshop, pp. 224-232, Eds. D. Sleeman & P. Edwards, Morgan Kaufman, San Mateo, CA.

10 Labeit, S., Gautel, M., Lakey, A., Trinick, J. (1992) Towards a molecular understanding of titin. EMBO J 11: 1711-6.

11 Patterson, C. (1988) Homology in classical and molecular biology. Mol Biol Evol 5: 603-25

12 Schank, R. C. (1991). Where's the AI? AI Magazine 12(4):38-49.

13 States, D. J. and Claverie, J.M. (1993) Computational Chemistry, in press.

14 Wootton, J. C. & Federhen, S. (1993). Statistics of Local Complexity in Amino Acid Sequences and Sequence Databases. Computers in Chemistry (in the press).

Proceedings of the Intelligent Systems for Molecular Biology (ISMB-93), AAAI Press, Menlo Park, CA

Computationally Efficient Cluster Representation in Molecular Sequence Megaclassification

David J. States*, Nomi L. Harris†, and Lawrence Hunter‡
*Institute for Biomedical Computing, Washington University, St. Louis, MO 63110
†Dept. of Pharmaceutical Chemistry, UCSF, San Francisco, CA 94143
and the ‡Lister Hill Center, National Library of Medicine, Bethesda, MD 20892
states@ibc.wustl.edu nomi@cgl.ucsf.edu hunter@nlm.nih.gov

Abstract

Molecular sequence megaclassification is a technique for automated protein sequence analysis and annotation. Implementation of the method has been limited by the need to store and randomly access a database of all the sequence pair similarities. More than 80,000 protein sequences are now present in the public databases, and the pair similarity data table for the full protein sequence database requires over 1 gigabyte of storage. In this paper we present a computationally efficient representation of groups based on a graph theory approach where sequence clusters are described by a minimal spanning tree of highest scoring similarity pairs. This representation allows a classification of N proteins to be stored in order(N) memory. The use of this minimal spanning tree representation simplifies analysis of groups, the description of group characteristics and the manual correction of artifacts resulting from false hits. The new tree representation also introduces new possibilities for artifact generation in sequence classification. Methods for detecting and removing these artifacts are discussed.

Introduction

Megaclassification of protein sequences is a useful tool for molecular sequence analysis (Hunter, Harris, and States, 1992; Harris, Hunter and States, 1992). Megaclassification involves automatically dividing a large sequence database into a collection of groups of related subsequences. These classes describe the database well with few ambiguously assigned sequence segments and clear distinctions between sequence clusters. Each group of protein subsequences may be associated with a particular function in the cell, and thus the classification can be used to predict the possible function of a novel protein.

The implementation of massive classification is computationally demanding. Although algorithmic speed is important, the main practical limitation is space complexity. We developed a massive classification algorithm, called HHS (Hunter, Harris, States, 1992), that can be used to classify very large sequence databases. HHS assembles sequence groups by using a sequence-comparison tool called BLAST (Altschul et al 1990), which generates pairwise similarity information for all pairs of sequences in the database. As the groups are assembled, the pair similarity database must be available for random access. This pair database requires over 500 megabytes of storage for the current sequence collections and grows with the square of the number of sequences. To make massive classification a feasible calculation, the pair information must reside in RAM; the five order of magnitude time penalty required to access magnetic disks is prohibitively slow. These space requirements are the main impediment to work in this area, so we sought to develop alternative algorithms for massive classification with reduced memory requirements.

A second issue that arises in the practical use of the HHS algorithm is its susceptibility to overaggregation due to false positive similarity judgments. Our algorithm does an approximate transitive closure on the similarity judgments, and a single false positive is enough to merge two unrelated groups of sequences. We take a variety of steps in the clustering algorithm to avoid this problem, including the use of sequence filters that eliminate repetitive and low-entropy sequences, such as XNU (States and Claverie, 1993) and seg (Wootton and Federhen, 1992). The use of these filters dramatically reduces the number of high scoring false positive alignments generated in the course of a sequence database self-comparison. However, these filters do not completely eliminate false positives. The problem is compounded by the fact that false positives often occur in sets. If a high scoring alignment is seen between two members of biologically unrelated sequence classes, sequence correlations within the classes often imply that many high scoring alignments will be observed between closely related members of the two classes. That means that increasing the strictness of the similarity measure (e.g., increasing the number of similar sequences required for two groups to be merged) does not solve the problem. Although testing of the method on synthetic data shows that this problem occurs in fewer than 1% of groups (Hunter, Harris & States, 1992), current databases produce many thousand groups, and overaggregation does occur.

Because the number of overaggregated groups can be expected to be relatively low (a few dozen out of thousands), it is plausible to identify incorrectly merged groups manually. However, this has proven to be a difficult task because of the size and complexity of the individual classes. The overaggregated groups are going to be the largest ones, and these can include several thousand sequences and millions of similarity pairs. We sought a method of representing these large groups that would clarify the sequence relationships within them and that would allow manual reviewers to more readily identify and eliminate false positive hits and falsely merged sequence classes.

A third related problem is that of how to build a total order over the members of each group. In contrast with many classification tasks, the classes or groups formed by our program don't have obvious definitions: each group is a set of protein subsequences that have been found to resemble each other. The similarity relationships within groups are often complex and are not guaranteed to be entirely self-consistent. Each sequence in a given class resembles some other sequence in the class; that is how they ended up together, but this may not be sufficient to generate a complete order of all the sequence segments in a class. In particular, the process of hit assembly prior to clustering allows the possibility of cyclic graph formation during the clustering phase of the HHS algorithm (Hunter, Harris, and States, 1992). If an unambiguous ordering could be generated, this ordering could be used to align all sequences in a group with each other, and we could fill in a consensus frequency matrix that shows the frequency of each amino acid at each position along the set of sequences. If desired, this could be used to represent the class as a single consensus sequence by taking the most common amino acid at each position.

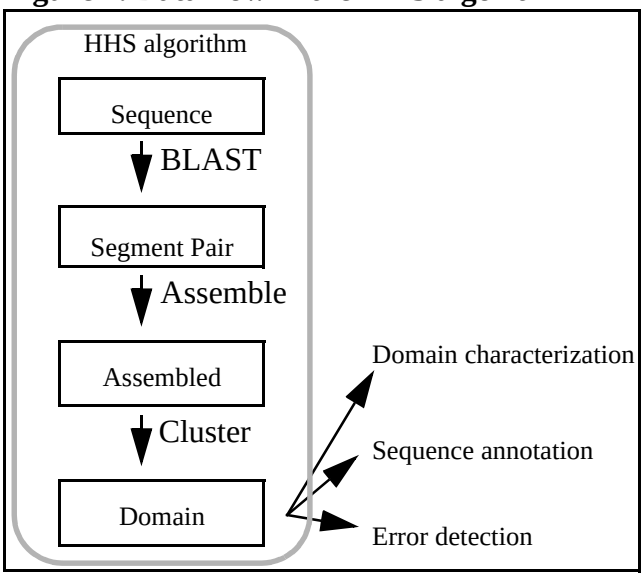
To address these multiple issues, we have developed an alternative classification algorithm which uses a minimal spanning tree of similarity relationships to build sequence classes. This approach dramatically reduces the random access memory requirements needed to implement the classification. In addition, the minimal spanning tree provides a more compact view of sequence relationships within a family that is useful in identifying false hits and removing them from the classification. Finally, it provides a method for unambiguously ordering the sequence segments within a group. In this paper we will describe the minimal spanning tree classification algorithm in greater detail, we will compare classifications generated by this approach with classifications generated storing the full pair similarity set, we will show how this representation can be used to facilitate manual editing of classifications, and we will discuss classification artifacts which arise as a result of using this representation.

Protein Sequence Megaclassification

Although many protein families and functional domains are known, many more have not yet been recognized, and there are errors and disagreements over some of the existing definitions of families and domains. In previous work, we reported on HHS, our algorithm for automatic clustering of large protein sequence databases. Our algorithm was applied to the largest collection of protein sequences that we could assemble, totaling about 17,000,000 amino acids. This classification resulted in the identification of more than 10,000 groups of protein subsequences, including families, domains, and some artifacts.

In this section, we describe the framework we use for classifying these databases, and introduce some of the difficulties involved. Figure 1 shows a data flow representation of the classification process.

Figure 1. Data flow in the HHS algorithm



Database Search and Hit Assembly

Binary similarity judgments are found using BLAST (Altschul et al 1990) to search the molecular sequence database against itself to generate a database of all similar sequence segments. These are assembled into sequence similarity pairs.

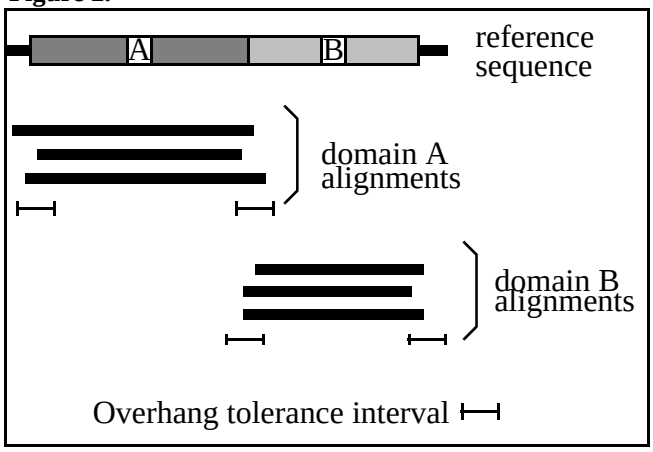
BLAST is a computationally efficient sequence similarity search tool that produces a list of statistically significant ungapped similarity segments for a pair of sequences, called the query and the subject (Altschul et al 1990). We used BLAST to search the molecular sequence database against itself to generate a list of all similar sequence segments. Biologically occurring insertion and deletion mutations may break a single region of similarity into several segments, each of which appears as a separate BLAST hit. The HHS algorithm compensates for this hit fragmentation by assembling together hits that belong to the same region of similarity. Overall, the database search phase of the calculation requires order($N_{sequence}^2$) time, but the database of similarities can be stored and updated incrementally.

Clustering Assembled Hits

After the assembly phase, the BLAST hits have been reduced to a somewhat smaller number of assembled hits. We now want to group these assembled hits into equivalence classes, forming the transitive closure of the pairwise similarity judgments. Hits that should be grouped together may have “ragged ends,” and be of somewhat different lengths.

Hits belong in the same group if they refer to the same region of similarity. In order to be grouped together, two hits should demonstrate significant overlap, but they need not coincide exactly. The non-overlapping portions of the hits are referred to as overhang.

Figure 2.



BLAST hits establish equality relations across proteins; the query and subject portions of a hit are nonrandomly similar. Constructing groups is a matter of building the transitive closure of the similarity judgments provided by BLAST. The ragged ends issue complicates the determination of whether two regions (within a protein) are in fact the same, and, therefore, whether hits that include those two regions should be placed in the same group. Building equivalence classes is then a matter of determining when two hits contain references to the same region. However, there are several complications that make building the transitive closure difficult. BLAST searching is probabilistic and therefore noisy. It can miss regions of similarity, and it can fragment a single region of similarity into multiple hits. Also, BLAST handles approximate matches in the content of the sequences, but it requires exact registration for matching, and its matches have fixed extent. We need to build groups that have approximately matching extents, and where the registration between regions of similarity is not perfect.

HHS address these issues by storing all of the similarity judgments about a sequence segment throughout the clustering calculation. Each new similarity judgement is tested against all of the previously saved similarities to see if any of them are consistent with clustering this new similarity into an existing group. In large groups, much of this similarity data is redundant; since all of the segments in a group are, by definition, related to each other, the ways in which they are related are also similar. The number of similarity judgements that must be saved is proportional to the square of the number of group members. Figure 3. shows that for large groups, the pair similarity dataset can be very large.

Figure 3.

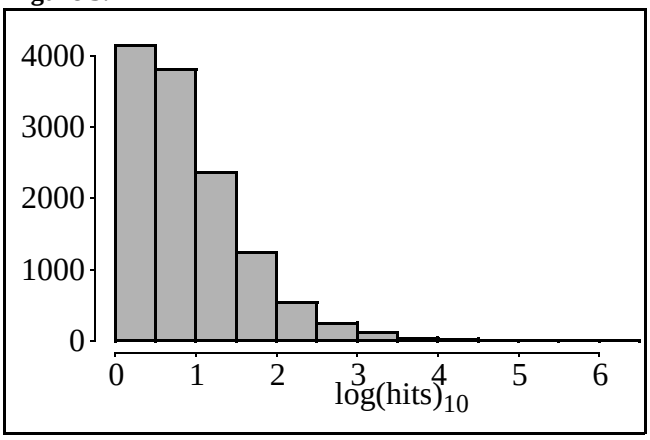
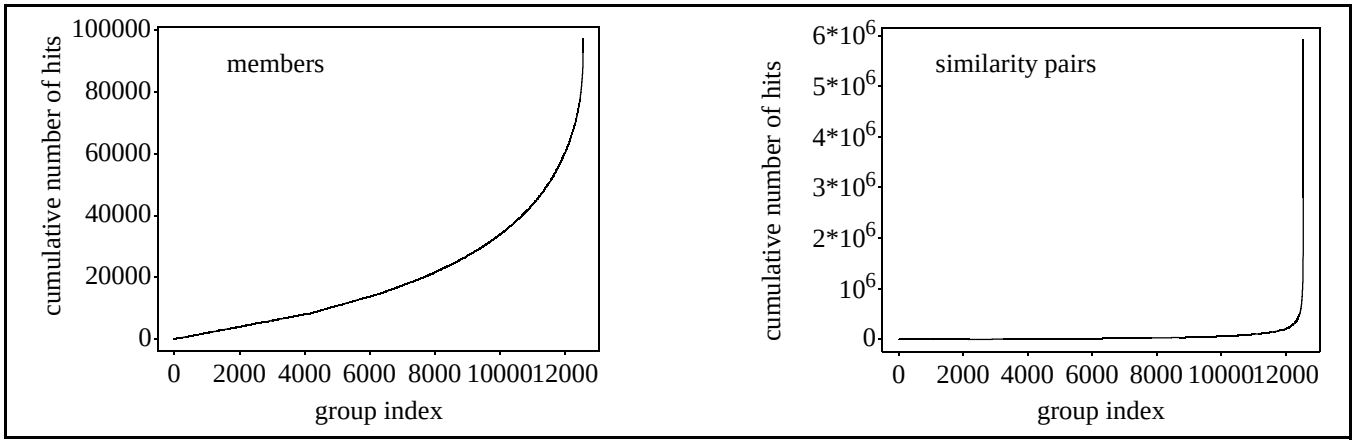


Figure 3. shows the distribution of the number of members per group in classification of the NCBI non-redundant sequence database (NRDB). The X-axis is on a logarithmic scale. The vast majority of groups contain only a few members, while a small number of groups have many members; the largest group contains over 2 million hits.

A small number of very large clusters account for most of the memory required to run the HHS algorithm. As Figure 4. demonstrates, these large groups are inefficient in terms of storage required per sequence.

The figure shows that a few large groups include the vast majority of similarity relationships, and the number of similarity values in these groups is out of proportion to the number of members they contain. This observation led us to modify our clustering method to reduce this redundancy. The modification also has salutary effects on the memory required to cluster and the human comprehensibility of the

Figure 4. Cumulative number of members and similarity pair



results, and provides a mechanism to impose a total order on the members of each group.

Computationally Efficient Class Representation: The HHS/MST Algorithm

It occurred to us that most of the similarity data saved for large groups was redundant, and it was clear that the storage of this excess data was limiting our ability to classify increasingly larger sequence databases. Recall that HHS works by approximate transitive closure: If a to-be-classified sequence is similar to a single member of a group, it is added to that group; if it is similar to members of more than one group, those groups are combined. HHS keeps track of all the similarity relationships between sequence in a group (by definition, there are no similarity relationships outside a group). The hope was that we could reduce this storage requirement by keeping only a subset of the similarity relationships within group, rather than all of them. The most aggressive way to do this is to keep only one similarity relationship for each member of a group.

If we take this aggressive approach, we can throw away all but one of the similarity relationships between that sequence and the other members of its group. Which similarity relationship should be kept? The highest scoring similarity pair for a sequence is an obvious candidate as the relationship to store. There are several reasons for choosing this pair. The sequence pair with the highest similarity score is likely to have diverged least evolutionarily. Since the information content of a sequence alignment declines with evolutionary divergence (Dayhoff et al, 1978; Altschul, 1990), the highest scoring pair is the most informative. Since the information content of the alignment is greatest, the highest scoring pair is likely to give the most accurate estimate for the endpoints of the aligned segments. The highest scoring pair is the similarity pair least likely to miss a region of similarity distal to an insertion or deletion mutation. The number of insertion and deletion mutations in an alignment correlates with the number of substitution mutations; high scoring pairs are likely to have fewer of each. If an insertion or deletion event has occurred in a closely related sequence pair, the distal segments are most likely to be recognizable for the most similar sequence pair.

To recognize a sequence segment as a member of a particular group, the segment must demonstrate similarity to a single member of the group. HHS stores all of the sequence similarity relationships within every class, and thus additional similarity relationships may modify the endpoints of the segment that is assigned to the sequence class. In some cases a new similarity relationship may be consistent with some, but not all, of the similarity hits already in a group. Testing a new hit against only a subset of the similarity data might, therefore, alter the group to which a segment is assigned, but in practice such cases are rare. To test how limiting the amount of similarity data stored might affect classifications, we implemented a classification in which only a single similarity relationship was retained for each new sequence segment.

The memory requirements and computational complexity of the classification algorithm can be analyzed by graph theory. Sequence segments may be considered to be nodes, and similarity relationships may be viewed as edges with the length of an edge being inversely proportional to the similarity score. A sequence class is then a connected graph. Representing the class by storing only the single highest scoring similarity relationship for each new sequence is equivalent to replacing the class relationship graph with a minimal spanning tree. This analogy is valid as long as the reduction to minimal spanning tree representation does not alter the segment endpoints for the sequence segments which are the nodes of the class. In practice, we have found that this condition is usually met. We refer to this algorithm as the minimal spanning tree variant of the HHS algorithm or HHS/MST.

The computational complexity of sequence classification is equivalent to the computational complexity of defining the minimal spanning trees in the forest of graphs defined by the full set of edges. This is a well known problem which has been analyzed in detail. The forest of minimal spanning trees can be generated by sorting the edges by length (computational complexity order(N_{edge} log(N_{edge})), taking them in order and rejecting any edge which generates a cyclic graph. By marking the nodes of each tree, the graph can be tested by cycles in constant time for each additional edge. A new edge will be incorporated into the forest at most once for each node. A new edge may merge two previous trees, and remarking the nodes of the tree will require time proportional to the number of members in either of the two merged groups.

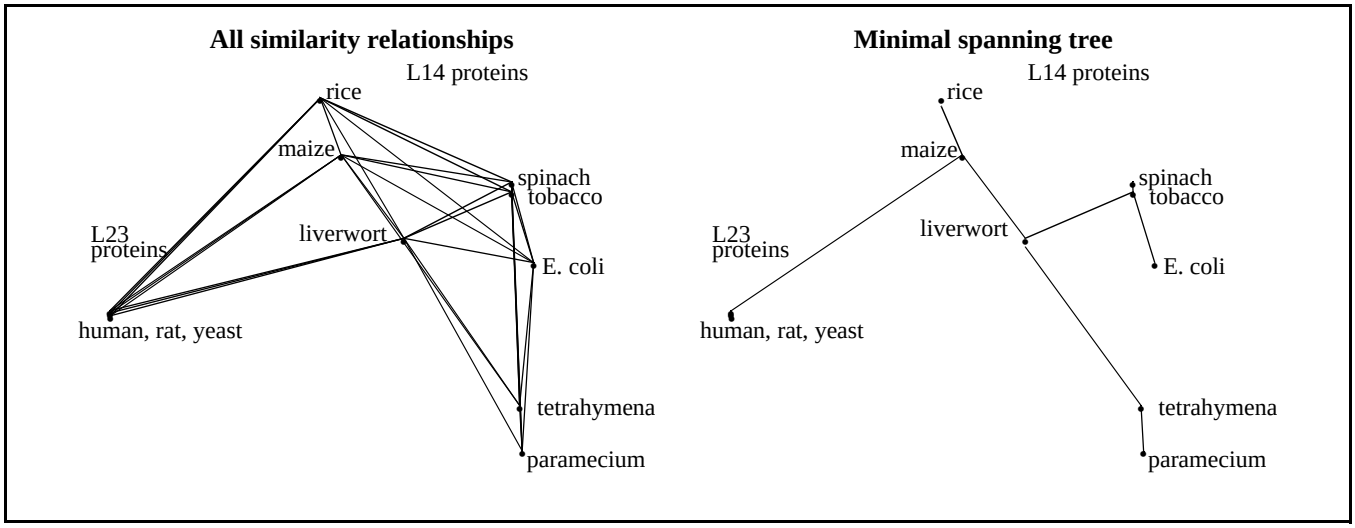
The storage requirements of the minimal spanning tree algorithm are proportional to the number of nodes in the forest. This contrasts with the HHS algorithm, in which the full set of edges is stored. Since the number of edges is proportional to the number of nodes squared, the use of a minimal spanning tree representation results in a dramatic reduction in storage requirements.

Testing the HHS/MST Algorithm

There is a possible problem with this approach. Recall that the extent of similar regions is used to determine whether a new sequence belongs in an existing group. Suppose a group contains a region of sequence A. Suppose further that sequence B has a region that is similar to part of sequence A. Whether sequence B will be added to the group depends on the extent of the overlap (and non-overlap) with A. If we keep only one similarity pair for A, then the extent of A in the group is the one associated with that similarity. If there is a wide range of extents of similarities for sequence A, and sequence B is at a different end of that distribution than the hit that was saved for A, then it is possible that using the HHS/MST method will cause B to fail to be added to the group. This could also cause a pair of groups to fail to be merged together.

We have reason to believe that this is not likely to be practically significant. The highest similarity score gener-

Figure 5. Fully connected and minimal spanning tree representations of a group



ally goes to the longest pair of sequences, so all of the extents will tend to be at the long end of the distribution. We also ran the two methods on the same dataset to compare the differences in classification.

Table 1.

Category	HHS	HHS/MST
families	139	140
mixed	11	10
domains	11	14
Total	161	164

Table 1. compares the results of classifications generated using the HHS and HHS/MST algorithms to classify the sequences in the Brookhaven Protein Data Bank (PDB). For the vast majority of the classes, the members included and the precise endpoints of the domains were exactly the same in the two classifications. There were two exceptions to this. One group that contained both whole protein hits and subprotein hits in the HHS classification was altered in the HHS/MST classification so that the domain hit was eliminated from the group, and this group therefore became a protein family class instead of a mixed domain/family class. In addition, three domain groups were defined in the HHS/MST classification that were not defined in the HHS classification. These additional subgroups resulted from cases where assembled alignments between distantly related members of a group spanned less than the full extent of the domain and were short enough that the endpoint length cutoff used in the classification did not allow these hits to be included in the domain. In these three cases, new groups were created which represent these conserved cores of sequence. These new groups are not simply artifacts of the HHS/MST algorithm because they provide additional information about regional sequence conservation within the parent domains. In this sense, the HHS/MST classification may actually be more informative than the full HHS classification.

The use of a minimal spanning tree representation provides a useful tool for generating subgroup descriptions. This is illustrated in Figure 5. This family contains both L14 and L23 ribosomal proteins. Viewing all of the similarity relationships within the group, it is difficult to distinguish between these two subgroups. When only the hits making up the minimal spanning tree are shown, the tightly clustered L23 subgroup is more apparent. In addition, the phylogenetic relationships of the L14 members are also more easily discerned.

The ability to generate subgroups is also useful in making functional and biological correlates. For example, the tyrosine kinase domains which are found in transmembrane receptors such as the insulin and epidermal growth factor receptors form a distinct subtree in the kinase class. Similarly, the trypsin and elastase subtrees of the serine proteases correlate with substrate preferences.

Using the MST for manual review of the classification

As described above, even when sequences are prefiltered to remove low entropy regions, false positive similarities can generate some overaggregated groups. Trying to manually screen the HHS produced classes and manually repair errors proved infeasible. The large size and complexity of the groups in which false positive hits occurred make it difficult to eliminate them by manual editing. If one false hit was present, there were often other false hits between proteins closely related to those in the initial false hit. Even if a false positive hit can be identified and eliminated, there is no guarantee that all of the false hits have been removed.

In HHS/MST classifications, false hits can easily be identified by searching the path which connected two biologically unrelated proteins in an artifactually merged group. Even for very large groups, only a few dozen edges were typically found. This is illustrated in Table 2 which

Table 2. False positive hit identification in a large group by link tracing

Segment span	Protein
(133 to 341)	MUSNCAMR precursor polypeptide >513435 0 NCA3_MOUSE
(57 to 543)	RATTAG1 axonal glycoprotein
(502 to 610)	RATNCAM14 neural cell adhesion molecule
(1 to 108)	HUMNCAM neural cell adhesion molecule secreted isoform
(485 to 681)	XELNCAMA cell adhesion molecule
(1 to 235)	HUMNCAMA N-CAM >1019770 1 A26883 Neural cell adhesion
(621 to 681)	XELNCAM neural cell adhesion molecule precursor
(82 to 951)	HUMTITINC2 titin
(248 to 940)	A40985 *Projectin - Fruit fly (Drosophila melanogaster) (fragment)
(2515 to 2738)	HUMTITINC3 titin
(251 to 606)	RATMLCK skeletal muscle light chain kinase
(1 to 368)	A05120 Myosin light chain kinase, skeletal muscle
(263 to 608)	RABMLCKA myosin light chain kinase >511296 0 KMLC_RA

shows the hits connecting an immunoglobulin-like neural cell adhesion molecule (NCAM) to a protein kinase domain The table lists a set of segments, each of which was linked by a similarity hit to the segments above and below it in the table. In this example, the hits, or edges, connecting an NCAM to a kinase were traced in the cluster tree. Hits to a set of "titin" proteins were seen to link the NCAMs and the kinases. Titins are large structural proteins (Labeit et al, 1992) containing several regions of low entropy sequence, and XNU was not successful in completely eliminating associated false hits. By manually deleting the hit from HUMTITINC3 to RATMLCK, the kinase domain family was correctly dissociated from the titins and NCAMs. Deleting the hit from XELNCAM to HUMTITINC2 removed the link from the cell adhesion molecules to the titins. The minimal spanning tree representation guaranteed that when a false positive hit was identified and eliminated from the dataset, the falsely merged groups were divided. If they were not, then a cycle would have been present in the graph and the original class representation would not have been a minimal spanning tree.

Finally, using HHS/MST makes it more difficult to detect a certain kind of database artifact that we discovered with HHS. This artifact arises as a result of technical difficulties in cDNA cloning: partial sequences for many proteins were present in the database along with complete sequences for the same proteins. The fragmented nature of these sequences often was not annotated and occasionally was unknown to the contributing author. For HHS classifications, these artifactual groups could be detected using post-classification analysis. The manifestation of the artifact was a pair of two nearly identical groups. In one group, each protein had hits with many other proteins. This was the correct group. In the corresponding artifactual group, one protein (the fragment) had hits with all the other proteins, but because the non-fragment proteins had longer regions of similarity (which are in the true group), none of these other proteins had hits of this size with anything but the fragment. This artifact produced an easily distinguishable star topology in the connectivity graph of the group. In addition, the members of an artifact group, other than the fragment, were all members of a corresponding true family group. In the HHS/MST classifications, automated recognition of these fragment artifacts has proven more difficult because much of the redundancy information used to discriminate between the true and artifactual group has been eliminated.

One of our goals in the use of a minimal spanning tree representation was a significant reduction in the storage requirements for the classification calculation. This was achieved. While classifications of the full NRDB using the HHS algorithm required in excess of 500 MB of RAM memory and required the use of a supercomputer with 512 MB of main memory, classifications using the minimal spanning tree representation could be performed in 60 MB of RAM and can be run easily on available workstations.

Discussion

Scalability of algorithms (Schank, 1991) and the ability to work in large and complex data sets (Almuallim and Dietterich, 1991) are critical issues in machine learning. One of our expressed goals in the sequence megaclassification project has been the application of machine learning and pattern induction techniques to large real world problems. The HHS algorithm was successful in attacking real world problems on datasets of interest to the biological community (Hunter, Harris, and States, 1992), but given the rapid growth in biological sequence data, even the quadratic scaling of memory requirements with dataset size in HHS has proven to be a significant limitation. In addition, some cases of real biological interest have proven to be impossible to analyze on available computing resources. In particu-

lar, although we have been able to classify the current protein sequence databases, much larger databases of nucleic acid sequence are also available; analysis of these datasets using HHS would require several gigabytes of RAM. Much of the progress in computational molecular sequence analysis has resulted from algorithms development. We sought and were successful in deriving an algorithmic solution to the limitation of the HHS approach. The use of the HHS/MST approach will allow nucleic acid sequence datasets and datasets containing both protein and protein coding nucleic acid sequences to be analyzed jointly.

The ability to work with combined protein and nucleic acid sequence databases is of particular importance in dealing with the classification artifacts created by the presence of fragmentary sequences in the database. It may be possible to recognize partial mRNA sequences, but there is no definitive way to recognize partial sequences by protein sequence classification alone. For example, the relationship of the src kinase domain to the kinase domain of the insulin receptor is entirely analogous to the relationship of a partial protein to its full parental sequence, but the proteins in the src/insulin receptor examples are full length sequences and the true relationship is an example of composite protein structure. Messenger RNAs (mRNAs) typically contain a number of distinctive features at their 5' end including ribosome binding sites and initiator codons. If these are absent, it is likely that the mRNA is a partial sequence. By jointly classifying protein and nucleic acid coding regions, such partial sequences can be recognized by criteria which are independent of the classification.

The reduced storage requirements of the HHS/MST algorithm will also be important in keeping pace with the rapidly expanding databases of molecular sequence. Sequence databases have been doubling in size every two years. While computing speed has been able to match this rate of growth to date, the corresponding pair similarity datasets quadruple every two years. The cost of RAM has fallen dramatically in recent years, but it has not fallen fast enough to accommodate the projected space requirements of a quadratic scaling calculation.

Improved ability to manually review and edit groups is an interesting benefit of the HHS/MST representation. In a sense, the requirement of a more compact representation forces a higher level view of the problem. Using the minimal spanning tree representation made it easier to find false positive hits and to manually edit and correct classifications.

The higher level view of the classification generated by HHS/MST also elucidates important relationships between sequences within a group. As the L14/L23 ribosomal protein example illustrated, there may be significant substructure within a group. Reducing group representation to a minimal number of strong similarity relationships highlights this extra level of structure.

Calculation of consensus sequences or sequence profile descriptions for groups is also facilitated by the spanning tree description of groups. When the full segmental pair similarity list is used, ambiguous ordering or alignment relationships were often generated by cycles in the similarity graph describing a group. Since the tree representation contains no cycles, these ambiguities are eliminated. Furthermore, the use of the spanning trees based on the highest scoring similarities optimizes the likelihood that the ordering and alignments defined for the group will be correct.

The tree representation implicit in the HHS/MST problem maps closely to the hierarchic organization of protein domains generated by the evolutionary process of gene duplication and mutation (Patterson, 1988; Felsenstein, 1988; Doolittle, 1992). The HHS/MST algorithm does not retain any notion of a parent-sibling relationship, and all of the nodes in the HHS/MST tree are currently extant proteins. Nevertheless, there is some similarity between the highest scoring links selected by HHS/MST and a true evolutionary tree. Homologs of closely related species are typically found near each other in HHS/MST trees, and the longer branches of HHS/MST trees frequently correspond with ancient divergence events between orthologous proteins. The HHS/MST classification algorithm may be a valuable tool in the exploration of the relationship between protein sequence, structure, and function.

Bibliography

1 Almuallim, H., & Dietterich, T. (1991) Learning with many irrelevant features. In Ninth National Conference on Artificial Intelligence, 547-552. Anaheim, CA: AAAI Press.

2 Altschul SF (1991). Amino acid substitution matrices from an information theoretic perspective. J Mol Biol 219: 555-65.

3 Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). A Basic Local Alignment Search Tool. Journal of Molecular Biology 215:403-410.

4 Dayhoff, M. O., Schwartz, R. M., & Orcutt, B. C. (1978). A Model of Evolutionary Change in Proteins. In Atlas of Protein Sequences and Structure.

5 Doolittle, R. (1992). Reconstructing History with Amino Acid Sequences. Prot. Sci., 1, 191-200.

6 Felsenstein, J. (1988). Phylogenies from molecular sequences: inference and reliability. Annu Rev Genet 22: 521-65.

7 Harris, N., Hunter, L., & States, D. (1992). Megaclassification: Discovering Motifs in Massive Datastreams. In Tenth National Conference on Artificial Intelligence (AAAI), pp. 837-842. San Jose, CA: AAAI Press.

8 Harris, N., States, D. & Hunter, L. (1993). ClassX: A Browsing Tool for Protein Sequence Megaclassifcations. In Proceedings of the Twenty-Sixth Hawaii International

Conference on System Sciences, pp 554-563. Los Alamitos, CA: IEEE Computer Society Press.

9 Hunter, L., Harris, N., & States, D. (1992). Efficient Classification of Massive, Unsegmented Datastreams. In International Machine Learning Workshop, pp. 224-232, Eds. D. Sleeman & P. Edwards, Morgan Kaufman, San Mateo, CA.

10 Labeit, S., Gautel, M., Lakey, A., Trinick, J. (1992) Towards a molecular understanding of titin. EMBO J 11: 1711-6.

11 Patterson, C. (1988) Homology in classical and molecular biology. Mol Biol Evol 5: 603-25

12 Schank, R. C. (1991). Where's the AI? AI Magazine 12(4):38-49.

13 States, D. J. and Claverie, J.M. (1993) Computational Chemistry, in press.

14 Wootton, J. C. & Federhen, S. (1993). Statistics of Local Complexity in Amino Acid Sequences and Sequence Databases. Computers in Chemistry (in the press).