Functional classification of proteins by pattern discovery and top-down clustering of primary sequences

by A. H. Liu
A. Califano

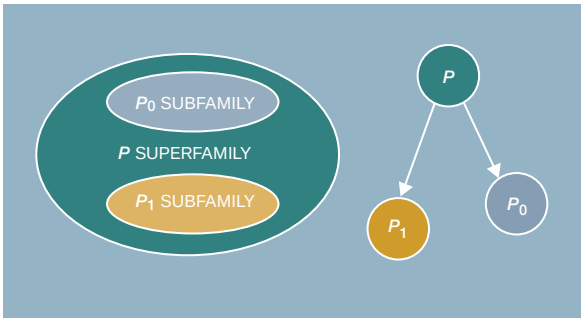
Given a functionally heterogeneous set of proteins, such as a large superfamily or an entire database, two important problems in biology are the automated inference of subsets of functionally related proteins and the identification of functional regions and residues. The former is typically performed in an unsupervised bottom-up manner, by clustering based on pair-wise sequence similarity. The latter is performed independently, in a supervised top-down manner starting from functional sets that have already been identified by either biological or computational means. Clearly, however, the two processes remain inextricably linked, because functional motifs and residues are related to corresponding functional clusters. This paper introduces a high-performance, top-down clustering technique and the corresponding system that determines functionally related clusters and functional motifs by coupling a pattern discovery algorithm, a statistical framework for the analysis of discovered patterns, and a motif refinement method based on hidden Markov models. Results are reported for the G protein-coupled receptor superfamily. These show that a significant majority of well-known functional sets and biologically relevant motifs are correctly recovered. They also show that a majority of the important functional residues reported in the literature occur in the inferred functional motifs. This technique has relevant implication in functional clustering and could be used as a highly predictive aid to mutagenesis experiments.

It has been shown[1] that the combination of an efficient, deterministic pattern discovery algorithm, SPLASH,[2] and a framework for the assessment of the statistical significance of the discovered patterns[3] has a high probability of identifying biologically significant protein motifs, defined as highly conserved, ungapped regions of a protein or DNA (deoxyribonucleic acid) sequence.[4] Biologically significant, in this context, means that mutations of some of the residues that are highly conserved in one of these motifs are likely to result in the loss of biological function, due to modifications in either the structural or the physiochemical properties of the protein.

The rationale behind this approach is that mutations that would result in a critical loss of biological function are less favored by evolution and, consequently, functionally and structurally relevant regions tend to be highly conserved across a corresponding protein family. This conservation can be detected as a pattern of conserved residues that would be unlikely to have occurred by chance. Complex protein families, consisting of several domains, each characterized by specific physiochemical properties, will therefore be characterized by large numbers of such statistically significant patterns.

This paper extends this approach to a top-down clustering method that can be used to organize large protein sets into subsets that are functionally related. For simplicity, we refer to protein sets identified by experimental means as *protein* or *functional groups* and those identified by computational means as *protein* or *functional classes*. This procedure is expected to separate the original sequence set into smaller and

Figure 1 Venn diagram representation of a superfamily and a functional subfamily



smaller subsets, characterized by an increasingly higher degree of functional relatedness. This is tested and reported in the results section against a manually generated taxonomy for the set of G protein-coupled receptor (GPCR) proteins (ground truth). It is also reported, in the section on related work, against the results of running ProtoMap,[5] a bottom-up clustering method, on the same set of proteins. Our results are found to be in substantial agreement with the ground truth and to outperform ProtoMap on the number of false positives and false negatives, as well as on the ratio between the number of approximately matched and the total number of identified classes.

Let us assume that P is a set of protein sequences, such as a large superfamily, consisting of several functionally distinct subfamilies, each one with a significant number of representatives. From the results of Hart,[1] the single most statistically significant regular expression pattern in the set, π_1, is expected to correspond to a motif that is both biologically significant and discriminative. That is, sequences matching the motif would be likely to be functionally distinct from those that do not match it. In this context, given a model of the motif, derived from the regular expression, such as a position-specific scoring matrix (PSSM)[6] or a profile hidden Markov model (HMM),[7] and a corresponding statistical criterion, such as a P-value or E-value,[8] a sequence is said to match the motif if it satisfies the corresponding statistical criterion.

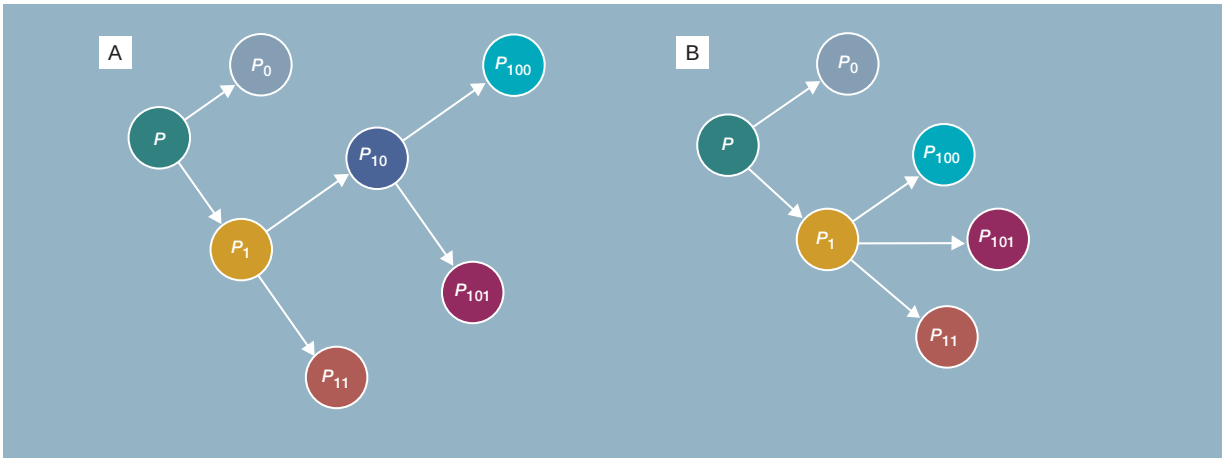
Based on such a *motif-criterion* combination, one can split the family P into two subsets: P_1, with sequences that match the motif, and P_0, with sequences that do not match the motif. This is shown in Figure 1. P_1 can be smaller than or equal to P, while P_0 is typically strictly smaller. This paper tests the hypothesis that an exhaustive, iterative application of this method leads to the simultaneous identification of both a significant number of subsets of functionally related sequences and their corresponding functional motifs.

We visualize the evolutionary process for a protein superfamily via a basic model, where a common ancestral gene evolves into a hierarchical gene family through recursive gene duplication and divergence events. The output of this model is a gene tree. Based on this model, the procedure in Figure 1 should be repeated iteratively on both P_0 and P_1, after "masking" the residues of π_1 in the sequences of P_1 to avoid discovering the same exact pattern again. The procedure can be stopped when either the set size becomes lower than a preset threshold or a statistically significant pattern can no longer be discovered.

This procedure can be used to construct a binary tree where each successive node corresponds to an increasing degree of functional similarity. An edge in the tree corresponds to the presence or absence in the child node of the motif identified in the parent node. This is shown in Figure 2A. By collapsing all edges that do not correspond to a match and lead to an internal node of the tree, a tree of variable "arity" can be produced such that internal nodes correspond only to the presence of a corresponding motif. This is shown in Figure 2B.

The advantages of this approach are threefold. First, functional sets are directly inferred from and related to specific motifs in sequence space. This offers important clues as to the functional relevance of the individual motifs. Second, high similarity in nonfunctional regions does not interfere with cluster selection. That is, two sequences that are highly similar may have a critical difference in a functional region and therefore not share a common function. Third, the procedure is extremely efficient because it relies on a hierarchical "divide and conquer" approach. The system constructs a tree during the clustering process, and all the nodes at the same level of the tree are independent of one another and thus can be "conquered" at the same time. Large superfamilies, with thousands of members, can be clustered in a few hours on workstation-level hardware. For this reason the method could be applied to the unsupervised clustering of large-scale databases.

Figure 2 Venn diagram of a superfamily and several increasingly smaller functional subfamilies



There are, however, several issues that may lower the performance of the approach when measured against known functional-family classification schemes. First, entire or partial domain insertion, rather than point mutation, may occur during evolution. Therefore, several functional properties may be highly intertwined across multiple families, and the best representation of functional relationships may not be a tree. This is addressed by a modification of our approach that represents functional relationships through a tree-graph combination. We report on this in a later section.

Second, those parameters of our system effective during the pattern discovery phase determine the types of patterns that will be discovered. Therefore, subtle, flexible,[9] or very small motifs, such as an individual catalytic residue, may be missed by the pattern discovery procedure. In a later section we study the parameter space to determine the robustness of the algorithm. Third, the use of any statistical criterion to determine class membership will result in some false-positive and false-negative results. Therefore, some functional family members may end up in erroneous branches of the tree. A "fuzzy" classification algorithm that attempts to minimize this effect is introduced in the methods section. Finally, one should consider that manual functional classifications of a large protein family typically show significant disagreement and that some functional sets may have a semantic rather than functional basis. This aspect is covered in detail in the results section.

This top-down clustering method is quite distinct from traditional approaches, such as COG,[10] DOMO,[11,12] HHS/MST,[13] ProDom,[14] and ProtoMap[5]—where shared functionality is inferred from pair-wise sequence homology. A top-down use of pattern discovery for the construction of motif dictionaries has also been proposed by Rigoutsos.[15] This approach is based on the discovery of exact regular expressions without statistical analysis or pattern refinement, and no exhaustive comparison of the functional motifs or of the functional residues to existing literature is reported.

To measure our method's performance, a suitably large protein sequence set has been analyzed, where the functional nature of the member proteins is known *a priori*—the G protein-coupled receptors (GPCRs). The GPCR superfamily comprises an important, large, and functionally diverse set of proteins that mediate the cellular responses to an enormous number of unique signaling molecules across the plasma membrane. They play fundamental roles in regulating the activity of virtually every body cell. Therefore, they constitute an almost ideal candidate set for this exercise.

Furthermore, studies of the deduced amino-acid sequences indicate that these proteins have marked homology and share a common membrane topology consisting of seven transmembrane helices. Upon binding of extracellular ligands, GPCRs interact with a specific subset of heterotrimeric G proteins that

can then, in their activated forms, inhibit or activate various effector enzymes and ion channels. This means that, although these proteins have a common action mechanism, they are highly specific in their targets. This selectivity (of both ligands and G proteins) should result in a number of highly selective motifs responsible for the binding of the specific molecules. A great wealth of information on these functional regions exists in the literature from site-directed mutagenesis experiments and other biological assays.[16–19] This information can also be used to assess the method's performance.

Finally, molecular cloning studies have shown that GPCRs form one of the largest protein families found in nature. In fact, more than 200 functionally distinct receptors in this gene family have been cloned and more than 1000 sequences or sequence fragments are available in the SWISS-PROT database.[20] This is again ideal for this analysis because distinct functional families are represented by a significant number of members.

The next section of the paper describes the methodology in detail. In particular, it describes how patterns discovered by SPLASH can be used to generate highly sensitive profile HMMs[7] and how these in turn are used to infer functional relationships. The following section describes the comparison of the resulting functional clustering against taxonomies reported in the literature and the comparison of the putative functional motifs against residues for which biological activity is also reported in the literature. The final section covers related work.

Methods

This section is devoted to a description of the pattern discovery and clustering steps.

Pattern discovery. SPLASH,[2] a novel pattern discovery algorithm, is used to identify conserved patterns in sets of protein sequences. This algorithm discovers all rigid regular expressions that occur in a set of sequences subject to the following constraints:

1. The characters of the regular expression are from an alphabet of amino acids, amino acid similarity classes, and a "don't-care" symbol. Similarity classes, such as `[ILMV]`, are defined as sets of amino acids that score above a given threshold with respect to the first amino acid in the similarity class, using a predefined scoring matrix such as PAM[21] or BLOSUM.[22] Individual amino acids in the regular expression match only an exact occurrence in the sequence; similarity classes match an occurrence of any of the amino acids in the similarity class; a "don't-care" symbol matches any residue. For instance, `C..[ILMV].[DE]` matches `ACRKMVDQP`, starting at the second amino acid, with `M` matching `[ILMV]` and `D` matching `[DE]`.
2. Patterns must occur at least j_0 times in the sequence set or in at least j_0 distinct sequences. In this paper, the former definition is used, even though more than one match in a single sequence occurs infrequently.
3. Patterns must satisfy a density constraint. That is, any substring of length l_0 in the pattern that does not start with a "don't-care" symbol must contain at least k_0 tokens, which are letters, similarity classes, but not "don't-care" symbols. If the pattern is shorter than l_0, it must contain at least k_0 tokens.
4. Patterns must contain at least t_0 tokens.
5. Patterns must be maximal. That is, no token can be added to the pattern without reducing its "support," defined as the number j of occurrences in the set.

The stability and performance of parameters such as j_0, l_0, k_0, and t_0 are examined in the methods section.

Patterns are assigned z-scores computed from the mean number and standard deviation of equivalent patterns that should have been discovered in a random database of similar composition.[3] Patterns are considered equivalent if they have the same support j, length k (number of full characters), and span l (total number of characters including "don't-care" symbols). Equivalent patterns are assigned identical z-scores. As reported by Stolovitzky,[3] z-scores are inversely proportional to the corresponding P-values of the pattern. Hart[1] shows that high z-score patterns tend to be biologically significant.

Given a set of sequences P, and a set of parameters as described by Hart,[1] SPLASH is run repeatedly until a minimum number, N_{-}, of statistically significant patterns are discovered. The algorithm starts by looking for patterns that occur in 100 percent of the sequences. The minimum support j_0 is then gradually reduced, by 5 percent of the number of sequences each time. For every new minimum support, both the density as specified and one-half of it (by doubling the window length) are attempted. These steps are repeated until a good sample of at least N_{-} statistically significant patterns is obtained or until a

minimum support $j_0 = 0$ is reached. As soon as at least N_- patterns are found, the most statistically significant one is selected.

Pattern refinement. The amino acid classes described in this section are not context-specific. That is, they are not likely to realize all the possible substitutions that would preserve function in a particular family. This may result in incomplete patterns. To minimize this effect, a pattern is extended by examining both the left- and right-flanking regions of all the occurrences of the pattern in the sequence set. The goal is to detect additional significant residue conservation that would not be discovered based on the similarity class definition. Given a pattern, sequences that match it are first rigidly aligned according to where the pattern occurs, as shown in Figure 3. For each position relative to the sequence multiple alignment, the residue statistics are analyzed to determine if there is substantial conservation. This is accomplished by computing the amino acid entropy over a small window and then by sliding the window, as shown in Figure 3, until the entropy increases by more than a predefined amount, ΔE. The window is initially positioned inside the pattern at its left or right boundary depending on the direction of the extension. The entropy is computed as:

$$E = \sum_{i=1}^{w} \sum_{j=1}^{20} - p_{ij} \log p_{ij} \qquad (1)$$

where p_{ij} is the probability of seeing the jth amino acid at the ith window position, computed from its frequency in the aligned set. The first sum is over all positions of the sliding window; the second is over the 20 amino acids. At the end of this process, patterns are extended both left and right all the way to the outer edge of the window, at the last window position considered before the entropy threshold was exceeded. For results reported in this paper, $w = 10$ and the cutoff delta ΔE is computed as:

$$\Delta E = (E_{\max} - E)/16 \qquad (2)$$

where E is the entropy computed over the first sliding window position, when it is still completely contained within the original pattern; $E_{\max}$ is the maximum possible entropy based on the database composition; and 16 is a heuristic factor.

As discussed in Califano,[2] regular expressions produced by the pattern discovery algorithm should be used only as seeds for sounder statistical models such as PSSMs or profile HMMs. The latter have the advantage of being based on a formal statistical framework,[7] which provides a consistent theory for scoring insertions and deletions. For this reason, they are the model we chose. For practical HMM construction and scoring purposes, the package HMMER has been used. This software is available at http://hmmer.wustl.edu/.

Figure 3 Pattern extension

A profile HMM is obtained by running HMMBuild on the set of occurrences of the extended pattern resulting from the previous step. This is a set of aligned, ungapped sequences. HMMBuild constructs a profile HMM using the maximum *a posteriori* (MAP)[23] construction algorithm to determine the length of the main model, and the maximum likelihood estimates[23] with Dirichlet mixture priors to estimate pseudo counts.[24,25] HMMCalibrate is then used to obtain close estimates of the score probability densities for the computed HMM model. This is performed by scoring a large number of synthetic random sequences with the profile HMM and by fitting the resulting score histogram with an extreme value distribution (EVD).[26,27]

HMMSearch is then used to identify all the sequences that contain regions similar to the extended occurrences of the pattern discovered by SPLASH.[2] The input is the entire set of sequences that was used in the pattern discovery phase at the current level of iteration. This set of sequences has not been previously aligned, and its alignment may contain gaps. The program generates a log likelihood ratio[23,28] for every sequence examined that indicates how well it aligns with the profile HMM compared to a random

set of sequences of similar size and composition. The value that the program reports for every sequence is an E-value, which is an estimate of the number of sequences expected to have an equal or greater log likelihood ratio in the random set.[28] The E-value accounts for the different statistics of sequences of different lengths according to the extreme value distribution. Sequences with an E-value equal to or below a first threshold e_1 are considered matches, while those with E-values above a second threshold e_2 ($e_2 \leq e_1$) are considered mismatches. A given set of sequences can thus be divided into a subset P_0, containing the mismatching sequences, and a subset P_1, containing the matching ones.

Sequence conservation across families is directly related to the functional or structural relevance of the conserved region.

In theory, further pattern refinement can be performed by iteratively refining the statistical model. For any of these iterations, another profile HMM would be obtained by running HMMBuild on the subset of sequences matching the profile HMM obtained in the previous iteration. HMMCalibrate and HMMSearch would then be run in exactly the same way as described previously. Notice that the first profile HMM is obtained from the set of occurrences of a SPLASH pattern, but any of the subsequent profile HMMs is obtained from a set of matches of a profile HMM. That implies that the construction of a second profile HMM may make the biggest difference. For time concerns as well, we choose to iteratively refine our statistical model only once by constructing a second profile HMM. Some results are reported in the next section.

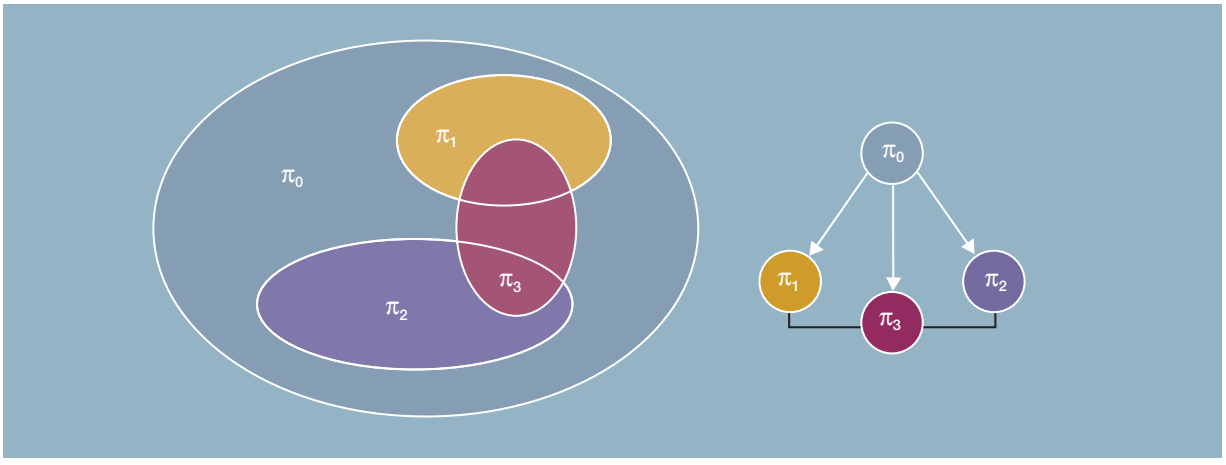
Functional clustering with a binary tree. As discussed in the introduction, the fundamental idea is that sequence conservation across functional families is directly related to the functional or structural relevance of the conserved region. Therefore, the more statistically unlikely a globally conserved region is, the more likely it is that there exists a significant functional or structural justification for that conservation. Ideally, one would want to first identify all existing patterns, from those conserved in large subsets to those conserved in just a handful of proteins, and then rank order them according to their statistical significance. This is impractical, however, because highly conserved patterns—patterns with a high support j—would result in a huge number of close variants that are conserved in subsets of the j sequences. This number can be shown to be exponential in j. Therefore, as described in an earlier subsection, motifs are identified starting at the highest j. We next describe how they are subsequently masked in the input set to prevent detection of related variants.

Binary tree-based functional clustering is performed as follows: given any sequence set P_i, pattern discovery is performed as described earlier. Then, the single most statistically significant pattern is selected, extended, and refined. Two new sets P_{i0} (mismatching) and P_{i1} (matching) are then generated from P_i by comparing the computed E-values against the two thresholds. The new sets become respectively the right- and left-child of P_i in a binary tree.

If $e_1 = e_2$ then the two sets have an empty intersection. Otherwise, the two sets may partially overlap over borderline cases. The first method is referred to as *exact clustering* and the second one as *fuzzy clustering*. It is evident that the setting of the E-value thresholds determines the performance of the system. The E-value thresholds may and probably should vary from iteration to iteration. This is because the distribution of E-values for a specific iteration depends on a combination of factors, including the database used to construct the profile HMM and the database against which to match the profile HMM. In this paper, however, we use constant E-value thresholds throughout a single experiment. Tighter thresholds (higher E-values) tend to result in more false negatives, while looser thresholds tend to result in more false positives. To achieve good performance, we should either adjust the thresholds intelligently or allow fuzziness in the classification, as in fuzzy clustering.

The procedure is repeated iteratively on each new node in the tree until either resulting sets contain fewer than a predefined number of sequences or statistically significant patterns can no longer be found. The root node of the tree contains all the sequences in the set of interest. By definition, matching sets P_{i1} are assigned to left branches, and mismatching sets P_{i0} to right branches. Finally, if for a tree node P_i, determined by a motif π_i, another motif $\pi_{i'}$ is found

Figure 4 Venn diagram representation of tree-graph pattern relationships



such that $P_{i1} = P_i$, then the two nodes are combined and the rigid motifs π_i and $\pi_{I'}$ may also be combined into a flexible motif.

Each node in the tree can be uniquely identified by a sequence of ones and zeros corresponding to the set of match and mismatch events leading to it. In the exact clustering, each protein belongs to only one leaf of the tree. In the fuzzy clustering, a protein may belong to more than one leaf. In the latter case, the sequence is assigned *a posteriori* only to the leaf that would result in the smallest sum of E-values over all the left branches in the path from the root. The closer two proteins are evolutionarily, the more instances of common motifs they are expected to have. Therefore, the distance between two leaves can be used as a measure of the phylogenetic distance between any pair of proteins in the two leaves.

Clustering with a tree-graph combination. In this approach, we account for motifs that may be supported by partially overlapping sequence sets. Nature follows specific rules in general, but never fails to make exceptions. Ideally, we may conceive that nature always generates a well-defined and clear-cut gene tree. Realistically, we should expect nature to generate an "imperfect hierarchy" of functional families with additional cross-branch connections, which may have been formed via domain insertions or even convergent evolution.

Therefore, given a set of proteins, if we consider how different proteins may share domains in various ways and thus how different patterns representing those domains may be related, we expect to observe two situations. We illustrate them in Figure 4, where we represent a pattern by its support set of matching sequences in a Venn diagram. In the first situation, sets corresponding to different patterns are either completely contained in one another (π_1 and π_0; π_2 and π_0; π_3 and π_0) or completely separated from one another (π_1 and π_2). In the second situation, sets could be partially overlapping (π_1 and π_3; π_2 and π_3).

Patterns that are completely contained in one another in terms of their support sets will be linked by directed edges and form a partial n-ary tree. Patterns that are partially overlapping in terms of their support sets will be linked by undirected edges and form a partial undirected graph. Such an overlap between two patterns corresponds to a set of proteins that possess two domains that are otherwise possessed by proteins that are considered to belong to two remote functional sets. The result is a combination of a tree and an undirected graph. Based on how the patterns overlap in terms of both their support sets and the patterns themselves, appropriate relationships can be determined for the proteins. This representation differs from the one suggested by Rigoutsos,[15] where a directed graph representation is suggested.

To construct the tree-graph combination, it may be intuitive to first construct a tree and then somehow establish the cross-branch connections. We decide

to adopt the following cleaner approach. Given a set of protein sequences, we simply discover one statistically significant pattern as described before. We then mask the occurrences of the pattern. In the recursive processing that follows, instead of splitting the current set of sequences into two subsets depending on whether or not the pattern occurs, we continue working with the same set of sequences. We do so until we can no longer discover any statistically significant pattern that satisfies all the constraints. This is also described in Hart.[1] In the resulting set of patterns, some may contain others while some may overlap others, in terms of their support sets, and a tree-graph combination could be constructed as described previously. Some preliminary results for this analysis are reported in the next section.

We discussed in previous subsections the construction of a pure tree, which can be carried out efficiently. We expect the pure tree to reflect the skeleton of the tree-graph combination or even correspond to the tree part of the combination. Therefore, on the one hand, we may use the tree to get a quick look at the result of evolution; on the other hand, we would want to use the graph to gain more detail from and insight into the evolution process. It will be interesting to examine the tree part and the graph part of the tree-graph combination separately. It will also be interesting to investigate any difference between the pure tree and the tree part of the tree-graph combination.

Protein classification. Given a set of proteins, an automated and unsupervised procedure has been introduced to establish a hierarchy of functional classes. Starting with the hierarchy, functional classification of an unknown protein can be performed in a rather straightforward way. One can either use the hierarchy as a decision tree and traverse it until a leaf is reached, or match the protein to the set of all motifs leading to a leaf of the tree and establish which branch is the most statistically significant. For this purpose, HMMSearch can also be used to compute the *E*-value for this protein against each one of the derived profile HMMs.

Results

This section analyzes the performance of the top-down clustering method against a set of GPCRs[29] obtained from GPCRDB,[15] excluding orphan, probable, and putative sequences. GPCRs form one of the largest and most functionally differentiated protein families and are an ideal test set for the performance of our method.

G protein-coupled receptors. The molecular mechanisms involved in GPCR function, particularly the molecular modes of receptor activation and G protein recognition and activation, have become an ever-increasing research focus. Mutagenesis and biophysical analysis of several of these receptors indicate that small molecule agonists and antagonists bind to hydrophobic pockets buried in the transmembrane core of a receptor. In contrast, peptide ligands bind to both the extracellular and transmembrane domains. Meanwhile, G proteins are typically the ones that bind to the intracellular domains.

A great wealth of information about GPCRs is available. GPCRDB[16] and GCRDb[17] are full-fledged databases specifically on the set of GPCRs. They are constructed manually by biologists. PRINTS[18] is a database of protein sequence fingerprints (sets of multiple alignment blocks), and it includes a comprehensive, hierarchical set of fingerprints for GPCRs. The fingerprints have been shown to have strong discriminative power for family membership. The database is constructed computationally with a supervised learning approach. Finally, the GPCR mutant database (GPCRMD)[19] is a database of mutation information on the set of GPCRs. It compiles a comprehensive list of mutagenesis experiments performed on GPCR sequences up to 1997, detailing all the pertinent information about each experiment. These information sources provide a basis on which to assess the performance of our system.

The performance of the clustering scheme is studied via three distinct methods. The goal is to show that (1) the system produces a hierarchical decomposition of GPCRs where the subsets are likely to overlap significantly with well-known functional subsets and (2) corresponding motifs identify regions and residues with important, family-specific functional roles.

First method. Functional classes identified by our approach are compared against a set of functional groups. Unfortunately, there is no agreement on a global taxonomy for GPCRs and different databases report partially overlapping functional groups based on biological rather than computational classification. For instance, PRINTS defines functional groups according to the GPCR fact book,[29] unlike GPCRDB, which organizes the set of GPCRs based on the pharmacological classification of receptors. The corre-

sponding hierarchical lists of GPCRs disagree considerably. As a result, a combination of subfamilies reported by either PRINTS or GPCRDB has been used as a reference database against which to compare the set of clusters identified by our system. This maximizes the number of potential functional groups that could be matched. The two hierarchical lists are merged by adopting the finer classification, whenever a discrepancy exists between them, while maintaining consistency with both hierarchical lists whenever possible. The result is a set of nonoverlapping groups, called *base groups* (*b*-groups), that cover the entire GPCR set, and a set of combinations of the base groups, called *composite groups* (*c*-groups), that either contain or are contained completely by one another. A list of these groups is available on the Web.[30] Only groups containing at least three member sequences are considered in this analysis.

Each node in the tree constructed by our system, called an *f*-class, is compared to the set of *b*- and *c*-groups based on the percent overlap of the list of member protein sequences. Any *f*-class that highly overlaps a *b*- or *c*-group, in the sense that the number of false positives and that of false negatives with respect to the group are both small, is assigned the functional label of the corresponding group. Overlap is computed as:

$$n_{fp} \leq n_{\max}, n_{fn} \leq n_{\max}; n_{\max} = \max(round(\alpha \cdot n), 1) \quad (3)$$

where n_{fp} is the number of false positives (*f*-class members not in the *b*- or *c*-group), n_{fn} is the number of false negatives (*b*- or *c*-group members not in the *f*-class), n is the total number of elements in the *b*- or *c*-group, and α is a percent coefficient. Setting $n_{\max}$ to at least one allows some tolerance even for very small groups. Results for $\alpha = 0.1$, 0.2, and 0.3 are reported in a later subsection.

Second method. The *f*-motifs (those associated with each *f*-class) are compared against the PRINTS fingerprints (*p*-motifs). The latter are produced by a supervised procedure where motifs are extracted from manually selected functional families. Matches provide some evidence that our unsupervised procedure is successful in identifying known functionally significant motifs. *F*-motifs that are not present in PRINTS, on the other hand, invite further analysis as they may be related to previously unknown functional regions.

Motifs are compared as follows: for each pair of *f*-motif and *p*-motif, it is determined whether their respective support sequence sets overlap. If they do, it is further determined whether there is any overlap between the regions where the *f*- and *p*-motifs are incident.

Third method. Individual residues in the *f*-motifs are compared against the existing database of functionally assayed residues reported in GPCRMD. If the residues in GPCRMD are incident on *f*-motifs, this provides some evidence that our method is effective in identifying functionally significant residues. In that case, those residues in the *f*-motifs that are not matched by any residues in GPCRMD present themselves as interesting targets for mutagenesis experiments. Results suggest that this approach, called "synthetic mutagenesis," is universally applicable to identify potential functional residues and as an aid to direct mutagenesis assays.

If the GPCRMD residue occurs in a sequence (*r*-sequence) that belongs to an *f*-class, the documented position of the residue is directly compared with the position currently under consideration in any occurrence of the corresponding *f*-motif. Otherwise, a profile HMM is constructed from all the occurrences of the *f*-motif, the *r*-sequence is aligned with the profile HMM by using HMMSearch, and the documented position of the residue is compared with the position currently under consideration in any occurrence of the *f*-motif. This is useful for determining functional residues that belong to regions that have high homology to a region identified by an *f*-motif but may have been missed due to the tree-splitting procedure.

Experimental results. In this section, we report experimental results for the following studies: the robustness of our method based on an exploration of the parameter space, the clustering performance based on the number of *b*- and *c*-groups that overlap with *f*-classes, PRINTS comparison results, and functional residues analysis. Also, for each *f*-motif, the locations of its occurrences are annotated with putative structural (transmembrane, intracellular, or extracellular) information. The functional significance of each of these regions is reported. These results are available on the Web.[30]

Algorithm robustness. Table 1 reports the results of the top-down clustering method for various values of the parameters. Some of the rows in the table are shaded for grouping purposes. The most relevant parameters of the system are the following:

Table 1 Results of parameter space analysis

Row Index	k_0, l_0	t_0	N_-	e_1, e_2	N_m *with* α = 0.1, 0.2, 0.3	N_P	N_{Res}
Exact Clustering							
1	4, 8	4	3	1, 1	53, 58, 74	854	338
2	4, 8	6	3	1, 1	100, 108, 122	1042	339
3	4, 8	8	3	1, 1	108, 121, 136	1005	318
4	4, 8	10	3	1, 1	115, 122, 139	994	347
5	4, 12	8	3	1, 1	109, 116, 134	1051	312
6	6, 12	8	3	1, 1	116, 123, 139	1080	334
7	8, 12	8	3	1, 1	106, 114, 135	1038	357
8	4, 8	8	1	1, 1	109, 121, 134	1072	335
9	4, 8	8	10	1, 1	115, 128, 144	1039	359
10	4, 8	8	20	1, 1	120, 129, 149	1045	371
11	4, 8	8	3	10, 10	108, 116, 134	1055	319
12	4, 12	6	3	10, 10	97, 107, 139	1089	330
13	4, 12	8	3	10, 10	105, 113, 134	1038	314
14	4, 12	8	10	10, 10	104, 108, 122	1079	352
15	4, 8	8	3	0.1, 0.1	122, 129, 147	1083	366
16	**4, 8**	**8**	**20**	**0.1, 0.1**	**121, 131, 151**	**983**	**371**
17	4, 8	12	3	1, 1	110, 118, 133	1056	342
18	4, 12	8	3	0.1, 0.1	116, 125, 138	1054	311
19	4, 12	8	10	0.1, 0.1	111, 119, 139	1023	310
With Iterative Pattern Refinement							
20	4, 8	8	3	1, 1	115, 129, 149	1044	323
21	4, 8	8	3	10, 10	120, 129, 146	1029	333
22	4, 8	8	3	0.1, 0.1	112, 124, 138	1040	320
Fuzzy Clustering							
23	4, 8	8	3	0.1, 10	112, 121, 140	1009	341
24	4, 8	8	20	0.05, 5	124, 134, 151	988	342

1. Density constraint parameters, k_0 and l_0. These are studied in rows 5 to 7.
2. Minimum number of tokens, t_0. This is studied in rows 1 to 4.
3. E-value thresholds for HMMSearch, e_1 and e_2. In the exact clustering case, $e_1 = e_2$, and the single E-value threshold is studied for three separate sets with corresponding values of 0.1, 1, and 10.
4. Minimum number of patterns required before selecting one, N_- (Rows 8 to 10).

The results of the analysis are combined in the last three columns of the table. N_m is the number of unique b- or c-groups that match an f-class with at least three member sequences, and there are 212 such groups. N_P is the number of unique PRINTS fingerprints that overlap an f-motif, and there are 1441 such fingerprints. N_{Res} is the number of functional residues in GPCRMD that are incident on f-motifs, and there are 581 such residues. As evident from this table, the results are quite stable with respect to the choice of values for the density constraint parameters and improve with larger values of the t_0 parameter and with lower values of the e_1 or e_2 parameter. This can be understood because more specific patterns are likely to reduce the chances that the HMMs are trained using false positive instances. Furthermore, a tighter E-value threshold is likely to minimize the number of false positives. It is also evident that a larger value of minimum number patterns, N_-, among which an optimal one is chosen, improves the performance of the algorithm. It is interesting that N_P generally goes down as N_m and N_{Res} both go up. We suspect that it happens because the patterns discovered by our system tend to overlap with the same PRINTS fingerprints repeatedly rather than scatter around hitting regions not covered by PRINTS fingerprints, whereas the PRINTS fingerprints might have covered a substantial portion of the entries in GPCRMD.

Rows 20 through 22 report the results for those experiments with iterative pattern refinement for different E-value thresholds. As can be seen from comparing such an experiment with its noniterative counterpart (Rows 3, 11, or 15), the performance of our system improves when the E-value thresholds are relatively loose but does not otherwise. One of the reasons may be that relatively tight thresholds tend to increase the number of false negatives, and a refined profile HMM built from a more selective

subset of sequences tends to further increase the number of false negatives.

Finally, the last two rows report the results for the fuzzy clustering method. There appears to be moderate but not significant improvement over the corresponding exact clustering technique. Notice, however, that with the fuzzy clustering, the size of the tree tends to be bigger and thus N_m is expected to be larger as well.

Identification of functional residues. As opposed to a supervised learning model, such as that used for generating PRINTS fingerprints, our technique makes no assumptions on the functional relatedness of a set of sequences and infers it based only on motif conservation. Since the functional clustering proposed in this paper is based on the occurrence of individual motifs, it is reasonable to assume that if the clustering is relatively successful in recovering functional groups, as shown by Table 1, then the corresponding motifs will contain at least some residues that have a functional nature. This is shown to be the case in Hart,[1] where the three catalytic residues of trypsin (a serine, a histidine, and an aspartic acid) belong, respectively, to the three most statistically significant patterns discovered from a set of 348 trypsin sequences.

Table 1 shows that a large majority of the 581 functional residues reported in GPCRMD correspond to sites on the *f*-motifs. There are a few cases where a residue reported in GPCRMD corresponds to a residue in an *f*-motif that does not occur in the *r*-sequence of the residue, but its set of occurrences align well with the *r*-sequence. As discussed earlier, this can happen if a sequence is misclassified in a node closer to the root of the tree, thus ending in the wrong branch. In this case, direct alignment with the *f*-motif usually shows that a match actually exists. A list of all residues reported in GPCRMD and all the relevant information, such as their biological literature and incident *f*-motifs, are available on the Web.[30]

In total, up to 64 percent of the functional residues are incident on *f*-motifs, for the best-match parameters. This result supports the claims about the usefulness of this technique as an aid to directed site mutagenesis experiments.

Best set of results so far. The best set of results from all the experiments performed is shown in Row 16 in boldface. It corresponds to the identification of 121 to 151 functional families out of 212, or 57 percent to 72 percent, depending on the group-match threshold. More results for this experiment are available on the Web.[30]

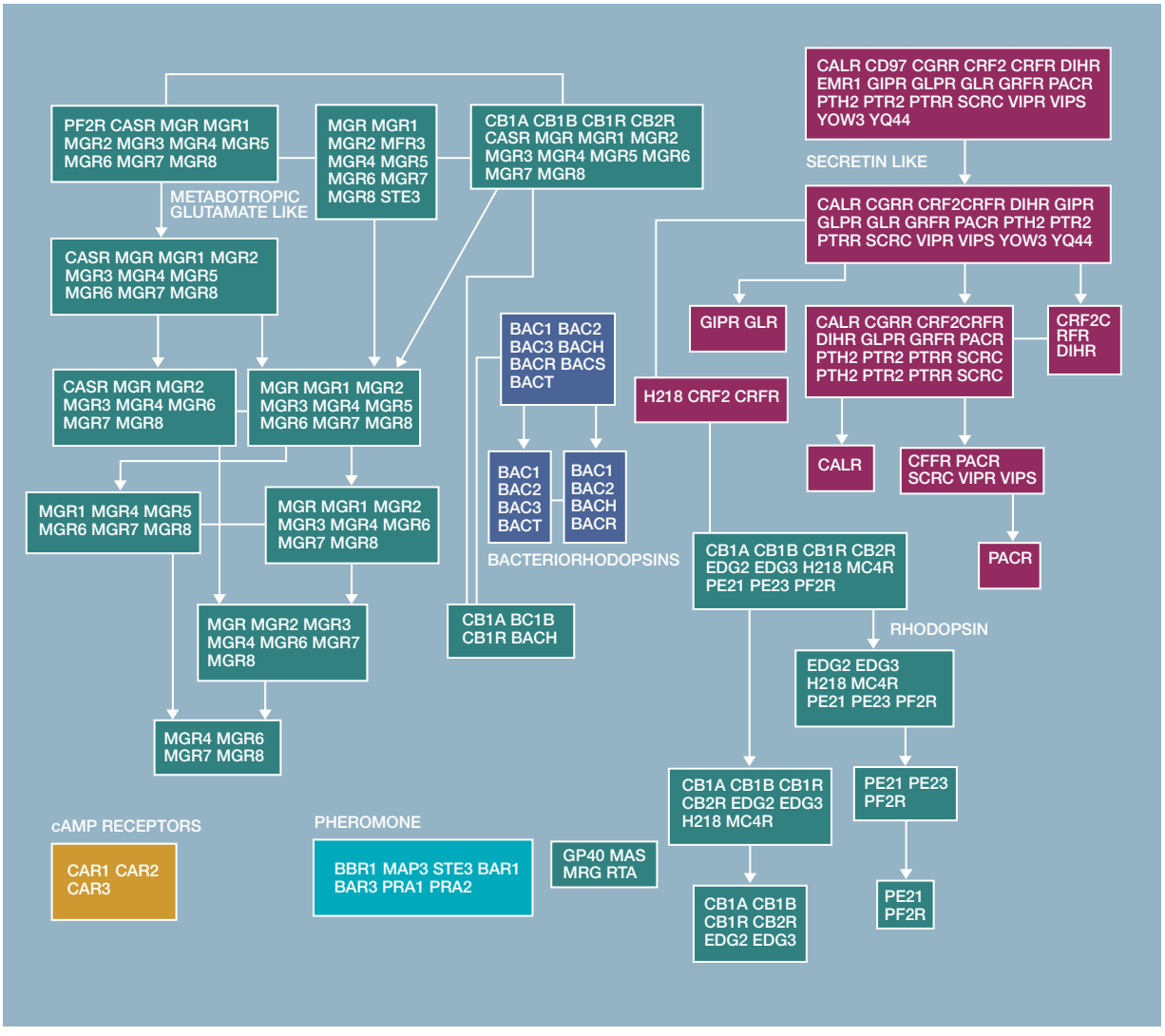
Tree-graph model. In Figure 5, we show a graphical representation of the proteins that do not contain the well-known DRY motif.[27] Individual proteins are identified by the corresponding gene names. Different colors are used to separate biologically different subfamilies.

Related work

A number of alternative approaches to construct various phylogenetic tree representations for various subsets of the GPCRs have been reported. GPCRDB organizes the set of GPCRs based on the pharmacological classification of receptors, as described previously. It then constructs a phylogenetic tree for each group of proteins at the lowest level of the classification, based on pair-wise alignment and a neighbor-joining algorithm via WHAT IF.[31] Results of this analysis are reported at http://www.gpcr.org/7tm/phylo/phylo.html. GCRDb[17] starts from a manually assembled high-level functional classification of the set of GPCRs. It then also constructs a phylogenetic tree for each group of proteins at the lowest level of the classification using the accepted-mutation parsimony method. In contrast to these partially supervised approaches, our proposed method attempts to build a tree for the entire set of GPCRs in an unsupervised manner.

There also exist a variety of sequence-based, automatic protein classification systems of varying scope and emphasis. Our system adopts an unsupervised learning approach, which means that classification is performed in the absence of any prior knowledge. This is an inherently more difficult problem than supervised learning. COG,[10] DOMO,[11,12] HHS/MST,[13] ProDom,[14] and ProtoMap[5] are among those that adopt an unsupervised learning approach. All of these systems, however, adopt a bottom-up or agglomerate approach, constructing small classes first using pairwise local alignment methods and iteratively merging them to form larger classes based on various linkage rules. This is a rather *ad hoc* process and is critically dependent on parameter selection. While our system naturally generates a complete hierarchical classification, all of these systems except ProtoMap[13] focus on sets of highly related proteins and generate classes only at the bottom level. Therefore, they automatically stop constructing larger classes by merging existing smaller classes when the result-

Figure 5 Example of a tree-graph combination



ing classes no longer contain only proteins that are highly related to one another. Rigoutsos[15] also adopts an unsupervised learning approach, and furthermore, the approach uses a pattern discovery method in building a dictionary of motifs. However, it focuses on sets of highly related proteins and does not attempt to build a model of relationships that may exist among the proteins sets.

Our system identifies and manipulates local, disconnected, but conserved regions in protein sequences such that all the proteins assigned to one protein class share the same set of conserved regions. We do so in the hope that the conserved regions correspond to functionally significant areas or particular protein domains. However, our scheme does not attempt to specifically delimit the potential protein domains and thus identify their exact boundaries and locations. In this sense, our system identifies "partial domains" that describe all the proteins in a protein class, just as PRINTS[18,32] and BLOCKS.[22,33,34] DOMO,[11,12] SBASE,[35] and ProDom[14] attempt to identify complete domains, although not necessarily all the proteins assigned to one protein class share the same set of po-

tential domains. The other systems tend to consider all the proteins assigned to one protein class to be globally related, where the global relationship may be formed via transitivity through sharing partial domains.

Our system uses profile HMMs to represent short, disconnected, but conserved regions, which are efficiently identified by discovering regular expressions in the full-length sequences. This allows for a combination of the statistically sound and sensitive HMM technique and an efficient pattern discovery method, such as SPLASH.[2] Pfam[36] is the only other system that uses profile HMMs to characterize protein classes. Unlike other model-based systems, however, it uses the full-length sequences as a training set. This results in a significant computational load, both during training and during the matching phase of the approach.

Our system produces a binary tree, where each node in the tree potentially corresponds to a protein class. Each left (matching) node in the tree is associated with a set of patterns represented by profile HMMs, which can be used in a statistically sound way to characterize all the proteins assigned to the corresponding protein class. The binary tree can then be used directly for sequence annotation. It can also be used directly for classification as a decision tree driven by the *E*-values generated by the profile HMMs at a node. PIMA[37] and ProtoMap[5] appear to be the only other systems that consider statistical significance in an explicit and consistent way. In particular, ProtoMap produces a full hierarchy of classes from the bottom up, by considering different thresholds of statistical significance (tighter thresholds corresponding to higher statistical significance result in lower-level functional classes) and always forming new classes at the current threshold level from the classes formed at the previous, higher threshold level.

Since ProtoMap is the only other automated, bottom-up clustering method that organizes a given set of proteins into a hierarchy of functional classes, and ProtoMap has been run on the set of GPCR proteins, we have carried out a performance comparison between the results generated by ProtoMap and those generated by our system. Specifically, we computed the number of total classification errors (sum of false positive and false negative errors with respect to a group) for these methods, and we did so both in terms of the percentage of matched groups and that of distinct matched classes. In this case, we refer to a protein set identified by either ProtoMap or our system as an *f*-class. There are about 25 percent more groups with low error rate (from one to five errors) reported by our system and, conversely, 25 percent more groups with high error rate (more than five errors) reported by ProtoMap. Also, the percentage of distinct matched *f*-classes is much smaller for ProtoMap (10 percent vs 40 percent), because this method reports a much larger number of *f*-classes (more than 700) and only 212 *b*- and *c*-groups can be matched. In other words, while ProtoMap generates many more *f*-classes, fewer are matched with low error rate to the biologically determined groups.

Conclusion

It has been shown that a combination of regular expression-based pattern discovery, pattern discovery statistics, and hidden Markov model-based pattern refinement and classification can be used to efficiently and accurately identify functional protein clusters in a top-down manner. Experimental results show that the approach is well-behaved with respect to the choice of parameter values and that a significant set of known functional families are successfully identified from the large, functionally differentiated GPCR superfamily. Due to the efficiency of the process and its parallel nature, this could be performed on larger databases such as SWISS-PROT.

Furthermore, it has been shown how the identification of functionally relevant regions in protein can lead to the identification of potential functional residues with high probability. This process, which we call synthetic mutagenesis, could be used to guide the reduction of cost, complexity, and time requirements of real mutagenesis experiments.

Acknowledgments

We would like to thank Gustavo Stolovitzky, Ajay Royyuru, and Reece Hart for many useful discussions and suggestions. We would also like to thank Walter L. Ruzzo, the University of Washington, for many helpful suggestions and discussions.

Cited references and notes

1. R. Hart, A. Royyuru, G. Stolovitzky, and A. Califano, "Systematic and Automated Discovery of Patterns in PROSITE Families," *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, Tokyo, Japan (April 8–11, 2000), pp. 147–154.
2. A. Califano, "SPLASH: Structural Pattern Localization and Analysis by Sequential Histograms," *Bioinformatics* **16**, 341–357 (2000).
3. G. Stolovitzky and A. Califano, Statistical Significance of Pat-

terns in Biosequences, available at http://www.research.ibm.com/splash/Papers/Pattern%20statistics.pdf.
4. A. Bairoch, "The PROSITE Dictionary of Sites and Patterns in Proteins, Its Current Status," *Nucleic Acids Research* **21**, No. 13, 3097–3103 (1993).
5. G. Yona, N. Linial, N. Tishby, and M. Linial, "A Map of the Protein Space—An Automatic Hierarchical Classification of All Protein Sequences," *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology*, Montreal, Canada (June 28–July 1, 1998), pp. 212–221.
6. M. Gribskov, R. Lüthy, and D. Eisenberg, "Profile Analysis," *Methods in Enzymology* **183**, 146–159 (1990).
7. A. Krough, B. Brown, I. S. Mian, K. Sjolander, and D. Haussler, "Hidden Markov Models in Computational Biology: Applications to Protein Modeling," *Journal of Computational Biology* **235**, 1501–1531 (1994).
8. The *P*-value is the probability of a hit occurring by chance; the *E*-value is the expected number of hits. The maximum *P*-value is 1.0, while the maximum *E*-value is the number of sequences in the database that was searched.
9. SPLASH can also discover flexible patterns but the trade-off between efficiency and accuracy is not favorable. In general, also, many reported flexible patterns have one or more rigid cores that can be successfully discovered by the algorithm.
10. R. L. Tatusov, E. V. Koonin, and D. J. Lipman, "A Genomic Perspective on Protein Families," *Science* **278**, 631–637 (1997).
11. J. Gracy and P. Argos, "Automated Protein Sequence Database Classification: I. Integration of Compositional Similarity Search, Local Similarity Search, and Multiple Sequence Alignment," *Bioinformatics* **14**, No. 2, 164–173 (1998).
12. J. Gracy and P. Argos, "Automated Protein Sequence Database Classification: II. Delineation of Domain Boundaries from Sequence Similarities," *Bioinformatics* **14**, No. 2, 174–187 (1998).
13. D. J. States, N. L. Harris, and L. Hunter, "Computationally Efficient Cluster Representation in Molecular Sequence Megaclassification," *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, Bethesda, MD (July 6–9, 1993), pp. 387–394.
14. E. L. L. Sonnhammer and D. Kahn, "Modular Arrangement of Proteins as Inferred from Analysis of Homology," *Protein Science* **3**, 482–492 (1994).
15. I. Rigoutsos, A. Floratos, C. Ouzounis, Y. Gao, and L. Parida, "Dictionary Building via Unsupervised Hierarchical Motif Discovery in the Sequence Space of Natural Proteins," *Proteins: Structure, Function, and Genetics* **37**, 264–277 (1999).
16. F. Horn, J. Weare, M. W. Beukers, S. Hörsch, A. Bairoch, W. Chen, Ø. Edvardsen, F. Campagne, and G. Vriend, "GPCRDB: An Information System for G Protein-Coupled Receptors," *Nucleic Acids Research* **26**, No. 1, 277–281 (1998).
17. L. F. Kolakowski, "GCRDb: A G Protein-Coupled Receptor Database," *Receptors Channels* **2**, 1–7 (1994).
18. T. K. Attwood, M. E. Beck, A. J. Bleasy, and D. J. Parry-Smith, "PRINTS—A Database of Protein Motif Fingerprints," *Nucleic Acids Research* **22**, No. 17, 3590–3596 (1994).
19. A. M. van Rhee and K. A. Jacobson, "Molecular Architecture of G Protein-Coupled Receptors," *Drug Development Research* **37**, 1–38 (1996).
20. See http://www.expasy.ch/sprot/.
21. R. M. Schwartz and M. O. Dayhoff, "Matrices for Detecting Distant Relationships," *Atlas of Protein Sequence and Structure*, M. O. Dayhoff, Editor (1978), pp. 353–358.
22. S. Henikoff and J. G. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," *Proceedings of the National Academy of Sciences (USA)* **89**, 10915–10919 (1992).
23. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, UK (1998).
24. C. Antoniak, "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems," *Annals of Statistics* **2**, 1152–1274 (1974).
25. M. Brown, R. Hughey, A. Krogh, I. S. Mian, K. Sjölander, and D. Haussler, "Using Dirichlet Mixture Priors to Derive Hidden Markov Models for Protein Families," *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, Bethesda, MD (July 6–9, 1993), pp. 47–55.
26. A. Dembo and S. Karlin, "Strong Limit Theorems of Empirical Functionals for Large Exceedances of Partial Sums of I.I.D. Variables," *Annals of Probability* **19**, No. 4, 1737–1955 (1991).
27. S. Karlin, A. Dembo, and T. Kawabata, "Statistical Composition of High-Scoring Segments from Molecular Sequences," *Annals of Statistics* **18**, No. 2, 571–581 (1990).
28. S. Karlin and S. F. Altschul, "Methods for Assessing the Statistical Significance of Molecular Sequence Features by Using General Scoring Schemes," *Proceedings of the National Academy of Sciences (USA)* **87**, 2264–2268 (1990).
29. *The G-Protein Linked Receptor Factsbook*, S. Watson and S. Arkinstall, Editors, Academic Press, New York (1994).
30. A. Liu and A. Califano, A Pattern Discovery-Based Hierarchical Taxonomy for GPCRs, http://www.research.ibm.com/splash/gpcr.
31. G. Vriend, "WHAT IF: A Molecular Modeling and Drug Program," *Journal of Molecular Graphics* **8**, 52–56 (1990).
32. F. Daeyaert, H. Moereels, and P. J. Lewi, "Classification and Identification of Proteins by Means of Common and Specific Amino Acid N-Tuples in Unaligned Sequences," *Computer Methods and Programs in Biomedicine* **56**, 221–233 (1998).
33. H. Moereels, P. J. Lewi, L. M. Koymans, and P. A. Janssen, "The α and ω of G-protein Coupled Receptors: A Novel Method for Classification, Part 1," *Receptors and Channels* **4**, No. 1, 19–30 (1996).
34. H. Moereels, P. J. Lewi, F. Daeyaert, E. Schenck, and P. A. J. Janssen, "The α and ω of G-protein Coupled Receptors: A Novel Method for Classification. Part 2: Bin Classification," *Receptors and Channels* **5**, Nos. 3–4, 139–148 (1997).
35. J. Murvail, K. Vlahovicekl, E. Barta, B. Cataletto, and S. Pongor, "The SBASE Protein Domain Library, Release 7.0: A Collection of Annotated Protein Sequence Segments," *Nucleic Acids Research* **28**, No. 1, 260–262 (2000).
36. E. L. L. Sonnhammer, S. R. Eddy, and R. Durbin, "Pfam: A Comprehensive Database of Protein Domain Families Based on Seed Alignments," *Proteins* **28**, 405–420 (1997).
37. R. F. Smith and T. F. Smith, "Automatic Generation of Primary Sequence Patterns from Sets of Related Protein Sequences," *Proceedings of the National Academy of Sciences (USA)* **87**, 118–122 (1990).

General references

W. C. Barker, J. S. Garavelli, P. B. Mcgarvey, C. R. Marzec, B. C. Orcutt, G. Y. Srinvasarao, L. L. Yeh, R. S. Ledley, H. Mewes, F. Pfeiffer, A. Tsugita, and C. Wu, "The PIR-International Protein Sequence Database," *Nucleic Acids Research* **27**, 39–43 (1999).

P. Bucher and A. Bairoch, "A Generalized Profile Syntax for Bi-

omolecular Sequences Motifs and Its Function in Automatic Sequence Interpretation," *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, Stanford, CA (August 14–17, 1994), pp. 53–61.

S. R. Eddy, "Hidden Markov Models," *Current Opinion in Structural Biology* **6**, 361–365 (1996).

C. M. Fraser, "Structure and Functional Analysis of G Protein-Coupled Receptors and Potential Diagnostic Ligands," *The Journal of Nuclear Medicine* **36**, No. 6, 17S–21S (1995).

L. Oliveira, A. C. M. Paiva, and G. Vriend, "A Common Motif in G-Protein-Coupled Seven Transmembrane Helix Receptors," *Journal of Computer-Aided Design* **7**, 649–658 (1993).

J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia, "Sequence Comparisons Using Multiple Sequences Detect Three Times as Many Remote Homologues as Pairwise Methods," *Journal of Molecular Biology* **284**, No. 4, 1201–1210 (1998).

C. D. Strader, T. M. Fong, M. P. Graziano, and M. R. Tota, "The Family of G-Protein-Coupled Receptors," *FASEB Journal* **9**, No. 9, 745–754 (1995).

C. D. Strader, M. T. Fong, M. R. Tota, and D. Underwood, "Structure and Function of G Protein-Coupled Receptors," *Annual Review of Biochemistry* **63**, 101–132 (1994).

J. Wess, "G-Protein-Coupled Receptors: Molecular Mechanisms Involved in Receptor Activation and Selectivity of G-Protein Recognition," *FASEB Journal* **11**, No. 5, 346–354 (1997).

C. H. Wu, S. Zhao, and H. Chen, "A Protein Class Database Organized with ProSite Protein Groups and PIR Superfamilies," *Journal of Computational Biology* **3**, 547–561 (1996).

Accepted for publication February 9, 2001.

Agatha H. Liu *IBM Research Division Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: ahliu@us.ibm.com).* Ms. Liu is currently a graduate student at the University of Washington. She studied sequencing by hybridization and repeat sequences in the human genome and received her master's degree in computer science and engineering in 1999. She is currently working toward her Ph.D. degree, studying functional classification of proteins based on primary sequences. She received a Graduate Research Fellowship from the National Science Foundation for her graduate study. Along with studying, she also works part-time for IBM Research as a co-op student at the Computational Biology Center. Her current research interests include functional annotation and classification of proteins, analysis of protein primary and secondary sequences, and analysis of protein expression data.

Andrea Califano *First Genetic Trust, Inc., 3 Parkway North Center, Suite 150 North, Deerfield, Illinois 60015 (electronic mail: acalifano@firstgenetic.net).* Dr. Califano was born in Napoli, Italy. He received the Laurea in Physics (*magna cum laude*) from the University of Florence, Italy, in 1985. He continued his thesis research on the chaotic behavior of high-dimensional dynamical systems as a research associate at the Istituto Nazionale di Ottica in Florence, Italy. In 1986 he spent six months as a visiting scientist at the Information Mechanics group at the Massachusetts Institute of Technology, Cambridge. From 1986 to 1990, he was a research staff member in the Exploratory Computer Vision group at the IBM Thomas J. Watson Research Center. In 1990 Dr. Califano became first the manager of the Computational Biology group and later, in 1997, the director of the IBM Computational Biology Center, a worldwide organization focused on bioinformatics, chemoinformatics, complex biological system modeling and simulation, pharmacogenomics, protein structure prediction, and molecular dynamics. He is currently founder and Chief Technology Officer of First Genetic Trust, Inc., a genetic banking company. His research interests are in the exploration and application of pattern analysis and association discovery algorithms to gene expression analysis and single nucleotide polymorphisms (SNPs). Dr. Califano has been an IEEE Fellow since 1997.