Sequence analysis

Bayesian search of functionally divergent protein subgroups and their function specific residues

Pekka Marttinen¹,*, Jukka Corander¹, Petri Törönen² and Liisa Holm²,³

¹Department of Mathematics and Statistics, PO Box 68, 00014 University of Helsinki, Finland, ²Institute of Biotechnology, PO Box 56, 00014 University of Helsinki, Finland and ³Department of Biological and Environmental Sciences, PO Box 56, 00014 University of Helsinki, Finland

Received on February 16, 2006; revised on July 23, 2006; accepted on July 24, 2006
Advance Access publication July 26, 2006
Associate Editor: Dmitrij Frishman

ABSTRACT

Motivation: The rapid increase in the amount of protein sequence data has created a need for an automated identification of evolutionarily related subgroups from large datasets. The existing methods typically require a priori specification of the number of putative groups, which defines the resolution of the classification solution.

Results: We introduce a Bayesian model-based approach to simultaneous identification of evolutionary groups and conserved parts of the protein sequences. The model-based approach provides an intuitive and efficient way of determining the number of groups from the sequence data, in contrast to the ad hoc methods often exploited for similar purposes. Our model recognizes the areas in the sequences that are relevant for the clustering and regards other areas as noise. We have implemented the method using a fast stochastic optimization algorithm which yields a clustering associated with the estimated maximum posterior probability. The method has been shown to have high specificity and sensitivity in simulated and real clustering tasks. With real datasets the method also highlights the residues close to the active site.

Availability: Software ‘kPax’ is available at http://www.rni.helsinki.fi/jic/softa.html
Contact: pekka.marttinen@helsinki.fi
Supplementary information: http://www.rni.helsinki.fi/~jic/softa.html

1 INTRODUCTION

Many finished and ongoing genome projects have produced massive amounts of potential protein sequence data, increasing the demand for automated sequence analysis methods. One particularly critical task in the automation context is the functional annotation of sequences. The functions of the proteins (Mirny and Gelfand, 2002). Still the existing clustering methods often require a selection of some parameter, such as the number of clusters, which has a strong effect on results. Therefore, it is a standard procedure to run clustering with various parameter settings, and to use one of the many possible clustering evaluation methods to find optimal results. For comparison of various ways for protein clustering see Donald and Shakhnovich (2005a).

A drawback of the standard clustering (and clustering evaluation) approaches is that they usually consider all the positions in the alignment equally important, whereas in reality only a small portion is associated with the active site of a protein. Often, only the amino acids within the active site show sequence similarity between distant relatives that still have the same biochemical function. Regular clustering would assign the same importance to all positions and would therefore be mainly influenced by the uninformative positions in the sequences.

We propose to separate the sequence positions into positions that are either informative (signal) or uninformative (noise) for clustering purposes. The clustering is allowed to optimize the results using only the positions that are selected as signal. Similar ideas have been utilized in different fields under names such as feature selection (e.g. Law et al., 2004) and sub-space clustering (Parsons et al., 2004). We present a fully Bayesian model that allows the simultaneous optimization of: number of clusters, the clustering of sequences and the selection of the signal preserving positions. These represent the positions that optimally illuminate diversity between the sequences, being therefore the potential specificity defining residues of the active site (e.g. Donald and Shakhnovich, 2005b). We consider using information provided by the model to further identify the specificity determining residues. We validate
the fundamental properties of the method with simulated datasets where the underlying cluster structure is known, and additionally, with two protein families for which an expert-derived classification to subfamilies and PFAM (Bateman et al., 2002) classification are available. In addition, we analyze the reporting of the functionally important residues and find it to show correlation with the distance from the active site. A close correspondence can be seen between our method and that of Vaithyanathan and Donn (2000), who developed a Bayesian method for text classification, with data divided into noise and signal dimensions. An earlier Bayesian method for protein sequence classification was introduced in Qu et al. (1998), however, they treated the number of classes as an a priori fixed quantity.

2 METHODS

Consider a set of \( n \) proteins whose sequences are aligned, e.g. by the ClustalW algorithm available at EBI (Thompson et al., 1994), or a more recent approach, Maxflow, introduced by Heger et al. (2004). Assume that the sequences are converted linearly into binary strings of length \( d \), where each element equal to unity reflects the existence of a particular amino acid at a specific location in the sequence. Thus, the resulting binary strings are in theory 20 times longer than the original amino acid sequences; however, the columns consisting only of zeros are removed. So, characteristic string \( x_h \) of protein \( h \) takes values in the space:

\[
x_h = (x_h^{(1)}, ..., x_h^{(d)}) \in \{0,1\}^d, h = 1, ..., n.
\]

Here the elements \( 1, ..., d \) of the binary strings are called attributes. The primary target of our analysis is the identification of a suitable integer \( k \) and a corresponding unsupervised classification (i.e. clustering) of the proteins into non-overlapping classes \( s_1, s_2, ..., s_k \). This is conveniently represented in mathematical terms by a partition \( S \) of the integer \( n \). The biological relevance of this representation stems from the assumed evolutionary relatedness among the proteins allocated to the same class of \( S \), which in turn is determined by statistical similarity in the characteristic strings. The general mathematical basis of unsupervised classification using a partition representation of strings from a finite alphabet has been investigated by Corander et al. (2006b). Specific applications of such a framework to classification of molecular marker data have been considered earlier by Corander et al. (2004, 2006a).

On the basis of our prior knowledge we expect that most of the attributes are such that the probabilities of observing their values equal to unity are the same in different classes. This kind of attributes are not informative for classification purposes, and hence we call them uninformative. On the other hand, if the probabilities of observing some attribute equal to unity differ between classes, the attribute is called informative. Let \( D \subseteq \{1, ..., d\} \) denote all the informative attributes. Furthermore, we expect that for each class \( s, i = 1, ..., k \), there exists a subgroup of informative attributes \( T_i \) that are characteristic for the class. These attributes correspond to conserved or group-specific attributes for which the corresponding attribute value is very likely to be equal to unity. We denote the groups of characteristic attributes by \( T = (T_1, ..., T_k) \).

Let \( x = (x_1, ..., x_n) \) represent all the observed data. Our predictive Bayesian model \( M \in M \) for the data consists of the three above specified components \( M = (S,D,T) \), where the model space \( M \) contains all the triplets of the specified type. We proceed by defining a joint probability

\[
p(x,M) = p(x | M)p(M),
\]

where \( p(x | M) \) is a predictive distribution of the data given model \( M \) and \( p(M) \) is a probability distribution representing our prior uncertainty about different models. Our goal is to find the model \( M \) that maximizes the posterior probability:

\[
p(M | x) = \frac{p(x | M)p(M)}{\sum_{M' \in M} p(x | M')p(M')}
\]

Thus, our classification framework aims at a simultaneous identification of the number of classes \( k \), allocation of proteins into the classes and the characteristic attributes of the classes that are best able to predict the patterns observed in the data.

With the assumption of conditional independence of the observations over different classes and attributes, we obtain the predictive representation

\[
p(x | M) = \int_0^1 \prod_{i=1}^k \prod_{j \in T_i} p(x_{ij} | T_i) \prod_{i=1}^k \prod_{j \notin T_i} \prod_{l \in D} p(x_{ij} | T_i, l) \prod_{l \in D} \prod_{i=1}^k \prod_{j \notin T_i} p(x_{ij} | T_i, l)
\]


Three distinct types of conditional probabilities are involved in this representation. First, \( p(x_{ij}, i = 1, ..., k, j \in D \cap T_i, l = 0, 1) \) is the probability that attribute \( j \) equals \( l \) in class \( i \), when we know that attribute \( j \) is characteristic for class \( i \). Second, \( q(x_{ij}, i = 1, ..., k, j \in D \cap T_i, l = 0, 1) \) are the corresponding probabilities for the uncharacteristic attributes in the class \( i \). Finally, \( w_{ij}, l \notin D, l = 0, 1 \) is the probability that an uninformative attribute equals \( l \) for any protein. The parameter \( \theta \) represents jointly all the quantitative parameters \( p(x_{ij}, q(x_{ij}), w_{ij}) \). The observed counts of zeros and ones for an informative attribute \( j \) in class \( i \) are represented by \( n_{ij}, l \in D, l = 0, 1 \). The observed counts of zeros and ones for an uninformative attribute \( j \) among all the proteins are represented by \( n_{ij}, l \notin D, l = 0, 1 \). Note that for a fixed \( \theta \) the integrand in (4) can be seen as proportional to a product of binomial distributions. Thus the prior beliefs \( p(\theta) \) for the parameters of the model are most naturally represented by beta distributions, the conjugate family for binomial likelihood (Bernardo and Smith, 1994):

\[
p(\theta | S, D, T) = \frac{\Gamma(\sum_{i=1}^k \lambda_i) \prod_{l=0}^1 \prod_{i=1}^k \Gamma(\lambda_i + n_{ij})}{\prod_{i=1}^k \Gamma(\lambda_i)}
\]

To complete the classification model, we need to assign a prior distribution for the layer \( (S,D,T) \). This is determined by

\[
p(S,D,T) = p(S)p(T,D | S),
\]
where we set the probability
\[ p(S) = c_1 \text{ (constant)}, \] (10)
according to an uninformative choice, that is, we give equal probability to every partition of \( n \). We further specify
\[ p(T, D | S) = c_2 \xi \sum_{i=1}^{n-1} |D_i| (1 - \xi) \sum_{j>i}^{n} |D_{ij}|, \] (11)
where \( c_2 \) is a constant, \( \xi \) is the probability given by the user that a certain attribute is characteristic for some cluster, and \(|\cdot|\) is the cardinality operator. This prior can be interpreted in the following way: Suppose that we are examining an informative attribute in an arbitrary class. The probability that the attribute is characteristic in this particular class has a Bernoulli distribution with success probability \( \xi \). The formula (11) is then simply the product probability of such Bernoulli random variables over all the informative attributes and classes. This prior is characterized by a penalty for complex models having many classes and many informative attributes.

Bayes factors (Kass and Raftery, 1995) allow us to compare the relative goodness of any two models. Bayes factor (BF) in favor of \( M_1 \) and against \( M_2 \) is given as follows:
\[ B_{12} = \frac{p(x | M_1)p(M_1)}{p(x | M_2)p(M_2)}. \] (12)

The uncertainty related to the allocation of any particular protein \( h \) in the optimal solution can be estimated in a similar fashion. Assume that in the optimal model the protein \( h \) belongs to a group \( i \) and we would like to compare this to a case where groups are kept similar except that protein \( h \) belongs now to some other group, say \( j \). If we assign \( x \) to \( i \) and to \( j \), say, then \( M_1 = (S_1, D_1, T_1) \) refers to the optimal model and \( M_2 = (S_2, D_2, T_2) \) refers to a competing model where the partition \( S_2 \) is the same as \( S_1 \) except for the allocation of the protein \( h \). \( D_2 \) and \( T_2 \) are chosen to correspond partition \( S_2 \). Now the BF (12) in favor of \( M_1 \) and against \( M_2 \) can be used as a measure of plausibility of allocating the protein \( h \) into the group \( i \) instead of the group \( j \). BFs can be also analogously computed attribute-wise. In this case they provide a measure of an attribute being informative, e.g. characteristic to some group, against the alternative hypothesis that the attribute is uninformative. Thereby, the values of the BFs can be used to detect the functional areas in the protein sequences. The values for the mentioned BFs are provided in our implementation.

The interpretations of the different model parameters and the choice of their actual values are further considered in the Implementation section.

### 3 ALGORITHM

Here we provide details for an algorithm that searches the model space \( \mathcal{M} \) to identify the optimal Bayesian solution to the classification problem, i.e. the triplet \((S, D, T)\) maximizing the posterior probability (3). Since the denominator in (3) is the same for all models it is sufficient to find the model maximizing the numerator which is given by (2). As it is possible to analytically evaluate both the terms \( p(x | M) \) and \( p(M) \), the problem can be viewed generally as an optimization problem for which numerous algorithms have been proposed in the literature. We have revisited a greedy stochastic optimization algorithm developed in Corander et al. (2006a) for unsupervised classification of molecular marker gene data. The algorithm of Corander et al. (2006a) implemented in the BAPS 3 software has in the literature been shown to perform very well in a wide variety of challenging situations.

The central search space of our algorithm contains all possible partitions \( S \) of \( n \). The search can be initialized either manually or with a state obtained by fast heuristic classification methods. The algorithm proceeds by repeatedly applying different kinds of transition operators to the current state (corresponding to a particular value of \( M \)). The changes that lead to an improvement in the posterior probability (3) are accepted. After each proposed change to the partition \( S \), the optimal groups of informative (\( D \)) and characteristic (\( T \)) attributes are identified, and using these, the putative partition is evaluated. The search terminates when no changes leading to an improvement in the posterior probability are available.

The different search operators available in our algorithm are:

1. Move a protein from one class to another.
2. Join two classes.
3. Using some heuristic distance measure between the proteins, divide a class to smaller classes.

Owing to the relatively complex objective function, the sole use of local changes such as those provided by the first alternative may easily yield a local maximum of the posterior. Therefore, it is of importance to have a variety of informative distance measures in the last alternative to guide the search and facilitate greater jumps in the search space. The different distance measures that we utilize in the splitting step are as follows:

1. Hamming distance between the sequences of the proteins.
2. Hamming distance between the protein sequences computed using only such attributes that equal unity for at least one of the two considered proteins.
3. Hamming distance of the sequences of the proteins, using only such attributes that are most diverse in the class (i.e. those attributes for which the number of zeros and ones is about equal). This distance is defined only between the members of the class that is being currently processed.
4. Hamming distance computed using only the attributes for which the total number of ones among all the proteins is above the general noise level.

In the core, the algorithm we use is a greedy stochastic search algorithm. Once a partition \( S \) is fixed, the values of \( D \) and \( T \) can be obtained exactly. However, as is the case with most greedy algorithms, there is no guarantee that the final solution \( S \) is really the optimal one. To obtain a higher level of certainty, the algorithm should be started from different initial assignments. If it happens that the algorithm ends up in two different maxima on different runs, the obtained solutions can be compared using Bayes factors. Alternative initial values of \( S \) can be easily sought by using some of the distance measures described earlier and dividing the proteins to some putative number of clusters on the basis of their heterogeneity. Also, the more different search operators the greedy algorithm has available at a certain point, the more likely it is to find the optimal solution. In our algorithm there are available local operators moving a single protein as well as global operators that treat groups of proteins.

The run-time of the algorithm depends strongly on the initial state of the search and the order in which different search operators are carried out. The most time-consuming search operator is the first one, which consecutively tries to move every protein to every other cluster. The time-complexity of this search operator is proportional to the product of the size of the dataset and the number of clusters. This can be used as a rule of thumb to estimate the total run-time of the algorithm. Table 1 shows approximate run-times of the algorithm on a 1.3 GHz P4 processor for datasets of varying size.
The question of how to choose the sums $\lambda_1 + \lambda_0$, $\mu_1 + \mu_0$, and $\tau_1 + \tau_0$, can be approached theoretically. The value of the sum is related to the amount of prior information fed into the model. If the sum is small, say one, the prior information becomes negligible already after a few observations. This kind of a choice may lead to contradictory conclusions where the model implicates that some attribute is characteristic in a class although every protein allocated has the attribute value equal to zero. This is due to the fact that the first observed zeros update the distribution of the characteristic attribute such that it is actually preferable to observe even more zeros. Therefore the values of the sums should not be too small. Choosing the values of the sums to be much larger than the number of observations, corresponds to a situation where the probabilities $p_{ij1}$, $q_{ij1}$, and $w_{ij1}$ have a point distribution with the probability mass completely concentrated at the expected value of the distribution. This kind of an approach might be reasonable, although not utilizing all the flexibility allowed by a model-based approach.

We have analyzed more than 200 simulated datasets with different choices of the values of the sums $\lambda_1 + \lambda_0$, $\mu_1 + \mu_0$, and $\tau_1 + \tau_0$. The simulations suggested that these values should not be constant but they should depend on the size of the corresponding evolutionary group that is identified from the data. This led us to the following prior distributions for $p_{ij1}$, $q_{ij1}$, and $w_{ij1}$:

$$p_{ij1} \sim \text{Beta}(0.95\phi_i, 0.05\phi_i),$$  \hspace{1cm} (13)

$$q_{ij1} \sim \text{Beta}(f * \phi_i, (1 - f) * \phi_i),$$  \hspace{1cm} (14)

$$w_{ij1} \sim \text{Beta}(f * n, (1 - f) * n),$$  \hspace{1cm} (15)

where $n$ is the total number of proteins and $f$ is the overall frequency of ones in the data. $\phi_i$ is the scaling factor that depends on the number of proteins in class $i$. We set

$$\phi_i = \max\{|s_i|, 10\},$$ \hspace{1cm} (16)

where $|s_i|$ is the number of proteins in class $i$. This choice of the hyperparameters preserves the earlier formulated expected values. The value of the sums $\lambda_1 + \lambda_0$, $\mu_1 + \mu_0$, is chosen to be the number of proteins in the underlying group, except for groups having size $<10$, for which a constant value 10 is used. This choice allows some flexibility for the observed counts of zeros and ones while preventing contradictory behavior of the model as described before. In the simulations such a strategy proved to have more resolution and ability to handle also very small evolutionary groups than the alternative strategy with the sums of the parameters fixed to some large value.

Recall that the amount of prior information fed into the model corresponds to the sum of the corresponding hyperparameters. Thus, the amount of prior information for a model of an informative attribute can be obtained by summing over the groups the values $\lambda_1 + \lambda_0$, or $\mu_1 + \mu_0$, depending on whether the attribute is characteristic or not in a group. If all the groups have sizes $>10$, it follows from (13), (14) and (16), that the amount of prior information in the model of an informative attribute is equal to the total number of proteins in the data. To make the comparison feasible with the competing model of the attribute $j$ being uninformative, in which case all the observations are described by a single pair of parameters, $(w_{ij1}, w_{ij0})$, the value of the sum of the corresponding hyperparameters $\nu_1 + \nu_0$ was in (15) also chosen to be equal to the

<table>
<thead>
<tr>
<th>Data</th>
<th>Proteins</th>
<th>Attributes</th>
<th>Clusters</th>
<th>Run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated</td>
<td>100</td>
<td>7000</td>
<td>5</td>
<td>20 min</td>
</tr>
<tr>
<td>Ureases</td>
<td>296</td>
<td>7344</td>
<td>23</td>
<td>2 h</td>
</tr>
<tr>
<td>Crotonases</td>
<td>915</td>
<td>18020</td>
<td>157</td>
<td>5 days</td>
</tr>
</tbody>
</table>

The sizes of the analyzed datasets with running times of the algorithm on a 1.3 GHz P4 processor. Run-time depends strongly on the initial state of the search algorithm. The presented run-times are approximations for a standard run using a default initial state.
RESULTS
Our method has been tested with real data and under different simulation scenarios. The results from the analyses of simulated data are presented collectively in Table 2. Each row in Table 2 summarizes the results from analyses of five different simulated datasets. The datasets were created by simulating proteins from five different groups, each having its characteristic attributes and frequencies for ones for characteristic and uncharacteristic attributes. The rows 5 and 6 in Table 2 show results from a more challenging scenario where the frequencies of ones in characteristic and uncharacteristic attributes were fixed so that the same values were used for every attribute in every group. In this kind of situation the small differences in the frequencies do not provide any help for the classification, but information is obtained only through the characteristic attributes. Even then the performance of the algorithm was surprisingly good, having the average values of Rand-index 0.86 and 0.90 in the two different settings, respectively. A closer inspection of the obtained partitions revealed that even in these simulations distinct classes were still identified with high accuracy. However, the number of classes was slightly overestimated so that some of the true underlying groups were further divided into subgroups. Dividing one large or two medium-sized groups into two yields a Rand-index value somewhere around 0.9. Comparison with the k-means shows that in most cases our method outperforms both versions of the k-means algorithm. In the last two rows where our method slightly overestimated the number of clusters, the second version of k-means algorithm benefited from knowing the true number clusters and got a bit higher Rand-index values than the presented method.

The last two columns of Table 2 show the average number of correctly inferred characteristic attributes for each individual and the average number of falsely identified characteristic attributes. Inspection of these columns shows that our method is capable of inferring the correct characteristic attributes for the groups with high accuracy. The lowest percentage of recognition of the correct attributes occurred on row 4, and even then, on average 9.1 attributes out of 10 were identified. Also, the number of falsely identified characteristic attributes is very low. Even when the number of clusters was slightly overestimated on rows 5 and 6, where inferred clusters were subgroups of the correct clusters, they had almost the same groups of characteristic attributes as the correct clusters.

Besides the simulated datasets we also tested the method with two protein families, for which we had an expert curated classification into sub-families: ureases (Heger and Holm, 2003) and crotonases (A. Sivakumar, personal communication). The sizes of the datasets are shown in Table 1. We evaluated the obtained partitions by comparing them to the expert classifications using Rand-index. We also used Pfam classification (Bateman et al., 2002) as another reference whenever it was available for the sequence. For comparison, we analyzed the same datasets with two simple standard clustering methods, k-means and linkage clustering (e.g. Ripley, 1996), and one state-of-art method, SPC (Tetko et al., 2005). k-Means and linkage clustering were run using the expert opinion of the number of underlying groups. Also, we tried k-means and linkage clustering with different distance measures, from which the distance 2 described in the Algorithm section had the best performance. SPC was run with default settings and with small
Table 3. Comparison of different clustering methods with reference clusterings

<table>
<thead>
<tr>
<th>Method</th>
<th>Urease (man.)</th>
<th>Urease (Pfam)</th>
<th>Crotonase (man.)</th>
<th>Crotonase (Pfam)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduced method</td>
<td>0.96</td>
<td>0.64</td>
<td>0.87</td>
<td>0.75</td>
</tr>
<tr>
<td>SPC</td>
<td>0.89</td>
<td>0.62</td>
<td>0.87</td>
<td>0.76</td>
</tr>
<tr>
<td>Kmeans2</td>
<td>0.96</td>
<td>0.64</td>
<td>0.89</td>
<td>0.78</td>
</tr>
<tr>
<td>Single linkage</td>
<td>0.68</td>
<td>0.61</td>
<td>0.84</td>
<td>0.70</td>
</tr>
<tr>
<td>Complete linkage</td>
<td>0.87</td>
<td>0.52</td>
<td>0.89</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Results given by five different clustering methods were compared with reference clusterings using Rand-index. The comparison was performed using two sets of proteins, urease and crotonase, and the reference clustering was either the expert hand-curated clustering (man.) or Pfam clustering. Kmeans2 refers to the k-means algorithm using distance measure 2, as described in the Algorithm section, with the correct number of clusters. Both the linkage based methods were also run with the correct number of clusters.

Table 4. The table shows the inferred numbers of clusters for urease and crotonase datasets by two different methods, the introduced method and SPC. Also, the expert opinion of the true number is shown

<table>
<thead>
<tr>
<th>Inferred numbers of clusters</th>
<th>Urease</th>
<th>Crotonase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert opinion</td>
<td>23</td>
<td>157</td>
</tr>
<tr>
<td>Introduced method</td>
<td>24</td>
<td>103</td>
</tr>
<tr>
<td>SPC</td>
<td>17</td>
<td>63</td>
</tr>
</tbody>
</table>

variations to them. From all the results only the most reasonable were chosen for the analysis. The results are collectively presented in Tables 3 and 4. The comparison of our method with SPC is the most interesting, since these are the two methods that are capable of inferring the underlying number of clusters. Table 3 shows that performances of the two methods do not differ much when measured with Rand-index. However, SPC did not give a classification to all of the proteins. For ureases, SPC found a classification for 264/296 proteins, and for crotonases for 623/915 proteins. The results for the Tables 3 and 4 were calculated using only these proteins. Alternative strategy would have been to consider all non-clustered proteins as singleton clusters, but this would have lead to a harsh overestimation in the number of clusters. Table 4 shows that the number of clusters as inferred by our method is closer to the expert opinion than that obtained from SPC analysis. However, it must be acknowledged that SPC works extremely fast compared with our method.

We also wanted to evaluate the selection of functionally important features from the protein sequences. Owing to space constraints we limit this analysis to the urease family. Our program enables two ways of monitoring the important amino acid positions: cluster-specific feature representation and global feature representation. Cluster-specific presentation shows sequences as separate clusters, with cluster-specific over-represented features. Amino acids in an important position in the alignment are represented with capital letter, whereas the rest of the alignment is shown in the lower case letters. Owing to the large size this is shown only in the additional material.

Global feature representation plots the Bayes factor (BF) between the signal model and noise model for each position of amino acid sequence, combining the importance of the position in all the clusters. Clustering actually produces 20 attribute values (one for each amino acid type) for each position. In order to maximally distinguish features of this information we combined two measures, (1) the maximum attribute value and (2) the sum of all attribute values for each alignment position. Results are shown in Figure 1a, showing high values for positions that were considered important by the clustering process. Notice the two types of well-performing areas: regions where the maximum value and the sum are high, and regions where the sum is high, although the maximum does not have a similarly high value. The first case corresponds to the regions where the entire signal is represented by one conserved amino acid type (such as positions 184 and 55), therefore being the functionally very critical positions. The latter case represents the positions where few amino acid types occurring in combination contain a considerable signal (positions such as 293 and 268), thus being the most potential candidates for specificity determining residues.

Similar information can be produced by calculating the Shannon entropy (SE) from the original sequence alignment (e.g. Weiss et al., 2000) without any clustering 1b, or calculating it separately for each cluster and averaging the cluster scores 1c (Mihalek et al., 2004). We used the obtained clustering result to create profile in 1c, so the differences between 1a and 1c are only caused by a different measure. There are also two ways of calculating the SE: comparing the number of each amino acid type to total number of amino acids in the position, and comparing the number of each amino acid type to total number of sequences in the dataset. Profile in 1b is calculated using the first method as it gave more reasonable results than the second (which is shown with a dotted line in 1b) whereas 1c shows both methods, as they give similar results.

To ease the comparison we have marked six residues from active site (five metal binding, one catalytic; swissprot entry PYRC_METJA) with asterisks to all the profiles. These positions differ so that 61, 63, 184, 222 and 295 are invariant, staying similar in most clusters, whereas position 147 varies and has either lysine, glutamate or no feature at all in different clusters. Entropy calculated from the original alignment 1b finds the invariant positions (ranks 6, 5, 2, 11, 1 for 61, 63, 184, 222 and 295) whereas the position 147 is totally lost (rank 138). SE, obtained with clustering 1c gives mainly similar peaks as 1b. Now the position 147 has obtained a weak peak. Still the most striking feature in 1c is the strong local bias in the baseline (average of local minimums), especially at region around position 200. Varying baseline is seen with both methods. Analysis of alignment (see additional files: urease_output.txt) revealed that this region has got mostly gaps. SE is based on maximum likelihood model and such models can have problems when there is a varying amount of information available, like in the positions that have larger number of gaps.

The maximum BF profile picks most invariant positions nicely (ranks 3, 4, 1, 5 for 61, 63, 184 and 295, respectively) with the surprising exception of position 222 (rank 26) that does not perform as well as with SE. Still the largest benefit of BF profile can be seen in the sum profile that can monitor the combinatorial signal as now the position 147 gets a clearer signal (rank 38). Notice also that
BF profiles do not suffer from the varying baseline level observed in the SE with clustering 1c.

Previous invariant positions are often easy to find from alignment simply by eye (Holm and Sander, 1998). The positions that are conserved only within clusters and vary significantly between them are harder to find. As these positions are not as well known, we used structural information (from PDB entry 1xge, PYRC_ECOLI) and calculated the minimum distance from every
residue to any heteroatom (metal ion and ligands). Obtained negative distance value profile is shown in 2d. We mapped the known positions from active site to this profile, showing that, apart from the position 147, all are placed within 2 Å from heteroatom. Exception in the results at the area around the 150th position is due to the bad quality of the alignment, and the distance results of this region seemed unreliable. The correct distance is manually added as another higher asterisk to the 147th position.

Obtained distance profile shows many regions close to the active site. We further highlighted the positions within top 30 results by marking the first and last position from close region with a red circle to the profiles. Note that these circles occasionally collide with known active site positions. Now it is clear that the BF sum profile gives better scores for areas close to the active site, although some of the positions are also picked by SE methods. This can be seen with regions at positions 63 and 295, where SE scores drop dramatically (practically to minimum) whereas the BF sum stays quite high, agreeing more with the distance profile. Region at position 90 gets clear peak from BF sum whereas the SE profiles do not show signal at all. Areas at positions 123 and 269 are close to stronger peaks in BF sum but are also elevated themselves, whereas in SE profiles these areas have no signal. SE methods pick correctly the point (246), but it is only BF sum that considers also the neighborhood of this point important, again showing more agreement with distance. Low score in 1b points that many of these positions cannot be found from the alignment without the clustering (see the alignment.txt in Supplementary Material). To further highlight the correlation between the distance and the BF sum, we show a scatter plot of these two values (Fig. 2). Notice that all the positions closer than 5 Å to the heteroatoms get score >50. Also most of the positions that have score >150 are within 10 Å from the active site. We observed an approximate border at the 10 Å distance where we have most of the signal in between 50 and 250, whereas beyond 10 Å signal seems to be from ~0 to 150. Especially the positions that get strong signal from BF and lie within 10 Å from the active site seem like good candidates for functional analysis (for a detailed analysis of some positions see Supplementary Material). Other methods did not show similar correlation in the scatter plots (see figure in Supplementary Material).

6 DISCUSSION

We have presented a method for simultaneous clustering of proteins into evolutionary groups and finding the active areas in the sequences of the proteins in different groups. The method has been shown to perform well in a variety of simulated scenarios and with two empirical datasets. The potential of using the method for identifying the functionally important residues has been highlighted by showing the correlation of the BF sum with the distance from the heteroatoms at the active site. Recognition of functionally important residues is one of the key problems in biosciences. Although the results obviously need evaluation with other structures, our results still show that the obtained BF sum should be among the methods considered, when analyzing the key residues. The greatest advantage of our method over other methods comes from the possibility of a probabilistic (predictive) comparison of any two solutions. This fully Bayesian approach also provides the user an estimate of the most likely number of different groups that are present in the data. The statistical power to detect different groups in the data is increased by the fact that the model is able to learn which areas in the sequences are most relevant for classification purposes. Also, biological information concerning the characteristics of the functional areas in the sequences is used in the formulation of the model.

It should be noted that the presented method can be applied at three different levels (1) use of the proposed algorithm and model to find the needed clustering solution and function specific residues, as done here, (2) evaluation of a large group of clustering results with the specified model and the selection of the optimum clustering and the corresponding function specific residues; and (3) analysis of manually selected optimum cluster solution for the selection of functionally important residues.

A considerable effort was devoted to the specification of the hyperparameters of the model. However, it should be stressed that, although this may appear as a tedious task, at least in our experiments all the reasonable values for the parameters gave results that were in accordance with each other. We have managed to define the parameters in a way that can be justified theoretically, and is supported by a large number of simulations. One advantage of scaling the hyperparameters as described in Section 4 is that the resulting model can be expected to perform well with datasets of widely varying magnitude. This will allow us to develop the software further in the future by simply considering the computational efficiency of the algorithm, while the model itself remains the same. One further advantage of our strategy is that it does not require any inputs regarding the parameters of the model from the user. This saves the potential time needed in testing a range of possible alternatives for the model (hyper)parameters.

Owing to the overwhelming size of the model space in a typical application, it seems to be beyond the capabilities of any thinkable method to provide a reliable estimate of the probability distribution.

Fig. 2. Scatterplot showing the correlation between the (negative) distance (Å) to the active site and the sum of log(BF) (Y). Vertical border line highlights the difference in the scores when the distance goes from <10 Å to >10 Å. Horizontal border at 150 highlights that most of positions with this score are within 10 Å distance from active site. Horizontal border at 50 highlights similarly that none of the positions within 5 Å and few within 10 Å gets BF sum <50.
over the whole model space. Therefore we suggested the use of Bayes factors to locally estimate the uncertainty related to the optimal solution. Bayes factors provide a way to compare any two models. They also allow us to estimate the uncertainty related to the allocation of any single protein into the classes. The properties of the obtained solution can also be assessed through the sets of characteristic attributes inferred for each group. First, they provide a way to detect a putative overestimation in the number of classes: if two classes have nearly equal sets of characteristic attributes, it is clear that in reality they are very close to each other. Second, if some class has none or very few characteristic attributes it may indicate that the class does not form an evolutionary group, but may consist of outlier proteins for which a classification is not clear. This latter kind of a behavior has been observed in some of the analyses that we have performed.

One drawback of our method is the requirement of a reliable multiple sequence alignment. Although some practical solutions for multiple sequence alignment exist, the problem still remains open for further research. Especially, no truly Bayesian model-based methods have been presented for the problem, and yet this would seem to provide a natural way to estimate the uncertainty. One of our future goals is to better take the uncertainty related to the alignment into account. We also consider the possibility of simultaneous multiple sequence alignment and clustering with the sequences. The model-based approach could be further utilized in comparison of the different clusters that are present in the estimated optimal model. In particular, it would be feasible to do a search (e.g. MCMC) in the space of different ways to partition the clusters into greater groups. Results of this kind of an analysis could be summarized using a dendrogram based on probabilities in a similar manner as in Corander et al. (2004), yielding a hierarchical clustering. Another issue where our method could be improved is to better utilize the biologically meaningful information related to the various types of conservation observed in amino acids. This kind of detailed information could be brought into our model by a careful modification of the prior distributions. However, even without the suggested improvements our method already shows a reasonable performance.

Another area where our method is behind the distance-based methods (e.g. SPC) is the time consumption. This is a result of the model-based approach and the resulting need to search through a vast space of different models. However, the current implementation of the method is capable of dealing with challenging real-life data in a reasonable time (e.g. crotonases). Also, we feel that the advantages of the model-based approach are worth the introduced computational cost. We plan to improve the computational efficiency of our method by implementing a parallel version of the algorithm that could utilize a multi-processor computer architecture. At the same time we will introduce more intelligent steps to the search algorithm, possibly by combining our method with the NMF method presented in Heger and Holm (2003). These advances will allow users to analyze datasets on much larger scale than presented here. We believe that the method presented in this paper has potential of becoming a standard tool in the analysis of protein sequence data.

ACKNOWLEDGEMENTS

This work was supported by the research funds of the University of Helsinki and Tekes (grant no. 40672/01).

Conflict of Interest: none declared.

REFERENCES


Corander,J. et al. (2006b) Bayesian unsupervised classification algorithms based on parallel search strategy. Pattern Recogn., under revision.


Teo,K.Y. et al. (2005) Super paramagnetic clustering of protein sequences. BMC Bioinformatics, 6, 82.

