Using Domain Knowledge and Shift of Origin in Clustering Similarity Data *

Boris Mirkin[†] Renata Camargo, Trevor Fenner, George Loizou[‡] Paul Kellam[§]

Abstract

Clustering is an activity purported to help in enhancing knowledge of the domain the data relate to. We describe a method for clustering similarity data derived within the data recovery framework. In this method, ADDI-S, one parameter, the similarity shift value, determines other clustering parameters such as the number of clusters.

Since similarity data do not involve entity features directly, the shift value is useful as a mechanism that relates domain knowledge to the clustering. The similarity shift value serves as a soft threshold that can be obtained as follows. We assume that domain knowledge can provide two sets of pairs of entities: those that should and those that should not be placed in the same clusters. This data may considerably narrow the choice of reasonable threshold values; this is illustrated using the problem of aggregating motif-defined homologous protein families over herpesvirus genomes. We further show that, in a situation in which there is an independent interpretation device (such as reconstruction of the evolutionary histories of protein families corresponding to clusters), this may lead to further reduction of choices for the clustering using the criterion of consistency among the interpretations.

This approach leads to a number of substantively meaningful results for herpesvirus data. In particular, we indicate a set of proteins that arguably represent descendants of the same gene despite having lost all similarity between their amino acid sequences. Nevertheless, this situation can be recognised if their corresponding neighbouring genes are always homologous.

1 Introduction

1.1 Clustering and Similarity Shift

Similarity data is an important data type that emerges naturally, for example, out of web interaction networks, as well as from the analysis of complex data, such as protein sequences or foldings. On the one hand, there have been a number of heuristic algorithms proposed for clustering similarity data, some recently reviewed in [7]. On the other hand, there exists a long standing tradition of data recovery criteria and methods for clustering similarity data (see, for instance, [16, 38, 27]). Clustering methods considered in this paper are within this second tradition and are, in essence, extensions of methods proposed in [27, 29].

These methods are based on modelling similarity data by weighted sums of partitions or clusters, the clusters and their weights being determined by minimizing the differences between the given similarity data and those generated by the putative model. We utilize the least-squares criterion for explicitly expressing the data recovery approach. We extract clusters one-by-one [26, 27],

[*] The authors thank the Wellcome Trust for its support under Grant 072831/Z/03/Z to Birkbeck University of London.

[†] For communications, use e-mail address: mirkin at dcs.bbk.ac.uk

[‡] All from School of Computer Science and Information Systems, Birkbeck, University of London, UK

[§] Centre for Virology, Department of Infection, University College London, UK

which not only finds clusters effectively, but also supplies meaningful estimates of their intensity and contribution to the data scatter. In a data recovery clustering model, there is a parameter analogous to the intercept of the regression line that plays the role of a similarity shift applied prior to clustering. This parameter is a kind of similarity threshold, so that entities whose similarity is less than it are unlikely to get combined in the same cluster. Its value, which may strongly affect the number and contents of the clusters, could be derived according to the least-squares criterion. However, as we shall illustrate, a better choice may be made by using domain knowledge.

A clustering method, ADDI-S, derived from the data recovery approach is applied to evolutionary intergenomic studies in which homologous protein families (HPFs) contain similar proteins from different genomes. These families are assumed to be inherited from the same ancestral gene and are therefore parsimoniously mapped to an evolutionary tree on the set of genomes under consideration, thereby reconstructing the HPFs' evolutionary histories and the ancestral genomes. Obviously, these histories may critically depend on the level of aggregation: a highly aggregated family intersecting all or almost all genomes would be mapped to the last common ancestor. However, if the family is partitioned, the parts would be mapped to different, more recent, ancestors. These two mappings would lead to two different histories of the function of the HPF under consideration. In other words, the evolutionary mapping of protein families is a rather powerful interpretation tool that can be used for fine tuning the similarity threshold/shift value by analysing the consistency of the reconstructed histories of different functions.

1.2 Neighbourhood Approach

We demonstrate how this approach can work in the framework of the analysis of HPFs over a set of 30 herpesvirus genomes representing three superfamilies residing in different tissues of humans and animals. Specifically, starting with HPFs that have been pre-clustered on the basis of a similar contiguous fragment in the virus database VIDA [2], we try to further aggregate them according to their overall similarity. We assign to each HPF, as its *neighbourhood list*, the set of proteins similar to those in the HPF, and then define similarity between HPFs according to the similarity of their neighbourhoods rather than of the sequences themselves. This neighbourhood approach is utilised for the following reasons.

(a) It is robust. Specifically, it may overcome the problem that conventional sequence similarity scores, obtained using tools such as PSI-BLAST [3], do not necessarily correspond to the 'real' homology between proteins (see examples in section 5.2).

(b) It is universal. Similarity between sets is a relatively well studied problem that, unlike sequence alignment problems, does not rely on empirical parameter values..

(c) It may have an evolutionary meaning in terms of the HPFs (see section 5.2).

1.3 Interaction with Domain Knowledge

To determine an appropriate value for the similarity shift, we analyse a set of pairs of HPFs whose functions are known. The expectation is that proteins with the same function should be more similar to each other than would be proteins with dissimilar functions. This should indicate an appropriate similarity value that could distinguish those pairs that should be in the same cluster from those that should not. The actual distribution of similarity scores turned out to be more complex than we had hoped, and two reasonable similarity shift values emerged: one which would guarantee that HPFs with dissimilar functions would be in different clusters, whereas the other would give the minimum relative error in separating protein pairs with similar and dissimilar functions. Both of these values are derived using domain knowledge. The final choice, however, requires further domain knowledge, viz. the consistency of the suggested reconstructions of ancestral genomes. In fact, with the current level of knowledge, both thresholds lead to very similar results. However, the latter shift value leads to more consistent reconstructions and was therefore selected. Among further conclusions, for example, is a situation in which some HPFs have little sequence similarity but should be taken as homologous because of evidence coming both from the reconstructed ancestors and the juxtaposition with homologous neighbouring genes.

1.4 Contents

The rest of the paper is organised as follows. Section 2 introduces the data recovery approach to clustering similarity data. The additive clustering model [38] is described in Section 3 and the ADDI-S method for one-by-one clustering [27] in Section 4. Section 5 is devoted to a description of the results of aggregating protein families with ADDI-S and mapping them onto an evolutionary tree of herpesviruses. The domain knowledge used to identify similarity shift values and insight gained from our approach are described in Sections 5.3 and 5.4. In Section 6 we conclude and outline possible future work.

2 Structuring and clustering using the data recovery approach

2.1 Similarity clustering: a review

Let I be a set of entities under consideration and let $A = (a_{ij})$ be a symmetric matrix characterising similarities (or, synonymously, proximities or interactions) between entities $i, j \in I$. The greater the value of a_{ij}, the greater is the similarity between i and j. A cluster is a set of highly similar entities whose similarity to entities outside of the cluster is low.

Similarity is a quantitative feature of pairs of individual entities. It should be noted in this regard that we distinguish between two types of quantitative features. For the first, both summation and averaging are meaningful operations. Physical characteristics, such as time and distance, are examples of this type. For the second type of feature, summation is meaningless, and only averaging is meaningful. Density and temperature are examples of this. We use the term *similarity* data for only this second type of feature. Features of the former type, admitting both summation and averaging, will be referred to as *flow* data. Some examples of similarity data are (i) individual judgements of similarity expressed using a fixed range, and (ii) the probability of both entities being generated from the same source. Some examples of flow data are (i) co-occurrence counts for disjoint categories, and (ii) values of transactions between the two entities. In our view, different clustering models should be used for these two types of data [29]. In this paper we only consider clustering models for similarity data.

Similarity clustering emerged quite early in graph theory, probably before the discipline of clustering itself. A graph may be thought of as a structural expression of similarity data, its nodes corresponding to entities with edges joining similar nodes. Cluster related graph-theoretic concepts include: (a) *connected component* (a maximal subset of nodes in which there is a path connecting each pair of nodes), (b) *bicomponent* (a maximal subset of nodes in which each pair of nodes belongs to a cycle), and (c) *clique* (a subset of nodes in which each pair of nodes is connected by an edge).

Other early clustering concepts include the B-coefficient method for clustering variables using their correlation matrix [18] and the Wrozlaw taxonomy [11]. These are precursors to the ADDI and ADDI-S methods [27], described later, and the single linkage method [16, 15], respectively.

Two more recent graph-theoretic concepts are also relevant: *maximum density subgraph* [12] and *min-multi-cut* in a weighted graph [13].

The density $g(S)$ of a subgraph $S \subset I$ is the ratio of the number of edges in S to the cardinality of S. For an edge weighted graph with weights specified by the matrix $A = (a_{ij})$, the density $g(S)$ is equal to the *Raleigh quotient* $s^T As/s^T s$, where $s = (s_i)$ is the characteristic vector of S, viz. $s_i = 1$ if $i \in S$ and $s_i = 0$ otherwise. A subgraph of maximum density represents a cluster. After removing such a cluster from the graph, a maximum density subgraph of the remaining graph can be found. This may be repeated until no "significant" clusters remain. Such an incomplete clustering procedure is natural for many types of data, including protein interaction networks. However, to our knowledge, this method has never been applied to such problems, probably because it involves rather extensive computations. A heuristic analogue can be found in [4]. We consider that the maximum density subgraph problem is of interest because it is a relaxation of the maximum clique problem and fits well into data recovery clustering (see section 2.3). The maximum value of the Raleigh quotient of a symmetric matrix over any real vector s is equal to the maximum eigenvalue and is attained at an eigen vector corresponding to this eigenvalue. This gives rise to *spectral clustering*, a method of clustering based on first finding a maximum eigenvector s^* and then defining the spectral cluster by $s_i = 1$ if $s_i^* > t$ and $s_i = 0$ otherwise for some threshold t. This method may have computational advantages when A is sparse. Unfortunately, this method does not necessarily produce an optimal cluster [29], but empirically it produces good clusters in most cases.

The concept of min-multi-cut is an extension of the max-flow min-cut concept in capacitated networks and, essentially, seeks a partition of nodes into classes having minimum summary similarities between classes or, equivalently, maximum summary similarities within classes. When similarities are non-negative, this criterion may often lead to a highly unbalanced partition with one huge class and a number of singleton classes. This can be somewhat alleviated by requiring certain pairs of entities to be in the same clusters and other pairs in different clusters. This line of research has led to using the *normalized cut*, proposed in [39], as a meaningful clustering criterion. The normalized cut criterion assumes that the set I should be split into two parts, S and $\bar{S}$, so that the normalized cut

$$nc(S) = a(S, \bar{S})/a(S, I) + a(S, \bar{S})/a(\bar{S}, I)$$

is minimized. Here $a(S, T)$ denotes the summary similarity between subsets S and T. The criterion $nc(S)$ can be expressed as a Raleigh quotient for a generalized eigenvalue problem [39], so the spectral clustering approach may be applied to minimizing the normalized cut.

It is probably worth mentioning that this criterion only applies to flow data. Flow data can be standardized using *Quetelet* coefficients [30]. If $A = (a_{ij})$ is a flow data matrix over I, then its

Quetelet transformation is defined by

$$q_{ij} = \frac{a_{ij}a_{++}}{a_{i+}a_{+j}} - 1,$$

where a_{i+} and a_{+j} are the sums of the flows a_{ij} over row i and column j, respectively, and a_{++} is the total of all the flows.

The aggregate Quetelet coefficient q_{ST} is defined similarly in terms of the aggregate flows from S and to T. Flow data, especially co-occurrence frequency data, have been successfully handled using the Correspondence Analysis approach [6, 35], a version of Principal Component Analysis that approximates the underlying Quetelet coefficients. It is easy to see that the normalized cut $nc(S)$ is the aggregate Quetelet coefficient $q_{S\bar{S}}$ plus a constant, which implies that minimising $nc(S)$ is equivalent to minimising $q_{S\bar{S}}$.

In the context of this paper, it is important to note that the user typically finds it meaningful, in the framework of domain knowledge, to define a similarity threshold α, such that entities i and j should be aggregated if $a_{ij} > \alpha$ but not if $a_{ij} < \alpha$. When this is the case, the data should be pre-processed to take the threshold into account.

There are two different ways of implementing this idea: (1) by zeroing all similarities a_{ij} that are less than α, or (2) by shifting the zero similarity to α by subtracting α from each similarity a_{ij}.

The former is popular, for example, in image analysis because it makes the similarity data sharper and sparser. However, we favour the latter as better fitting in with the additive structure recovery models presented later. In fact, the similarity shift originated from these models (see, for example, [25, 26]).

2.2 The Additive Structuring Model and Iterative Extraction

To represent a set of structures assumed to underly the similarity matrix A, we use the terminology of binary relations since these are naturally represented by an “ideal” similarity matrix. A binary relation on the set I can be defined by a (0,1) matrix $R = (r_{ij})$ such that $r_{ij} = 1$ if i and j are related and $r_{ij} = 0$ otherwise. Partitions, rankings and subsets can be represented by equivalence, ordering and square relations, respectively. A quantitative expression of the intensity of a relation can be modelled by a real value λ. So a relation of intensity λ is represented by the product λR.

Given a set of binary relations $\mathcal{R}$ defined by a general property (for example, equivalence or order relations), an additive structuring model for a given $N \times N$ similarity matrix $A = (a_{ij})$ is defined by the equations

$$a_{ij} = \sum_{k=0}^{K} \lambda_k r_{ij}^k + e_{ij}, \text{ for } i, j \in I, \tag{1}$$

where $R^k = (r_{ij}^k) \in \mathcal{R}$ and λ_k is the intensity of R^k; the number of relations $K+1$ in (1) is typically assumed to be much smaller than $|I|$, the cardinality of I. The goal is to minimise the residuals e_{ij} with respect to the unknown relations R^k and intensities λ_k. In some problems, the intensities λ_k may be given, based on substantive or model considerations.

In certain cases, we may require one of the relations R^k to be the universal relation, for which $r_{ij}^k = 1$ for all $i, j \in I$. The corresponding intensity λ_k then plays the role of an intercept in the model (1), similar to that in linear regression. Conventionally, we relabel the universal relation as R^0 and denote its matrix by $\mathbf{1}$. The intercept value λ_0 may be interpreted as a similarity shift, with the shifted similarity matrix $A' = (a'_{ij})$ defined by $a'_{ij} = a_{ij} - \lambda_0$. Equation (1) for the shifted model has a'_{ij} on the left and the sum on the right starting from $k = 1$.

To minimise the residuals in (1), the least-squares criterion can be applied. Moreover, we can employ the greedy heuristic of extracting the relations R^k one by one in order to reduce the amount of computation. This may be particularly useful if the relations R^k contribute very unequally to the data as, for example, when the λ_k vary significantly. At step k, $k = 0, 1, 2, ..., K$, we find R^k using an algorithm for minimising

$$L^2(R) = \sum_{i,j \in I} (a^k_{ij} - \lambda r_{ij})^2 \tag{2}$$

over $R \in \mathcal{R}$ and λ (unless pre-specified). Given R, the optimal value of λ is equal to the average similarity a^k_{ij} over all related pairs (i, j), i.e. those for which $r_{ij} = 1$. The complexity of this minimization problem depends on the type of relations in $\mathcal{R}$. Therefore, in some cases, we only find a local minimum of (2). The similarity matrix $A^k = (a^k_{ij})$ is updated after each step by subtracting $\lambda_k R^k$ from it. At the start, $A^0 = A$ and, at the end, $A^{K+1} = (e_{ij})$, the matrix of residuals.

This method, which will be referred to as ITEX (ITerative EXtraction), was first proposed in [26] as a method for "categorical factor analysis", and was called SEFIT in [28].

When the λ_k are not pre-specified, then, at each step, the residual similarity matrix is orthogonal to the relation extracted. This implies the following Pythagorean decomposition [28, 29]:

$$\sum_{i,j \in I} a^2_{ij} = \sum_{k=0}^{K} \lambda^2_k \sum_{i,j \in I} r^k_{ij} + \sum_{i,j \in I} e^2_{ij} \tag{3}$$

This equation additively decomposes the data scatter into the contributions of the extracted relations R^k ("explained" by the model) and the minimised residual square error (the "unexplained' part).

The decomposition (3) makes it possible to prove that the residual part converges to zero under relatively mild and easily checked assumptions on the solutions found at each iteration [28, 29].

Obviously, our convention implies that, when ITEX is applied to the shifted model, the universal relation R^0 must be extracted first. In this case, the optimal value of λ_0 will be equal to $\bar{a}$, the average of the similarities in A.

2.3 Additive Clustering Model

The additive clustering model in [38] is the special case of the shifted version of the model (1) that emerges when $\mathcal{R}$ consists of square relations, each corresponding to a subset $S \subseteq I$. Specifically, let $s = (s_i)$ be the characteristic vector of S. Then the square relation 'i and j belong to S' can be represented by $r = (r_{ij}) = (s_i s_j)$. The universal relation $R^0 = \mathbf{1}$, used in the shifted model, is the square relation corresponding to the universal cluster I.

When we assume that the similarities in A are generated by a set of 'additive clusters' $S^k \subseteq I$, $k = 0, 1, ..., K$, in such a way that each a_{ij} approximates the sum of the intensities of those clusters that contain both i and j, the shifted version of (1) becomes:

$$a_{ij} = \sum_{k=1}^{K} \lambda_k s^k_i s^k_j + \lambda_0 + e_{ij}, \tag{4}$$

where $s^k = (s_i^k)$ are the membership vectors of the unknown clusters S^k, $k = 1, 2, ..., K$, and e_{ij} are the residuals to be minimised. In this model, introduced in [38], the intensities λ_k, $k = 1, 2, ..., K$, and the shift λ_0 also have to be optimally determined. In the more general formulation of 'categorical factor analysis' [26, 27], these values may be user specified.

We note that the role of the intercept λ_0 in (4) is three-fold:

1. it is an intercept of the bilinear model, similar to that in linear regression;

2. it is the intensity of the universal cluster I;

3. it is a 'soft' similarity threshold in the sense that the shifted similarity matrix a'_{ij} is used to determine the clusters S^k, $k = 1, 2, ..., K$. This role is of special interest when λ_0 is user specified.

When the one-by-one ITEX strategy is applied to fitting (4) with none of the λs pre-specified, the data scatter decomposition (3) holds for the optimal values of λ_k. In this case, λ_k is equal to $\bar{a}_k$, the average of the residual similarities a_{ij}^k for $i, j \in S^k$. Substituting $s_i^k s_j^k$ for r_{ij}^k and $\bar{a}_k$ for λ_k, (3) can be written in the form:

$$(A, A) = \sum_{k=0}^{K} [s^{kT} A^k s^k / s^{kT} s^k]^2 + (E, E) \tag{5}$$

The inner products (A, A) and (E, E) denote the sums of the squares of the elements of the matrices, considering A and E as vectors; these are conventionally expressed as the traces (sums of diagonal elements) of the products $A^T A$ and $E^T E$, respectively.

3 Approximate Partitioning

In this section, we restrict the additive clustering model to nonoverlapping clusters.

If the clusters S^k, $k = 1, ..., K$, are mutually disjoint (so the membership vectors s^k are mutually orthogonal), the optimal intensity λ_k depends only on the elements a'_{ij}, $i, j \in S^k$, of the shifted matrix $A' = A - \lambda_0 \mathbf{1}$ and not on the residual matrix A^k. The following decomposition of A' corresponding to (5) then holds and is independent of the the order of the clusters.

$$(A', A') = \sum_{k=1}^{K} [s^{kT} A' s^k / s^{kT} s^k]^2 + (E, E). \tag{6}$$

Since $A' = A - \lambda_0 \mathbf{1}$, it follows that

$$(A, A) = 2\lambda_0 (\bar{a} - \lambda_0/2)(\mathbf{1}, \mathbf{1}) + \sum_{k=1}^{K} [s^{kT} A' s^k / s^{kT} s^k]^2 + (E, E) \tag{7}$$

When λ_0 is not pre-specified and must be found according to the least-squares criterion, its optimal value, found by differentiating (7) with respect to λ_0, is:

$$\lambda_0 = \frac{\sum_{i,j \in I} a_{ij}(1 - s_{ij})}{\sum_{i,j \in I}(1 - s_{ij})}, \tag{8}$$

where $s_{ij} = \sum_{k=1}^{K} s_i^k s_j^k$ (so $s_{ij} = 1$ if both i and j belong to S^k for some $k = 1, 2, ..., K$ and $s_{ij} = 0$ otherwise).

Thus, the optimal λ_0 is the average of the similarities a_{ij} for i and j belonging to different clusters.

Equation (6) is analogous to the representation of the trace of $A'^T A'$ as the sum of the squares of the eigenvalues of A' because the terms are squares of the Raleigh quotients

$$g(s^k) = s^{kT} A' s^k / s^{kT} s^k. \tag{9}$$

which are attained at zero/one rather than arbitrary vectors s^k.

According to (6), an optimal partition with weights λ_k adjusted according to the least-squares criterion must maximise the sum of the cluster contributions $g(s^k)^2$, that is,

$$\sum_{k=1}^{K} g^2(s^k) = \sum_{k=1}^{K} (\sum_{i,j \in S^k} a'_{ij} / N_k)^2 \tag{10}$$

where $N_k = |S_k|$, the cardinality of S_k.

An "unsquared" version of this criterion comes from applying the data recovery approach to an entity-to-feature data matrix [30], which leads to

$$\sum_{k=1}^{K} g(S^k) = \sum_{k=1}^{K} \sum_{i,j \in S^k} a_{ij} / N_k \tag{11}$$

as the contribution of the clusters to the entity-to-feature data scatter. The similarity a_{ij} is defined, in this approach, as the inner product of the feature vectors corresponding to entities i and j. In matrix terms, if Y is an entity-to-feature data matrix then A is defined as $A = YY^T$. The difference between criteria (10) and (11) is somewhat similar to that between the spectral decomposition of $A = YY^T$ and singular-value decomposition of Y.

In contrast to (11), criterion (10) has never been analysed, either theoretically or experimentally.

To illustrate the difference between preset and optimal values of the shift λ_0 when model (4) is used for approximate partitioning, let us consider the similarity data between eight entities in Table 1.

For $\lambda_0 = 2$, the only positive values of $a'_{ij} = a_{ij} - \lambda_0$ are within clusters 1-2-3, 4-5, and 6-7-8 plus similarities between entity 4 and both 6 and 7. These positive extra-cluster similarities lead

Table 1: Illustrative similarities between eight entities; self-similarity is not defined.

Entity	1	2	3	4	5	6	7	8
1	-	4.33	5.60	-0.20	-0.16	-0.21	-0.49	0.17
2	4.33	-	4.93	0.79	0.06	1.22	-0.10	-0.45
3	5.60	4.93	-	0.21	0.79	-1.20	-0.15	0.80
4	-0.20	0.79	0.21	-	4.62	3.29	2.80	0.32
5	-0.16	0.06	0.79	4.62	-	-1.00	0.25	-0.08
6	-0.21	1.22	-1.20	3.29	-1.00	-	5.96	4.38
7	-0.49	-0.10	-0.15	2.80	0.25	5.96	-	5.23
8	0.17	-0.45	0.80	0.32	-0.08	4.38	5.23	-

to differences in the clustering if λ_0 is changed. At the average similarity shift $\lambda_0 = \bar{a} = 1.49$, these three clusters with respective intensities 3.46, 3.13 and 3.70 form the optimal partition. This partition contributes 37.1% to the original data scatter. For the global optimum partition, the $\lambda_0 = 0.49$ and entity 4 joins the cluster 6-7-8. The optimal partition then consists of clusters 1-2-3 (with intensity 4.47), 4-6-7-8 (with intensity 3.17), and singleton 5 (since self-similarity is not defined, the intensity has no meaning). This contributes 65.6% of the data scatter. The rather large difference between the two contributions to the data scatter is mainly due to the difference between the contributions due to λ_0, i.e., the first term on the right-hand side of (7).

4 One Cluster Clustering

In this section, we turn to the problem of applying ITEX to the additive clustering. This involves extracting a single cluster from, possibly residual, similarity data presented in the form of a symmetric matrix A, assuming that any requred shift λ_0 has already been made. For the sake of simplicity, in this section, we assume that the diagonal entries a_{ii} are all zero.

4.1 Pre-specified Intensity

We first consider the case in which the intensity λ of the cluster to be found is pre-specified. Remembering that $s_i^2 = s_i$ for any 0/1 variable s_i, criterion (2) can be expressed as

$$L^2(S) = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2 = \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j \tag{12}$$

Since $\sum_{i,j} a_{ij}^2$ is constant, for $\lambda > 0$, minimizing (12) is equivalent to maximizing the summary within-cluster similarity after subtracting the threshold value $\pi = \lambda/2$, i.e.,

$$f(S, \pi) = \sum_{i,j \in I} (a_{ij} - \pi) s_i s_j = \sum_{i,j \in S} (a_{ij} - \pi). \tag{13}$$

This criterion implies that, for an entity i to be added to or removed from the S under consideration, the difference between the value of (13) for the resulting set and its value for S, $f(S \pm i, \pi) - f(S, \pi)$, is equal to $\pm 2 f(i, S, \pi)$ where

$$f(i, S, \pi) = \sum_{j \in S} (a_{ij} - \pi) = \sum_{j \in S} a_{ij} - \pi |S|$$

.

This gives rise to a local search algorithm for maximizing (13): start with $S = \{i^*, j^*\}$ such that $a_{i^* j^*}$ is maximum element in S, provided that $a_{i^* j^*} > \pi$. An element $i \notin S$ may be added to S if $f(i, S, \pi) > 0$; similarly, an element $i \in S$ may be removed from S if $f(i, S, \pi) < 0$. The greedy procedure ADDI [27] iteratively finds an $i \notin S$ maximising $+f(i, S, \pi)$ and an $i \in S$ maximizing $-f(i, S, \pi)$, and takes the i giving the larger value. The iterations stop when this larger value is negative. The resulting S is returned along with its contribution to the data scatter, $4\pi \sum_{i \in S} f(i, S, \pi)$. To reduce the dependence on the initial S, a version of ADDI can be utilised by starting from the singleton $S = \{i\}$, for each $i \in I$, and finally selecting the S that contributes most to the data scatter, i.e. minimises the square error $L^2(S)$ (12).

The algorithm CAST [5], popular in bioinformatics, is a version of the ADDI algorithm, in which $f(i,S,\pi)$ is reformulated as $\sum_{j\in S} a_{ij} - \pi|S|$ and $\sum_{j\in S} a_{ij}$ is referred to as the affinity of i to S.

Another property of the criterion is that $f(i,S,\pi) > 0$ if and only if the average similarity between a given $i \in I$ and the elements of S is greater than π, which means that the final cluster S produced by ADDI/CAST is rather tight: the average similarities between $i \in I$ and S is at least π if $i \in S$ and no greater than π if $i \notin S$ [27].

Intuitively, changing the threshold π should lead to corresponding changes in the optimal S: the greater π is, the smaller S will be [27].

4.2 Optimal Intensity

When λ in (12) is not fixed but chosen to further minimise the criterion, it is easy to prove that:

$$L^2(S) = (A,A) - [s^TAs/s^Ts]^2, \tag{14}$$

in line with the decomposition (6), with $K = 1$ ans $L^2(S) = (E,E)$. The proof is based on the fact that the optimal λ is the average similarity $a(S)$ within S, i.e.,

$$\lambda = a(S) = s^TAs/[s^Ts]^2, \tag{15}$$

since $s^Ts = |S|$.

The decomposition (14) implies that the optimal cluster S must maximise the criterion

$$g^2(S) = [s^TAs/s^Ts]^2 = a^2(S)|S|^2 \tag{16}$$

According to (16), the maximum of $g^2(S)$ may correspond to either positive or negative value of $a(S)$. The latter case may emerge when the similarity shift λ_0 is large and corresponds to S being the so-called *anti-cluster* [29]. In this paper, we do not consider this case, but focus on maximising (16) only for positive $a(S)$. This is equivalent to maximising the Raleigh quotient,

$$g(S) = s^TAs/s^Ts = a(S)|S| \tag{17}$$

To maximise $g(S)$, one may utilise the ADDI-S algorithm [27], which is the same as the algorithm ADDI/CAST, described above, except that the threshold π is recalculated after each step as $\pi = a(S)/2$, corresponding to the optimal λ in (**??**).

A property of the resulting cluster S, similar to that for the constant threshold case, holds: the average similarity between i and S is at least half the within-cluster average similarity $a(S)/2$ if $i \in S$, and at most $a(S)/2$ if $i \notin S$.

To obtain a set of (not necessarily disjoint) clusters within the framework of the additive clustering model, one may use ITEX by repeatedly extracting a cluster S using ADDI-S and then replacing A by the residual matrix $A - a(S)ss^T$.

We can apply this method to the partitioning problem, by repeatedly using ADDI-S to find a cluster S and then removing from consideration all the entities in S. The process stops when the similarity matrix on the remaining entities has no positive entries. The result is a set of non-overlapping clusters S_k, $k = 1, ..., K$, each assigned with its intensity $a(S_k)$, and also the remaining unclustered entities in I.

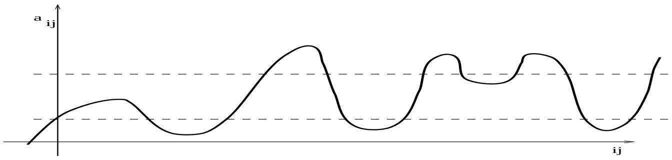

Figure 1: A pattern of clustering depending on the subtracted similarity shift λ_0.

ADDI-S utilises no ad hoc parameters, so the number of clusters is determined by the process of clustering itself. However, changing the similarity shift λ_0 may affect the clustering results, which can be of advantage in contrasting within- and between- cluster similarities. Figure 1 demonstrates the effect of changing a positive similarity a_{ij} to $a'_{ij} = a_{ij} - \lambda_0$ for $\lambda_0 > 0$; small similarities $a_{ij} < \lambda_0$ are transformed into negative similarities a'_{ij}.

5 Domain knowledge in determining similarity shift

5.1 Aggregation of proteins in protein families

In this section, we apply ADDI-S above to the aggregation of proteins in the so-called homologous protein families (HPFs) combining proteins of the same function and considerable sequence similarity from different genomes. The concept of homologous protein family, HPF, can be considered an empirical expression of the concept of gene as a unit of heredity in the intergenomic evolutionary studies. As such the HPF is an important instrument in the analysis of the evolutionary history of the function that it bears. The evolutionary history of a set of genomes under consideration is depicted as an evolutionary tree, or phylogeny, whose leaves are labelled by genomes of the set, and internal nodes correspond to hypothetical ancestors. An HPF can be mapped to the tree in the following natural way. First, the HPF is assigned to the leaves corresponding to genomes containing its members. Then the pattern of belongingness can be iteratively extended to all the ancestor nodes in a most parsimonious or most likely way. For example, if each child of a node bears a protein from the HPF then the node itself should bear the same gene itself, because it is highly unlikely that the same gene emerged in the children independently. Exact formulations of the algorithms can be found in [**?**, 33]. Having annotated the evolutionary tree nodes with hypothetical evolutionary histories of various HPFs, realistic conclusions of possible histories and mechanisms of evolution of biomolecular function may be drawn for the purposes of both theoretical research and medical practice.

Assignment of proteins to HPFs is often determined with a large manual component because

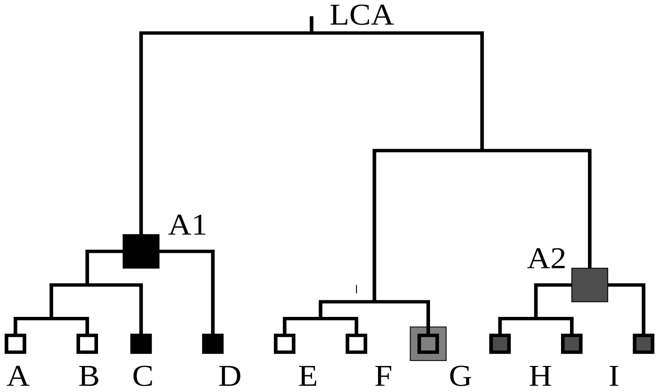

Figure 2: An evolutionary tree over genomes A to L with three protein families shown in black, grey and patterned tones present in genomes C, D (one family), G (another family) and H, I, L (third family). Their reconstructed ancestors are shown with greater boxes in nodes A1, G and A2. If, however, these three families would be recognised as parts of the same family, then their reconstructed ancestor ought be pplaced at the root, node LCA.

the degree of similarity between proteins within an alignment of protein sequences is not always sufficient to automatically identify the families. Significant protein similarity over the full length of the protein is often insufficient to group proteins into families, especially for rapidly evolving organisms such as bacteria and viruses.

This is why a two-stage strategy for identifying HPFs has been considered in [33]. According to this strategy, HPFs are created, first, as groups of proteins that have a common motif, a contingent fragment of protein sequence that is similar in all HPFs members. This motif represents a relatively well conserved segment of the genetic material that can be associated with a protein function. Obviously such, motif defined, HPFs may be overly fragmented since many proteins are multifunctional and thus could bear resemblances to different proteins at different fragments.

The fragmented HPFs may lead then to wrong reconstructions of functional histories such as presented in Figure 2: Reconstructed ancestral nodes of the first emergence of each of the three HPFs labeled by differently patterned boxes are shown with greater boxes in nodes A1, G and A2. These histories, however, may be due to an erroneous aggregation: The three HPFs may, in fact, bear similar proteins and thus should be combined into a single aggregate HPF whose origin then ought to be in the ultimate ancestor corresponding to the tree root.

Therefore, the next stage of the strategy is to cluster the first stage motif-based HPFs into larger aggregations based on whole sequence similarity. Since entities at this stage are not single proteins but protein families, we need to score similarities between families rather than single proteins. This

Table 2: List of 30 herpesvirus genomes under consideration.

#	VIDA Ref.	Genome	GenBank Ref.
		Alphaherpesvirinae	
01	CeHV-1	Cercopithecine hv 1	NC_004812
02	HHV-1	Human hv 1/simplex 1	NC_001806
03	HHV-2	Human hv 2/simplex 2	NC_001798
04	EHV-4	Equid hv 4	NC_001844
05	EHV-1	Equid hv 1	NC_001491
06	BoHV-1	Bovine hv 1	NC_001847
07	BoHV-5	Bovine hv 5	NC_005261
08	CeHV-7	Cercopithecine hV 7	NC_002686
09	HHV-3	Human hv 3/varicella-zoster	NC_001348
10	MeHV-1	Meleagrid hv 1	NC_002641
11	GaHV-2	Gallid hv 2/Marek's disease	NC_002229
12	GaHV-3	Gallid hv 3	NC_002577
13	PsHV-1	Psittacid hv 1	NC_005264
		Betaherpesvirinae	
14	HHV-6	Human hv 6	NC_001664
15	HHV-7	Human hv 7	NC_001716
16	HHV-5	Human hv 5/cytomegalovirus	NC_006273
17	ChCMV	Chimpanzee cytomegalovirus	NC_003521
18	MuHV-2	Murid hv 2/rat cytomegalovirus	NC_002512
19	TuHV	Tupaiid hv	NC_002794
		Gammaherpesvirinae	
20	HVS-2	Saimiriine hv 2	NC_001350
21	AtHV-3	Ateline hv 3	NC_001987
22	EHV-2	Equid hv 2	NC_001650
23	BoHV-4	Bovine hv 4	NC_002665
24	MuHV-4	Murid hv 4/murine hv 68	NC_001826
25	RRV-17577	Macaca mulatta rhadinovirus	NC_003401
26	HHV-8	Human hv 8/Kaposi's sarcoma	NC_003409
27	AIHV-1	Alcelaphine hv 1	NC_002531
28	CeHV-15	Cercopithecine hv 15	NC_006146
29	HHV-4	Human hv 4/Epstein-Barr	NC_001345
30	CaHV-3	Callitrichine hv 3	NC_004367

issue will be covered in the next section after the data we deal with are described in greater detail.

5.2 Neighbourhood similarity between HPFs

The data for this analysis come from studies of herpesvirus - a pathogene highly affecting both animals and humans. A set of 30 complete herpesvirus genomes covering the so-called α, β and γ herpesvirus superfamilies that differ by the tissue in which the virus resides, have been extracted from the herpesvirus database VIDA, release 3 [2] (see Table 2); and an evolutionary tree has been built over the genomes for the conserved DNA polymerase gene using the PHYLIP package [10] (see Figure 3). This tree agrees well, within the uncertainty limits, with the previously published instances of herpesvirus phylogenies and, moreover, all of the results reported here hold for the other topology as well (see details in [33]).

A set of 740 homologous protein families (HPFs) represented in these 30 genomes have been extracted from the VIDA database [2]. Each VIDA HPF is defined by a conserved fragment in proteins constituting the HPF; these were computed using the algorithm XDOM [14, 2]. In this way, each HPF is proposed to represent a basal functional grouping, whose origin can be mapped to the evolutionary tree under the assumption that the function is inherited according to the tree topology. As discussed above, such motif based protein family assignment can suffer from fragmentation of protein families and from the non-assignment of proteins to a family due to lack of pair-wise similarity.

To further aggregate the VIDA HPFs, we have to develop a system for scoring similarity between them. A most straightforward idea would be to score first similarities between proteins belonging to different HPFs with follow-up averaging them. Another approach would dwell on the property of VIDA HPFs that they may overlap, sometimes significantly, because different HPFs can be defined by different fragments of the same sequences. According to this approach, similarity between HPFs should reflect set-theoretic similarity between them as 'bags' of proteins. We accept an intermediate approach: we measure set-theoretic similarity but between HPF neighbourhoods defined by using a whole-sequence alignment tool PSI-BLAST [3] rather than between HPFs themselves. Given an HPF, this approach works as follows. First, for every protein from the HPF a list of similar proteins is created using PSI-BLAST. Second, these lists are combined according to a majority rule. The resulting set of proteins constitutes the HPF's neighbourhood. Note that it consists of proteins, not of HPFs. Third step is computing matrix of a set-similarity index values by applying it to the HPF neighbourhoods, for every pair of HPFs.

There are several features of this approach that made us to use it.

One of them is the issue of relying on the accuracy of alignment of protein sequences in scoring similarity between them. Alignment tools, including PSI-BLAST [3] which we utilise, rely on a number of user-defined parameter values, that are specified for default options based on experiments. These parameter values work quite well when sequences are similar indeed. However, there is a great uncertainty in their values at proteins that are not homologous, which is a typical case when proteins are from different HPFs. Therefore, by limiting action of PSI-BLAST to aligning only similar sequences, we avoid the uncertainty and arbitrariness of similarity estimates at distant protein sequences.

Another feature relates to the idea that neighbourhoods may give more reliable information on functional aspects of proteins. There are many examples of proteins, especially virus encoded proteins, whose pair-wise similarity is low, but which are known to be functionally related and which have many common homologues. For example the glycoprotein H like protein of murine herpesvirus 4 (gi: 1246777) and the UL22 protein of Bovine herpesvirus 1 (gi: 1491636) have minimal sequence identity (15%, identified on the second PSI-BLAST iteration), and have been initially assigned to separate HPFs within the VIDA database, namely HPFs 12 and 42 [2]. However, their sets of homologous protein neighbours (with 20% or greater sequence identity), contain 25 and 20 sequences, respectively, and have 14 common proteins, making the overlap between the homologous protein lists quite significant: the average relative overlap is 63% (14/25=56% in one of the sets and 14/20=70% in the other). To alleviate the issue, PSI-BLAST runs are conventionally reiterated for accruing distantly related proteins into families. This, however, may import irrelevant proteins or proteins that are not within the organism group under investigation. An HPF obtained in this way requires manual curation, but the overlap between the neighbourhood lists suggests that our computational strategy may be useful in overcoming the issue.

One more feature of our approach relates to the stage of combining individual neighbourhoods of protein sequences into an HPF neighbourhood. The set of HPF member proteins covers an evolutionary time span during which they have developed from a hypothetical ancestor. It is assumed that the greater the difference between sequences, the greater the time at which they diverged. This phenomenon should be reflected in the composition of the neighbourhood lists. That means that we can regulate the time span taken into account by choosing different majority thresholds when combining the neighbourhoods. This may provide an alternative to the way PSI-BLAST seeks for more distant relatives by relying on statistical frequency profiles [3].

The idea of employing neigbourhoods to measure similarities between entities is not original. It is used in information retrieval starting probably from work [41] and generalising to what is referred to as the ”semantic similarity” in the natural language processing [20]. It has been employed in bioinformatics as well, mostly in the analysis of gene expression data (see, for example, [40]). In the perspective of clustering of complex data, this approach allows for a unified framework of between-subset similarities rather than individual frameworks of domain-specific similarity measures.

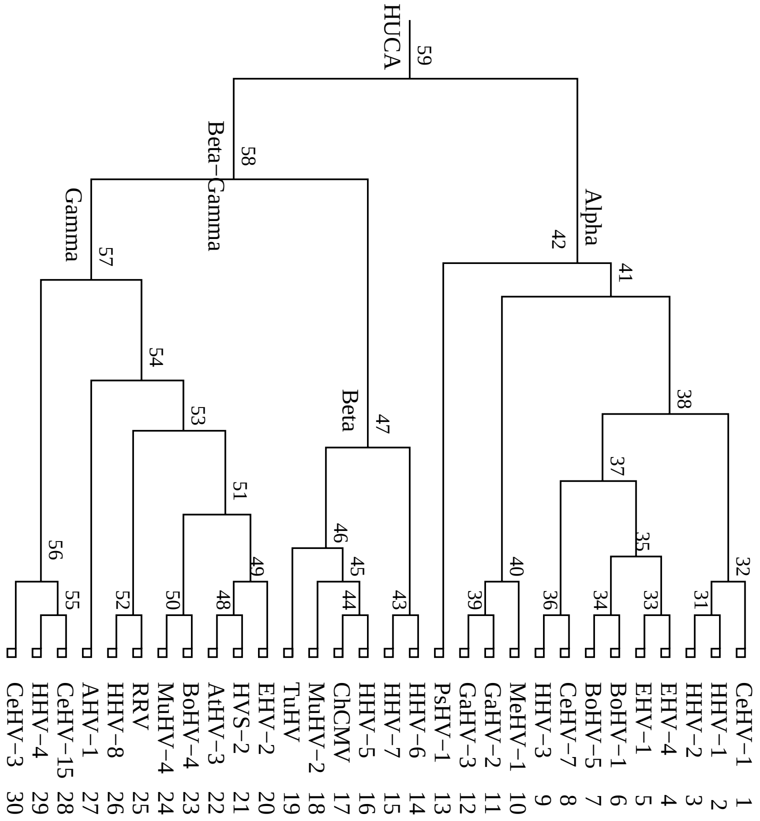

Figure 3: Herpesvirus evolutionary tree. The root corresponds to the herpesvirus ultimate common ancestor (HUCA); its child on the right to the ancestor of α superfamily, and the child on the left, to the common ancestor of β and γ superfamilies.

Let us describe in more detail how we take neighbourhoods of HPF members and combine them all into a majority set. .

Given a query protein sequence p, we utilise the PSI-BLAST program [3] to sort all protein sequences under consideration (we use those in the NCBI Entrez web site [36]) by their similarity to the query sequence. An initial fragment of this sorted list, defined by a contrasting cut-off similarity value, is identified. The list of all those proteins from this fragment that also belong in our collection of herpesvirus genome protein sequences makes the homology neighbourhood (HN) of p, denoted by $l(p)$.

Given a protein family h consisting of m proteins p_1, p_2,...,p_m, with herpesvirus constrained HN sets $l(p_1)$, $l(p_2)$, ...,$l(p_m)$ assigned to each of them, we aggregate these sets by using the majority rule. Let us assign a membership score $s(p)$ to each sequence; $s(p)$ being defined as the proportion of the HN sets $l(p_1)$,..,$l(p_m)$ to which p belongs; this is 1 if p belongs to all m of the sets.

Given $t > 0$, the t-majority list $M_t(h)$ is defined as the set of those p for which $s(p) \geq t$. For $t = 1/2$, $M_{1/2}(h)$ is the so-called *simple* majority list. As t decreases, the size of $M_t(h)$ can only increase, so that for $t \leq 1/m$ the t-majority list $M_t(h)$ is the set-theoretic union of sets $l(p_i)$ for all $p_i \in h$.

To measure similarity between two HPFs represented by their HN sets of protein sequences, $L1$ and $L2$, one should rely on the quantities involved: the size of the overlap between $L1$ and $L2$, denoted by n, the number of elements in $L1$ denoted by $n1$, and the number of elements in $L2$ denoted by $n2$. To take into account the relative size of the overlap, we use the average proportion of the overlap, $mbc = \frac{1}{2}(\frac{n}{n1} + \frac{n}{n2})$, known as the Maryland Bridge coefficient [32]. This index is co-monotone with the popular Jaccard coefficient $J = \frac{n}{n1+n2-n}$, but does not suffer from the intrinsic flaw of the Jaccard coefficient, which systematically underestimates the similarity [32].

To determine an appropriate value for majority threshold t, we accept the view that the proteins in an HPF have developed over a period of time; thus, the longer the time period spanned by the t-majority list proteins, the smaller should be the value chosen for t.

Specifically, in the case under consideration, the majority threshold has been set at the level of 20%, i.e. $t = 1/5$, based on analysis of clusterings of HPFs produced at neighbourhoods defined at different thresholds. Specifically,:

1. The median mbc similarity value between clusterings corresponding to "neighbouring" majority thresholds 1/6 and 1/5 is 0.98; 1/5 and 1/4, 1.00; 1/4 and 1/3, 0/99; 1/3 and 1/2, 0.96. The average mbc similarity value varies similarly, taking its maximum at the majority thresholds 1/5 and 1/4. The similarity between clusterings at non "neighbouring" thresholds slightly decreases, though overall clusterings produced at different similarity shift levels differ little.. The sets of unclustered entities behave similarly.

2. The clustering found over 1/5=20%-majority lists is "central" among other clusterings; it is more similar to the other clusterings than at any other of the considered majority thresholds.

3. The clustering found over 20%-majority lists is more similar than the others to clusterings produced with the homology lists obtained with the iterated PSI-BLAST search [3], starting from a random protein in an HPF. Repeated PSI-BLAST search, over an averaged profile of the first search results, allows one to catch more distant homologues to the query sequence [3]. The median similarity between the clustering at 20%-majority lists and the clustering found at HPF neighbourhood lists of the first iteration is 0.91; lists of the second iteration, 0.82; and lists of the third iteration, 0.50. (We take these results to support our view that repeated iterations of BLAST may need manual curation.)

The similarity between two clusterings as sets of clusters is defined by the index mbc applied to the situation when entities are clusters and two clusters are considered the same if they are either equal or one is part of the other differing by not more than two elements.

We therefore used our method of generating and assessing similarity between lists of homologous proteins to check the validity of the starting HPFs and merge HPFs those with similar neighbourhoods into aggregate protein families (APFs). This however is not quite straightforward exercise because results highly depend on the similarity shift value. In further sections we describe how the domain knowledge may help in choosing right shift similarity value.

5.3 Utilising domain knowledge

At different similarity shifts we get different numbers of clusters of HPFs. Specifically, at the zero similarity shift, $b = 0$, there are 99 non-trivial clusters. The number of clusters rises to 107 at

$b = 0.10$ and then gradually falls down, from $b = 0.40$, when the similarity shift is risen further so that there are 29 non-singleton clusters at $b = 0.97$. Note that the latter number corresponds to the situation when HN sets of the clustered HPFs are practically the same: to overlap at the level of 97% or higher, majority lists of less than 30 elements in an HPF (this is true for almost all HPFs) must be equal to each other.

The optimal similarity value is almost 0 (0.017, to be exact), which implies that to choose a right similarity shift, one should involve external knowledge of the domain. Such external knowledge, independent of sequence similarity estimates, is knowledge of functional activities of the proteins under consideration. Each HPF is supposed to have a function (for examples of function see Table 3 below), though unfortunately functions of most part of proteins available are unknown. If functions are known, as it happens to at least some HPFs, then we can play on those HPFs that are synonymous. Two proteins are considered synonymous if they are consistently named between the herpesvirus genomes and/or they share the same known function. Such proteins should therefore belong in the same aggregate protein family. Two proteins are considered non-synomymous if they have different functions and thus should belong to different protein families. Thus, what needs to be done is to identify HPFs with known function and form pairs of those with clearly synonymous function and those whose functions clearly differ (this may not be necessarily a straightforward exercise because authors of different submissions of data to databases tend to use different terminology).

Sequence similarity values should be high between synonymous sequences and should be low between non-synonymous sequences. The similarity shift value should be taken between these groups so that similarities between not synonymous HPFs get negative after the shift while those between synonymous HPFs remain positive.

To implement this idea, we analysed 287 available pairs of HPFs with known function and positive similarity value. Among them, no non-synonymous pair has a greater *mbc* similarity than 0.66, which should imply that the shift value $b = 0.67$ confers specificity for the production of APFs.

Unfortunately, the situation is less clear cut for synonymous proteins. Out of the 86 synonymous pairs available, there are 24 pairs (28%) that have their mutual similarity value less than 0.67. Thus

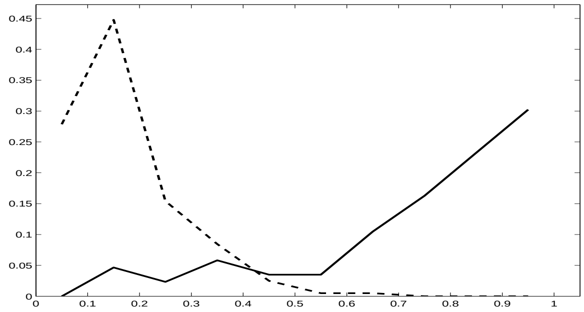

Figure 4: Empirical percentage frequency functions (y-values) for the sets of synonymous pairs (solid line) and non-synonymous pairs (dashed line). The x-values represent the *mbc* similarity.

at the similarity shift at 0.67, 28% of the synonymous pairs will not be identified as such suggesting that at this similarity shift the method would lack sensitivity. To choose a similarity shift that minimises the error in assigning negative and positive similarity values, one needs to compare the distribution of similarity values in the set of synonymous pairs with that in the set of non-synonymous pairs. As Figure 4 shows, the graphs intersect when the similarity value *mbc* is 0.42. The number of synonymous pairs whose similarity falls into the wrong side of 0.42 (that is, less than 0.42) falls to 11, whereas the number of non-synonymous pairs whose similarity is higher than 0.42 increases to 7 (from 0 at 0.67), which leads to the minimum summary error rate of 16%, at $b = 0.42$.

Thus the external knowledge of synonymity/non-synonymity among pairs of HPFs suplies us with two candidates for the similarity shift values: (a) $b = 0.67$ to guarantee specificity in that non-synonymous HPFs are not be clustered together, and (b) $b = 0.42$ to ensure the minimum misclassification error rate.

The two similarity shift values indicated, i.e. $b = 0.67$ and $b = 0.42$, lead to somewhat different but rather compatible clusterings of the set of 740 HPFs under consideration. There are 80 APF clusters comprising original 180 HPFs and leaving 560 HPFs unclustered at $b = 0.67$. There are 102 APF clusters over original 249 HPFs, and 491 HPFs unclustered at $b = 0.42$.

The first 80 clusters extracted at similarity shift $b = 0.42$ correspond one-to-one to the 80 clusters obtained at $b = 0.67$. All 22 of the additional clusters extracted at $b = 0.42$ are doublets with *mbc* similarity values of 0.50 to 0.62 (implying that there is a gap in *mbc* similarity values between 0.42 and 0.50).

The aggregation found at $b = 0.67$ suggests $560 + 80 = 640$ APFs altogether whereas $b = 0.42$

leads to a smaller total, $491 + 102 = 593$. Which one is correct? Probably that better fitting into the domain knowledge.

5.4 Advancing domain knowledge

Luckily, we have a strong interpretation tool, the mapping of HPFs to the evolutionary tree resulting in their evolutionary histories. This tool supplies us with reconstructed HPF contents of all the genome ancestors according to the tree. Of these, currently most useful are reconstructions of the most ancient genomes, those of ancestors of superfamilies α, β and γ as well as the more universal common ancestors, HUCA and $\beta\gamma$. This is because the similarities and differences among herpesvirus species are somewhat better understood at this level.

The multitude of reconstructed histories may provide us with an additional criterion for choosing right aggregate HPFs. This additional criterion is consistency among the histories as well as domain knowledge.

The reconstructions of the five ancestors with APFs found at similarity shifts $b = 0.42$ and $b = 0.67$ are essentially the same.

The only exception is the common ancestor of the α superfamily, which gains three more APFs when b changes from $b = 0.67$ to $b = 0.42$. These are APF81 comprised of HPFs 9 and 504, both of glycoprotein C; APF82 comprised of HPF 38 and HPF 736, both of glycoprotein I; and APF84 comprised of HPF 47 and HPF 205, both of glycoprotein L. Unfortunately, at the current state of domain knowledge, we cannot interpret the phenomenon of simultaneously gaining three glycoprotein families in terms of the α herpesvirus activities alone.

We can, however, look at the mutual positions of genes bearing these proteins within the virus genomic circular structures.

We find that in all 13 genomes comprising α superfamily in our data, gene bearing glycoprotein E always immediately precedes that bearing glycoprotein I. This by itself may be considered a strong indication that there must be a mechanism in the superfamily involving both glycoproteins that has been developed already in the α ancestor. Moreover, it appears, glycoprotein E corresponds to an aggregate protein family comprised of HPF 26 and HPF 301 (at both levels of the similarity shift, 0.67 and 0.42) that has been mapped by our alogorithm to the ancestral α node [33]. This leads us to conclude that glycoprotein I must also belong to the α ancestor, thus implying that similarity shift = 0.42, at which glycoprotein I's aggregate family falls in α ancestor, better fits to the knowledge added than $b = 0.67$, at which glycoprotein I's HPFs are not aggregated and are mapped into more recent ancestors. Aa additional supporting evidence comes from glycoprotein D's aggregate family comprising HPF 4 and HPF 45 at both similarity shift values. It is also mapped into α ancestor. And, moreover, its gene immediately precedes the gene of glycoprotein I in almost all (eleven) genomes of the α superfamily. (In two genomes, CeHV-7 and HHV-3, the preceding gene is of protein kinase rather than of glycoprotein D, which itself may lead to some speculations of possible mechanisms underlying such a substitution.)

5.5 Final results

Therefore, we accept the value $b = 0.42$ and corresponding number of protein families, after aggregation, 593.

Now we can draw structural conclusions from the mapping of the aggregate families to the evolutionary tree; some of them are presented below.

The common ancestor of herpesviruses, HUCA, according to our reconstruction, should comprise 45 HPFs aggregated to 29 APFs, i.e. 29 protein families. These are well studied proteins with only three of the participating families, HPFs 17, 23 and 107, of no known function. Our HUCA is consistent with the work of others, D-HUCA[8, 9], but does not include all the protein families assigned by Davidson et al. This concurs with our view that our approach, relying only on sequence similarity alone, is conservative.

Typical relations between our mapping results and D-HUCA are illustrated in Table 3

In some cases, it is clear that the fragmented HPFs fail to aggregate at that level of moving from the α, β and γ-ancestor into HUCA because of almost zero sequence similarity between them. For example, all three ancestors, of each α-, β-, and γ families, have a glycoprotein L. However, the corresponding HPFs, 47, 50 and 296, have no significant sequence similarity and, thus, cannot be combined together, even in terms of the neighbourhood lists. Still, at the genome organisation level, illustrated on Figure 5, each of the glycoprotein L genes always exactly precedes the corresponding Uracil-DNA glycosylase gene, which is mapped into HUCA. This suggests these

Table 3: Comparison between a previously determined herpesvirus common ancestor D-HUCA's [8, 9] list of functions in the herpesvirus ancestor (two columns on the right) versus the results from the mapping of HPF/APFs (first four columns), with function descriptions taken from VIDA.

Mapping	A/HPF	Function	Description	HSV-1 Gene	D-HUCA
					Peripheral Enzymes
HUCA	8	Nucleotide repair/ metabolism	uracil-DNA glycosylase, HHV-1 UL2	UL2	Uracil-DNA glycose
HUCA	24	Nucleotide repair metabolism	RNA reductase large subunit, HSV-1 UL39	UL39	RNA reductase; large subunit
HUCA	33	Nucleotide repair metabolism	RNA reductase small subunit, HHV-1 UL40	UL40	RNA reductase small subunit
HUCA	APF 10			UL23	Thymidine Kinase
	2	*Nucleotide repair/ metabolism*	*thymidine kinase*		
	27	*”*	*thymidine kinase*		
HUCA	43	Nucleotide repair/ metabolism	dUTPase, HHV-8 ORF54	UL50	dUTPase
					Surface and Membrane
HUCA	20	Membrane glycoprotein	glycoprotein M, HHV-1 UL10	UL10	Glycoprotein M; complexed with glycoprotein N
HUCA	3	Membrane glycoprotein	glycoprotein B, HHV-1 UL27	UL27	Glycoprotein B
HUCA	APF 3			UL22	Glycoprotein H; complexed with glycoprotein L
	42	*Membrane/ glycoprotein*	*glycoprotein H, HHV-1 UL22*		
	12	*”*	*glycoprotein H, HHV-8 ORF22*		
	531	*”*	*glycoprotein H, HHV-8 ORF22*		
Node 32	267	Virion protein	envelope protein, HHV-1 UL49A	UL49A	Glycoprotein N; complexed with glycoprotein M
ALPHA	47	Membrane glycoprotein	glycoprotein L, HHV-1 UL1	UL1	Glycoprotein L; complexed with glycoprotein H
BETA	50	”	glycoprotein L, HHV-5 UL115		
GAMMA	114	”	glycoprotein L, HHV-8 ORF47		
GAMMA	296	”	glycoprotein L, MuHV-4 ORF47		

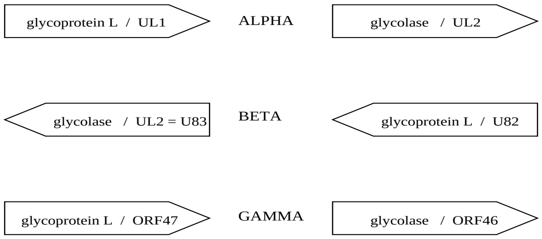

Figure 5: Positional homology between glycoprotein L sites in the herpesvirus superfamilies α, β and γ. The homology suggests that the glycoprotein L gene co-functions with the glycosylase gene and thus the former, like the latter, should be mapped to HUCA.

are common ancestral genes indeed; just they have undergone sequence change to a level where sequence similarity is no longer sufficient to assign homology. Putting the corresponding gene UL2 into D-HUCA has been based on experimental evidence that in the α-, β-, and γ families, glycoprotein L sequences in HPF 47, 50 and 296 functionally complex with glycoprotein H [9].

This is a clear example of a situation in which sequence similarity is not indicative of the homology so that association between the proteins can be seen only at a higher level of gene arrangement in genomes. We do not know any other example of such a situation in the published literature.

Concerning other four superfamily ancestors in our study, α, $\beta\gamma$, β and γ, we can claim that only the contents of the α superfamily is relatively well studied. Of its 33 gained HPFs (plus the inherited HUCA contents) only 9 are of unknown function.

This pattern is not repeated in the $\beta\gamma$ ancestor, with 10 gains (plus the inherited HUCA) of which only 2 are of known function. Similarly, of 31 additional gains at β-ancestor, only 10 have known function and of 32 additional gains at the γ-ancestor, the function is known for only 9. Together, these three ancestors, $\beta\gamma$, β and γ, received 73 gains of which 52, more than 70%, are of unknown function.

This shows that so far researchers in the area concentrated their efforts more on commun features among the herpesviridae. The mechanisms separating the three superfamilies, especially those for β and γ, are yet to be investigated. Our reconstructions give clear indications of what proteins should be studied next.

Conclusion

Clustering is an activity purported to help in enhancing knowledge of the domain the data relate to. Typically, this comes via set of features assigned to entities that are to be clustered; the features reflect the knowledge and are to be used in interpreting cluster results. In situations in which entities are supplied with their similarities only, entity features still can be used for interpretation, too. However, the general knowledge of the data is much weaker in this case, which is reflected, indeed, in the choice of similarity rather than feature data and, respectively, leading to lack of sensible features to look at when interpreting results. In such a situation, data recovery clustering supplies a reasonable device for reflection of the domain knowledge, the soft similarity threshold that serves as the similarity shift value. This value, in the data recovery clustering context, determines other clustering parameters such as the number of clusters. The domain knowledge, even if rather weak, can produce two sets of pairs of entities: those that should and those that should not fall into the same clusters. This may lead to considerably narrowing down the choice of reasonble threshold values as shown in the previous section in which herpesvirus data from VIDA database are analysed. We further show that in a situation in which there is an independent interpretation device such as reconstruction of the evolutionary history of the protein family corresponding to a cluster, the clusters lead to enhancing knowledge with a set of interpretations. These may allow further reduction of choices for the clusterings using the criterion of consistency among the interpretations.

As an independent result, we have come to a set of proteins that represent descendants of the same gene but have lost all the similarity between their amino acid sequences. Still, their positioning in metabolic processes have been caught on the higher syntactic level, of the gene arrangement within genomes.

This shows that a possile direction for further work can be application of similar principles for clustering and interpreting at other genomic databases.

References

[1] Alba, M.M., Das, R., Orengo, C. and Kellam, P. (2001a) Genomewide function conservation and phylogeny in the herpeviridae, *Genome Research*, 11, 43-53.

[2] Alba, M.M., Lee, D., Pearl, F.M., Shepherd, A.J., Martin, N., Orengo, C. and Kellam, P. (2001b) VIDA: A virus database system for the organisation of animal virus genome open reading frames, *Nucleic Acid Research*, 29, 133-136.

[3] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Research*, 25, 3389-3402.

[4] Bader, G.D. and Hogue, C.W.V. (2003) An automated method for finding molecular complexes in large protein interaction networks, *BMC Bioinformatics*, 4:2.

[5] A. Ben-Dor, R. Shamir, Z. Yakhini, Clustering gene expression patterns, *Journal of Computational Biology*, **6**, 281-297, 1999.

[6] J.P. Benzecri (1992) *Correspondence Analysis Handbook*, New York: Marcel Dekker.

[7] S. Brohée and J. van Helden (2006) Evaluation of clustering algorithms for protein-protein interaction networks, *BMC Bioinformatics*, 7:488 (http://www/biomedcentral.com/1471-2105/7/488.

[8] Davison, A.J. (2002) Evolution of the herpesviruses, *Veterinary Microbiology*, 86, 69-88.

[9] Davison, A.J., Dargan, D.J. and Stow, N.D. (2002) Fundamental and accessory systems in herpesvirus: Review, *Antiviral Research*, 56, 1-11.

[10] J. Felsenstein, *PHYLIP 3.6: Phylogeny Inference Package*, http://evolution.genetics. washington.edu/phylip/, 2001.

[11] K. Florek, J. Lukaszewicz, H. Perkal, H. Steinhaus, and S. Zubrzycki (1951) Sur la liason et la division des points d'un ensemble fini, *Colloquium Mathematicum, 2*, 282-285.

[12] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan (1989) A fast parametric maximum flow algorithm and applications, *SIAM Journal on Computing, 18*, 30-55.

[13] N. Garg, V. V. Vazirani, M. Yannakakis (1996) Approximate Max-Flow Min-(Multi)Cut theorems and their applications, *SIAM Journal on Computing, 25*, n.2,235-251.

[14] J. Gouzy, P. Eugene, E.A. Greene, D. Khan, and F. Corpet (1997) XDOM, a graphical tool to analyse domain arrangements in any set of protein sequences, *Comput. Appl. Biosciences*, 13, 601-608.

[15] J.C. Gower and G.J.S. Ross (1969) Minimum spanning trees and single linkage cluster analysis *Applied Statistics, 18*, 54-64.

[16] J.A. Hartigan (1967) Representation of similarity matrices by trees, *J. Amer. Stat. Assoc., 62*, 1140-1158.

[17] R. Holzerlandt, C. Orengo, P. Kellam, and M.M. Alba (2002) Identification of new herpesvirus gene homologs in the human genome, *Genome Research*, 12, 1739-1748.

[18] K.J. Holzinger and H.H. Harman (1941) *Factor Analysis*, University of Chicago Press, Chicago.

[19] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth (1996) From data mining to knowledge discovery: An overview. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining*, Menlo Park, Ca: AAAI Press/The MIT Press, 1-37.

[20] Inkpen, D., and Desilits, A. (2005) Semantic similarity for detecting recognition errors in automatic speech transcripts, *Conference on Empirical Methods in Natural Language Processing*, Vancouver, Canada.

[21] A.K. Jain and R.C. Dubes (1988) *Algorithms for Clustering Data*, Englewood Cliffs, NJ: Prentice Hall.

[22] Jarvis, R. A., and Patrick, E. A. (1973) Clustering using a similarity measure based on shared nearest neighbors, *IEEE Trans. Comput., 22*, 1025–1034.

[23] Jenner, R., Mar Alba, M., Boshoff, C. and Kellam, P. (2001) Kaposis sarcoma-associated herpesvirus latent and lytic gene expression as revealed by DNA arrays, *J. Virol*, 75(2), 891-902

[24] Hideya Kawaji, Yoichi Takenaka, Hideo Matsuda (2004) Graph-based clustering for finding distant relationships in a large set of protein sequences, *Bioinformatics*, 20(2), 243-252.

[25] V. Kupershtoh, B. Mirkin, and V. Trofimov (1976) Sum of within partition similarities as a clustering criterion, Automation and Remote Control, 37, n.2, 548-553.

[26] Mirkin, B. (1976) *Analysis of Categorical Features*, Finansy i Statistika Publishers, Moscow, 166 p. (In Russian)

[27] Mirkin, B. (1987) Additive clustering and qualitative factor analysis methods for similarity matrices, *Journal of Classification, 4*, 7-31; Erratum (1989), *6*, 271-272.

[28] B. Mirkin (1990) A sequential fitting procedure for linear data analysis models, *Journal of Classification, 7*, 167-195.

[29] B. Mirkin (1996) *Mathematical Classification and Clustering*, Dordrecht: Kluwer Academic Press.

[30] Mirkin, B. (2005) *Clustering for Data Mining: A Data Recovery Approach*, Chapman and Hall, Boca Raton.

[31] Mirkin, B., Fenner, T., Galperin, M. and Koonin, E. (2003) Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes, *BMC Evolutionary Biology*, 3:2 (www.biomedcentral.com/1471-2148/3/2/).

[32] Mirkin, B. and Koonin, E. (2003) A top-down method for building genome classification trees with linear binary hierarchies, In M. Janowitz, J.-F. Lapointe, F. McMorris, B. Mirkin, and F. Roberts (Eds.) *Bioconsensus*, DIMACS Series, V. 61, Providence: AMS, 97-112.

[33] B. Mirkin, R. Camargo, T. Fenner, G. Loizou and P. Kellam (2006) Aggregating homologous protein families in evolutionary reconstructions of herpesviruses, In D. Ashlock (Ed.) *Pro-*

ceedings of the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, Piscataway NJ, 255-262.

[34] Montague, M.G. and Hutchison III, C.A. (2000) Gene content phylogeny of herepsviruses. *Proc. Natl. Acad. Sci*, 97(10), 5334-5339.

[35] F. Murtagh (2005) *Correspondence Analysis and Data Coding with JAVA and R*, Chapman & Hall/CRC, Boca Raton, FL.

[36] *NCBI GenBank/Entrez web site*, http://www.ncbi.nlm.nih. gov/entrez, 2006.

[37] Nei, M. and Kumar, S. (2000) *Molecular Evolution and Phylogenetics*, Oxford University Press.

[38] R.N. Shepard and P. Arabie (1979) Additive clustering: representation of similarities as combinations of overlapping properties, *Psychological Review, 86*, 87-123.

[39] J. Shi and J. Malik (2000) Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*, n. 8, 888-905.

[40] Smid, M., Dorssers, L.C.J., and Jenster, G. (2003) Venn Mapping: clustering of heterologous microarray data based on the number of co-occurring differentially expressed genes, *Bioinformatics*, 19, no. 16, 2065-2071.

[41] Small, H. (1973) Co-citation in the scientific literature: A new measure of the relationship between two documents, *Journal of the American Society for Information Science*, 24, 265-269.

[42] Snel, B., Bork, P. and Huynen, M.A. (2002) Genomes in flux: The evolution of archaeal and proteobacterial gene content, *Genome Research*, 12, 17-25.

[43] Tatusov, R.L., Galperin, M.Y., Natale, D.A. and Koonin, E.V. (2000) The COG database: a tool for genome-scale analysis of protein function and evolution, *Nucleic Acids Research*, 28, no.1, 33-36.

[44] Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice, *Nucleic Acids Research*, 22, 4673-4680.