
Clustering Proteins and Reconstructing Evolutionary Events

Boris Mirkin^{1,2}

¹ School of Computer Science, Birkbeck University of London, Malet Street, London, WC1 7HX, UK, mirkin@dcs.bbk.ac.uk

² Department of Applied Mathematics, Higher School of Economics, Kirpichnaya 33/5, Moscow, RF, bmirkin@yandex.ru

Summary. The issue of clustering proteins into homologous families has attracted considerable attention by researchers. On one side, many databases of protein families have been developed by using relatively simple clustering methods and a lot of manual curation. On the other side, more elaborated clustering approaches have been used, yet with a very limited degree of success. This paper advocates an approach to clustering protein families involving the knowledge of protein functions to adjust the parameter of similarity scale shift. We proceed to reconstruct HPF evolutionary histories to both further narrow down the choice of the cluster solution and interpret clusters.

Key words: Clustering, Protein family, Neighborhood, Significance threshold, Scale shift, Reconstructed history.

1 Introduction: Clustering and knowledge feedback

Clustering is conventionally applied for deriving protein families (see, for example, [23, 22, 3, 11, 6, 17, 19, 5, 20, 18]).

Our clustering method falls within the so-called data recovery approach applied to the similarity data. According to this approach similarity data are considered as weighted sums of “ideal” structures such as partitions or clusters, the clusters and their intensity weights being determined by minimizing the differences between the given similarity data and those generated by the putative model. We extract clusters one-by-one [14, 15], to both facilitate the search and supply 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 prior similarity shift. This parameter also is a ‘soft’ similarity threshold, so that entities whose similarity is less than its value are unlikely to get combined in the same cluster. The parameter’s value may strongly affect the number and contents of the clusters, and it can be derived according to the least-squares

criterion. However, as we shall illustrate, a better choice may be made by using proteomics knowledge.

A clustering method, derived from the data recovery approach, can be applied to evolutionary intergenomic studies in which clusters are interpreted as homologous protein families (HPFs). The proteins in each of these families are assumed to be inherited from the same ancestral, so that an HPF can be parsimoniously mapped to an evolutionary tree on the set of genomes under consideration, thereby reconstructing the HPF's evolutionary history. Obviously, the reconstructed 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. As the level of aggregation of proteins depends on the value of the similarity threshold/shift, the evolutionary mapping of protein families can be used for fine tuning that value by analyzing the consistency between the reconstructed histories and other data available.

To determine an appropriate value for the similarity shift, we analyze 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, so that not one but two reasonable similarity shift values emerged: one would guarantee that HPFs with dissimilar functions would be in different clusters, whereas the other would give the minimum error in separating protein pairs with similar and dissimilar functions. The final choice, however, requires further knowledge of the genomes, viz. the consistency between the suggested ancestral reconstructions and gene arrangements.

Therefore, our approach involves two phases of interference of the clustering and proteome knowledge: one, passive, takes in the knowledge to adjust the values of a clustering parameter, and the second, active, makes use of cluster-based evolutionary histories of protein functions.

The rest of the paper is organized as follows. Section 2 describes the data recovery approach to clustering similarity data. Section 3 is devoted to a description of the results of clustering protein families by using the knowledge of protein functions to identify similarity shift values. Section 4 describes some results involving the reconstructed evolutionary histories. In Section 5 we conclude and outline possible future work.

2 Clustering using the data recovery approach

2.1 Additive clustering and one-by-one iterative extraction

Let I be a set of entities under consideration and let $A = (a_{ij})$ be a symmetric matrix of similarities (or, synonymously, proximities or interactions) between entities $i, j \in I$.

The additive clustering model [21, 13, 14] assumes that the similarities in A are generated by a set of ‘additive clusters’ $S^k \subseteq I$, $k = 0, 1, \dots, K$, in such a way that each a_{ij} approximates the sum of the intensities of those clusters that contain both i and j :

$$a_{ij} = \sum_{k=1}^K \lambda_k s_i^k s_j^k + \lambda_0 + e_{ij}, \quad (1)$$

where $s^k = (s_i^k)$ are the membership vectors of the unknown clusters S^k and λ_k are their intensities, $k = 1, 2, \dots, K$; e_{ij} are the residuals to be minimised.

The intercept value λ_0 can be interpreted as the intensity of the universal cluster $S_0 = I$ that must be part of the solution and, on the other hand, it has a meaning of the similarity shift, with the shifted similarity matrix $A' = (a'_{ij})$ defined by $a'_{ij} = a_{ij} - \lambda_0$. Equation (1) for the shifted model can be rewritten in an obvious way so that it expresses a'_{ij} through clusters $k = 1, \dots, K$ by moving λ_0 onto the left. The role of the intercept λ_0 in (1) as a ‘soft’ similarity threshold is of special interest when λ_0 is user specified because the shifted similarity matrix a'_{ij} may lead to different clusters at different λ_0 values.

To fit model (1), we apply one-by-one cluster extracting strategy by minimizing, at each step $k = 1, \dots, K$ criterion

$$L^2(S, \lambda) = \sum_{i,j \in I} (a'_{ij} - \lambda s_i s_j)^2 \quad (2)$$

and setting the found solutions S and λ as S_k and λ_k , respectively. Obviously, the optimal λ_k is the average of residual similarities a'_{ij} within S_k . The residual similarities a'_{ij} are updated after each step k by subtracting $\lambda_k s_{ik} s_{jk}$.

This strategy leads to the following decomposition of the data scatter into the contributions of the extracted clusters S^k (“explained” by the model) and the minimized residual square error (the “unexplained” part) [14]:

$$(A', A') = \sum_{k=1}^K [s^{kT} A^k s^k / s^{kT} s^k]^2 + (E, E) \quad (3)$$

The inner products (A', A') and (E, E) denote the sums of the squares of the elements of the matrices, considering them as vectors.

2.2 One cluster clustering

In this section, we turn to the problem of minimization of (2) for extraction of a single cluster. It should be noted that if A is not symmetric, it can be equivalently changed for symmetric $\hat{A} = (A + A^T)/2$ [13, 15]. For the sake of simplicity, in this section, we assume that the diagonal entries a_{ii} are all zero.

Pre-specified intensity

When the intensity λ of the cluster to be found is pre-specified, criterion (2) can be expressed as

$$L^2(S, \lambda) = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2 = \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j \quad (4)$$

For $\lambda > 0$, minimizing (4) is equivalent to maximizing the sum on the right,

$$f(S, \pi) = \sum_{i,j \in I} (a_{ij} - \pi) s_i s_j = \sum_{i,j \in S} (a_{ij} - \pi). \quad (5)$$

This implies that, for any entity i to be added to or removed from the S under consideration, the difference between the value of (5) at the resulting set and its value at S , $f(S \pm i, \pi) - f(S, \pi)$, is equal to $\pm 2f(i, S, \pi)$ 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 (5): 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 [14] iteratively finds an $i \notin S$ maximizing $+f(i, S, \pi)$ 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 utilized by starting from singleton $S = \{i\}$, for each $i \in I$, and finally selecting, from all S found at different i , that S that contributes most to the data scatter, i.e. minimizes the square error L^2 (2).

The algorithm CAST [4], 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 $i \notin S$ [14].

Changing the threshold π should lead to corresponding changes in the optimal S : the greater π is, the smaller S will be [14].

Optimal intensity

When λ in (4) is not fixed but chosen to further minimize the criterion, it is easy to prove that:

$$L^2(S, \lambda) = (A, A) - [s^T A s / s^T s]^2, \quad (6)$$

The proof is based on the fact that the optimal λ is the average similarity $a(S)$ within S , i.e.,

$$\lambda = a(S) = s^T A s / [s^T s]^2, \quad (7)$$

since $s^T s = |S|$.

The decomposition (6) implies that the optimal cluster S must maximize the criterion

$$g^2(S) = [s^T A s / s^T s]^2 = a^2(S) |S|^2 \quad (8)$$

or its square root, the Raleigh quotient,

$$g(S) = s^T A s / s^T s = a(S) |S| \quad (9)$$

over all binary vectors s .

To maximize $g(S)$, one may utilize the ADDI-S algorithm [14], which is a version of the algorithm ADDI/CAST, described above, in which the threshold π is recalculated after each step as $\pi = a(S)/2$, corresponding to the optimal λ in (7).

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 $i \notin S$.

ADDI-S utilizes 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.

3 Proteome knowledge in determining similarity shift

3.1 Protein families and evolutionary tree

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 [22, 1]. 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 one-to-one 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 [16].

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 [16, 17]. 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 the degree of similarity between proteins within an alignment of protein sequences, typically, with PSI-BLAST [2] or the like, is not always sufficient to automatically identify the families. This is why a two-stage strategy for identifying HPFs has been considered in [17]. 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 by using a software such as the XDOM [9, 1]. 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 (i) some functional sites, contiguous in the spatial fold, may correspond to dis-contiguous fragments of protein sequences, and (ii) multifunctional proteins may bear resemblances to different proteins at different places.

The fragmented HPFs may lead then to wrong reconstructions of functional histories because if they bear similar proteins and thus should be combined into a single aggregate HPF, then its origin ought to be in the ultimate ancestor corresponding to the tree root rather than in separate subtrees of the phylogeny.

Therefore, the next stage of the strategy is to cluster the first stage HPFs into larger aggregations. Since entities at this stage are not single proteins but protein families, one needs to score similarities between families rather than single proteins, which we do by using the set-theoretic similarity – not between HPFs themselves – but rather between their neighborhoods defined by using PSI-BLAST [2]. 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 into a set of proteins, the neighborhood, according to a majority rule. Third, a set-similarity index values are computed between the HPF neighborhoods. One can notice such advantages of this approach as

- Accuracy of protein alignments because only neighboring proteins are aligned here;
- Better capturing functional properties of the proteins. 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 assigned to separate HPFs within the VIDA database [1]. However, their sets of homologous protein neighbors (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 63% on average.
- The evolutionary timing can be caught up at different majority thresholds [17] as an alternative to relying on statistical frequency profiles in PSI-BLAST [2].

The idea of employing neighborhoods to measure similarities between entities stems from earlier work, see for example [10]. Our clustering model leads to the index of average overlap $mbc = (n/n1 + n/n2)$ for scoring the similarity between subsets of sizes $n1$ and $n2$ whose overlap is of size n .

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 by authors of [17] from the herpesvirus database VIDA [1] and an evolutionary tree has been built over the genomes using the neighbor joining algorithm from PHYLIP package [8] (see Figure ??). This tree totally agrees with the previously published instances of herpesvirus phylogenies [7, 12], except for the uncertainty fragments acknowledged in these publications. A set of 740 homologous protein families (HPFs) represented in these 30 genomes have been extracted from the VIDA database too [1].

3.2 Utilizing knowledge of proteome

To choose a right λ_0 value in model (1), one should use the external knowledge of the proteome, independent of sequence similarity estimates, for example, of functional activities of the proteins. Each HPF is supposed to have a biomolecular function (for examples of function see Table 1 below), though unfortunately functions of most proteins are unknown yet. We can use those HPFs that have similar functions versus those that are not to choose the 'right' level of the similarity shift. Operationally, we consider proteins as functionally similar if they are consistently named between the herpesvirus genomes and/or they share the same known function. The similarity shift value should be taken such that similarities between dissimilar HPFs get negative after the shift while those between similar HPFs remain positive. To implement this idea, we analyzed 287 available pairs of HPFs with known function and

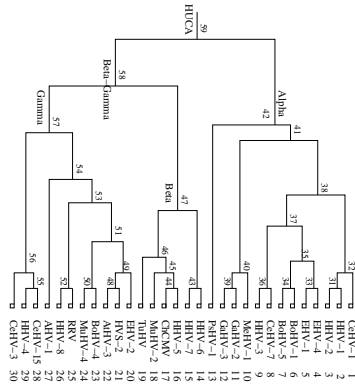


Fig. 1. Herpesvirus genomes evolutionary tree analyzed. The root corresponds to the herpesvirus ultimate common ancestor (HUCA); its child on the right to the ancestor of α superfamily, and the child on the left, to the common ancestor of β and γ superfamilies. The numbers are labels of different nodes on the tree.

positive similarity value. Among them, no dissimilar pair has a greater *mbc* similarity than 0.66, which should imply that the shift value $\lambda_0 = 0.67$ confers specificity for the production of APFs.

Unfortunately, the situation is less clear cut for the functionally similar proteins. Out of the 86 similar pairs available, there are 24 pairs (28%) that have their mutual similarity value less than 0.67. Thus at the similarity shift at 0.67, 28% of the similar pairs will not be identified as such, that is, at this similarity shift the method would lack sensitivity. To choose a similarity shift that minimizes the error in assigning negative and positive similarity values, one needs to compare the distribution of similarity values in the set of functionally similar pairs with that in the set of dissimilar pairs. As Figure 2 shows, the graphs intersect when the similarity value *mbc* is 0.42.

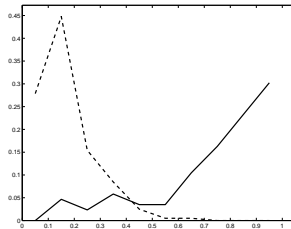


Fig. 2. Empirical frequency functions for the sets of functionally similar pairs (solid line) and dissimilar pairs (dashed line). The *x*-values represent the *mbc* similarity.

Thus the external knowledge of functional similarity between some HPFs supplies us with two candidates for the similarity shift values, 0.67 and 0.42.

There are 80 APF 0.67-clusters comprising original 180 HPFs and leaving 560 HPFs unclustered, and 102 0.42-APF clusters over original 249 HPFs, and 491 HPFs unclustered. The first 80 0.42-clusters correspond one-to-one to the 80 0.67-clusters. Which one is more suitable? To answer this, we are going to develop and use more knowledge of the genomes.

4 Advancing genome knowledge

4.1 Reconstructed histories of HPFs

For the further analysis, we utilize the evolutionary histories of HPFs over the evolutionary tree. These histories have been derived using the principle of maximum parsimony [16]. These histories supply us with the 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 deeper levels.

The reconstructions of the five ancestors with APFs found at the two similarity shift values, $\lambda_0 = 0.42$ and $\lambda_0 = 0.67$, are essentially the same. The only exception is the common ancestor of the α superfamily, which gains three more APFs when λ_0 decreases from 0.67 to 0.42. These are: (i) APF81 comprised of HPFs 9 and 504, both of glycoprotein C; (ii) APF82 comprised of HPF 38 and HPF 736, both of glycoprotein I; and (iii) 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 the 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 of 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 algorithm to the ancestral α node [17]. This leads us to conclude that glycoprotein I must also belong to the α ancestor, thus implying that similarity shift $\lambda_0 = 0.42$, better fits to the knowledge added by the reconstruction than $\lambda_0 = 0.67$, because at which glycoprotein I's aggregate family falls in α ancestor only at the former value.

4.2 Derived ancestors of herpes proteins

The analysis of glycoproteins in the α superfamily has led us to accept the value $\lambda_0 = 0.42$ and the corresponding number of protein families, after aggregation, 593. Some of the structural conclusions from the mapping of the aggregate 0.42-families to the evolutionary tree are presented in Table 1 taken from [18].

Table 1. Some previously determined herpesvirus common ancestor D-HUCA's [7, 12] functions within membrane glycoproteins in the herpesvirus ancestor (two columns on the right) versus the results from the mapping of our clusters (three columns on the left); with function descriptions taken from VIDA.

Mapping	H/APF	Description, gp – – glycoprotein	HSV-1 Gene	D-HUCA
HUCA	20	gp M, HHV-1 UL10	UL10	gp M; compl. with gp N
HUCA	3	gp B, HHV-1 UL27	UL27	gp B
HUCA	APF 3: 42 12 531	<i>gp H, HHV-1 UL22</i> <i>gp H, HHV-8 ORF22</i> <i>gp H, HHV-8 ORF22</i>	UL22	gp H; compl. with gp L
ALPHA	47	gp L, HHV-1 UL1	UL1	gp L; compl. with gp H
BETA	50	gp L, HHV-5 UL115		
GAMMA	114	gp L, HHV-8 ORF47		
GAMMA	296	gp L, MuHV-4 ORF47		

The common ancestor of herpesviruses, HUCA, according to our reconstruction, should be comprised of 29 protein families. These all are well studied proteins except only three of the participating families of no known function.

Relations between our mapping results and D-HUCA are illustrated in Table 1: the fragmented HPFs, having been aggregated into APF3, fall into HUCA, yet some HPFs clearly fail to aggregate (47, 50, 114 and 296). The ancestor of each α -, β -, and γ family, has a glycoprotein L, so that the corresponding gene may have been present in HUCA as well. The HPFs have no significant sequence similarity nor common neighbors and, thus, cannot be combined together by clustering alone. Yet, at the genome organisation level each of the glycoprotein L genes always exactly precedes the corresponding Uracil-DNA glycosylase gene, which is mapped into HUCA, according to our reconstruction. This suggests that these are common ancestral genes indeed; just they have undergone sequence change to a level where sequence similarity is no longer sufficient to assign homology.

Concerning other four superfamily ancestors in our study, α , $\beta\gamma$, β and γ , our reconstructions show that only the contents of the α superfamily is relatively well studied. This means that the mechanisms separating the three

superfamilies, especially those for β and γ , are yet to be investigated. Our reconstructed histories give clear indications of what proteins should be studied next.

5 Conclusion

Clustering is an activity purported to help in enhancing knowledge of the area the data relate to. Typically, this comes via a set of features assigned to the entities; the features reflect the knowledge and are to be used in interpreting cluster results. In proteomic studies, entities are frequently supplied with their similarities only, lacking any sensible features to look at when interpreting results. In such a situation, data recovery clustering supplies a reasonable device for reflection of the knowledge of proteome, the similarity shift value. Using two sets of protein pairs, those that should and those that should not fall into the same clusters, may lead to considerably narrowing down the choice of reasonable shift values, as shown above. One more step is in using the parsimonious reconstruction of the evolutionary history of the clusters. This may allow both further reduction of choices by confronting the reconstructions with the gene arrangement within genomes and interpretation of the clusters.

A possible direction for further work can be application of similar principles for clustering and interpreting of protein families at other sets of related genomes.

References

1. Alba, M.M., Lee, D., Pearl, F.M., Shepherd, A.J., Martin, N., Orengo, C. and Kellam, P. (2001) VIDA: A virus database system for the organisation of animal virus genome open reading frames, *Nucleic Acid Research*, 29, 133-136.
2. 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.
3. 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.
4. A. Ben-Dor, R. Shamir, Z. Yakhini, Clustering gene expression patterns, *Journal of Computational Biology*, 6, 281-297, 1999.
5. Brown, D.P., Krishnamurty, N. and Sjolander, K. (2007) Automated protein subfamily identification and classification, *PloS Computational Biology*, 3, 8: e160, 1526-1538.
6. Chen, Y., Reilly, K.D., Sprague, A.P., and Guan, Z (2006) SEQOPTICS: a protein sequence clustering system, *BMC Bioinformatics*, 2006, 7 (Suppl. 4): S10.
7. Davison, A.J. (2002) Evolution of the herpesviruses, *Veterinary Microbiology*, 86, 69-88.
8. J. Felsenstein, *PHYLIP 3.6: Phylogeny Inference Package*, <http://evolution.genetics.washington.edu/phylip/>, 2001.

9. 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.
10. Jarvis, R. A., and Patrick, E. A. (1973) Clustering using a similarity measure based on shared nearest neighbors, *IEEE Trans. Comput.*, 22, 1025-1034.
11. 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.
12. McGeoch, D.J., Rixon, F.J., and Davison, A.J. (2006) Topics in herpesvirus genomics and evolution, *Virus Research*, 117, 90-104.
13. Mirkin, B. (1976) *Analysis of Categorical Features*, Moscow: Statistika Publishers (in Russian).
14. Mirkin, B. (1987) Additive clustering and qualitative factor analysis methods for similarity matrices, *Journal of Classification*, 4, 7-31; Erratum (1989), 6, 271-272.
15. Mirkin, B. (1996) *Mathematical Classification and Clustering*, Dordrecht: Kluwer Academic Press.
16. 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/).
17. Mirkin, B., Camargo, R., Fenner, T., Loizou, G. and Kellam, P. (2006) Aggregating homologous protein families in evolutionary reconstructions of herpesviruses, In D. Ashlock (Ed.) *Proceedings of the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Piscataway NJ, 255-262.
18. Mirkin, B., Camargo, R., Fenner, T., Loizou, G. and Kellam, P. (2009) Similarity clustering and domain knowledge in reconstruction of evolutionary gene histories in herpesvirus (submitted).
19. Paccanaro, A., Casbon, J.A., and Saqi M. (2006) Spectral clustering of protein sequences, *Nucleic Acids Research*, 34:1571-1580.
20. Poptsova, M.S. and Gogarten, J.P. (2007) BranchClust: a phylogenetic algorithm for selecting gene families, *BMC Bioinformatics*, 8:120.
21. R.N. Shepard and P. Arabie (1979) Additive clustering: representation of similarities by overlapping properties, *Psychological Review*, 86, 87-123.
22. 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.
23. 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.