Theor Chem Acc (2010) 125:569–581
DOI 10.1007/s00214-009-0614-0

REGULAR ARTICLE

Similarity clustering of proteins using substantive knowledge and reconstruction of evolutionary gene histories in herpesvirus

Boris Mirkin · Renata Camargo · Trevor Fenner · George Loizou · Paul Kellam

Received: 27 January 2009 / Accepted: 18 July 2009 / Published online: 5 August 2009

Abstract The issue of clustering proteins into homologous protein families (HPFs) has attracted considerable attention by researchers. On one side, many databases of protein families have been developed by using popular sequence alignment tools and relatively simple clustering methods followed by extensive manual curation. On the other side, more elaborate clustering approaches have been used, yet with a very limited degree of success. This paper advocates an approach to clustering protein families involving knowledge of the protein functions to adjust the parameter of similarity scale shift. One more source of external information is utilised as we proceed to reconstruct HPF evolutionary histories over an evolutionary tree; the consistency between these histories and information on gene arrangement in the genomes is used to narrow down the choice of the clustering.

Dedicated to Professor Sandor Suhai on the occasion of his 65th birthday and published as part of the Suhai Festschrift Issue.

B. Mirkin (✉) · R. Camargo · T. Fenner · G. Loizou
School of Computer Science and Information Systems, Birkbeck College, University of London, London, UK
e-mail: mirkin@dcs.bbk.ac.uk

B. Mirkin
Division of Applied Mathematics, Higher School of Economics, Moscow, Russia

P. Kellam
Centre for Virology, Department of Infection, University College London, London, UK

1 Introduction

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 proteins and protein families (see, for example, [4, 7, 8, 18, 30, 32, 35, 37, 38]). On the other hand, there exists a long-standing tradition of data recovery criteria and methods for clustering similarity data (see, for instance, [14, 22, 33]). Clustering methods considered in this paper are within this second tradition and are, in essence, extensions of methods proposed in [22, 23].

Before describing our approach and computational results, we are going to point out the considerations related to tackling issues in biology with a computational approach—these are due to the different perspectives taken by the biologist and computer scientist when looking at the very basic concepts involved. The difference comes from the notion that a biologist is interested in describing the real phenomena in all their varieties, whereas a computer scientist needs a tangible general principle for transformation of the available data into an output (similar issues are treated in [2]). Consider, for example, such concepts as 'homologous proteins', 'evolutionary tree' and 'principle of parsimony'. For the biologist, homologous proteins are, by their very definition, those descended from a common ancestor. For the computer scientist this sounds somewhat cryptic—it is not operational, at least in the beginning, because the ancestry is not pre-specified and can be quite murky, especially for viruses. Establishing homology for proteins is an outcome rather than pre-requisite for the computer scientist, who is thus left to rely mostly on the proteins' similarity. This is not that problematic when data

of the protein structures are available; it is the protein's folding that mainly effects its function, and the folds are much more conservative than the protein amino acid sequence composition. Unfortunately, in our case the folds are largely unknown; only about 7% of proteins under study have their representatives among the Protein Data Bank (PDB) structures [6]. That means that we are left with protein sequence similarity as the major device for working on homology. We do realise that common ancestry does not necessarily imply sequence similarity. The question is: how far can this bring us? How much of the homology can be derived from the sequence similarity alone? It should be pointed out that the usage of sequence similarity based on computational tools for pair-wise or multiple alignment, along with heavy manual curation, dominates in building databases of homologous protein families such as COG [38] or VIDA [1].

To address the issue, we utilise our method PARS [24] for reconstructing evolutionary histories of the sets of proteins that are hypothetically homologous. This brings us to two further biologically charged terms: the evolution and the principle of parsimony. A biologist may claim that the concept of evolutionary tree is not quite applicable here because, first, the virus is not strictly speaking a life form and, second, the very concept of a fully resolved evolutionary tree suggests the idea of progress as a constant accumulation of complexity, which is not necessarily true for the herpesvirus. A computer scientist, who has no problems considering the evolution of computers or other technical devices, would fail to understand the first argument. As to the second argument, the computer scientist would say that the evolutionary tree bears a representation of the evolution, and that whatever events are at odds with the tree structure could be mapped to the tree as an additional annotation of its nodes and edges. With respect to the principle of parsimony, a biologist would justly claim that this principle applies in real evolution very rarely, if at all, to which a computer scientist would answer that this principle, an embodiment of Occam's razor, is just a heuristic tool for tackling a data reconstruction problem when there is not enough substantive information available.

The list of possible misconceptions between biologists and computer scientists can be further extended. They would boil down to the following. The computer scientist needs a general principle implemented in the algorithm to start computation, and the biologist would point out that there are many exceptions to the principle. The computer scientist would reply that there must be some conditions implying the exceptions, and these might become less mysterious if one analyses discrepancies between a solution based on the initial principle and the real world data. This paper is an example of such an approach; we start from a general principle and then use additional information to computationally advance in modelling a complex molecular biology phenomenon.

Fortunately, for the herpesvirus genomes, there exists a reliable reconstruction of their ancestral genome based on various types of evidence [9, 20]. This allows us to judge the quality of the computer-produced "homologous protein families" (that we refer to as HPFs or APFs later in the text) by comparing those that have made it to the last ancestor of herpesvirus genomes according to our method with those in the reconstruction of [20].

Accordingly, the following sections pursue several lines of attacking the issue of the computational reconstruction of protein families utilising, initially, sequence similarity estimates and then whatever additional information is available for the purpose.

First of all, we translate sequence similarity estimates into set similarity estimates by moving from the usage of protein sequences and VIDA database HPFs to their lists of neighbouring proteins. This allows us to restrict the usage of alignment scores only to cases of similar sequences, for which the alignments are reliable. We can then set similarity scores, which are more reliable, to develop a clustering method based on modelling similarity data using within-cluster intensities. We extract clusters one-by-one, which not only finds them effectively, but also supplies meaningful estimates of their intensity and contribution to the data scatter. The clusters found in this way are interpreted as HPFs, so that we can map their histories to the available evolutionary tree. The reconstruction has not brought any unexpected items to the last common ancestor of herpesvirus, HUCA in [20], which is encouraging. Yet, it has left many of the ancestral HUCA genes of [20] to 'emerge' at much more recent nodes—because our original HPFs were too fragmentary.

As another line of attack, we utilise a parameter of our clustering model, analogous to the intercept of the regression line, that plays the role of a similarity shift applied prior to clustering. This parameter is also a kind of similarity threshold, so that entities whose similarity is less than this value are unlikely to get combined in the same cluster. The similarity shift is therefore substantively interpretable unlike parameters used in other methods, such as the number of clusters. Nevertheless, its value may strongly affect the number and contents of the clusters. 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, 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. Both of these values are derived using proteome knowledge.

The line of attack we employ uses the consistency between the suggested reconstructions of ancestral genomes and information on gene arrangement within them.

The remainder of the paper is organised as follows. Section 2 describes our data recovery model for clustering similarity data and a clustering method, ADDI-S one-by-one clustering, derived from the model. Section 3 is devoted to a description of the results of aggregating protein families with ADDI-S by using the neighbourhood approach to measuring similarity. The substantive knowledge used to identify similarity shift values is described in Sect. 3.3. Results of mapping clusters onto an evolutionary tree of herpesviruses and insight gained from our approach are described in Sect. 4. In the conclusion we outline possible future work.

The preliminary report of some of the work described here, which previously appeared as a conference paper [26], was more focused on the algorithms for evolutionary reconstruction. In the current paper we focus on the clustering process and, in particular, our clustering model. In addition, we resolve the outstanding problem of the final selection of the relevant clusters by utilising our evolutionary reconstructions.

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, 22, 33] assumes that the similarities in A are generated by a set of 'additive clusters' $S^k \subseteq I$, $k = 0, 1, \ldots, K$, in such a way that each a_{ij} approximates the sum of the intensities of those clusters that contain both i and j:

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

where $s^k = (s_i^k)$ are the membership vectors of the unknown clusters S^k and $\lambda_k > 0$ are their intensities, $k = 1, 2, \ldots, K$; e_{ij} are the residuals to be minimised.

The intercept value $\lambda_0 \geq 0$ 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 can also be thought of as a similarity shift, with the shifted similarity matrix $A' = (a'_{ij})$ defined by $a'_{ij} = a_{ij} - \lambda_0$. Moving λ_0 onto the lefthand side of Eq. 1 yields the equivalent equation for the shifted similarities a'_{ij}. The role of the intercept λ_0 in Eq. 1 as a 'soft' similarity threshold is of special interest when λ_0 is user-specified because the shifted similarity matrix A' may lead to different clusters for different values of λ_0.

To fit the model (Eq. 1), we apply a one-by-one cluster-extracting strategy by minimising, at each step $k = 1, \ldots, K$, the criterion

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

and then define S_k and λ_k to be the solutions found for S and λ, respectively. It is easy to show that the optimal λ_k is the average of the residual similarities a'_{ij} within S_k. The residual similarities a'_{ij} are updated after each step k by subtracting $\lambda_k s_{ik} s_{jk}$.

When the clusters are required to be disjoint, this strategy can be implemented by removing the entities in the cluster S_k from the set I and reducing the size of the matrix A' accordingly, after each step k.

The method, in both versions, leads to a decomposition of the data scatter into the contributions of the extracted clusters S^k ("explained" by the model) and the minimised residual square error (the "unexplained" part) [22].

2.2 One cluster clustering

In this section, we turn to the problem of minimisation of Eq. 2 for extraction of a single cluster. Note that from now on we use A to denote the shifted similarity matrix. It should be noted that if A is not symmetric, it can be equivalently replaced by the symmetric $\hat{A} = (A + A^{\mathrm{T}})/2$ [21, 23], so we assume that A is symmetric. For the sake of simplicity, we also assume that the diagonal entries a_{ii} are all zero.

2.2.1 Pre-specified intensity

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

$$\begin{aligned} L^2(S, \lambda) &= \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2 \\ &= \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j. \end{aligned} \tag{3}$$

Since $\lambda > 0$, minimising Eq. 3 is equivalent to maximising the sum on the right,

$$f(S,\pi) = \sum_{i,j\in I}(a_{ij} - \pi)s_i s_j = \sum_{i,j\in S}(a_{ij} - \pi), \quad (4)$$

where $\pi = \lambda/2$.

This implies that, for any entity i to be added to or removed from the S under consideration, the difference between the value of Eq. 4 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 maximising Eq. 4: start with $S = \{i^*, j^*\}$ such that $a_{i^*j^*}$ is a maximum element in A, 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 [22] iteratively finds an $i \notin S$ maximising $+f(i, S, \pi)$ and an $i \in S$ maximising $-f(i, S, \pi)$, 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 that starts with the singleton $S = \{i\}$, for each $i \in I$ in turn, and finally selects the resulting S that contributes most to the data scatter, i.e. the one that minimises the square error (Eq. 3).

The algorithm CAST [5], popular in bioinformatics, is a version of the ADDI algorithm in which $f(i, S, \pi)$ is written 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 similarity between $i \in I$ and S is at least π if $i \in S$ and no greater than π if $i \notin S$ [22].

Changing the threshold π should lead to corresponding changes in the optimal S: the greater π is, the smaller S will be [22].

2.2.2 Optimal intensity

When λ in Eq. 3 is not fixed but chosen to further minimise the criterion, it is easy to prove that:

$$L^2(S,\lambda) = (A,A) - [s^{\mathrm{T}}As/s^{\mathrm{T}}s]^2, \quad (5)$$

where the inner product (A,A) denotes the data scatter, i.e. the sum of the squares of the elements of the matrix A.

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

$$\lambda = a(S) = s^{\mathrm{T}}As/[s^{\mathrm{T}}s]^2, \quad (6)$$

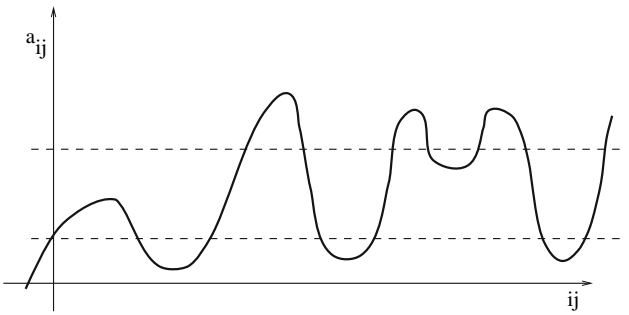

Fig. 1 A pattern of clustering depending on the subtracted similarity shift λ_0 represented by a *dashed line* on the graph. The y values denote the similarity values, and the x values depict, for purely illustrative purposes, the pairs of entities (i,j). Parts of the similarity curve over one of the *dashed lines* represent clusters found with the corresponding value λ_0. The higher the line, the smaller the clusters

since $s^{\mathrm{T}}s = |S|$.

The decomposition (Eq. 5) implies that the optimal cluster S must maximise the criterion

$$g^2(S) = [s^{\mathrm{T}}As/s^{\mathrm{T}}s]^2 = a^2(S)|S|^2 \quad (7)$$

or, equivalently, its square root, the Rayleigh quotient

$$g(S) = s^{\mathrm{T}}As/s^{\mathrm{T}}s = a(S)|S|, \quad (8)$$

over all binary vectors s.

To maximise $g(S)$, one may utilise the ADDI-S algorithm [22], which is a version of the algorithm ADDI/CAST, described above, in which the threshold π is recalculated after each addition/removal of an element to/from S as half of the optimal λ in Eq. 6. In an analogous manner to ADDI, we apply ADDI-S starting from each of the singletons $\{i\}$ in turn, with $\pi = 0$, and finally selecting the most contributing cluster.

A similar property to that for the constant threshold case holds for the resulting cluster S; 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 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 shifted similarities a'_{ij}.

3 Proteome knowledge in determining similarity shift

3.1 Aggregation of proteins in protein families

In this section, we consider the aggregation of proteins into homologous protein families (HPFs), which combine

proteins from different genomes that have the same function and considerable sequence similarity. The concept of HPF can be considered as an empirical expression of the concept of a gene as a unit of heredity in 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 whose 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 those genomes containing its members. Then the pattern of membership 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, because it is highly unlikely that the same gene emerged in the children independently. Exact formulations of the algorithms, PARS for maximum parsimony and MALS for maximum likelihood can be found in [24, 26]. Having annotated the evolutionary tree nodes with hypothetical evolutionary histories of various HPFs, realistic conclusions concerning possible histories and mechanisms of the 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 is not always sufficient to automatically identify the families, especially for rapidly evolving organisms such as bacteria and viruses.

This is why a two-stage strategy for identifying HPFs has been considered by the authors in [26]. According to this strategy, HPFs are created, first, as groups of proteins that have a common motif, a contiguous fragment of protein sequence that is similar in all members of the HPF. 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 (1) some functional sites, contiguous in the spatial fold, may correspond to discontiguous fragments of protein sequences, and (2) different fragments of multi-functional proteins may bear resemblances to unrelated proteins.

The fragmented HPFs may then lead to incorrect reconstructions of functional histories, such as those presented in Fig. 2. The reconstructed ancestral nodes exhibit the first emergence of the HPFs, labelled by differently patterned squares and shown with circles at nodes A1 and A2. These histories, however, may be due to an erroneous aggregation; the three HPFs may, in fact, bear similar

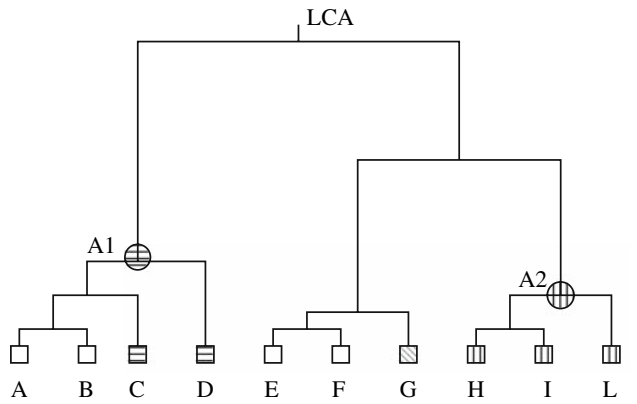

Fig. 2 An evolutionary tree over genomes *A* to *L* with three protein families (shown differently patterned, with *white squares* denoting genomes in which all three families are absent) present in genomes *C*, *D* (Family 1), *H*, *I*, *L* (Family 2), and *G* (Family 3). The reconstructed ancestors for Families 1 and 2 are shown with *circles* at nodes *A1* and *A2*, respectively. That in *G* remains as is. If, however, these three families were recognised as parts of the same family, then their reconstructed ancestor ought to be placed at the root node *LCA*

proteins and thus should be combined into a single aggregate HPF, whose origin then ought to be in the root of the tree corresponding to the ultimate ancestor LCA.

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, we need to score similarities between families rather than single proteins. This issue will be covered in the next section, after the data we deal with are described in a greater detail.

3.2 Neighbourhood similarity between HPFs

The data for this analysis come from studies of herpesvirus—a pathogene significantly affecting both animals and humans. A set of 30 complete herpesvirus genomes covering the so-called α, β and γ herpesvirus superfamilies, which differ by the tissue in which the virus resides, have been extracted from the herpesvirus database VIDA, release 3 [1] (see Table 1), and an evolutionary tree has been built over the genomes for the conserved DNA polymerase gene, using the neighbour-joining procedure from the PHYLIP package [11] (see Fig. 3). This tree agrees well on the set of coinciding genomes, within the acknowledged uncertainty limits, with previously published herpesvirus phylogenies [19, 20] and, moreover, all of the results reported here also hold for the other topology.

A set of 740 homologous protein families (HPFs) represented in these 30 genomes has been extracted from the VIDA database [1]. Each VIDA HPF is defined by a conserved fragment in the proteins constituting the HPF; these were computed using the XDOM software [1, 13]. In

Table 1 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

this way, each HPF is assumed 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 pointed out above, such motif-based protein family assignment can suffer both from fragmentation and from the non-assignment of proteins to a family due to lack of pairwise similarity.

To further aggregate the VIDA HPFs, we have to develop a system for scoring the similarity between them. Perhaps the most straightforward idea would be to first score similarities between proteins belonging to different HPFs and follow-up by averaging them. Another approach would dwell on the fact that VIDA HPFs 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 the set-theoretic similarity between them as 'bags' of proteins. We follow an intermediate approach by using the set-theoretic similarity, not between the VIDA HPFs themselves, but rather between their neighbourhoods defined by using the popular sequence alignment tool PSI-BLAST [3]. Given a VIDA 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 HPFs. The third step is to compute a matrix of set-similarity values between the HPF neighbourhoods for each pair of HPFs.

There are several features that encouraged us to use this approach. One of them is the issue of the accuracy of the alignment of protein sequences in scoring the similarity between them. Alignment tools, including PSI-BLAST [3], which we utilise, rely on a number of user-defined parameter values, which are usually specified by default options based on experiments. These parameter values work quite well when sequences are indeed similar. However, there is great uncertainty about the appropriate values when proteins are less similar, which is a typical situation when proteins are from different HPFs. Therefore, by limiting the use of PSI-BLAST to align only similar sequences, we avoid the uncertainty and arbitrariness of similarity estimates for 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 neighbours. 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 were initially assigned to separate HPFs within the VIDA database, namely HPFs 12 and 42 [1]. However, their sets of protein neighbours (with 20% or greater sequence identity) contain 25 and 20 sequences, respectively, and have 14 common proteins, making the overlap between the neighbourhood lists quite significant: the average relative overlap is 63% ($14/25 = 56\%$ in one set and $14/20 = 70\%$ in the other). To alleviate this difficulty, PSI-BLAST runs are conventionally iterated in order to accrue distantly related proteins. 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

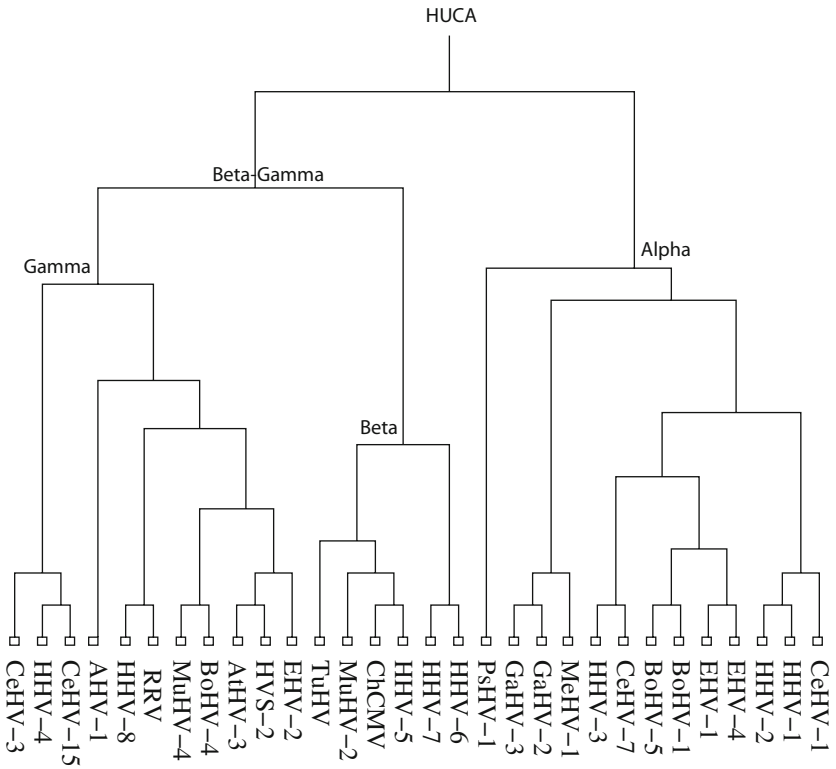

Fig. 3 Herpesvirus evolutionary tree. The root corresponds to the herpesvirus ultimate common ancestor (HUCA); its child on the right to the ancestor of α superfamily, and the child on the left, to the common ancestor of β and γ superfamilies

lists suggests that our computational strategy may be useful in overcoming this problem.

A further feature relates to the combining of 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 more distant relatives by relying on the statistical frequency profiles [3].

The idea of employing neigbourhoods to measure similarities between entities is not new. It has been used in information retrieval, originating probably from the work in [16, 34]. It has been employed in bioinformatics as well, mostly in the analysis of gene expression data (see, for example, [35]). From the perspective of clustering complex data, this approach allows for a unified framework of between-subset similarities rather than individual frameworks of specific similarity measures.

Let us describe in more detail how we compute the neighbourhoods of HPF members and combine them 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 GenBank at the NCBI Entrez web site [28]) 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 are also present in our collection of herpesvirus genome protein sequences constitutes the "homology neighbourhood" (HN) of p, denoted by $l(p)$.

To measure similarity between two HPFs, we compare their HN sets, $L1$ and $L2$, by relying 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 $n2$. The most popular similarity index is the Jaccard coefficient $J = \frac{n}{n1+n2-n}$; this reasonably takes into account all three numbers, but suffers from an intrinsic flaw by systematically underestimating the similarity [25]. In the literature, a number of symmetric versions of the most natural indexes, the relative sizes of the overlap, $\frac{n}{n1}$ and $\frac{n}{n2}$, have been proposed. However, we can take these directly to measure similarity of $L2$ to $L1$ by $\frac{n}{n1}$ and similarity of $L1$ to $L2$ by $\frac{n}{n2}$. We may obtain a symmetric measure by simply using their average, corresponding to the symmetrisation of the matrix A in the context of our clustering model, as described in Sect. 2. This measure, $mbc = \frac{1}{2}(\frac{n}{n1} + \frac{n}{n2})$, known as the Maryland Bridge coefficient, alleviates the problems related to the Jaccard coefficient [25].

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

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

Given t, $0 < t \leq 1$, 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 the $l(p_i)$ for all $p_i \in h$, and obviously the $M_1(h)$ is their intersection.

To determine an appropriate value for the 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 (we do not, at this stage, take into account the fact that the speed of evolution may be different in different parts of the tree at different times).

In the case under consideration, the majority threshold was set at the level of 20%, i.e. $t = 1/5$, based on the analysis of clusterings of VIDA HPFs produced for neighbourhoods defined at different thresholds. More specifically, we first computed, for all HPFs, t-majority lists for $t = 1/2$, $1/3$, $1/4$,.... For each t, we obtained the similarity matrix for the HPFs using the mbc index between the majority lists; then we clustered the HPFs by using the disjoint cluster version of one-by-one iterative extraction with the ADDI-S algorithm described in Sect. 2, for values of the similarity shift λ_0 between 0 and 1, at intervals of 0.05. We then analysed the similarities between the clusterings obtained for different values of t.

The reasons for choosing the majority threshold $t = 1/5$ were:

1. Given two threshold values, t_1 and t_2, we computed the median of the mbc similarity values for all pairs of clusterings, one for t_1 and one for t_2, for different values of λ_0. These medians for "neighbouring" t values were: 0.98 for $t_1 = 1/6$ and $t_2 = 1/5$; 1.00 for $t_1 = 1/5$ and $t_2 = 1/4$; 0.99, for $t_1 = 1/4$ and $t_2 = 1/3$; 0.96 for $t_1 = 1/3$ and $t_2 = 1/2$. The average mbc similarity value varied similarly, taking its maximum for $t_1 = 1/5$ and $t_2 = 1/4$. The median similarity between clusterings at non "neighbouring" t_1 and t_2 values were slightly lower. Overall, clusterings produced for the different values of λ_0 did not differ much.
2. The clustering found for $t = 1/5$ is "central" in the sense that it is more similar to the other clusterings than is the case for any of the other thresholds considered.
3. The clustering found for $t = 1/5$ is more similar than those for the other thresholds to clusterings produced by using the mbc similarities between the homology lists obtained with the iterated PSI-BLAST search [3], starting from random proteins in an HPF (The iterated 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 clusterings for $t = 1/5$ and the clusterings found for HPF neighbourhood lists was 0.91 after the first iteration; 0.82 after the second iteration; and 0.50 after the third iteration. We consider these results as supporting our view that repeated iterations of PSI-BLAST may need manual curation.).

3.3 Utilising proteome knowledge

A cluster of VIDA HPFs will be referred to as an aggregate protein family (APF). For different similarity shifts we obtain different numbers of clusters of HPFs. Specifically, for the zero similarity shift, $\lambda_0 = 0$, there are 99 non-singleton clusters. As λ_0 increases, the number of clusters rises to 107 for $\lambda_0 = 0.10$ and then eventually gradually decreases from $\lambda_0 = 0.40$ onwards, so that there are only 29 non-singleton clusters for $\lambda_0 = 0.97$. Note that this latter number corresponds to the situation when the HN sets of the clustered HPFs are practically the same; to overlap at the level of 97% or more, any two majority lists with fewer than 30 elements (this is true for almost all HPFs) must be identical.

To choose an appropriate λ_0 value, we involve substantive knowledge, independent of sequence similarity estimates, namely, knowledge of functional activities of the proteins under consideration. Each HPF is supposed to have a function (for examples of function see Table 2 below), though unfortunately the functions of most proteins available are still unknown. When the functions are known, however, we can use knowledge of which HPFs have similar functions and which do not. Two proteins are considered to have similar functions if they are consistently named between the herpesvirus genomes and/or they share the same known function. Such proteins should therefore belong to the same APF. Two proteins should not belong to the same HPF if they have different functions. Therefore, HPFs with known function should be identified to form pairs of those with clearly similar function and those whose

Table 2 Comparison of the lists of functions in the herpesvirus common ancestor between the previously determined ancestor D-HUCA [9, 20] (last two columns) and that resulting from mapping our 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 glycosylase
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	*Nucleotide repair/ metabolism*	*thymidine kinase*	UL23	Thymidine Kinase
	2				
	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; comp- lexed 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		

functions clearly differ (this may not necessarily be a straightforward exercise because authors of different submissions to protein databases tend to use different terminologies).

Sequence similarity values should be high between functionally similar sequences and should be low between sequences of proteins with different functions. The similarity shift value should therefore be chosen so that similarities between HPFs with different functions become negative after the shift, while those between functionally similar HPFs remain positive. To implement this idea, we analysed 287 available pairs of HPFs with known function and positive mbc similarity values. Among them, no pair with different functions has an mbc similarity value greater 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 functionally similar proteins. Out of the 86 such pairs available, there are 24 pairs (28%) for which their similarity value is less than 0.67. Thus for $\lambda_0 = 0.67$, 28% of the functionally similar pairs will not be identified as such, suggesting that with 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 for the set of functionally similar pairs with that for the set of functionally dissimilar pairs. As Fig. 4 shows, the graphs intersect when the *mbc* similarity value is 0.42. The number of functionally similar pairs whose similarity is less than 0.42, decreases to 11 (from 24 at $\lambda_0 = 0.67$), whereas the number of non-synonymous pairs whose similarity is higher than 0.42 increases to 7 (from 0 at $\lambda_0 = 0.67$); this yields the minimum summary error rate of 16% when $\lambda_0 = 0.42$.

Thus external knowledge of functionally similar or dissimilar pairs of HPFs supplies us with two reasonable candidates for the similarity shift value:

1. $\lambda_0 = 0.67$ to guarantee specificity so that functionally dissimilar HPFs will not be clustered together, and

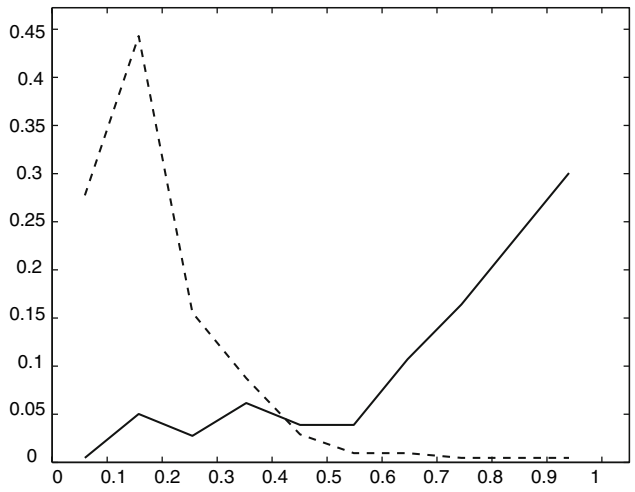

Fig. 4 Empirical percentage frequency functions (*y*-values) for the sets of functionally similar pairs (*solid line*) and pairs with different functions (*dashed line*). The *x*-values represent the *mbc* similarity

2. $\lambda_0 = 0.42$ to ensure the minimum misclassification error rate.

These two similarity shift values lead to somewhat different, but fairly compatible clusterings of the set of 740 HPFs under consideration. There are 80 APF clusters comprising 180 original HPFs, leaving 560 HPFs unclustered when $\lambda_0 = 0.67$. There are 102 APF clusters over 249 original HPFs, leaving 491 HPFs unclustered when $\lambda_0 = 0.42$. The first 80 clusters extracted when $\lambda_0 = 0.42$ correspond one-to-one to the 80 clusters obtained when $\lambda_0 = 0.67$. All 22 of the additional clusters extracted when $\lambda_0 = 0.42$ are doublets with *mbc* similarity values between 0.50 and 0.62 (implying that there is a gap in *mbc* similarity values between 0.42 and 0.50).

The aggregation found when $\lambda_0 = 0.67$ suggests $560 + 80 = 640$ APFs altogether, whereas $\lambda_0 = 0.42$ leads to a smaller total of $491 + 102 = 593$. Which one is more appropriate? Probably that which better accords with the substantive knowledge.

4 Advancing proteome knowledge

4.1 Evolutionary histories of HPFs

For the following analysis, we utilise the evolutionary histories of HPFs over the evolutionary tree. These histories have been derived using our algorithm PARS implementing the principle of maximum parsimony [24] in a rather straightforward way, because an overwhelming majority of the herpesvirus genome HPFs are consistent with the topology of the tree, each occurring in a subtree with just a few gaps.

Only for 17 of the HPFs did the PARS-reconstructed histories involve more than one gain, thus indicating possible horizontal transfers. This is obviously a very conservative estimate, as it is based only on cases of clear-cut deviation from parsimony; a recent paper [12] found a larger number of possible cases of horizontal transfer by using a different method that takes into account atypical fragments.

The reconstructed histories supply us with the reconstructed HPF contents of all the genome ancestors on the tree. Of these, currently the most useful are reconstructions of the most ancient genomes, the ancestors of superfamilies α,β and γ, as well as the more universal common ancestors, HUCA and $\beta\gamma$. This is because the properties of herpesvirus species are somewhat better understood at this level.

The multitude of reconstructed histories may provide an additional criterion for choosing an appropriate level of aggregation. This additional criterion is the consistency between the histories and substantive knowledge.

The reconstructions of the five ancestors in terms of the APFs found at the two similarity shift values, 0.42 and 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: (1) APF81 comprised of HPFs 9 and 504, both of glycoprotein C; (2) APF82 comprised of HPF 38 and HPF 736, both of glycoprotein I; and (3) APF84 comprised of HPF 47 and HPF 205, both of glycoprotein L. Unfortunately, with the current state of substantive knowledge, we cannot interpret the phenomenon of simultaneously gaining these three glycoprotein families in terms of α herpesvirus properties alone.

We can, however, examine the mutual positions of genes encoding these proteins within the circular structures of the virus genomes. We find that, in all of the 13 genomes in our data belonging to the α superfamily, the gene encoding glycoprotein E always immediately precedes that encoding glycoprotein I. This, by itself, may be considered a strong indication of the existence of some mechanism, involving both glycoproteins, that was already developed in the α ancestor. Moreover, for both $\lambda_0 = 0.67$ and $\lambda_0 = 0.42$, the APF comprised of HPF 26 and HPF 301, which both correspond to glycoprotein E, has been mapped by the PARS algorithm to the node corresponding to the α ancestor [26]. This leads us to conclude that glycoprotein I must also be present in the α ancestor. HPFs 38 and 736, both corresponding to glycoprotein I, are aggregated together as APF82 only when $\lambda_0 = 0.42$, but not when $\lambda_0 = 0.67$. However, neither HPF is mapped to the α ancestor node, whereas APF82 is. This implies that $\lambda_0 = 0.42$ is more in agreement with the knowledge gained from the reconstructed histories than is $\lambda_0 = 0.67$. Additional supporting evidence comes from the glycoprotein D

APF, comprising HPF 4 and HPF 45 for both similarity shift values, which is also mapped to the α ancestor. Moreover, the corresponding gene immediately follows that of glycoprotein I in 11 of the 13 genomes in the α superfamily (in two genomes, CeHV-7 and HHV-3, the preceding gene corresponds to protein kinase rather than glycoprotein D, which itself may lead to some speculations of possible mechanisms underlying such a substitution in the clade).

4.2 Herpesprotein ancestors

The analysis of glycoproteins in the reconstructed ancestor of the α superfamily leads us to accept the value $\lambda_0 = 0.42$ and, thus, the corresponding number of protein families, after aggregation, is 593. One can now draw structural conclusions from the mapping of the aggregate families to the evolutionary tree; some of these are presented below.

According to our reconstruction, HUCA, the common ancestor of herpesviruses, should comprise 45 HPFs aggregated to 29 APFs. These are well-studied proteins with only three of the participating families, HPFs 17, 23 and 107, of unknown function. Our HUCA is consistent with D-HUCA, the reconstracted ancestor in [9, 10], but does not include all the protein families assigned to D-HUCA, which indicates that more information needs to be taken into consideration in our computation.

Typical relations between our mapping results and D-HUCA are illustrated in Table 2. One can see that APF10 and APF3 have been mapped to HUCA, although their constituent HPFs were not. On the other hand, from the last rows of the table, we see that the glycoprotein L HPFs fail to aggregate and move from the α, β and γ ancestors into HUCA.

Let us examine these HPFs in greater detail. The three ancestors, of the α-, β-, and γ families, all contain a glycoprotein L, thus suggesting that the corresponding gene may have been present in HUCA as well. However, the corresponding HPFs, 47 (together with 205), 50 and 296, have no significant sequence similarity, and thus cannot be combined together computationally, even by using the majority lists. Yet, at the genome organisation level, illustrated in Fig. 5, each of the glycoprotein L genes immediately precedes the corresponding Uracil-DNA glycosylase gene, which was mapped to HUCA by PARS. This suggests that these are indeed common ancestral genes, just that they have undergone sequence change to such an extent that sequence similarity is no longer sufficient to assign homology. Assigning the corresponding gene UL2 to D-HUCA was based on additional experimental evidence that the glycoprotein L proteins in HPFs 47, 50 and 296 functionally complex with glycoprotein H in the α-, β-, and γ families, respectively [10] (In Fig. 5 the

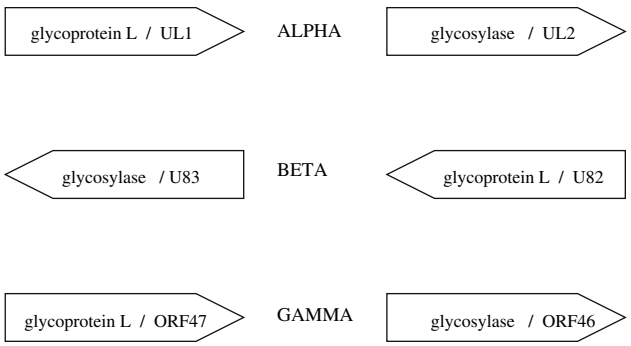

Fig. 5 Positional homology between glycoprotein L sites in the herpesvirus superfamilies α, β and γ. The homology suggests that the glycoprotein L gene co-functions with the glycosylase gene and thus the former, like the latter, should be mapped to HUCA

denotations of the submission authors are used—UL2, U83 and ORF46 are synonymous.).

Of the other four superfamily ancestors in our study, α, βγ,β and γ, according to our reconstructions, only the contents of the α superfamily have been relatively well studied. Of the 33 HPFs gained there, only 9 are of unknown function. This pattern is not repeated for the other ancestors. For only 2 of the 10 genes gained in the βγ ancestor is the function known, and similarly for only 10 out of 31 for the β-ancestor, and 9 out of 32 for the γ-ancestor. Together, at these three ancestors, βγ, β and γ, there were 73 gains of which 52, more than 70%, are of unknown function. This indicates that, so far, researchers have tended to concentrate their efforts on common features of all 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.

5 Conclusion

Clustering is an activity purported to help in enhancing knowledge of the field to which the data relate. Typically, this comes via a set of features assigned to the entities that are to be clustered; the features reflect knowledge and are to be used in interpreting results of clustering. In bioinformatics, especially proteomic studies, entities are frequently supplied only with their similarities, and are lacking sensible features to consider when interpreting the results. In such a situation, data recovery clustering supplies a reasonable device for reflecting on the substantive knowledge: the soft similarity threshold that serves as the similarity shift value. This value determines other clustering parameters such as the number of clusters. The substantive knowledge can produce two sets of pairs of entities: those that should and those that should not be

assigned to the same clusters. This may lead to significantly narrowing the choice of reasonable threshold values. We further showed that in a situation in which there is an independent interpretation device, such as the reconstruction of the evolutionary history of the protein family corresponding to a cluster, the clusters could be further aggregated by using gene arrangement data.

Using the above, we not only developed a computational approach to building HPFs, but also produced biologically relevant results, such as the hypothetical contents of the α, β and γ ancestors, including the noted link between the I, E and D glycoproteins in the α ancestor.

A possible direction for further work could be the application of similar principles for clustering and interpreting protein families in other genomic databases. Although most of the effort in clustering protein sequences goes into applying and testing the clustering algorithm against a protein-fold family database such as SCOP, this predictably does not yield good results (see, for example, [31]). The reason for this is that the sequence similarity is not enough to uncover the homology. Our approach, however, considerably narrows down the field by using only a set of related genomes. This allows us to utilise additional information such as phylogeny, functional similarity and gene arrangement. To expand this approach to other areas, one needs a clustering interpretation tool, which is not feasible unless knowledge of the proteome is appropriately structured, for example, by means of a reliable evolutionary tree. Another issue that may hinder this approach is that of functional similarity. In larger genomes, paralogous sequences belonging to the same HPF may bear different functions; this would make it more difficult to choose a separating scale shift value—but this would depend on the database: for example, all proteins of the same HPF in the COG database [38] are assigned the same function, in spite of the fact that there are paralogs among them. The possibility of systematically using gene arrangement as additional information should also be further explored.

Acknowledgments The authors thank the Wellcome Trust for its support under Grant 072831/Z/03/Z to Birkbeck, University of London. We are grateful to the anonymous reviewers for many helpful remarks and suggestions, and also to the editors for their efficient management of the reviewing process.

References

1. Alba MM, Lee D, Pearl FM, Shepherd AJ, Martin N, Orengo C, Kellam P (2001b) VIDA: A virus database system for the organisation of animal virus genome open reading frames. Nucleic Acids Res 29:133–136
2. Allaby RG, Woodwark M (2004) Phylogenetics in the bioinformatics culture of understanding. Comp Funct Genomics 5:128–146
3. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402
4. Bader GD, Hogue CWV (2003) An automated method for finding molecular complexes in large protein interaction networks. Bioinformatics 4:2. http://www.biomedcentral.com/1471-2105/4/2, doi:10.1186/1471-2105-4-2
5. Ben-Dor A, Shamir R, Yakhini Z (1990) Clustering gene expression patterns. J Comput Biol 6:281–297
6. Berman HM, Henrick K, Nakamura H (2003) Announcing the worldwide Protein Data Bank. Nat Struct Biol 10(12):980
7. Brohée S, van Helden J (2006) Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7:488. http://www.biomedcentral.com/1471-2105/7/488, doi:10.1186/1471-2105-7-488
8. Brown DP, Krishnamurty N, Sjolander K (2007) Automated protein subfamily identification and classification. PLoS Comput Biol 3(8):e160, 1526–1538
9. Davison AJ (2002) Evolution of the herpesviruses. Vet Microbiol 86:69–88
10. Davison AJ, Dargan DJ, Stow ND (2002) Fundamental and accessory systems in herpesvirus: review. Antiviral Res 56:1–11
11. Felsenstein J (2001) PHYLIP 3.6: Phylogeny Inference Package. http://evolution.genetics.washington.edu/phylip.html
12. Fu M, Deng R, Wang J, Wang X (2008) Detection and analysis of horizontal gene transfer in herpesvirus. Virus Res 131(1):65–76
13. Gouzy J, Eugene P, Greene EA, Khan D, Corpet F (1997) XDOM, a graphical tool to analyse domain arrangements in any set of protein sequences. Comput Appl Biosci 13:601–608
14. Hartigan JA (1967) Representation of similarity matrices by trees. J Am Stat Assoc 62:1140–1158
15. Holzerlandt R, Orengo C, Kellam P, Alba MM (2002) Identification of new herpesvirus gene homologs in the human genome. Genome Res 12:1739–1748
16. Jarvis RA, Patrick EA (1973) Clustering using a similarity measure based on shared nearest neighbours. IEEE Trans Comput 22:1025–1034
17. Jenner R, Mar Alba M, Boshoff C, Kellam P (2001) Kaposis sarcoma-associated herpesvirus latent and lytic gene expression as revealed by DNA arrays. J Virol 75(2):891–902
18. Kawaji H, Takenaka Y, Matsuda H (2004) Graph-based clustering for finding distant relationships in a large set of protein sequences. Bioinformatics 20(2):243–252
19. McGeoch DJ, Dolan A, Ralph AC (2000) Toward a comprehensive phylogeny for mammalian and avian herpesviruses. J Virol 74:10401–10406
20. McGeoch DJ, Rixon FJ, Davison AJ (2006) Topics in herpesvirus genomics and evolution. Virus Res 117:90–104
21. Mirkin B (1976) Analysis of categorical features. Finansy i Statistika Publishers, Moscow (In Russian)
22. Mirkin B (1987) Additive clustering and qualitative factor analysis methods for similarity matrices. J Classification 4:7–31; Erratum (1989) 6:271–272
23. Mirkin B (1996) Mathematical classification and clustering. Kluwer, Dordrecht
24. Mirkin B, Fenner T, Galperin M, 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 Evol Biol 3:2. http://www.biomedcentral.com/1471-2148/3/2/, doi:10.1186/1471-2148-3-2
25. Mirkin B, Koonin E (2003) A top-down method for building genome classification trees with linear binary hierarchies. In: Janowitz M, Lapointe J-F, McMorris F, Mirkin B, Roberts F (eds)

Bioconsensus. DIMACS Series, vol 61, AMS, Providence, pp 97–112
26. Mirkin B, Camargo R, Fenner T, Loizou G, Kellam P (2006) Aggregating homologous protein families in evolutionary reconstructions of herpesviruses. In: Ashlock D (Ed) Proceedings of the 2006 IEEE symposium on computational intelligence in bioinformatics and computational biology, Piscataway, pp 255–262
27. Montague MG, Hutchison III CA (2000) Gene content phylogeny of herepsviruses. Proc Natl Acad Sci 97(10):5334–5339
28. NCBI GenBank/Entrez web site (2006) http://www.ncbi.nlm.nih.gov/entrez
29. NCBI Viral Genome Resources (2009) http://www.ncbi.nlm.nih.gov/genomes/VIRUSES/viruses.html
30. Notredame C (2007) Recent evolutions of multiple sequence alignment algorithms. PLoS Comput Biol 3(8):e123. http://www.ploscompbiol.org/article/info, doi:10.1371/journal.pcbi.0030123
31. Paccanaro A, Casbon JA, Saqi M (2006) Spectral clustering of protein sequences. Nucleic Acids Res 34:1571–1580
32. Poptsova MS, Gogarten JP (2007) BranchClust: a phylogenetic algorithm for selecting gene families. BMC Bioinformatics 8:120. http://www.biomedcentral.com/1471-2105/8/120/additional/, doi:10.1186/1471-2105-8-120
33. Shepard RN, Arabie P (1979) Additive clustering: representation of similarities as combinations of overlapping properties. Psychol Rev 86:87–123
34. Small H (1973) Co-citation in the scientific literature: a new measure of the relationship between two documents. J Am Soc Inf Sci 24:265–269
35. Smid M, Dorssers LCJ, Jenster G (2003) Venn Mapping: clustering of heterologous microarray data based on the number of co-occurring differentially expressed genes. Bioinformatics 19(16):2065–2071
36. Snel B, Bork P, Huynen MA (2002) Genomes in flux: the evolution of archaeal and proteobacterial gene content. Genome Res 12:17–25
37. Swift S, Tucker A, Vinciotti V, Martin N, Orengo C, Liu X, Kellam P (2004) Consensus clustering and functional interpretation of gene expression data. Genome Biol 5:R94. http://genomebiology.com/2004/5/11/R94, doi:10.1186/gb-2004-5-11-r94
38. Tatusov RL, Galperin MY, Natale DA, Koonin EV (2000) The COG database: a tool for genome-scale analysis of protein function and evolution. Nucleic Acids Res 28(1):33–36
39. Thompson JD, Higgins DG, Gibson TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Res 22:4673–4680