Family relationships: should consensus reign?—consensus clustering for protein families

Macha Nikolski* and David J. Sherman
CNRS/LaBRI, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France

ABSTRACT

Motivation: Reliable identification of protein families is key to phylogenetic analysis, functional annotation and the exploration of protein function diversity in a given phylogenetic branch. As more and more complete genomes are sequenced, there is a need for powerful and reliable algorithms facilitating protein families construction.

Results: We have formulated the problem of protein families construction as an instance of consensus clustering, for which we designed a novel algorithm that is computationally efficient in practice and produces high quality results. Our algorithm uses an election method to construct consensus families from competing clustering computations.

Our consensus clustering algorithm is tailored to serve the specific needs of comparative genomics projects. First, it provides a robust means to incorporate results from different and complementary clustering methods, thus avoiding the need for an a priori choice that may introduce computational bias in the results. Second, it is suited to large-scale projects due to the practical efficiency. And third, it produces high quality results where families tend to represent groupings by biological function.

Availability: This method has been used for Génolevures project to compute protein families of Hemiascomycetous yeasts. The data are available online at http://cbi.labri.fr/Genolevures/fam/ Supplementary information: Supplementary data are available at http://cbi.labri.fr/Genolevures/fam/ Contact: macha@labri.fr

1 BACKGROUND

1.1 Overview

When confronted with computation of protein families, one must decide which algorithm to use, evaluate the quality of the result and decide which computed families—if any—correspond to real protein families. We argue that it is the comparison of different results that produces the set of the most plausible families.

Algorithmic means for establishing protein families on a large scale are based on a notion of similarity. Often the only available similarity is sequence similarity; some authors go so far as to define the notion of a protein family itself via the similarity of sequences (Doolittle, 1981). Use of structural similarities (Holm and Sander, 1996) is currently limited by the relatively small number of structures available in PDB (Berman et al., 2000), but by studying sequence variation among members of such families (Koehl and Levitt, 2002) one can guide the use of sequence similarity information.

To perform similarity-based detection of families one first selects the pairwise similarities to be used and then applies an algorithm that uses these similarities to group proteins into families. Algorithms for detection of protein families are unsupervised [see for example van Dongen (2000a)] or supervised [see for example Grundy et al., (1997) and Bejerano and Yona (2001)] learning procedures. Pairwise similarities can be provided by sequence alignments (BLAST (Altschul et al., 1990), Smith-Waterman (Smith and Waterman, 1981)) or by more sophisticated approaches using domain architecture databases (Bateman et al., 2002; Servant et al., 2002; Mulder et al., 2003). Other approaches use domain order as a fingerprint for the protein (Geer et al., 2002).

Methods that quantify similarity by using some attribute of the best BLAST hit and use single-linkage clustering are not always successful. Such straightforward approaches may group together dissimilar multidomain proteins that share a common domain (Smith and Zhang 1997), and can be fooled by promiscuous domains (Doolittle, 1995; Marcotte et al., 1999). Several graph-based methods have been proposed to overcome some of these limitations of single-linkage clustering (Matsuda et al., 1999; Enright and Ouzounis, 2000; Enright et al., 2002).

Reliability of computed clusters has been assessed by Zhang and Zhao, who studied the reliability of hierarchical clusters from gene expression data (Zhang and Zhao, 2000) and proposed a resampling algorithm for building a consensus tree, and van Dongen (van Dongen, 2000b), who introduced criteria for evaluating clustering results using a novel inter-cluster distance. It must be noted that the quality of a protein family computation cannot be truly assessed in the absence of an objective external criterion based on biological knowledge.

Here we describe a method that integrates results of multiple classifications in a single scheme, using an election algorithm. First, we formulate the problem as an instance of consensus clustering. We propose a heuristic that efficiently realizes the computation in practice. Finally, we compare our method with others and illustrate the quality of results in the case of protein families computed for Génolevures Hemiascomycetous yeasts.

1.2 Agreement between clusterings

Definition 1. A clustering \( \Pi \) is a partition of a dataset \( D = \{d_i\} \) into disjoint subsets \( \Pi_1, \ldots, \Pi_k \) called clusters, such that \( D = \bigcup_i \Pi_i \).

In what follows we will use the terms clustering and partition, and cluster and subset, interchangeably.

Let the number of elements in \( D \) and in cluster \( \Pi_i \) be \( n \) and \( n_i \) respectively. Let \( \Pi' \) be a second partition of \( D, \Pi' = \{\Pi'_1, \ldots, \Pi'_l\}, \) with cluster sizes \( n'_i \).

Most criteria for comparing clusterings are based on the confusion matrix (Kohavi and Provost, 1998), which measures the size of the intersection between the clusterings.

\[
M_{ij} = |\Pi_i \cap \Pi'_j| 
\]

Traditional measures of the proportion of agreements between two clusterings are the Rand index (Rand, 1971) and the Jaccard
As the expected value of the Rand index does not take a constant value, the adjusted Rand index (Hubert and Arabie, 1985) is preferred; it assumes the generalized hyper-geometric distribution as the model of randomness and provides an index bounded above by 1 with a constant expected value 0.

A well-known measure on the space of partitions is the equivalence mismatch coefficient $emc$ ([Mirkin, 1996] p. 241), which is precisely the Hamming distance for binary vectors if the set of all pairs $(d, a)$ is enumerated, and a partition is represented by a characteristic vector defined on this enumeration. One may interpret $emc$ in terms of edge numbers of complete graphs, where it represents the number of node moves needed to convert one partition into another.

Another category of criteria for comparing clusterings is based on set cardinality. The projection number $p$ of $\Pi$ onto $\Pi'$, defined as $p(\Pi) = \sum_i \max_j M_{ij}$, is an asymmetric index of the degree of refinement of one partition versus another. The same index normalized by $n$ is introduced in Ben-Hur and Guyon (2003) and a symmetric version is defined in Meila (2003). Laresen and Aone (Laresen and Aone, 1999) use the following asymmetric criterion:

$$L(\Pi, \Pi') = \frac{1}{k} \sum_{i} \max_j \frac{2M_{ij}}{n_i + n_j}.$$ 

Based on the projection number van Dongen introduces a distance (van Dongen, 2000a) between two partitions $\Pi$ and $\Pi'$:

$$d(\Pi, \Pi') = 2n - p(\Pi) - p(\Pi').$$ (1)

This gives the shortest distance between $\Pi$ and $\Pi'$ in a certain undirected weighted graph constructed on the set of all partitions of $D$, where two partitions are connected by an edge if one can be obtained from the other by joining two of its sets. The weight of the edge is the size of the smallest of two sets. In this construction $d(\Pi, \Pi')$ is the length of the shortest (weighted) path between $\Pi$ and $\Pi'$.

### 1.3 Consensus clustering

Combining the strengths of different clustering algorithms is the focus of research on clustering ensembles. This problem can be seen as finding the median partition with respect to given partitions.

**Consensus clustering (CC):** Given $k$ partitions, $\Pi_1', \ldots, \Pi_k'$, find a consensus partition $\Pi$ that minimizes $S = \sum_{i=1}^k d(\Pi', \Pi)$, where $d$ is a distance.

The first mathematical treatment goes back to Régnier (Régnier, 1965), where the problem was transformed into an integer problem and a branch and bound solution was proposed. The problem is proven to be NP-complete (Barthélemy and Leclerc, 1995). Wakabayashi gives an in-depth analysis of the median partition problem in Wakabayashi (1998) and concludes that approximating relations with a transitive one is NP-complete in general.

CC is NP-complete in general, yet it is not known whether it is NP-complete for any particular $k$. The case of $k = 1$ is trivial. The case of $k = 2$ is also simple since any of the partitions solves the problem optimally. Nothing is known for $k \geq 3$.

If the partition $\Pi$ of the dataset $D$, $|D| = n$ to discover is not necessarily one of the original partitions $\Pi_1, \ldots, \Pi_k$, then the size of the potential search space corresponds to the Bell numbers (Bell, 1934) or equivalently to the sum of Stirling numbers of second kind for all $m$, $1 \leq m \leq n$ [see Knuth (1992) on Stirling (1749)].

In the light of its hardness, exactly solving large instances of CC is intractable. Even so, exact methods have been implemented. Grötschel and Wakabayashi (Grötschel and Wakabayashi, 1989) give a cutting planes solution, which works well for $n$ in the hundreds.

A variety of approximations have been applied to this problem. For example Filkov (Filkov and Skiena, 2003) proposes a simple approximation that works by dramatically reducing the search-space.

**Factor-2 approximation:** Given an instance of CC, select a partition $\Pi$ among partitions $\Pi_1, \ldots, \Pi_k$ that minimizes $S = \sum_{i=1}^k d(\Pi_i, \Pi)$.

The time complexity is $O(k^2 n)$ since it takes $O(n)$ to compute the distance between any two partitions, and there are $O(k^2)$ pairs. This algorithm is a factor-2 approximation to CC.

Many heuristics based on local search have been studied in application to CC, such as simulated annealing that explores the search space by one-element moves, and a greedy algorithm (Filkov and Skiena, 2003).

The problem of finding a consensus clustering can be approached from a graph-based, combinatorial or statistical perspective. Several methods exist for discovering a consensus clustering solution from many multiple partitions: co-association methods (Fred and Jain., 2002), voting approach (Topchy et al., 2004b), information-theoretic approach (Meila, 2003), hypergraph partitioning methods (Strehl and Ghosh, 2003) and use of mixture models (Topchy et al., 2004a).

### 2 RESULTS

#### 2.1 Comparing protein families

Given a set of $m$ proteomes $P_1, \ldots, P_m$, we are interested in computing the protein families. A proteome $P = \{p_i\}$ is defined as a set of proteins. The universe of proteins $P$ is defined over the proteomes under study as $P = \bigcup P_i$. In what follows we suppose that a protein $p$ can appear only in one proteome and that it can appear only once.

As described earlier any method for protein family computation first selects the support data, then computes the families. We suppose that the filtering algorithm, $\alpha$, defines a subset $P^\alpha = \bigcup P_{i} \subseteq P$, the set of proteins over which a family computation will be performed.

**2.1.1 Distance and similarity**

**Definition 2.** A family computation $\mathcal{F}$ over $P^\alpha$ defines a partitioning of $P^\alpha$ into disjoint subsets $\mathcal{F} = \{f_1, \ldots, f_k\}$ such that $P^\alpha = \bigcup f_i$. The subsets $f_i$ are called families.

Given two family computations $\mathcal{F}^1 = \{f_1^1, \ldots, f_{k_1}^1\}$ over $P^\alpha$ and $\mathcal{F}^2 = \{f_1^2, \ldots, f_{k_2}^2\}$ over $P^\alpha$, the confusion matrix can be used to

---

1 When the cluster size is small, the Rand index tends toward 1 as the number of clusters increases.
The similarity score for FRel is defined as

\[ M_{ij} = |f_i^1 \cap f_j^1|, f_i^1 \in F^1, f_j^1 \in F^2. \]

Note that the fact that relative complements \( \partial P^1 \Delta P^2 \) and \( \partial P^2 \Delta P^1 \) may not be empty is not a problem since the only implication is that certain \( M_{ij} \) values will be 0.

The confusion matrix can be equivalently represented by a bipartite graph in a straightforward manner (Figure 1). Let \( \text{FRel} = (V, E) \), where the vertex set \( V = V^1 \cup V^2 \) is labeled by family names in \( F^1 \) and \( F^2 \), respectively, and the \( E \) is the set of edges weighted by the values in \( M_{ij} \) where \( M_{ij} > 0 \). That is, for two families \( f_i^1 \in F^1 \) and \( f_j^2 \in F^2 \) with an element in common, two vertices with labels \( f_i^1 \) and \( f_j^2 \) are created as well as an edge \( e = (f_i^1, f_j^2) \) such that \( w(e) = M_{ij} \).

Given a graph \( \text{FRel} \), its strongly connected components \( \{c_1, \ldots, c_n\} \) represent relationships between families in \( F^1 \) and \( F^2 \).

DEFINITION 3. The similarity score of a strongly connected component \( c = (V^1_i \cup V^2_i, E_c) \), where \( V^1_i \subseteq V^1, V^2_i \subseteq V^2 \) and \( E_c \subseteq E \) is

\[ s(c) = \max_{e \in E_c} w(e). \]  

The similarity score for \( \text{FRel} \) is defined as

\[ s(\text{FRel}) = \frac{1}{n} \sum_{i=1}^{n} s(c_i). \]  

This measure is symmetric and provides information on what is preserved between two family computations.

2.1.2 Making a choice The basic case we have just considered consists of two family computations \( F^1 \) and \( F^2 \). Connected components \( \{c_1, \ldots, c_m\} \) define all the local distances between families in two original partitions. Let us suppose that the comparison \( \text{FRel}^{1,2} \) has exactly \( k \) non-singletons (indices \([1..k]):\{c_1, \ldots, c_k, \ldots, c_m\} \).

Having made the comparison \( \text{FRel}^{1,2} \) how do we choose the families that will form the solution? We propose to consider solutions locally, connected component by connected component, using exclusively families in \( F^1 \) or in \( F^2 \). That is for a non-singleton component \( c = (V^1_i \cup V^2_i, E_c) \) we are only allowed to pick either \( V^1_i \) or \( V^2_i \). The rationale behind this choice strategy is that we are not allowed to invent classifications \( ab \) initio, but only to choose between the existing.

DEFINITION 4. A family choice \( F^s \) is defined by \( \{V^1_s, \ldots, V^k_s\} \cup \{V_{k+1}, \ldots, V_m\} \), where \( c \) is a vector of size \( k \) with elements taking values in \([1,2]\).

A family choice \( F^s \) defines a partition of a certain subset of the protein set \( \mathbb{P} \), namely the partition containing proteins of the families in \( F^s \). The number of family choices is equal to the number of different vectors \( c \) of size \( k \), that is \( 2^k \).

Lemma 1 indicates that all the consensus choice partitions are equivalent w.r.t. the reference set. This implies that the concrete choice is dependent on the needs of the application (see for a discussion Section 3).

2.2 Consensus for families Let us consider \( n \) family computations \( \{F^1, \ldots, F^n\} \) over their respective support sets of proteins \( \mathbb{P}^1, \ldots, \mathbb{P}^n \). We denote by \( \mathbb{P} = \bigcup F \) the set of all families.

2.2.1 Rules of the game Having made all the \( n(n-1)/2 \) possible comparisons \( \text{FRel}^{ij} \), how do we exploit this result for families computation?

As CC is NP-complete, the key to the approximation algorithm is to reduce the search-space. We propose to do so by only considering the families belonging to \( \mathbb{P} \) in the manner analogous to the discussion in Section 2.1.2.

Then any acceptable solution \( F \) is composed only of the families from \( F \) that are disjoint, that is \( \forall i \neq j, f_i \cap f_j = \emptyset \). In this manner an acceptable solution defines a partition of some subset of \( \mathbb{P} \). The set of all acceptable solutions \( \mathbb{F} \) is a subset of the powerset of \( F \), \( \mathbb{F} \subseteq 2^F \). The inclusion is strict since only empty intersections are accepted. Thus we have \(|\mathbb{F}| < |2^F| = 2^n|^F|^\).

While the size of \( \mathbb{F} \) is dramatically smaller than the whole search-space (whose size is equal to the \( n \)-th Bell number), this reduction of search-space is not sufficient and the corresponding exact algorithm remains NP-complete.

2.2.2 Definitions

DEFINITION 6. A protein \( p \in \mathbb{P} \) belongs to at most \( n \) distinct families in \( \mathbb{F} \) defined by function \( \varphi \):

\[ \varphi : \mathbb{P} \rightarrow 2^\mathbb{F}, \varphi(p) = \{f \in F^i \mid p \in f\}, |F_p| \leq n. \]

Let us now consider all pairwise comparisons \( \text{FRel}^{ij} \). A comparison graph \( \text{FRel}^{ij} \) can be decomposed into its \( m^2 \) connected components as defined in Section 2.1. A partition \( F^{ij} = \{c_1, \ldots, c_k\} \) of size \( k \leq m^2 \). The set of connected components over all of the comparison graphs is defined as

\[ \mathbb{C} = \bigcup_{i=0}^{n(n-1)/2} C^{ij} = \{c_k\}, k \leq m^2. \]

DEFINITION 7. A family \( f \) can appear in at most \( n(n-1)/2 \) distinct connected components in \( \mathbb{C} \) defined by function \( \kappa \):

\[ \kappa : F \rightarrow 2^\mathbb{C}, \kappa(f) = \{c \mid \exists f \in c s.t. p \in f\}. \]

We call the set \( C_p \), the set of \( p \)-components.

Then, for a protein \( p \) we can define by composition the set of \( p \)-components as the set of all connected components where \( p \) is a member. We will denote this function \( \sigma = \varphi \circ \kappa \).

\[ \sigma : \mathbb{P} \rightarrow 2^\mathbb{C}, \sigma(p) = \{c \mid \exists f \in c s.t. p \in f\} \]

Note that for \( C_p \), the inverse image given by \( \sigma \) induces a set of proteins \( P = \{p_f\} = \sigma^{-1}(C_p) \) such that all \( p \in P \) belong to some family \( f \) that itself belongs to a component \( c \in C_p \). We will call this set of proteins a support set for a given \( p \)-component.
2.2.3 Consensus

Let \( R \) be the set of all maximal conflict regions. A conflict region \( R \) has a support set of proteins \( P_R \) in \( \mathbb{P} \), \( P_R = \bigcup_{c \in P_c} \sigma^{-1}(c) \). \( P_R \) has \( n \) subsets (\( n \) being the number of family computations) defined by \( P'_R = \{ \varphi^{-1}(f) \mid \exists c \in R s.t. f \in c \} \), each corresponding to the proteins belonging to families of a particular family computation \( F \).

Consensus clustering over the reduced search space defined in Section 2.2.1 can be reformulated, using the definitions of Section 2.2.2, in terms of minimum set cover. The idea is to cover exactly one of the sets among \( P_R \) and \( \{ P'_c \} \) with the support sets of the connected components constituting a given conflict region. A cost function \( w \) on \( C \) can be given by the van Dongen distance [Equation (1)] or our similarity score [Equation (2)], for example.

**Minimum exact cover (MDC):** Let \( R \) be a maximal conflict region. Let \( P_R \) and \( \{ P'_c \} \) be the support set of proteins of \( R \) and the family computation-specific sets of proteins, respectively. We denote by \( P_c = \sigma^{-1}(c) \) the support set of proteins for a connected component that is included in the region \( R \). Find a subset of connected components \( C \subseteq R \) such that (1) every protein from \( \bigcup_{c \in C} P_c \) appears in \( \bigcup_{c \in C} P'_c \) and in only one of \( P'_c \), (2) there exists \( P \) among \( P_R \) and \( \{ P'_c \} \) entirely covered by the support sets in \( C \), that is \( P' = \bigcup_{c \in C} P'_c \), and (3) \( C \) minimizes \( S = \sum_c w(c), c \in C \).

The consensus family computation can then be defined as a MDC solution for all \( R \in \mathbb{R} \). This problem is a variant of minimum cover problem with an additional condition for the sets to be disjoint and is known to be NP-complete (Garey and Johnson, 1979).

2.2.4 Efficient heuristic

Two further relaxations make the problem tractable. First, we do not require to have all the \( n(n-1)/2 \) comparisons. Second, we do not require to have an exact cover. It is the second criterion that allows our heuristic to run in polynomial time. The obvious consequence of this relaxation is that not all of the proteins in \( \mathbb{P} \) will be found in the result. This is not an issue in the context of protein families since there is no reason to suppose that all proteins from \( \mathbb{P} \) belong to families of cardinality higher than 1.

Let us consider \( n \) family computations and \( m \leq n(n-1)/2 \) family comparisons \( \text{FRel}^{ij} \). Each element \( c_i, c_j \in C \) has a distance (say, the van Dongen distance) value \( d_i \) and a similarity value \( s_i \). Starting

---

*That is for any \( c_i, c_j \in C \), their intersection is empty.*
from the maximal conflict regions in R we build a conflict graph (Figure 1).

Definition 9. For family comparisons FRef \( f \) we define a conflict graph \( G = (V, E) \) as follows.

- The vertex set \( V \) is labeled by elements from \( C \), and we have \( |V| = |C| \).
- The edge set \( E \) is defined by all non-empty intersections between connected components \( c_i, c_j \in C \), that is \( e = (c_i,c_j) \in E \) if and only if \( c_i \cap c_j \neq \emptyset \).

We distinguish two disjoint classes of edges \( E = E_p \cup E_e \) those that are only based on proteins \( E_p \) and those that correspond to families \( E_e \). The latter means that there is a non-empty intersection on the level of families, that is \( \kappa^{-1}(c_i) \cap \kappa^{-1}(c_j) \neq \emptyset \), while the former is the complement \( E \cap E_p \).

By construction the set of connected components of \( G \) is exactly the set of conflict regions \( R \).

The consensus computation works by resolution of conflicts for all connected components of a conflict graph \( G \) and is given in the algorithm 1. The conflict resolution is done by a voting procedure vote.

**Algorithm 1** Family computation algorithm by weak consensus

Require: \( G = (V, E_1 \cup E_2) \) the conflict graph, a voting procedure vote
Let \( L \) be an empty list of components

for all \( C \) connected component of \( G \) do

# First, determine \( D_i \) and \( S_i \) scores for all components of this conflict region
for all \( c_i \) vertex in \( C \) do

Let \( D_i = 0, S_i = 0, n = 0 \)
for all \( (c_i,c_j) \) such that \( (c_i,c_j) \in E_2 \) do

\[ D_j = d_j, S_j = s_j, n + + \]
end for

\( D_i = D_i + D_j / n \)
\( S_i = S_i + S_j / n \)
end for

# Second, determine vote for all proteins of this conflict region
for all \( p \) protein of \( \sigma^{-1}(C) \) do

Let \( V_p \) be the set of votes for \( p \) that is equal to \( \sigma(p) \) ordered by \( D_i \) and by \( S_i \) \nend for

Let \( C_n \) be the result in order of vote(\( V_p, \ldots, V_{p_n} \)), where \( k = |\sigma^{-1}(C)| \)

# Third, compute the most consensual cover
Let \( P = \emptyset \) be the set of covered proteins
for all \( i \in C_n \) in winning order do

if \( \sigma^{-1}(c_i) \cap P = \emptyset \) then

\( P = P \cup \sigma^{-1}(c_i) \)
push \( L, c \)
end if
end for
end for

return \( L \)

3 APPLICATION AND DISCUSSION

The algorithm 1 has been used to calculate protein families for \( S\.cerevisiae \) and the four fully sequenced species of \( H\.saccharomyces \) yeasts in the context of the Généoleures project (Dujon et al., 2004; Sherman et al., 2006). The following is a summary of the key steps in this application. For a detailed presentation see the online supplement at http://cbi.labri.fr/ Genoleures/fam/method.html

Data, alignments and filtering The data consist of five proteomes of \( S\.cerevisiae, C\.glabrata, K\.lactis, D\.hansenii \) and \( Y\.lipolytica \). These proteomes taken together comprise the total set of proteins \( |P| = 29,426 \). We produced complementary alignments using Blast and Smith-Waterman filtered according to scoring criteria and an approximation to homeomorphy [see (Wu et al., 2004)]. These criteria produce four datasets of pairwise similarity: \( \beta^{SNH} \) corresponding to Smith-Waterman homeomorphic alignments, \( \beta^{SH} \) to Smith-Waterman alignments without the restriction of homeomorphy, \( \beta^{BH} \) to Blast homeomorphic alignments and \( \beta^{SNH} \) to Blast non-homeomorphic alignments.

Clustering TribeMCL software (Enright et al., 2002) was used for clustering. Three different inflation coefficients (1.2, 2.4 and 4.0) have been applied to each dataset, thus resulting in 12 different clustering results.

Parameter tuning Four parameters were added to the algorithm 1 in order to make it efficient in practice. The first three parameters help to break too large conflict regions. The last one realizes the choice policy.

(1) \( S_{max} \) is the maximal allowed size of the support protein set for conflict regions.
(2) \( f_{max} \) is the maximal number of families allowed in a conflict region.
(3) \( p \) is the percentile above which families are removed from \( R \); it is determined based on the distribution of family sizes in a given conflict region \( R \).
(4) A choice strategy (c.f. section 2.1.2). Given a connected component \( e = (V^1, V^2, E_e) \), either
determine \( V^1 \) such that \( |V^1| = \max(|V^1|, |V^2|) \) or systematically choose \( V^2 \) such that \( |V^2| = \min(|V^1|, |V^2|) \).

The voting procedure vote has been implemented as Condorcet election (de Caritat marquis de Condorcet, 1995).

Consensus families Various parameter ranges were tested resulting in 256 family computations. The final result contains 4389 protein families. Evaluation of the result was performed by comparison with manually curated literature standards, to reciprocal best hit partitions (RBH) (Rivera et al., 1998), and to Biofacet (Glemet and Codani, 1997) single-linkage clustering; see the online supplement for details. Results using our consensus method were qualitatively and quantitatively more satisfying and have been adopted by the Consortium.

The first example is provided by families GLR.2644 and GLS.60, respectively, homologs to RPL24 ribosomal-like protein 24 and homologs to ribosomal protein L30 of the large (60S) ribosomal
subunit. Instead of producing two groups of homologs, RBH clustering produces three groups corresponding to three species groups. A second example shown in the supplement illustrates the too-large families that are typically produced by blind MCL or RBH clustering, in this case an impossible family with 406 members. Our consensus algorithm splits this group into 30 families which are in fact different sets of protein kinases, annotated as ‘SNF1 protein serine/threonine kinase’ (GLC.1371, 5 members), ‘BCK1 ser/thr protein kinase’ (GLC.1809, 6 members), ‘PKC1 protein Kinase C’ (GLC.1902, 5 members), ‘cytoplasmic serine/threonine protein kinase’ (GLC.2023, 7 members), ‘protein kinase involved in morphogenesis and septin checkpoints’ (GLC.2551, 10 members), ‘PSK1 and PSK2 proteins, PAS domain containing S/T protein kinases’ (GLC.298, 7 members), etc.

These families are currently used in a variety of specific studies by the Génolevures Consortium and will be expanded as new yeast genomes are sequenced.

ACKNOWLEDGEMENTS

The present work was done as part of the the Génolevures Consortium. The authors thank all the members of the Consortium for numerous, friendly and creative discussions, and in particular Philippe Baret and André Goffeau for the discussions on the transporter families, and Hélène Ferry-Dumazet for the work on standard protein families. Support for this work was provided by the Région Aquitaine program Génotypage et Génomique comparée, the French Ministry of Research and Education through ATIP Jeunes Chercheurs, the ACI IMPBio Génolevures En Ligne and the French ANR ‘GENARISE’ project. Génolevures is supported by CNRS (GDR 2354).

Conflict of Interest: none declared.

REFERENCES