The method based on pairwise correlation method (PCM)

Here, we treat each pLSSP as a multivariate observation, or realization of a set of variables *{X(i)}*, one for each query residue, *i*. The *X(i)* takes on value 1 if residue *i* is included in a pLSSP, and 0 otherwise. The matrix element *A(k,i)* is interpreted as the *k*-th observation for residue *i*. The observed correlation *R(i,j)* between *X(i)* and *X(j)* becomes

$$R(i,j) = \frac{\sum_k A(k,i)A(k,j) - (1/n)\sum_k A(k,i)\sum_k A(k,j)}{\sqrt{\left[\sum_k A(k,i)^2 - (1/n)\left(\sum_k A(k,i)\right)^2\right]\left[\sum_k A(k,j)^2 - (1/n)\left(\sum_k A(k,j)\right)^2\right]}} \quad (16)$$

where the sums are taken over the *n* observations (pLSSPs). Since $N(i,j) = \sum_k A(k,i)A(k,j)$, and defining $N_i = \sum_k A(k,i)$ as the number of pLSSPs including residue *i*, the above equation simplifies to:

$$R(i,j) = \frac{N(i,j) - N_i N_j / n}{\sqrt{\left(N_i - N_i^2/n\right)\left(N_j - N_j^2/n\right)}}. \quad (17)$$

Note that *R* is essentially a normalized version of the *N*-matrix. Matrix *R* has the advantage of being nearly independent of the number of observed pLSSPs for this query or for particular pairs of residues. The correlation values fall in the interval [-1, 1], with positive values now considered evidence in favor of joining the particular residue pair into the same domain, and negative values considered evidence against. When N_i or N_j is zero, *R(i,j)* is undefined and arbitrarily set to zero.

The objective function or score, *Q*, summarizing the evidence for or against a particular assignment of the query protein residues into domains is

$$Q = \sum_i \sum_{j>i} M(i,j)R(i,j), \quad (18)$$

where *M* is the equivalence relation matrix defined in eq. (12) for the trial domain assignment *H*.

Note that *Q* adds up positive terms where the current domain assignment agrees with the evidence, and negative terms where that assignment is contradicted by the evidence. Unassigned residues, or residues in singleton domains, do not contribute to *Q*.

Maximizing *Q* over all possible domain assignments, *H*, is in fact a hard computational problem. A simple, "greedy" algorithm leading to an approximate solution is:

1) Initially, assign each residue to a separate domain, i.e. *H* is the identity matrix, and *Q=0* in this case.

2) Find two domains, which when joined together, maximally increases *Q*. This is the "greedy" step. Join these two domains, thereby reducing the number of columns of *H* by 1.

3) Repeat step 2) stopping before *Q* starts to decrease, or when all residues are joined into a single domain.

This greedy algorithm is essentially identical to hierarchical clustering using the average distance method for joining clusters, provided we define distance metric to be -*Q*. This violates the convention that distances must be positive, but does not affect the clustering algorithm, as implemented here. The algorithm is efficiently implemented in MATLAB.

This approach, while extremely simple, is surprisingly effective. However, some additional post-processing can improve the overall performance. We observe that this algorithm may result in domains with very small length (one or a few residues). As a final step, we simply de-assign residues from domains smaller than a cutoff of 20 residues, even if this reduces *Q* slightly. There is also no guarantee that a local trap in the maximization has been avoided, and hence that *Q* might not be completely optimized. We use a reassignment algorithm to further improve the *Q* score by testing the re-assignment of each residue, in turn, to other domains. This often has the effect of "cleaning up" the ends of domains.

PCM, as described above, displays a bias against detecting single domain proteins, especially in cases such as 1b67A, where many long pLSSPs are aligned to a small query structure. This

may stem from a subtle divergence between the statistical model and reality. Our model assumes that the observations are a random sample from the underlying multivariate distribution. In fact, however, pLSSPs for a particular query are collected only when they are aligned to at least some of the residues in the query; pLSSPs are by definition excluded when they do not align anywhere. The sample is thus censored or biased. The net effect of this bias is to reduce the effective sample size, *n*, from what it would be for a truly random sample.

A remedy for this bias is to simply inflate *n*, in cases where a strong bias is suspected, as with 1b67A. In these cases, we also note that the average fraction of the pLSSPs including each query residue, $q = \sum_{k}^{n} \sum_{i}^{m} A(k,i) \Big/ (n * m)$ is much higher than for most other proteins. Therefore, we replace *n* with a larger value, $n' = \max(n, n * q / p)$, where *p* is the typical average fraction for other structures. Here we choose to set *p* to about 0.16, close to the average *q* of the 15-chain set, for the calculations used here.

Viewing *p* as a tuning parameter, we observe that, as *p* tends to 0, *n*' tends to infinity, and the correlation matrix in (15) simplifies to

$$R(i,j)^{\infty} = \frac{N(i,j)}{\sqrt{N(i,i)N(j,j)}}, \qquad (19)$$

a scaled version of *N(i,j)* with all non-negative values. Thus, with *p=0*, the clustering algorithm would classify every protein as single domain. Appropriate choice of the parameter *p* between 0 and 1, will mitigate the sampling bias and allow for improved detection of single domain proteins.