Sequence analysis

A structure and evolution-guided Monte Carlo sequence selection strategy for multiple alignment-based analysis of proteins

I. Mihalek*, I. Res̆ and O. Lichtarge

Department of Molecular and Human Genetics, Baylor College of Medicine, One Baylor Plaza, Houston, TX 77030, USA

Received on May 18, 2005; revised on November 15, 2005; accepted on November 16, 2005
Advance Access publication November 22, 2005
Associate Editor: Keith A Crandall

ABSTRACT
Motivation: Various multiple sequence alignment-based methods have been proposed to detect functional surfaces in proteins, such as active sites or protein interfaces. The effect that the choice of sequences has on the conclusions of such analysis has seldom been discussed. In particular, no method has been discussed in terms of its ability to optimize the sequence selection for the reliable detection of functional surfaces.

Results: Here we propose, for the case of proteins with known structure, a heuristic Metropolis Monte Carlo strategy to select sequences from a large set of homologues, in order to improve detection of functional surfaces. The quantity guiding the optimization is the clustering of residues which are under increased evolutionary pressure, according to the sample of sequences under consideration. We show that we can either improve the overlap of our prediction with known functional surfaces in comparison with the sequence similarity criteria of selection or match the quality of prediction obtained through more elaborate non-structure-based-methods of sequence selection. For the purpose of demonstration we use a set of 50 homodimerizing enzymes which were co-crystallized with their substrates and cofactors.

Contact: imihalek@bcm.tmc.edu
Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

Predictions made using multiple sequence alignment-based methods are critically dependent on the input selection of sequences—on the breadth and the depth of the associated sequence similarity tree. This important fact is often left unstated, even though multiple alignments have been used since the pioneering days of the computational study of nucleotide sequences (Ouzounis and Valencia, 2003). Researchers turn to alignments to obtain (sometimes contradictory) answers to questions about enzyme active sites (Todd et al., 2001; Bartlett et al., 2002), DNA binding sites (Jones et al., 2003; Raviscioni et al., 2005), protein–protein interfaces (Grishin and Phillips, 1994; Lichtarge et al., 1996a; Elcock and McCammon, 1998; Valdar and Thornton, 2001; Fariselli et al., 2002; Caffrey et al., 2004; Bradford and Westhead, 2005) and structurally important residues (Miry and Shakhnovich, 2001; Larson et al., 2002), usually relying on similarity cutoffs (Rost, 1999; Wilson et al., 2000) to select a representative sample of sequences. Here we show that such sequence selection is often suboptimal and may be improved, in the cases when structure is known, using a Metropolis Monte Carlo (MC) type of optimization.

The question that closely motivates this work is determining functional surfaces in proteins. A generic analysis we have in mind starts with a multiple sequence alignment, selects a certain percentage of residues that appear, based on their variation pattern, to be under strong evolutionary constraints and then predicts them to be involved in the physiological functioning of the protein. Deciding which residues are the ‘topmost’ (or more generally, their relative rank) depends on two choices the investigator has to make: (1) which sequences to include in the alignment and (2) how to rank the residues. As long as the ranking reasonably captures the evolutionary behavior of residues, the former often has the stronger impact on the quality of prediction (Mihalek et al., 2006). Without any knowledge about the individual sequences (an issue relevant for automated applications), it is impossible to know in a general case which (sub)selection will suit the analysis the best. Trying them all is a problem which grows exponentially with the number of sequences; therefore, we turn to statistical sampling of the homologue space.

We suggest a MC (see, e.g. Leach, 2001) approach to select sequences for optimized prediction of functional surface in proteins with a known structure. Given a quantity that correlates well with the quality of the (functional surface) prediction, a point to which we will return shortly, we can optimize the sequence selection for that quantity. To efficiently search through the sequence space, we use the fact that similar sequences tend to cluster hierarchically, reflecting their evolutionary relatedness. If one sequence proves to be a bad contribution to the analysis, so will its close homologues. Therefore, our MC approach considers the contributions of whole subtrees at once.

As our measure of the quality of the sequence sample we choose the clustering of evolutionarily privileged residues, quantified by the selection clustering weight (Mihalek et al., 2003; see also Madabushi et al., 2002). This quantity correlates well with the overlap with the functional surface, a non-trivial finding which will be discussed in Section 3.1.

When making a blind prediction of a functional surface in the protein, we show, it is statistically advantageous to side with sequence selections which rank residues so that they cluster in a highly non-random way at all levels of evolutionary divergence. In the process we demonstrate that the clustering is a sufficiently strong cue to perform a self-guided, automatic selection of sequences for the comparative analysis appropriate for a given structure.

*To whom correspondence should be addressed.
2 METHODS

2.1 The Metropolis Monte Carlo optimization

Let \( S \) be a set of sequences \( S = \{s_1, s_2, \ldots, s_n\} \), which we will also refer to as a selection of sequences. Let \( A(S) \) be a function which assigns a (real) number to each set \( S \). We seek to optimize \( A \) in the set of all possible selections \( S \) from a superset of sequences \( \Sigma = \{s_1, s_2, \ldots, s_N\} \), by the following algorithm:

1. Build a guiding similarity tree on the initial set of sequences. Each sequence is now associated with a leaf of the tree. Also, associate with each leaf a flag with two states: ‘selected’ and ‘not selected’.
2. Make an initial selection of sequences and store it as ‘the best’.
3. Calculate \( A^{\text{init}} \) based on the initial selection of sequences.
4. Set \( A^0 = A^{\text{best}} = A^{\text{init}} \).
5. For \( i = 1, \ldots, i_{\text{max}} \):
   a. \{Pick a leaf at random, and flip its selection flag\} or \{Pick randomly a node \( n \) in the guiding tree and flip the flags in the leaves of the related subtree.\}
   b. Calculate \( A^i \) using the selected sequences.
   c. If \( A^i > A^{i-1} \), or if a randomly drawn number in the interval \([0, 1]\) is smaller than \( \exp(A^i - A^{i-1})/a \), do nothing (accept the step); otherwise flip back \{the flag in the selected leaf\}, or \{all the flags in the leaves of the subtree rooted in \( n \)\} (reject the step).
   d. If \( A^i > A^{\text{best}} \), set \( A^{\text{best}} = A^i \) and store the current selection of sequences as the best.
   e. Calculate the correlation between the current result and the previous one; if it stays high for some cutoff number of steps, output the current selection of sequences and reinitialize the simulation (steps 2–4).
6. Cluster all the output selections according to the correlation.
7. From each cluster pick a representative with the highest \( A_i \).
8. As a final selection choose the representative selection with the highest average similarity to the protein under study.

In the implementation used in this work, in step 1, the UPGMA (see e.g. Waterman, 2000) tree-building method was used.

When initializing the simulation (step 2) several strategies are possible: selecting the whole tree, a random subtree or a subtree with some prescribed average sequence similarity which contains the query sequence. Except in the scaling measurements (Section 3.3), we use the last strategy, with the prescribed average sequence similarity close to 40%. [In the scaling measurements the initial selection (step 2) encompassed the whole tree to provide an upper bound for the time requirements.]

In the examples below, \( i_{\text{max}} \), the maximum number of MC steps, is set to 10 and 100 samplings per tree node. (The comparison of results shows that \( i_{\text{max}} = 10 \) per node suffices to produce a reasonable and relevant selection of sequences; Section 3.3.).

In steps 5a and c, we suggest two alternative MC stepping strategies: picking individual leaves or picking whole tree branches at once. As it turns out, in practice there is little difference between the two, except in the cases of the largest alignments, where node-level picking does seem to increase the sampling capability of the simulation.

The constant \( a \) (in step 5c), which in simulations of physical systems plays the role of temperature, is set in such a way to optimize the acceptance to rejection ratio. In our numerical experiments we find the value of 0.01 suitable, since it keeps the ratio in the range between 1/2 and 1/5 for different proteins.

In step 5e, the correlation we consider is between the residue ranking (Section 2.3) in the current and in the previous steps. If the Spearman correlation between the ranks stays better than 0.999 for 20 steps, we reset the simulation as indicated in step 5e. This step is intended to prevent the simulation from spending too much time exploring selections of sequences which give essentially the same ranking of residues.

The ultimate goal of the sequence selection procedure, we recall, is to rank the residues in some reasonable way. The restarted simulation may, and often does, end up with sequence selections which result in an almost identical or highly correlated residue ranking. In step 8, such selections are grouped (clustered) if the resulting residue rankings are >90% correlated (using Spearman correlation coefficient). From each group of solutions, the representative resulting in the highest \( A \) is selected.

To prevent the simulation from picking a selection of remote homologues that share the same structure with the query, in step 8 we choose from the representatives (picked in step 7) the sequence selection with the highest average similarity to the query. This choice differs from choosing the absolutely highest \( A \) in no more than 10% of the cases.

2.2 Initial selection of sequences

In this work we consider two sets of sequences from two different sources. As a standard we use HSSP, a database of sequence selections/alignments obtained by carefully rethinking the similarity cutoffs for sequences of different lengths (Sander and Schneider, 1991). The HSSP alignments have the sequences already ‘selected’—they are nearly optimal according to our criteria, so as the main test we use the initial sets of sequences created using three iterations of PSI-BLAST (Altschul et al., 1997), with the default \( E \)-value cutoff, on the UniProt (2005, http://www.expasy.org/) database of proteins. The resulting sets are often too big to be aligned by a standard alignment method such as ClustalW (Thompson et al., 1994) so we resort to a hidden Markov model (HMMER, http://hmmer.wustl.edu/), profile-based alignment.

One situation that our approach cannot handle at this stage is the large number of sequence fragments which are sometimes returned by the database search. Therefore, as a preprocessing step, we remove from both sets all the sequences which are >40% shorter than the query. Also, for the sake of computational time, rather than necessity, for each pair of sequences that is 98% identical we remove the shorter one.

Throughout the simulation, as we make various random selections, the relative alignment of the sequences remains the same. Even though the method would undoubtedly profit from using a reliable method for the alignment of close homologues, such as ClustalW (Thompson et al., 1994) or T-Coffee (Notredame et al., 2000), after each \( X \) steps, currently these are orders of magnitude too slow to be used as a part of a simulation.

2.3 Residue ranking

The sequence selection quality score we ultimately use relies on the existence of a residue ranking function—a function which assigns a score to each residue, and according to which they can be sorted in the order of the presumably decreasing evolutionary pressure they experience. The algorithm described in Section 2.1 is not inherently tied to a particular method of residue ranking. Out of many methods proposed in the literature (Valdar, 2002; Soyer and Goldstein, 2004; Lichtarge et al., 1996b) we choose real-valued evolutionary trace (rET). rET is a method to rank the evolutionary importance of residues in a protein family which is based on the column variation in multiple sequence alignments (MSAs) and evolutionary information extracted from the underlying phylogenetic trees. The first step in rank calculation is to form subalignments that correspond to nodes in the tree. Information entropy is calculated for the initial MSA, and then corrected with the contributions from subalignment entropies. This subdivision of an MSA into smaller alignments reflects the tree topology, and therefore the evolutionary variation information within it. The score for a residue belonging to column \( i \) in an MSA is given by

\[
p_i = 1 + \sum_{x=1}^{n} \sum_{y=1}^{N-x} \left( -\sum_{a=1}^{20} f_{ax} \ln f_{ixy} \right),
\]
where \( f_{ia} \) is the frequency of amino acid of type \( a \) within a subalignment corresponding to group \( p \) at the level in which the sequence similarity tree is divided into \( n \) groups. Namely, the nodes (labeled by \( n \)) are assumed to be numbered in the order of increasing distance from the root, and each one of them has associated with it a division of the tree into \( n \) groups (subtrees). When \( N = 2 \) (no evolutionary information included in the form of subalignments) the expression in Equation (1) reduces to the information entropy of column \( i \) in the MSA (up to an additive factor of 1). Note that the same numerical value of \( p \) may be assigned to several residues. Further details can be found in Mihalek et al. (2004).

### 2.4 Residue clustering measure

Once the residues are ranked, we can construct a selection (or indicator) function \( S \), which assigns the value of 1 if the residue \( i \) belongs to the top fraction \( c \) of all residues in the protein. The fraction \( c \) will commonly be expressed as percentage and referred to as coverage. The actual number of different \( c \) is case dependent, and ranges from 1 (all residues share the same rank) to \( L \), the length of the peptide chain (if each residue belongs to a different rank).

Using \( S \), we define clustering weight (Mihalek et al., 2003), \( w_c \):

\[
 w_c = \sum_{i<j} S_i(i)S_j(j)A(i,j)(j-i).
\]

(2)

In this equation \( \sum_{i<j} \) stands for the sum over all different pairs of residues \((i,j)\) on the chain of length \( L \); \( A(i,j) \) is an adjacency matrix [which assigns 1 to any pair of residues \((i,j)\) which are neighbors on the three dimensional structure, and 0 otherwise]. It is worth emphasizing here that \( A(i,j) \) is the place where the protein structure enters our consideration.

The average and standard deviation for \( w_c \) in the ensemble of random residue choices can be found analytically (Mihalek et al., 2003), which allows us to determine the z-score,

\[
 z_c = \frac{w_c - \langle w_c \rangle}{\sigma_{w_c}}
\]

(3)

a quantity measuring how far \( w_c \) is from the random average, in units of standard deviation \( \sigma_{w_c} \). We then find a numerical approximation, \( A_{\text{clustering}} \), for the integral of the z-score of \( w_c \) over all coverages implied by the residue ranking:

\[
 A_{\text{clustering}} = \frac{1}{L} \sum_{i=1}^{L} z_c^i(i).
\]

(4)

The sum goes over \( L \) bins of the width \( 1/L \), with \( z_c^i(i) \) denoting \( z_c \) in the \( i \)-th bin.

(Note that since there are \( L \) residues on the chain, \( 1/L \) is the smallest fraction by which the coverage may change.) \( A_{\text{clustering}} \) can be interpreted as the area under the curve of \( z_c \) as a function of coverage (and that is why we choose to denote it as \( A \)). In the case of ranking functions which poorly distribute ranks across residues some bins will be empty—to them we assign the smaller of the values in the neighboring non-empty bins. By summing over all coverages we are rewarding residue scoring functions which rank the residues in a spatially ordered way, such that the clustering remains outstanding as the new residues are added with increasing \( c \). \( A_{\text{clustering}} \) is the quantity which is used as the optimization parameter in the MC simulation.

### 2.5 A posteriori measure of prediction quality

Our test set is chosen in such a way to enable us to find a good a posteriori estimate of the quality of prediction: the proximity to the ligands with which these proteins were co-crystallized enables us to outline the functional surface and compare our prediction with the average in the ensemble of random guesses.

The reliability of the detection of a functional surface may be expressed in terms of the overlap \( w_c \), the fraction of residues selected on the surface which, in hindsight, belongs to the functional surface of the protein. The probability of a particular overlap in the random ensemble is given by the hypergeometric distribution, commonly illustrated as the probability of selecting \( g \) ‘good’ beads from an urn containing \( G \) good and \( B \) bad ones if the total selection size is \( g \). In the case at hand, ‘good beads’ are residues known to be functionally important, and ‘bad beads’ compose the rest of the protein surface. The average value \( \langle w_c \rangle \) can easily be seen to be equal to

\[
 \langle w_c \rangle = \frac{G}{G+B}.
\]

(5)

The index \( s \) is attached to the selection size symbol \( \ell \) as a reminder that \( s \) is limited to surface residues only [in contrast to the case of Equation (2), where the selected residues can be anywhere on the protein]. The standard deviation \( \sigma_{w_c} \) can also be found in closed form (see e.g. http://mathworld.wolfram.com/HypergeometricDistribution.html), enabling us to calculate the z-score for the success rate of our prediction compared with the random guess:

\[
 z_0 = \frac{w_c - \langle w_c \rangle}{\sigma_{w_c}}.
\]

(6)

Taking each selection (to be exact, its surface) to be a prediction, we define in a way analogous to Equation (4)

\[
 A_{\text{overlap}} = \frac{1}{L} \sum_{i=1}^{L} z_c^i(i),
\]

(7)

where \( z_c^i \) now stands for the z-score of the overlap with the geometrically determined functional site, compared with the random picks on the surface, in the \( i \)-th bin.

### 2.6 Dataset

The set of homodimerizing enzymes was created manually, mostly with the help of the PDBSum database (Laskowski et al., 2005). A complex was used if all ligands were at least 50% similar to the natural ligand. Missing ligands of five atoms or smaller were ignored. We also required that the functional surface thus determined does not amount to more than one half of the total number of residues in the protein. The chosen enzymes all have a pairwise sequence identity \( \leq 25\% \). The final set consists of the proteins with the following Protein Databank (PDB; Berman et al., 2000) identifiers: 1a59, 1a35, 1a12, 1a1j, 1a9n, 1a9q, 1b0t, 1c0g, 1dam, 1d1g, 1d2x, 1e7y, 1e9c, 1g5s, 1h16, 1h7t, 1hkv, 1hrk, 1l79, 1l7j, 1l9c, 1lce, 1ker, 1l5w, 1l9w, 1lbm, 1lbe, 1lxy, 1m4n, 1m7s, 1m7p, 1m9n, 1mmu, 1nc1, 1nyw, 1o8b, 1one, 1p5t, 1qin, 1r30, 1ump, 1vtk, 1w1u, 1xp1, 2bif, 2dor and 6gst. (More information about individual proteins can be found in the Supplementary material).

### 2.7 Restriction to homodimerizing enzymes

As will be discussed below, the cluster scoring approach does not seem to be capable of distinguishing homologues oligomerizing in different ways. Therefore, as a part of the testing procedure we restrict the initial set of sequences to the ones containing the word ‘homodimer’ in the correspondingUniProt entry, and to their closest homologues. In order to do that, we traverse the guiding tree from leaves up, assigning to each internal node the label ‘unknown’ if both of its children are labeled ‘unknown’, ‘homo/hetero-N-mer’ if at least one of its children is labeled ‘homo/hetero-N-mer’ while the other one may be ‘unknown’, and ‘mixed’ otherwise. In the ensuing top-down traversal, the subtrees labeled ‘unknown’, ‘heterodimer’ or ‘homo/hetero-N-mer’ where \( N \) is not 2 are cut out from the tree.

### 3 RESULTS AND DISCUSSION

#### 3.1 Correlation between \( A_{\text{clustering}} \) and \( A_{\text{overlap}} \)

For our proposed method to work, it needs to be established, first and foremost, that the quantities \( A_{\text{clustering}} \) and \( A_{\text{overlap}} \) are related. It can be demonstrated (7) that this is so at least in the case of homodimerizing enzymes. In Figure 1 we show, through repeated random sampling of sequences from the HSSP alignment with removed
fragments, that the pairs ($A_{clustering}$, $A_{overlap}$) are correlated with correlation coefficients better than 0.5 in the majority of cases. This correlation range, as shown below, is sufficient to improve the input selection, without major degradation of the result in a statistically inevitable number of failures. The MC simulation itself would in principle work with any scoring function which assigns a number proportional to the quality of a sequence selection (which we propose to measure independently using $A_{overlap}$).

Furthermore, in a good selection of sequences, represented here by the HSSP selection minus the fragments, both $A_{clustering}$ and $A_{overlap}$ are expected to be high. For the HSSP selections of homologues for our test set the distributions of $A$s are shown in Figure 2. All distributions ($A_{clustering}$ and $A_{overlap}$ for the interface and for the catalytic site) are convincingly shifted to the right, as expected.

Knowing that $A_{overlap}$ and $A_{clustering}$ are correlated, and that both are large in a well chosen set of sequences, we set out to improve $A_{overlap}$ by improving $A_{clustering}$ through selection of the input sequence set.

3.3 Non-uniqueness of the selected sequence set

It is important to keep in mind that the scoring function we are using ($A_{clustering}$) is not directly aware of the selected sequences, but rather of the resulting residue ranking only. So, to pick an illustrative example, in the case of enolase from *Saccharomyces cerevisiae* (PDB identifier 1one), the MC algorithm reports two selections outstanding in their $A_{clustering}$: 229 sequences belonging mostly to taxonomical groups of Archaea, Firmicutes, Alphaproteobacteria, Gammaproteobacteria and Fungi, representing respectively 8, 16, 5, 11 and 11% of the alignment, and 184 sequences where the breakdown of the largest taxonomical groups follows the ratio 13, 16, 9, 10 and 12% (the other groups, mostly bacterial, contribute <5% of the alignment each). The two selections result in two residue rankings which are 96% correlated according to the Spearman criterion. They increase $A_{clustering}$ from the initial 5.1 for the UniProt/Psiblast selection to 9.7 and 9.9, respectively, increasing the overlap function $A_{overlap}$ with the homodimerizing interface from 1.3 to 4.2 and 3.4, while slightly degrading the overlap with the catalytic site from 5.2 to 4.6 and 4.5, respectively. From the practical standpoint, the two solutions are indistinguishable.

As a consequence, the problem has possibly very many closely related solutions, which are not the results of identical sequence selections. Looking at it in that light, the goal of the method is to get rid of the many inappropriate sequence selections rather than to pinpoint the single optimal one. Multiple solutions necessarily appear, for the minimal reason that each sequence may be replaceable by one or more of its close homologues, at least within the resolution provided by the $A_{overlap}$ and $A_{clustering}$ correlation.

3.3 Performance properties of the method

3.3.1 The choice of the MC step

Because the available sequence data are heavily geared toward a limited number of model organisms, a search in a database will, most often, result in uneven sampling of branches of the evolutionary tree. This motivates our design of the MC sampling of the guiding tree, using node selection in the algorithm step 5a, and results in faster convergence and better sampling compared with the straightforward random picking of sequences (leaves), at least in the case of large alignments. We illustrate the point on the first 100 steps in the MC simulation for the protein 1a59, alphabetically first in our test set (Fig. 3). The node picking strategy detects the rough position of the local optimum more quickly, after which point both strategies work comparably well.

3.3.2 The optimal number of MC steps

Since the problem possesses multiple local minima (Section 3.2), the algorithm may in principle find a different sequence selection at each restart, even though the resulting ranking of residues is highly correlated between several runs. (An alternative possibility would be to try and traverse the sequence space with some higher ‘effective temperature’ parameter $a$, and settle for the absolutely best selection found along the way. We choose not to do so, because the simulation which lingers around single optimum indicates its relative stability.) The question then arises of whether there are still better local optima that remain unexplored. The MC approach does not guarantee finding the global optimum, and different tree topologies in each individual case do
not yield to a straightforward generalization of the search process. We note, though, that in the majority of cases in our test set after 10 MC steps per node of the guiding tree we find a sequence selection which results in at least 90% of the maximal $A_{\text{clustering}}$ found in performing 100 MC steps per node (Fig. 4). The inset in the same figure indicates that the necessary number of steps depends weakly or perhaps not at all on the alignment length. This is not surprising since the number of steps depends prominently on the input sampler and the underlying tree topology, while the alignment length appears only as a part of the scoring function.

### 3.3.3 Scaling with the number of sequences

The Metropolis MC search through the sequence space scales as $O(N^3)$ with the number of sequences $N$, as expected from the triple sum in Equation (1) (note that the probability of finding all amino acid types in the innermost sum also grows proportionally to $N$). The actual polynomial constants, however, change non-trivially on a case-by-case basis. To illustrate the point, we consider the example of two superficially very similar cases of acetylglutamate kinase (PDB identifier 1gs5) and methylenetetrahydrofolate dehydrogenase (1digA). In the case of the former the initial alignment is 258 positions long and consists of 668 sequences with the average identity of 20%, while the numbers for the latter alignment were: 285 positions, 767 sequences and the average identity of 16%. Judging by the statistical properties of the initial sequence samplers, one might naively expect that 1gs5 will be handled more expediently; however, this turns out not to be the case. From each alignment we draw at random a subset of sequences and measure the CPU time necessary to go through 500 MC steps. The results are shown in the upper panel of Figure 5. While both simulations scale approximately cubically with the input number of sequences, the increase is much steeper for 1gs5. The difference in behavior can be traced back to the typical (or average) number of sequences that the simulation handles at each step (lower panel of Figure 5). This can, in turn, be related to the ‘appropriateness’ of the initial set as the whole: the $A_{\text{clustering}}$ in the case of 1gs5 is 5 to begin with, while in the case of 1dig it is −1, indicating that the pool of ‘good’ sequences is much larger for 1gs5, and more of them are handled at once in each MC step—the effective size of the problem is larger in this case.

### 3.3.4 Scaling with the length of the protein chain

The system behaves very nearly as $O(L^2)$ in the length of the alignment $L$, in agreement with the double sum in Equation (2), and is shown in Figure 6 for the example of maltodextrin phosphorilase, PDB identifier 153w. The quadratic dependence is thus related to our choice of structure-dependent scoring function $w_c$.

### 3.4 Selecting sequences toward functional surface prediction

The importance of our approach lies in enabling us to increase the quality of a blind prediction of the functional parts of a protein.
surface through sequence selection, without explicitly relying on sequence identity/similarity arguments. Rather, we incorporate the information about the protein structure and about residue variability into a single scoring function [Equation (2)] which we then optimize by choosing sequences from a broad set of (possibly distant) homologues.

To illustrate the capabilities of the method, we divide the functional surface into two groups: catalytic sites and homodimer interfaces. For the catalytic site cases, we compare the prediction after 10 and 100 MC steps per node (of the guiding tree; Fig. 7). The proteins from our test set are sorted in the order of increasing initial $A_{overlap}$ to emphasize that the method proves most useful when the initial selection includes many poorly related sequences that obscure the existence of evolutionary pressure on the functional surface. The two distributions in Figure 7 (arrow tails and arrow heads; before and after the MC optimization) are demonstrably not the same: they are different with the Kolmogorov–Smirnov (Press et al., 1992) $p$-value of less than one half percent (i.e. the probability of the two sets of numbers to be fair samples from the same distribution is $<0.5\%$).

In Figure 7 we also show the comparison between the improvement in catalytic site predictions by the MC optimized PsiBlast and by HSSP selections. We notice that the MC and HSSP sequence selections result in predictions of very similar quality. While the HSSP retains sequences which are estimated as homologous through statistically adjusted similarity cutoff (Sander and Schneider, 1991), our approach does not rely on the sequence similarity directly, but rather uses the clustering on the structure of residues under evolutionary pressure as a quantity guiding the optimization.

The problem turns out to be less tractable in the case of homodimerizing interfaces. The results for the same MC selections shown in Figure 7 are shown in Figure 8, the $y$-axis now showing the change in the interface detection with respect to the initial PsiBlast selection. The selections which improve the clustering obviously do not always work well toward detection of the interface. The reason can be traced back to the oligomer composition of the initial sequence selection: even though the queries are all homodimers, their close homologues may be monomers, tetramers, or heterodimers, for example, thus not preserving the same interface. This does not necessarily affect the clustering of the top scoring residues. After restricting the initial selection to sequences annotated in the UniProt database as homodimers and to their closest homologues (Section 2.7), we obtain the results shown in Figure 9. The largest part of the improvement now comes from the restriction to homodimers (dashed arrows). The change in prediction after the MC optimization is indicated as full line arrows in the same figure. While the MC optimization does not produce a distribution of predictions significantly different from the one obtained by restriction to homodimers, it is still helpful in the cases of bad (or non-existent) initial overlap. The approach can thus be used as a prefilter when looking for the oligomerization interfaces, since it can...
detect potentially very bad cases without significantly degrading prediction in the rest.

What is the meaning of the sequence selections that the algorithm ends up with? As one would hope, the content of the set is typically shifted toward orthologues and taxonomical relatives of the query protein. To quote two easily interpretable examples (see Supplementary material), the selection for protein urocanate hydratase (1w1u) shifts toward Gammaproteobacteria, the group that the crystallized protein belongs to, and becomes almost completely recognizable as the set of orthologous proteins; in the pro-

In Figure 10 we show the higher proficiency of the MC optimization than the specialization for a particular oligomerization type.

We have noted in step 8 in the algorithm layout that the simulation may end up with a selection of sequences which are mutually highly homologous but poorly related to the query. This observation—that there might be several protein families for which the highest ranking residues form statistically significant but not entirely overlapping clusters—leads to the possibility of difference analysis of ET data, an idea already explored in Madabushi et al. (2004).

Beyond doubt, the sequence selection strategy serves in part to compensate for the shortcomings of the residue scoring method. What makes this discussion interesting, however, is the fact that we can pinpoint the relevant sequences by considering the physical placement of the residues which the selection highlights as relevant for the protein. The hope is that, eventually, the careful physical analysis of the protein might reveal that the variability of a residue depends on the depth of the free energy well it sits in, within a monomer or as a part of the complex, and that properly selected homologues indeed reflect those constraints. (Notably, the HSSP alignments which we use as the gold standard, and which were selected through an independent line of reasoning, indeed show high degree of clustering of slowly varying residues, Fig. 1.) The use of clustering as a guide in sequence selection thus amounts to reverse engineering implications of physical constraints within the folded protein.

In a case without other reliable experimental cues, we conclude, it is statistically advantageous and computationally feasible to look for a selection of sequences which optimizes the clustering score for highly ranked residues.

In the process we learn that selecting sequences by the clustering criterion might turn out distinctly disadvantageous in trying to detect a protein–protein interface, if the initial sample contains sequences with arbitrary oligomerization properties (Figures 8 and 9). Clustering of the core residues in a protein family with a specified fold goes much further back in the evolutionary history than the specialization for a particular oligomerization type.

We have demonstrated that it is possible to find the optimization function, formally unrelated to the a posteriori measure of prediction quality. We have used it in a MC strategy to optimize, in a self-guided way, sequence selection in multiple alignments toward detection of functional surfaces in proteins. Thus, finally, we hope we have motivated further research into methods taking
sequence selection a step further beyond simple similarity arguments.

ACKNOWLEDGEMENTS

This work was supported by grants from the National Institute of Health (GM066099), the National Science Foundation (DBI #0318415) and the March of Dimes (MOD FY03-93).

Conflict of Interest: none declared.

REFERENCES


