Obtaining statistical evidence for underlying protein molecular mechanisms

Andrew F. Neuwald
1. Overview

In the *Novum Organum* [2] Francis Bacon described, as the first step in applying the scientific method, the compilation of observational data, followed by the categorization of these observations and the generation of hypotheses. Our tools facilitate this process by extensively characterizing protein sequence and structural constraints leading to plausible hypotheses. From this may then follow the accumulation of additional empirical results through further experimentation with the goal of better understanding how protein molecular machines work at the atomic level. Our core hypothesis is that characterizing the most statistically significant constraints will reveal otherwise overlooked protein properties responsible for underlying molecular mechanisms. Because we seek to identify properties that biochemical and structural studies have thus far failed to identify, we allow the data itself to reveal its most statistically surprising features without making assumptions about what should be found. We argue that, in the absence of relevant biochemical studies, it is only possible to directly link individual residues to other residues and such residue sets to structural features. Hence, our approach focuses on these (observed) properties rather than on predicting (unobserved) biochemical properties. Augmenting a knowledge of biochemical and structural properties with visualization of functionally imposed constraints in this way can lead to plausible hypotheses for experimental design.

Our tools are applied to protein superfamilies that have diverged into subgroups, each of which fills a functional niche compatible with the superfamily’s common “core” structure. Within each subgroup, proteins from distinct phyla often conserve residues at (non-active site) sequence positions that other related proteins fail to conserve. Often a subgroup, $G$, is composed of smaller subgroups, each of which conserves both residues due to $G$’s functional constraints and other residues due to constraints imposed by its own, more specialized function. Repeated rounds of this evolutionary process have led to hierarchically interrelated patterns of correlated residues and, in some cases, to networks of functionally critical residues embedded within structurally defined clusters. Moreover, for proteins sharing a common core structure, 3D contacts between pairs of (often non-conserved) residues generally produce correlated substitution patterns: Over evolutionary time substitutions at one residue position often result in compensating substitutions at other positions to maintain critical interactions. This leads to a correspondence between covarying residue pairs and 3D contacts. Hence, our tools can characterize statistically significant constraints appearing as residues co-conserved in functionally related subgroups, as subtle pairwise correlations, and as correlations among these sequence features or with structural features. These can also be applied, of course, to proteins that functionally interact with proteins of primary interest.

A fundamental question in biology is: How do proteins sharing a common structural core perform entirely different functions? For example, AAA+ ATPases mediate diverse cellular activities, including membrane fusion, DNA replication, microtubule dynamics, intracellular transport, transcriptional activation, protein refolding or degradation, and the assembly and disassembly of protein complexes. By identifying diverse types of statistically significant constraints within sequence and structural data, our approach both addresses this question and obtains evidence for how such molecular machines work. This can suggest plausible hypotheses regarding the roles of various categories of residues responsible for functional specificity. Such residues may be remote from an enzyme’s active site and may often mediate function through dynamic interactions with each other or with specific cofactors and substrates. For this reason, it can be helpful to apply our tools in conjunction with molecular dynamics simulations. Characterizing protein properties in this way may help guide protein engineering efforts, provide insights into the molecular basis of human disease, identify potential antibiotic target sites in bacterial proteins, and aid the design of highly target-specific drugs.

This document describes a suite of programs for identifying determinants of protein functional specificity that were developed in collaboration with statisticians Stephen F. Altschul (NCBI, NIH)(Gibbs sampler, GISMO, BPPS, STARC, SIPRIS, DARC, cCOMPASS), Jun S. Liu (Harvard)(Gibbs sampler, GISMO, BPPS), and Charles E. Lawrence (Brown University)(Gibbs sampler).
2. Statistics and probability in biology

Due to the inherent variability of biological systems, a protein, such as the tumor suppressor p53 shown above, does not correspond to a specific sequence, but instead must be defined probabilistically as an evolutionary ensemble of sequences sharing a common cellular function. This ensemble can be modeled statistically as a high dimensional probability distribution over every possible sequence. A generative statistical model (e.g., a hidden Markov model or HMM) can ‘emit’ sequences with probabilities defined by such a distribution. HMMs are described in section 3 below.

**Analogy between our approach and classical genetics.** Classical geneticists obtained evidence for pairs of linearly ordered genes based on patterns of inherited traits in the absence of any direct cytological or molecular data. Likewise, our approach obtains evidence for underlying protein mechanisms based on patterns within sequence and structural data. Just as correlations associated with inherited traits are due to each gene’s chromosomal location, correlations associated with protein sequence and structure are due to each residue’s biochemical function. Just as hypotheses generated by early genetic analyses have been validated by cytological and genomic studies, our goal is to generate plausible hypotheses that can be validated by biochemical and biophysical studies. However, just as genetic analysis works well for some organisms (e.g., peas), but not for others (e.g., hawkweed), there are protein superfamilies for which our approach works much better than for others. Therefore, it is best to focus on those superfamilies most amenable to analysis, of which there are many.

**Frequentist vs Bayesian statistics.** We utilize both frequentist and Bayesian statistics. A frequentist assigns probabilities to data, not to hypotheses. Hence, the frequentist looks for the probability of observing the data given a model (termed the null hypothesis). A frequentist often focuses on p-values. A p-value (green shaded area in the figure on the right) is the probability of an observed or more extreme result assuming that the null hypothesis is true.

In contrast, a Bayesian assigns probabilities to hypotheses, arguing that all that we know for certain is the observed data; we can’t observe the underlying model generating the data. Bayesian statistics incorporate prior probabilities and updates model probabilities as more data become available. Prior probabilities provide a way to incorporate either general knowledge about the model or else no information at all, the latter being termed a non-informed prior where each outcome is treated as equally likely a priori.
Neuwald 3

Note that a Bayesian approach tells us what can reasonably be inferred given the available data. For example, consider a simple experiment that probabilistically predicts a coin’s true property based on the observed data (i.e., the coin flips). Such a probabilistic (or ‘blurry’) description most accurately represents what the data is telling us. Without such statistical criteria, one can easily be misled either by being too cautious or by not being careful enough.

**Bayesian version of the scientific method.** Our tools are largely based on Bayesian Markov chain Monte Carlo (MCMC) sampling, an approach that considers multiple alternative models concurrently: Instead of a single hypothesis regarding a specific model (termed \( M_i \)) being consistent or inconsistent with the outcome of a particular experiment, every possible model is assigned a specific probability of being correct given many empirical observations (termed \( D \)), which in our case corresponds to sequence and structural data. Typically, one cannot compute probabilities for every model, so Bayesian procedures rely—as does the scientific method itself—on iterating between hypothesis testing and model refinement using MCMC sampling. In each iterative step, \( n_x \) alternative hypotheses regarding a particular modeled property, \( x \), are evaluated (while other properties are held constant) by computing model probabilities over the full range of possible hypotheses for that property (i.e., \( P(M_j | D) \) for \( 0 < j \leq n_x \)). That property of the model is then updated according to the outcome of these ‘experiments’ (i.e., proportional to the computed probabilities for each hypothesis). MCMC sampling iterates through all model properties in this way until convergence on the most probable models. When the underlying probability distribution is multimodal, as in the figure shown, the sampler can be run from multiple starting points to find a near global optimum.
3. Multiple Sequence Alignment (the Gibbs sampler and GISMO)

**Multiple sequence alignment (MSA) via MCMC sampling.** The following describes how the Gibbs sampler identifies and aligns ungapped conserved regions (termed ‘blocks’) within protein sequences. The sampler starts by arbitrarily selecting a candidate site in each sequence; these will of course be very unlikely to correspond to conserved regions. Next, it iteratively realigns each site proportional to its similarity with the other aligned blocks, which, over time, favors the alignment of conserved over unconserved regions. Eventually the sampler converges on alignment of the conserved regions; this typically requires application of simulated annealing where the sampling “temperature” is gradually lowered (i.e., by distorting the sampling probabilities to favor higher probability sites and disfavor lower probability sites more strongly).

The Gibbs sampler is particularly useful for detecting subtly conserved repeats, as illustrated by our discovery of β-propeller-like repeats in UV-damaged DNA-binding protein [3], and of HEAT repeats in certain chromosome condensation components [4]. The β-propeller prediction was later confirmed through crystal structure analysis [5]. The HEAT-repeat prediction led to our discovery (based on the presence of similar HEAT repeats in hypothetical proteins) of a new vertebrate condensin component [6]. To detect subtly conserved blocks in a set of protein sequences, however, it is important to eliminate closely related sequences, which would introduce very strong signals to mislead the sampler. This is done either using our purge program [7] or using the (heuristic) cd-hit program (by Weizhong Li) [8], both of which take as input a set of fasta formatted protein sequences.

How Gibbs sampling works is analogous to how crystallization occurs based on statistical thermodynamics. Starting from an arbitrary ‘state’ a thermodynamic ‘system’ evolves due to internal ‘forces’ and random fluctuations. The attractive forces gradually bring similar ‘objects’ together. So, just as some molecules begin to aggregate, some sequences are aligned more or less correctly. Eventually both ‘systems’ reach equilibrium and given an appropriate temperature the molecules crystallize out of solution and the conserved regions are aligned.

**MCMC sampling and assignment uncertainty.** An additional advantage of MCMC sampling is that it can provide a measure of model uncertainty as illustrated by the flipped coin example above. This involves continuing to sample among alternative models after convergence, in which case uncertain aligned blocks will tend to change but more certain aligned blocks will tend to be retained. The frequency with which states are sampled estimates the probability of that prediction being correct. For example, the figure on the left highlights in red and orange those β-strands in bacterial porins that the Gibbs sampler assigned predictive probabilities ≥ 0.75 and ≥ 0.25, respectively. These porins form homotrimers, arranged as shown in panel B, with conserved repeats located at non-interacting surface regions.
**GISMO** (Gibbs Sampler for Multi-alignment Optimization) uses MCMC sampling to search for a protein hidden Markov Model (HMM) that is most likely to have generated a set of input sequences [9,10]. In the process, it multiply aligns the sequences while also inferring position specific gap penalties. It tends to align sequences as conserved regions and can easily span across large insert regions without fragmenting the rest of the alignment, as the figure on the right illustrates. Unlike most MSA programs, GISMO will not align unrelated (e.g., randomly generated) sequences (see below).

An HMM is an abstract statistical model that probabilistically generates strings of characters in an alphabet, which for protein sequences consists of the 20 amino acid residues. The architecture of a GISMO HMM is shown on the right. It consists of 5 types of states: a start (S) and an end (E) state, and delete (D), insert (I) and match (M) states; and of transition arrows between states. Transition arrows have assigned probabilities such that the sum of the probabilities out of each state sum to 1. An HMM generates a sequence by stochastically following a path through the HMM: Beginning in the S state, subsequent states are sampled based on the transition probabilities; upon transition into an insertion (I) or a match (M) state, the HMM emits a character in the alphabet (i.e., an amino acid residue) based on the emission probabilities assigned to that state. This process is terminated when the end state (E) is reached with the output being the sequence of residues emitted.

There are three algorithmic operations associated with such an HMM: (1) An arbitrary sequence is scored against an HMM as the probability of the HMM having generated that sequence. This requires summing probabilities over all possible paths through the HMM that would have generated that sequence; this is done using the forward algorithm, which is similar to the Smith-Waterman (SW) algorithm[11]. (2) A given sequence is optimally aligned against an HMM by finding the path through the HMM that is most likely to have generated that sequence. This is done using the Viterbi algorithm, which is also similar to the SW algorithm. (3) No tractable exact algorithm is known to find the HMM transition and emission probabilities most likely to have generated a given set of input sequences—in other words, to derive the maximum likelihood estimate of the parameters of the HMM given a dataset of sequences. However, one can use Markov chain Monte Carlo sampling to obtain a near optimal solution.

GISMO uses MCMC sampling in this way to search for an optimal HMM and an associated optimal MSA by probabilistically sampling HMM architectures and corresponding parameter settings based on a given set of input sequences. This process begins by arbitrarily aligning the sequences and then using this alignment to parameterize an initial HMM, termed $M_0$. An HMM is parameterized by estimating transition and emission probabilities based on the observed residue frequencies in the MSA, where each aligned column contains those residues emitted upon transition into a corresponding match (M) state or, for gaps, to transitions into a corresponding delete (D) state and where each insertion corresponds to transitions into a corresponding insert (I) state. Next, GISMO iteratively samples an MSA based on the HMM and then re-parameterizes the evolving HMM based on the sampled MSA. Specifically, in the $i^{th}$ iteration GISMO computes the probability of each of $n_e$ alternative (slightly different) HMMs having generated the input sequence.
sequence data $D$ (i.e., it computes $P(M_{i,j} | D)$ for $0 < j \leq n_x$); it then samples one $M_{i,j}$ (and a corresponding MSA) proportional to these probabilities. Finally, it stops upon convergence on a high probability (ideally, nearly optimal) HMM (and MSA).

On average, GISMO aligned 408 benchmark sequence sets more accurately than did six of the most popular MSA programs (panel A in Fig. 3.1), especially for the most difficult to align sequence sets. Unlike these other programs, which are deterministic and thus return the same MSA for each run, GISMO is stochastic (panel B). However, the program-to-program variability (panel C) is greater than the run-to-run variation for GISMO (panel B).

**Figure 3.1.** Alignment quality among various MSA programs based on their SP-scores, which vary from 0 (no correctly aligned sequence pairs) to 1 (all pairs aligned correctly). **A.** The sorted (lowest to highest) SP-scores obtained by six MSA programs. **B.** Run-to-run variability in SP-scores over six GISMO runs. Test set data points are sorted along the x-axis by the SP-score obtained for each set on the first run (red data points) out of six. **C.** SP-scores for the six programs analyzed, sorted by the GISMO score on each test set. GISMO SP-scores (for a single run) are shown in red. Each red data point and the five black data points (one point for each program) plotted in the same column correspond to the same test sequence set.

**Version 2 of GISMO** can run a user-specified number of threads in parallel, which allows for a more thorough exploration of the posterior probability distribution over possible HMMs. Rather than speeding up the program, this is designed to improve MSA accuracy by improving its ability to find a more nearly optimal MSA. **Appendix 1** provides mathematical details regarding GISMO. The GAMBIT program will attempt to further improve a GISMO-generated MSA by continuing to apply MCMC sampling, which can also provide a measure of alignment uncertainty. The eCOMPASS (evaluative Comparison of Multiple Protein Alignments by Statistical Score) program [12] evaluates the relative quality of two alignments of the same sequences based on direct coupling analysis (DCA), which is described below.

**Example.** An early MCMC sampler implemented in conjunction with a database search procedure identified very subtly conserved motifs within sequences that included lysophospholipid acyltransferases. The MSA obtained in this way is shown here:

<table>
<thead>
<tr>
<th>start</th>
<th>motif A</th>
<th>motif B</th>
<th>motif C</th>
<th>motif D</th>
<th>motif E</th>
<th>end</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>220</td>
<td>Ph.B (Pf40)</td>
</tr>
<tr>
<td>1</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>220</td>
<td>Ph.B (Pf40)</td>
</tr>
<tr>
<td>2</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>220</td>
<td>Ph.B (Pf40)</td>
</tr>
<tr>
<td>3</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>220</td>
<td>Ph.B (Pf40)</td>
</tr>
<tr>
<td>4</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>AKCVRRNLGRLTGGHLDYRA</td>
<td>220</td>
<td>Ph.B (Pf40)</td>
</tr>
</tbody>
</table>

Among the sequences found was tafazzin, which is associated with Barth syndrome, an inherited cardiomyopathic disorder in children. This predicted that Barth syndrome was due to an acyltransferase deficiency, which then was clinically confirmed (see J Pediatr. 2002. 141(5): 729-733), and which later led to potential treatments for this disease (see J Lipid Res. 2003. 44(3): 560-566), including the first-ever
approved treatment for Barth syndrome: Elamipretide, a small mitochondrially-targeted tetrapeptide that appears to reduce the production of toxic reactive oxygen species, thereby stabilizing cardiolipin. Abnormal cardiolipin is a specific diagnostic marker of cardiomyopathies caused by Barth syndrome mutations leading to alterations in the fatty acid composition of several phospholipids. The latest version of Gismo creates a much better alignment of these proteins, as illustrated here:

Statistical significance. An important feature of Gismo, which most other MSA programs lack, is that it will not align sequences lacking statistically significant sequence similarity—though it will align very subtly conserved regions that might appear to lack significant similarity. Such subtle similarity is important for detecting distant relationships that reflect functional and/or mechanistic properties. A MAFFT alignment of randomly shuffled (PH domain) sequences, which someone might misinterpret as being biologically relevant, but that Gismo will not align, is shown here:
4. Protein sequence divergence and functional specificity

Proteins sharing a common structural core and, in the case of many enzymes, catalytic residues a well, typically have diverged into subgroups involve in distinct cellular functions. In some cases, these functions are very different, as is illustrated in Fig. 4.1 for soluble chloroperoxidase (CPO) and integral membrane associated type 2 phosphatidic acid phosphatase (PAP2). CPO, which catalyzes the production of hypochlorous acid (i.e., bleach), is produced by a parasitic fungus to break down the host cell wall. Membrane-associated phosphatidic acid phosphatase (or PAP), on the other hand, is involved in lipid metabolism and signal transduction and catalyzes the dephosphorylation of phosphatidic acid to produce diacylglycerol and orthophosphate. These enzymes nevertheless share subtle sequence similarity (Fig. 4.1A) and a common catalytic core (Fig. 4.1B,C). The CPO reaction is facilitated by a vanadate ion bound to the active site. Because vanadate is a transition state analog of phosphate, this structural relationship suggests a catalytic mechanism for PAP2 (see Fig. 4.1D). It also raises the question: Which divergent residues within CPO and PAP2 are responsible for their distinct functions? More generally, which divergent residues among members of a given protein superfamily are responsible for functional specificity?

Fig. 4.1. Structural relationship between soluble haloperoxidases and integral membrane phosphatases [ref]. A. Homologous regions within haloperoxidases and phosphatidic acid phosphatases PAPs with catalytic residues highlighted in red. The alignment was obtained using Gibbs sampling. B. The vanadium binding domain of chloroperoxidase (CPO) (pdb: IVNC). The vanadate ion (orange), which is a transition state analog of phosphate, is bound from below by His-496 and from above by azide (not shown). The backbone is shaded red proportional to the degree of sequence conservation. C. Model of the active site of mouse PAP2 based on the CPO structure. Coloring scheme: vanadate and transition state phosphate, orange; residue sidechains interacting with vanadate or phosphate, cyan; Predicted (PMID: 7795533) trans-membrane regions in PAP2, blue; diacylglycerol moiety of phosphatidic acid, light gray. D. Proposed catalytic mechanism for PAP2-related phosphatases. Residue positions in parentheses indicate corresponding active site residues within CPO.

Regarding this second question, consider Ran GTPase, which is essential for the translocation of RNA and proteins through the nuclear pore complex and that shares a catalytic core with other P-loop GTPase. A sequence alignment of Ran proteins reveals a very high degree of conservation across diverse eukaryotic phyla and ‘kingdoms’ (Fig. 4.2). Most of the conserved residues are not directly involved in catalysis and differ from the residues conserved in other P-loop GTPases. At some conserved positions, even minor side-chain modifications—such as substitution of leucine for isoleucine, which merely involves rearrangement
of a methyl group—appear to be consistently eliminated by natural selection. This suggests that such residues establish functionally important interactions with precise geometric and/or chemical constraints important for functional specificity.

Likewise, within an alignment of all P-loop GTPases, there are residues conserved within each protein family as well as subgroup conserved residues that span multiple families; this is illustrated schematically in the figure on the right, where similarly colored horizontal bars correspond to families, vertical bars correspond to conserved patterns, and brackets denote subgroups of families sharing conserved residue that serve as candidate functional specificity determinants. There are dozens of P-loop GTPase families that can be aligned and analyzed in this way.
5. Sequence detection & alignment using a curated hierarchical MSA

**MAPGAPS.** Even the best MSA methods will fail to correctly align some protein sequences. Various Ras-like GTPases, for example, conserve an Arg-Glu salt bridge where the arginine is sometimes preceded by a deletion or an insertion and is sometimes followed by a 15-residue insertion. Hence, aligning this arginine correctly is impossible without manual curation, which is prohibitively time consuming for all but the smallest MSAs. The MAPGAPS (Multiply-Aligned Profiles for Global Alignment of Protein Sequences) program [13] addresses this problem by detecting and aligning database sequences using as the query a curated hierarchical MSA (hiMSA). An hiMSA consists of a set of subgroup MSAs for a specific protein superfamily that are multiply aligned to each other using template MSAs—one for the superfamily and one for each family, subfamily, etc. in the hierarchy. After converting the hiMSA into a set of multiply aligned profiles, MAPGAPS uses these both to detect and globally align database sequences related to the superfamily. It relies on Karlin–Altschul statistics as a measure of significance and on PSI-BLAST (and other) heuristics for speed.

The NCBI maintains a conserved domain database (CDD) [14] of manually curated hiMSAs for various protein superfamilies that can be downloaded as query hiMSAs for MAPGAPS [15] after conversion into MAPGAPS-readable format using the CDD2MGS program. We typically perform a MAPGAPS search on the NCBI fasta-formatted non-redundant (nr) [16] and pdbaa databases. To speed up a MAPGAPS search and alignment, it is best to split up the database sequences into smaller sets (of ≤ 250,000 sequences) using our fasplit program and then run a grid node MAPGAPS search on each set separately. MAPGAPS generated MSAs can be concatenated into a single file and then merged into one MSA using TweakCMA with the -m option. Note that our programs create and utilize MSAs in cma-format, which is more compact than most other MSA formats and thereby facilitates the creation of very large MSAs. The ConvertMSA program can be used to convert MSAs from cma to fasta format and vice versa. Often cma-formatted MSAs require some additional processing prior to being used as input by our programs, for which TweakCMA can again be used. For example, it is recommended that sequence redundancy be reduced to between 90-98% sequence identity. The following table describes some other commonly used TweakCMA options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cdhit=&lt;int&gt;</td>
<td>heuristically remove all but one sequence among those sharing ≥ &lt;int&gt; percent identity using cd-hit (range: 40-100)</td>
</tr>
<tr>
<td>-csq</td>
<td>output a cma formatted consensus sequence corresponding to the input MSA</td>
</tr>
<tr>
<td>-hsw</td>
<td>create a *.hsw file (as required by various programs) that down weights aligned sequences for redundancy</td>
</tr>
<tr>
<td>-m</td>
<td>merge concatenated MSAs (all with the same number of columns) into a single MSA</td>
</tr>
<tr>
<td>-mincol=&lt;real&gt;</td>
<td>output only those aligned seqs with &gt;= &lt;real&gt; fraction of column matches</td>
</tr>
<tr>
<td>-phyla</td>
<td>show the phyla represented in the MSA</td>
</tr>
<tr>
<td>-pdb</td>
<td>output a cma file with pdb sequences only</td>
</tr>
<tr>
<td>-rpdb</td>
<td>remove pdb sequences from the input alignment</td>
</tr>
<tr>
<td>-rm=&lt;int&gt;</td>
<td>remove all but &lt;int&gt; seqs from an alignment; this can speed up analyses by pruning down a large MSA</td>
</tr>
<tr>
<td>-U&lt;int&gt;</td>
<td>remove all but one aligned sequence among those sharing ≥ &lt;int&gt; percent identity using an exhaustive algorithm</td>
</tr>
<tr>
<td>-U</td>
<td>remove identical sequences from a cma file (keeping first occurrences)</td>
</tr>
</tbody>
</table>

The AddPhylum program can be used to annotate nr input sequences with taxids, and with class, phylum and kingdom designations—information that can be used by our other programs; this requires downloading the NCBI taxdump.tar.gz, and prot.accession2taxid.gz files [17] and, for the pdbaa sequences,
the pdb.accession2taxid file. Taxonomic information is added after each sequence identifier (i.e., `<seq_id>`) using the syntax: `<seq_id> {x(t)p;c(k)>}` where x is the starting residue position for N-terminal truncated sequences; t is the sequence’s NCBI taxonomy identifier as an unsigned integer; p and c are the names of the phylum and class, respectively; and k ∈ {A,B,V,M,F,E} indicates the kingdom, where A=Archaea, B=bacteria, V=plants, M=metazoa, F=fungi, and E=protozoa.

**LAPIS.** The (preliminary) LAPIS (Lots of Accurately-aligned Proteins Initiated from Scratch) program takes as input a fasta formatted set of unaligned protein sequences belonging to a specific superfamily and outputs a large MSA. It applies the following steps: (1) Using cd-hit, remove redundant sequences from the input set and select a relatively small set of representative sequences. (2) Run GISMO on the representative set to obtain a starting MSA. (3) Create a hiMSA of depth 1 with each sequence serving as a subgroup ‘MSA’. (4) Run MAPGAPS using the initial hiMSA as the query and the input sequence set as the database to create expanded MSAs for each subgroup. And (5) Run MAPGAPS again using the expanded hiMSA as the query to generate the output MSA. Thus, LAPIS provides a way to create very large, relatively accurate MSAs for those superfamilies lacking a manually curated hiMSA. By running the following BPPS program on this MSA, one can obtain a starting, representative hiMSA for manual curation. Note, however, that LAPIS is still under development and thus is not quite ready for public release.
6. Characterizing protein functional determinants (BPPS)

After obtaining a large, reliably accurate MSA (typically consisting of at least tens of thousands of sequences), the next step is to characterize statistically significant sequence constraints. **Bayesian Partitioning with Pattern Selection (BPPS)** [18,19] does this by partitioning an MSA into a **hierarchical MSA (hiMSA)** (Fig. 6.1A) where hierarchically-arranged subgroups are each defined by ‘pattern residues’ that best distinguish the subgroup’s aligned sequences from sequences in other, closely related subgroups that generally lack that pattern (Fig. 6.1B). Presumably, such residues play key roles in subgroup-specific functions. Using MCMC sampling [20] with simulated annealing [21], the BPPS sampler [22,23] searches for the mode of the posterior probability distribution over possible hierarchies and, as an aid to biological interpretation, visualizes pattern residues within representative sequences and structures (as illustrated in Fig. 6.2 below). Among these are residues with well-characterized functions, which is not our primary focus. Instead, our focus is on pattern residues of unknown functional relevance—structural interactions among which may suggest hypotheses regarding underlying mechanisms [24-34].

**Figure 6.1.** Basic features of a protein hiMSA. A. Schematic diagram of that part of an hiMSA corresponding to the lineage from the root node to leaf node 9. There is one such diagram for each leaf node in the hierarchy. Horizontal lines represent aligned sequences and are color-coded by level in the hierarchy. Thin light gray horizontal lines represent non-homologous and deleted regions. Vertical lines represent the residue pattern positions upon which the hierarchy is based and are similarly color-coded by levels. On either side are the subtrees for each level of the hierarchy. The colored, gray, and white nodes in each tree correspond, respectively, to their **foreground**, **background**, and non-participating partitions (as explained in panel B). The background for the entire superfamily (not shown) is based on standard amino acid frequencies. B. Schematic of a BPPS “contrast alignment” corresponding to the (green) subfamily tree rooted at node 8 in panel A. There is one such contrast alignment for each node in a hierarchy. Sequences assigned to node 8’s subtree (green nodes of upper right tree in panel A) constitute the ‘foreground’ partition, those assigned to the rest of node 8’s parent subtree (dark gray nodes in panel A) constitute the ‘background’ partition, and the remaining sequences constitute an omitted, non-participating partition. Horizontal bars represent sequences; these are colored as are the corresponding nodes of the ‘subfamily’ tree in panel A. Green vertical bars represent foreground pattern residues (shown below each bar); these diverge from (or contrast with) the background residues at those positions (white vertical bars). Red vertical bars heights quantify the degree of divergence.

BPPS runs in various modes; the example shown in Fig. 6.2 corresponding to mode ‘1’. To obtain the sort of hiMSA shown schematically in Fig. 6.1, the output from mode ‘1’ must be used as the input to BPPS modes ‘2’ and ‘3’, which expand each of the subgroup alignments to include regions corresponding to insertions relative to the common core defined for the parent node. The following table describes these and other BPPS modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial hierarchical partitioning of MSA into subgroups</td>
</tr>
<tr>
<td>2</td>
<td>Create a hiMSA as in Fig. 6.1 using a mode 1 checkpoint file</td>
</tr>
<tr>
<td>3</td>
<td>Create contrast alignments for a specified node’s lineage</td>
</tr>
<tr>
<td>A</td>
<td>Run modes 1-3 using default options with optional mapping of pattern residues to structures</td>
</tr>
<tr>
<td>Q</td>
<td>Perform a query-centric run in mode 1</td>
</tr>
<tr>
<td>E</td>
<td>Evaluate the consistency between two BPPS-generated hiMSAs</td>
</tr>
<tr>
<td>H</td>
<td>Run BPPS with a user curated hyperpartition (which need not correspond to a tree) and seed MSAs</td>
</tr>
</tbody>
</table>
**BPPS mode ‘1’** is used when starting from an input MSA. In this mode, BPPS explores the ‘space’ of domain hierarchies by attaching, moving, inserting, and deleting nodes corresponding to functionally divergent subgroups. Each such hierarchy corresponds to a tree, but for reasons explained in the next section, are represented as a **hyperpartition (hpt)**, which (for a tree) consists of an \((n + 1) \times n\) matrix with rows corresponding to nodes in the tree plus an additional node corresponding to a ‘rejected’ category for sequences lacking superfamily features and with columns corresponding to contrast alignments (see Fig. 6.1B). To the right is shown a hpt for globins, where foreground and background partitions are indicated by ‘+’ and ‘–’ symbols, respectively, and omitted sequence sets are left blank.

At the end of each row, the sequence set designation for each node is given with the number of sequences assigned to that set in parentheses. An asterisk at the end of a line indicates an internal (non-leaf) node. The lines below the hpt matrix further annotate columns. Each line lists: the numbers of foreground and background sequences; the associated log-probability ratio (LPR) in nat, in nat per sequence (nps) and in nat per weighted sequence (npws); and the number of pattern positions (i.e., columns of the MSA) for the contrast alignment. The bottom line gives the total LPR, the simulated annealing pseudo-temperature, and the fraction of ‘failed nodes’—that is, those nodes that lack statistical significance. Common user-modifiable BPPS parameters are the following:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-heatmap</td>
<td>generate heatmaps showing, for each pattern, the degree of conservation among hierarchy nodes</td>
</tr>
<tr>
<td>-ichk</td>
<td>save intermediate checkpoint files during the run</td>
</tr>
<tr>
<td>-maxcol=&lt;int&gt;</td>
<td>set the maximum number of pattern positions per node to &lt;int&gt; (default: 25)</td>
</tr>
<tr>
<td>-minsize=&lt;int&gt;</td>
<td>set the minimum set size to &lt;int&gt; sequences (default: 50)</td>
</tr>
<tr>
<td>-minnats=&lt;real&gt;</td>
<td>set the minimum required nats per weighted sequence to &lt;real&gt; (default: 2.0)</td>
</tr>
<tr>
<td>-maxdepth=&lt;int&gt;</td>
<td>set the maximum tree depth to &lt;int&gt; (default: 2)</td>
</tr>
<tr>
<td>-run=P</td>
<td>Create additional output files (requires a checkpoint file)</td>
</tr>
<tr>
<td>-seed=&lt;int&gt;</td>
<td>Set the seed for the random number generator to &lt;int&gt;</td>
</tr>
</tbody>
</table>

The -heatmap option reveals how strongly the pattern residues are conserved in the foreground, background and omitted partitions for each contrast alignment (see image on the right). It also reveals pattern residues that are ‘cross conserved’; that is, are also conserved in some of the background or omitted subgroups. If so, then a user can create a new hyperpartition (that need not correspond to a tree) that can better model the patterns of conservation (as discussed in the BPPS-H section below). This option requires that PyMOL be installed and on your path.
Example 6.1: Ras-like GTPases. Phosphate-binding loop (P-loop) GTPases [35] bind to GDP or GTP via Walker A (G-K-[ST]) residues, which corresponds to the P-loop, and Walker B (D-x-x-G) residues, the conserved aspartate (D) of which interacts (indirectly through a water molecule) with the Mg\(^{2+}\) ion that coordinates with nucleotide phosphate groups. An important subgroup of P-loop GTPases are Ras-like GTPases, which includes Ras, Rab, Rho/Rac, Ran, Arf, and Arf-like (Arl) GTPases and α subunits of heterotrimeric G proteins. Ras-like GTPases [36] function as on-off switches within eukaryotic signaling pathways regulating diverse cellular processes, including vesicle transport, embryonic development, the sensation of vision, odor, taste and pain, microtubule assembly and cell division. These GTPases are associated both with guanine nucleotide exchange factors (GEFs), which turn them on by mediating the exchange of GTP for GDP, and with GTPase activating proteins (GAPs), which turn them off by stimulating hydrolysis of GTP to GDP. The on- or off-state of Ras-like GTPases is communicated through conformational changes within their switch I and II regions, which detect the presence or absence of the γ phosphate of bound guanine nucleotide.

Figure 6.2. BPPS-1 analysis of Ras-like GTPases. A-D. Four contrast alignments highlighting pattern residues that most distinguish the subgroups to which Rab GTPases belong [33,34]. A. Alignment highlighting all residues that are conserved among the 11 aligned sequences representative of Rab-like GTPases. The bullets above specific columns in the alignment indicate residue positions that are well conserved within the displayed sequences but that are not distinctive of the subgroups corresponding to the B, C and D contrast alignments; the heights of the red bars above these positions indicate the degree of conservation relative to
standard amino acid background frequencies. The leftmost column gives the phylum to which each sequence belongs and is color coded by kingdom (metazoans, red; protozoans, cyan; fungi, dark yellow; plants, green; bacteria, purple; archaea, blue). B. Contrast alignment highlighting pattern positions distinctive of P-loop GTPases, with corresponding pattern residues indicated below the alignment; directly below these, corresponding residue frequencies are given in integer tenths. A ‘7’, for example, indicates that the corresponding residue occurs in 70-80% of the sequences. Above highlighted columns are red bars quantifying the selective constraints imposed on pattern positions. The leftmost column gives the protein identifiers for the 11 representative sequences shown and below this column is given the name of the foreground set and (in parentheses) the total number of sequences assigned to the foreground.

C. Highlighted residues distinguishing Ras-like GTPases from other P-loop GTPases [35]. The format used is as described for alignment B, except that corresponding information regarding the background sequence set is also provided. D. Residues distinguishing the Rab-like subfamily from other Ras-like GTPases. E. Structural locations of subgroup-specific residues within Rab11A bound to a GTP analog [37]. Residues generally conserved in all P-loop GTPases (magenta sidechains) bind to GTP or to a GTP-bound Mg2+ ion. Residues with orange and yellow sidechains correspond to Ras-like family and Rab-like subfamily GTPases—that is, to pattern residues highlighted in C and D, respectively. Five of the Ras-like residues occur in the Switch II region that undergoes conformational changes associated with signal transduction. These five Ras-like residues [34] mutually-interact near the C-terminal end of the switch II region. As described below, the Rab-like subfamily residues (yellow side-chains) form aromatic CH–π interactions hypothesized to stabilize glycine ‘flexible hinges’ within nucleotide binding loops, thereby facilitating nucleotide binding and release [33]. In crystal structures these form distinct Rab11A conformation [Figs 6.2F, G] that have been proposed to play a role in the switching mechanism [34] via repositioning of two other Ras-like residues (Gln70 and Glu71 of Rab11A) involved in GTP hydrolysis [39] and in nucleotide exchange [40,41], respectively.

Shown in Fig. 6.2 are four BPPS-generated contrast alignments corresponding to the lineage for Rab-like subfamily, which consists of Rab, Rho and Ran GTPases. The highlighted positions within aligned Rab-like sequences reveal discriminating features distinctive of P-loop GTPases [34] (Fig. 6.2B), of the Ras-like family [33] (Fig. 6.2C), and of the Rab-like subfamily (Fig. 6.2D)—as well as conserved residues that fail to be distinctive of any of these categories (Fig. 6.2A). Roles for several of the Ras-like GTPase pattern residues in GTP hydrolysis and nucleotide exchange were previously proposed. However, BPPS identifies five additional Ras-like residues: (i) the pattern [RK]-x-[ILV] preceding the P-loop, (ii) the pattern [WF] directly preceding the Walker B aspartate, and (iii) the pattern [YF]-[YF] at the C-terminal end of the switch II region. These residues form an aromatic pocket around the negative-dipole moment at the end of the switch II helix with the positively charged pattern residue inserted into the pocket. This helix is oriented in a specific direction away from the GTPase core but is reoriented upon rearrangement of the charge-dipole pocket (Figs 6.2F,G). The charge-dipole pocket occurs in both the on and off states and both the charge-dipole pocket and an alternative configuration occur within the unit cell of a single crystal structure of Rab5a GTPase in the off state. Thus, the charge-dipole pocket configuration is closely associated, not with the on or off state, but rather with formation of an outward-oriented helix and, as a result, with restructuring of the switch II N-terminal region, which plays a critical role both in sensing the on/off state and in mediating GTP hydrolysis and nucleotide exchange via the two other Ras-like residues Gln70 and Glu71 of Rab11A in Figs 6.2F,G.

An unusual homodimeric Rab27 configuration (Fig. 6.3A), in which the switch I, switch II and inter-switch regions of each subunit are exchanged and protrude out like an antenna, can explain why the switch II residues are conserved: In this configuration both subunits form charge-dipole pockets (Fig. 6.3B), which thus are structurally compatible with the outward-directed switch II helices. An association between the charge-dipole pocket and such switch II restructuring is suggested by comparisons between this unusual, homodimeric form of GDP-bound Rab27 and monomeric forms of Rab family GTPases. In the homodimeric form, the switch II region forms a long α helix that is directed away from the structural core of one subunit and toward the structural core of the other subunit. Presumably this helix lacks the conformational strain typically imposed on monomeric GTPases—the switch II regions of which need to bend around and reconnect to the structural core. Hence the charge-dipole pocket is structurally compatible with formation of this unusual, outward-directed switch II helix. Likewise, within monomeric Ras-like GTPases formation of a charge-dipole pocket is associated with an outward-directed switch II helix (Fig. 6.3C), whereas its disruption is associated with disruption of this outward-directed helix (Fig. 6.3D)—a theme that occurs repeatedly in many other Ras-like GTPases [ref].
Neuwald 16

BPPS also identifies (Fig. 6.2D) both two Rab-like pattern residues (forming a ‘T-A’ motif) previously proposed to perform a role in nucleotide exchange [39] and four other previously unidentified Rab-like residues forming a structural component, termed the glycine brace [33]. These include an aromatic residue that forms a CH-π interaction with a Ras-like conserved glycine at the start of the guanine-binding loop (Gly123; see figure on the right) and a second aromatic residue (nearly always a tryptophan) that forms CH-π and NH-π interactions with a conserved glycine at the start of the phosphate-binding P-loop (Gly18 in the figure). Such aromatic interactions are believed to stabilize the β-strand conformation of glycine [42]. The two other Rab-like residues (typically an aspartate and a serine or threonine), together with a conserved buried water molecule, form a network of interactions connecting the two aromatic residues. These observations suggest that the two glycine residues serve as hinges for the P-loop and for the guanine-binding loop and that the glycine brace facilitates guanine nucleotide binding and release by either interacting with or dissociating from these glycine hinges. Consistent with this notion, these aromatic-glycine interactions are disrupted in the structure of Ran GTPase bound to its exchange factor RCC1 (pdb_id: 1i2m) [43] (not shown).

**BPPS H-mode.** Some superfamilies exhibit complex patterns of residue constraints that cannot be modeled hierarchically (i.e., as a tree). This can occur when some proteins in different subtypes inherit certain biochemical properties from a common ancestor that are lost in other members of each subtree due to a relaxation of selective constraints. This is seen, for example, for DNA clamp loader AAA+ domains [18], where subunits B, C and D of eukaryotic replication factor C (RFC) share constraints both with RFC-A (because these all hydrolyze ATP) but not with inactive RFC-E, and with RFC-E (because these all trans-activate ATP hydrolysis) but not with RFC-A (which does not). Moreover, all RFC subunits share constraints both in common with and distinct from bacterial clamp loader γ subunits. For such complex relationships, the
BPPS H-mode may be used, the output of which is illustrated for RFC subunits in Fig. 6.4; this mode requires as input a user-defined hyperpartition (edited using the edit_hpt program) along with a representative ‘seed’ alignment for each subgroup. The H-mode can also rerun an analysis using a larger (updated) MSA of the same superfamily and of the same length as for a previous analysis. The format required for an input hyperpartition (hpt) is illustrated and described in Fig. 6.5; a corresponding output hpt is shown in Fig. 6.6.

(a) Active RFCs vs. catalytically impaired RFCs

(b) AAA+ ATPases vs. all other proteins

(c) Eukaryotic & archaeal vs. bacterial clamploaders

(d) Active RFCs adjacent to an ATP site vs. other RFCs

(e) RFC-A vs. CTF18 + RFC-B

(f) ATP-adjacent clamp loader vs. active AAA+ ATPases

Fig. 6.4. BPPS H-mode generated contrast alignments showing sequence constraints distinctive of various types of replication factor C (RFC) clamp-loader subunits [44]. The sequences shown span a functionally crucial region of RFC subunits, within which most of the strongest constraints occur. Each alignment is highlighted to reveal constraints distinguishing a specific subgroup of RFC subunits from closely related subgroups. The alignments include representative RFC subunits from fungi (budding yeast), animals (human), protozoans (malaria parasite) and plants (mouse-ear cress). As for Fig. 6.2, the conserved patterns and corresponding frequencies that are distinctive of the foreground are shown directly below each alignment and, below this, the conserved patterns distinctive of the background are shown in gray. (a) Constraints most distinguishing active RFC ATPases from catalytically inactive RFC-E subunits. (b) Constraints most distinguishing active AAA+ ATPases from all other proteins. (c) Constraints most distinguishing eukaryotic and archaeal clamp loader RFC subunits from bacterial clamp-loader γ subunits. (d) Constraints most distinguishing active RFC ATPases that interact with the ATP site of an adjacent RFC subunit from other RFC subunits. (e) Constraints most distinguishing the large RFC subunit (RFC-A) from both an alternative large RFC subunit (CTF18) [45] and the structurally adjacent small RFC subunit (RFC-B). (f) Constraint most distinguishing all (eukaryotic, archaeal, bacteriophage and bacterial) clamp-loader ATPases interacting with an adjacent ATP site from other AAA+ ATPases.
Figure 6.5. Input hyperpartition (hpt) illustrated for AAA+ RFC proteins. There are two sections: the first section (labeled ‘HyperParTition’) shows the columns and rows of the hpt. The first row consists of two types of characters (‘!’ and ‘*’) corresponding to the columns of the hpt, where ‘!’ indicates that a contrast alignment for that column should or should not be printed out. The rows of the hpt correspond to the subgroups being modeled. Each row consists of a series of ‘!', '!', and 'o' characters, which indicate that the subgroup in that row is assigned to the foreground, background, or omitted partition for that column’s contrast alignment. This string is followed by the row number, the name of the subgroup, and a ‘!', '.', or '?' character, where a ‘!’ and '.' indicates that the contrast alignments for which that subgroup is in the foreground should or should not be printed, respectively; and where a ‘?’ indicates that the row does not correspond to a specific subgroup (and thus that the contrast alignment should not be printed). The second section (labeled as ‘Settings’) lists the names of the contrast alignment for each column along with (optionally) parameter settings specific to each column; this typically includes a seed pattern.

Figure 6.6. Output hyperpartition (hpt) for AAA+ RFC proteins. Based on the specifications provided by the input hpt and an input MSA, BPPS returns an optimized set of underlying constraints. The output again has two sections: The first section (labeled ‘HyperPartition’) shows the columns and rows of the hpt. The rows again correspond to the subgroups, where each row consists of a series of ‘!’ and ‘o’ characters, corresponding to the foreground, background, or omitted partition for that column’s contrast alignment, followed by the row number, the subgroup name and (parenthetically) the number of sequences assigned to that subgroup. The second section lists statistics for the contrast alignments corresponding to the columns of the hpt. The information in each of these rows consists of: the column number; the number of foreground and background sequences; the log probability ratio (LPR) in nats; the nats per sequence (nps); the nats per weighted sequence (npws); the number of pattern positions (i.e., discriminating columns in the MSA); and the name of the contrast alignment. The last line gives the total LPR (in nats); the final sampling ‘temperature’; and the number of contrast alignments that failed (i.e., that lack statistical significance).

Running BPPS in H-mode can create PyMOL session files showing the structural locations of subgroup-specific residues, as illustrated in Fig. 6.7, which shows the locations of constrained residues identified in Fig. 6.4.

Figure 6.7. Structural features associated with the six categories of RFC constraints shown in Fig. 6.4. The central hole of the RFC complex, through which DNA presumably is thread, is located at the bottom of each panel. (a) Conformation of RFC-B when bound to ATP and the clamp [5]. In all four active RFC ATPases, the arginine corresponding to Arg84 in RFC-B forms hydrogen bonds with DNA.
bonds with main-chain oxygen atoms on either side of the putative catalytic base, as shown. Oxygen, nitrogen and (predicted) hydrogen atoms establishing hydrogen bonds (broken lines) or involved in ionic interactions are shown in red, blue and white, respectively. Ionic and van der Waals interactions are shown as dot clouds. Residues with cyan side chains distinguish active RFC ATPases from catalytically impaired RFC-E subunits; the magenta residue corresponds to the putative catalytic base shared by all active AAAC ATPases; orange residues distinguish all RFC subunits from bacterial clamp-loader ATPases; and yellow residues distinguish active RFC ATPases that interact with an adjacent ATP site from other RFC subunits. (b) A conformation of ADP-bound RFCS, the archaeal small subunit corresponding to RFC-B [31]. The Walker B-interacting arginine (Arg84 in RFC-B or Arg88 in RFCS) is repositioned and, on the basis of a model of the RFC–DNA–clamp complex, could interact with DNA thread through the clamp. (c) Regions of interaction between RFC-A, RFC-B and RFC-C subunits in the crystal structure of the RFC–ATP–clamp complex. The residue side chains corresponding to the NxSD motif have been omitted for clarity. The phenylalanine in RFC-B (Phe96) seems to form a hydrophobic pocket for Lys109. Blue residues distinguish RFC-A from other RFC subunits; the red residue most clearly distinguishes all clamp-loader subunits interacting with an adjacent ATP site from other AAAC ATPases.

**The get_pdb, TweakPDB, and vsi2pml programs.** BPPS and several other programs can map the structural locations of pattern residues (as in Fig. 6.2E–G); this requires as input structural coordinate files, which may be retrieved using the get_pdb program, which also adds modeled hydrogen atoms to identify hydrogen bonds automatically based on geometric criteria. TweakPDB can be used to analyze or modify a pdb-formated coordinate file. To conserved disc space, BPPS may store pattern residue locations for various proteins of known structure within a *.vsi file, from which the vsi2pml program can generate corresponding PyMOL session (*.pse) files, automatically. See the Auxiliary Programs section below for detailed descriptions of get_pdb, tweakPDB and vsi2pml.
7. Correspondence between sequence and structural constraints (SIPRIS)

SIPRIS (Structurally Interacting Pattern Residues’ Inferred Significance) identifies correlations between BPPS- and structurally-defined residue sets. SIPRIS takes as input a set of BPPS-defined pattern residues and corresponding structural coordinates for proteins of interest and, for each structure, identifies statistically significant clusters or hydrogen bond networks of pattern-residues. The simplest cluster is predefined, for instance, by residues at the interface between subunits or in contact with substrate, as illustrated in Fig. 7.1 for N-acetyltransferases, which transfer an acetyl group from CoA to a target substrate.

Figure 7.1. SIPRIS [46,47] estimates the statistical significance of all but two of the 25 residues most distinctive of the glucosamine-6-phosphate N-acetyltransferase (Gna1) family; these either (A) occur at the interface between homodimeric subunits (blue & pink cartoon traces) \( (p = 8.5 \times 10^{-7}) \) or (B) interact with substrate (shown in cyan) \( (p = 6.8 \times 10^{-5}) \) (pdb_id: 4ag9 [48]). The remaining two residues (not shown) are: Lys116, which may position CoA for catalysis by interacting with a CoA phosphate group, and Cys141, which covalently links to the sulfur atom of CoA within Gna1 [48].

SIPRIS can also select the optimal, structurally-defined cluster among a nested set of clusters: Starting from the highest scoring BPPS pattern residue or from a user-defined residue, molecule or atom, SIPRIS first creates an initial cluster by sequentially adding “structurally-adjacent” residues until reaching a pattern residue. (Structural adjacency can be defined based either on the closest hydrogen bond to a current member of the cluster or on the distance either from the starting residue or from another current member of the cluster.) Next, it adds more residues in this way until reaching the next pattern residue; this is the second cluster. Finally, it repeats this process until generating a cluster containing all BPPS residues. From this nested set, SIPRIS selects the cluster that most significantly overlaps with the BPPS-defined residue set (after adjusting for the number of hypotheses considered); this is illustrated for Rab, Rho and Ran (R3) GTPases in Fig. 7.2.

Fig. 7.2. SIPRIS defined GTPase clusters. A. Rab/Rho/Ran (R3) hydrogen-bond network (red sidechains) in Rh1 (pdb_id: 3reB [49]; \( p = 6.2 \times 10^{-5} \)). This includes a salt bridge (R137-E163) and CH-π interactions (G27-W114; G131-F99) hypothesized to modulate nucleotide exchange by stabilizing the P-loop and guanine binding loop [19,33,42]. These loops (bright green backbone traces) harbor residues (yellow sidechains) that bind to guanine-nucleotide and that are distinctive of all GTPTases. B. A Rab4 R3 hydrogen bond network (red sidechains) \( (p = 2.6 \times 10^{-6}) \) and a Rab-specific cluster (orange sidechains) \( (p = 2.9 \times 10^{-5}) \) contacting the Rabenosyn-5 effector (pdb_id: 1z0kA [50]).
A **BPPS-SIPRIS comparative analysis** of exonuclease-endonuclease phosphatases (EEP) [51] identifies residues presumably responsible for the functional specificity of APE1 endonucleases (Fig. 7.3) and of inositol polyphosphate 5-phosphatases (INPP5) (Fig. 7.4), which cleave phosphodiester bonds in nucleic acids and phospholipids, respectively. These analyses suggest that the same core structure and catalytic residues can mediate very different reactions by interacting with a network of family-specific residues, thereby presumably forming a substrate-specific ‘reaction chamber’.

**Figure 7.3.** BPPS-SIPRIS analysis of APE1 endonuclease bound to DNA with an abasic site analog (pdb_id: 5dfi) [52]. APE1 incises an abasic site phosphodiester backbone in DNA. A proposed mechanism involves superfamily-conserved active site residues forming hydrogen bonds with abasic site phosphate group oxygen atoms [53]. A cluster of EEP superfamily residues (yellow sidechains; \( p = 5.2 \times 10^{-6} \)) and a hydrogen bond network of exonIII-AP-endo family residues (red sidechains; \( p = 1.6 \times 10^{-6} \)), both of which are centered on the abasic site (circled). Family residues aggregate between the catalytic core and a loop containing DNA-interacting basic residues (magenta sidechains); they insert into the DNA major groove to form a kink that engulfs the abasic-DNA strand and thus appear to form a substrate-specific “reaction chamber”. Nitrosation of exonIII-AP-endo residue Cys310 results in dissociation of APE1 from DNA and relocation to the cytoplasm [54]; thus, the associated hydrogen-bond network may communicate the nitrosation signal to the DNA binding site. **B**. Close up of the APE1 active site. (For clarity not all residues are shown.) Hydrogen bond atoms use CPK coloring.

**Figure 7.4.** BPPS-SIPRIS analysis of the inositol polyphosphate 5-phosphatase INPP5B. **A.** EEP superfamily, INPP5 family and INPP5B subfamily residue sidechains: yellow, red, and orange, respectively; ligands, cyan; hydrogen bond atoms, CPK coloring. **A**. An INPP5 hydrogen bond network (\( p = 1.1 \times 10^{-7} \)) that forms a secondary shell around the active site and is hypothesized to recognize inositol polyphosphates having phosphate groups attached at positions 4 and 5 of the inositol ring. This network is adjacent to a superfamily hydrogen bond network (\( p = 3.2 \times 10^{-8} \)). **B.** INPP5B bound to the reaction product, phosphatidylinositol 3,4-bisphosphate (pdb_id: 4cml) [55]. SPIRIS clustering results: EEP, \( p = 5.8 \times 10^{-13} \); INPP5, \( p = 3.9 \times 10^{-7} \); INPP5B, \( p = 0.0021 \). The INPP5B subfamily network lies between the proposed membrane interface [55] and the EEP catalytic core, suggesting a role in sequestering specific membrane-associated substrate from the lipid bilayer [56]. **B.** INPP5 hydrogen bond network within INPP5B (pdb_id: 3mtc). **C.** View focusing on the substrate 4-phosphate group. INPP5 enzymes cleave the 5-phosphate, but require for recognition the 4-phosphate, which directly interacts with three INPP5-family basic residues (K503, K516, and R518). **D.** INPP5B (pdb_id: 5a7i [57]), INPP5-family residues most remote from the catalytic core are part of a cleft to which a phosphate is bound. This site may bind a molecule similar to the known substrate and may be allosterically linked to the active site via the network of INPP5 residues. **E.** The INPP5B-like OCRL protein with glycerol bound to a site analogous to that indicated in (D) (pdb_id: 4cmn [55]).
8. Correspondence between pairwise MSA correlations and structural interactions (DCA, STARC)

STARC: Statistical Tool for Analysis of Residue Couplings. Predicting residue-to-residue structural contacts from a multiple alignment covariance matrix has been a topic of study for some time [58,59], the rationale being that residue mutations occurring at one position often result in compensatory mutations at other, structurally interacting positions. A problem with this straightforward approach is that residue positions may be correlated transitively: if position \( i \) interacts with position \( j \) and \( j \) with \( k \), then residues at \( i \) and \( k \) may be correlated even though they fail to interact directly. A critical breakthrough came with Direct Coupling Analysis (DCA) [60-64], which distinguishes direct from indirect correlations [62,65,66]. DCA has been validated through numerous studies on globular [63,65,67] and membrane [60,61,68,69] proteins, and on protein internal repeats [70] and complexes [71-73]. STARC [74] estimates the statistical significance (as \( S = -\log_{10}(p) \)) of the correspondence between high scoring DC-pairs and 3D structural contacts; it can be used to evaluate DCA methods [74] and the potential functional relevance of alternative conformations and of homomeric and heteromeric interactions. Our DARC and SPARC programs (see below) incorporate both DCA and STARC. Prior to running STARC they perform DCA using pseudo-likelihood maximum entropy optimization [75], which outperformed [74] DCA methods based on sparse inverse covariance estimation [76] and on multivariate Gaussian modeling [77].
9. Multidimensional query-centric analysis of protein constraints (DARC)

DARC [81] (Deep Analysis of Residue Constraints) performs a multidimensional analysis that combines DCA, STARC, BPPS, and SIPRIS into a single, query-centric program for identifying and visualizing constraints as superfamly and functional-subgroup conserved residues, as family-specific, high DC-scoring residue pairs, and as correlations of these with each other and with structure. It does this by: (1) characterizing BPPS pattern residues and high DC-scoring residue pairs most distinctive of subgroup along the query sequence’s lineage. (2) Visualizing pattern residues within representative aligned sequences. (3) Automatically generating PyMOL [82,83] session files showing the structural locations of constrained residues and of high DC-scoring pairs. (4) Determining how functional subgroup-specific residues and high DC-scoring pairs correlate with each other and with structure using SIPRIS. We illustrate DARC for bacterial DNA clamp loaders.

The bacterial DNA clamp loader AAA+ complex loads ring-shaped sliding β-clamps onto DNA to keep polymerase attached during replication; it contains one δ, three γ, and one δ’ AAA+ subunits semi-circularly arranged in the order: δ-γ1-γ2-γ3-δ’. Only γ is active, though both γ and δ’ functionally influence an adjacent γ ATPase domain. Hence, γ and δ’ share certain (γ/δ’) constraints, while γ is subject to additional (γ-) constraints absent from δ’. In the presence of ATP, the clamp loader binds to and opens the β-clamp and, upon binding to DNA, ATP hydrolysis occurs, leading to closing of the β-clamp onto DNA [84-86].

![Figure 9.1](https://example.com/figure9_1.png)

**Fig. 9.1.** Residue constraints within the E. coli DNA clamp loader complex bound to primer DNA and to an ATP analog (pdb_id: 3glf [87]). A. View of the γ- and γ/δ'-residues (red and blue sidechains, respectively) at the γ1/γ2 interface and of the top γ/δ'-DC-scoring pairs linking together (within γ2) the DNA-binding α2 and α3 N-termini (magenta rods) and the β-clamp binding loops (purple rods). The γ-residues cluster around the catalytic base (yellow sidechain circled in red; \( p = 1.8 \times 10^{-9} \)). The γ/δ'-residues cluster around L140 (circled in blue; \( p = 6.7 \times 10^{-10} \)). B. Close up of constrained residues linking the ATP, DNA, and clamp binding sites. Color scheme: γ/δ'-residues, blue. γ-residues, red. Backbones of γ1, γ2 and γ3, blue, dark yellow, and pink, respectively. β-clamp binding loops in γ1, γ2, and γ3, marine blue, orange, and bright pink, respectively. Helices α2, α3, and α4 of γ2, orange. Walker B catalytic residues in γ1, yellow. ATP analog ADP•BeF3 and DNA, cyan; Zn++, gray; Mg++, green.

The following functionally-congruent features interconnect the ATP, DNA, and clamp binding sites [81]: (1) Centered on the catalytic base near the γ-phosphate group of ATP are (SIPRIS-identified) γ- and γ/δ'-residue clusters on opposite sides of each γ-to-γ/δ' interface (Fig. 9.1A); the α3 helix C-terminus harbors γ/δ'-residues that interact with the adjacent ATP binding site (Fig. 9.1B). (2) The N-termini of the α2 and α3 helices bind DNA (due to their partial positive charges), harbor γ/δ'-residues interacting with DNA, and are entwined by six top γ/δ'-DC-scoring pairs (i.e., computed using a subgroup alignment of γ and δ’ sequences). And (3) the clamp-binding loop attaches to the α2 helix C-terminus, harbors two top...
γ/δ’-DC-scoring pairs, and forms hydrogen bonds with γ- and γ/δ’-residues. The positively charged sidechain of the γ/δ’-residue, K121, could interact with the α2 C-terminal negative dipole. Since DNA interacts with the α2 N-terminal positive dipole, α2 may act as a levering rod between the clamp binding site and DNA. Likewise, α3 may act as a rod linking the ATP-binding site to DNA. Together, these features appear to constitute an allosteric network coupling DNA binding to ATP hydrolysis & clamp loading [81]. The corresponding contrast alignments are shown in Fig. 19.1 of Appendix 7: Using a MSA of 463,471 AAA+ proteins and the E. coli δ’ clamp loader subunit as the query, BPPS identified those residues that most distinguish both γ and δ’ from other AAA+ proteins (Fig. 19.1C). Using the γ/δ’ sub-MSA and the E. coli γ subunit as a query, BPPS identified those residues that most distinguish γ from δ’ (Fig. 19.1E). DARC saves a BPPS checkpoint file that can later be used to initiate a deeper analysis by expanding subtrees within the query’s lineage.
Figure 9.2. DARC-generated alignments highlighting all residues conserved in \( \gamma \) and \( \delta' \) clamp loader proteins and residues distinctive of the AAA+ superfamily, of the \( \gamma + \delta' \) subgroup, and of \( \gamma \) but not \( \delta' \). These are shown using five versions of the
same representative set of γ proteins (in panels a-e) and of δ' proteins (in panels a to c). Residues are highlighted to indicate amino acid biochemical properties based on the following color code: red font with yellow highlight, non-polar (AVILMWFY); blue font with yellow highlight, cysteine (C); red, acidic (DE); cyan, basic (KR); magenta, polar (STNQ); green, glycine (G); blue, histidine (H); black, proline (P). Non-conserved positions in panels (a) and (d) and non-pattern residues in panels (b), (c) and (e) are shown in gray font. The leftmost columns in panels (b), (c) and (e) give the NCBI sequence identifiers; these are colored the same as the residue sidechains in Figure 2 of the paper.

a. Alignment highlighting all γ + δ' conserved residues. Those sequences above the line within the alignment correspond to representative γ proteins from the (distinct) phyla denoted in the leftmost column; the first sequence corresponds to the E. coli γ subunit (pdb_id: 3glfB). Those sequences below the line correspond to representative δ' proteins from distinct phyla, the first sequence of which corresponds to the E. coli δ' subunit (pdb_id: 3glfE), which was used as the DARC query. The positions listed at the bottom correspond to the E. coli γ subunit. b. BPPS contrast alignment showing the same sequences as in panel (a), but highlighting only those residues most distinctive of the AAA+ superfamily. The heights of the red bars above each highlighted column estimate the selective pressure imposed on pattern residues at that position using a semi-logarithmic scale. Directly below the aligned sequences, the characteristic AAA+ residues at each position are shown and, directly below these, corresponding frequencies are given in integer tenths. A ‘7’, for example, indicates that the corresponding residue occurs in 70-80% of the 452,949 AAA+ sequences in the alignment. Below this is shown the residue positions and sequence of the E. coli γ subunit (with the Thr 165 residue that was mutated to Val highlighted in red), and shown below these are predicted secondary structure elements (symbol: H, helix; E, strand), helix and strand designations, and AAA+ structural motifs (red font) and putative clamp binding loops C1 and C2 (green font). Secondary structure assignments were calculated for the E. coli γ subunit using DSSP [PMID 6667333]. c. BPPS contrast using the same format as in panel (b) to highlight those residues that most distinguish γ and δ subunits from other AAA+ proteins. d. DARC-generated alignment highlighting all residues conserved in γ. e. DARC-generated alignment highlighting residues distinguishing γ subunits from δ' subunits. A few of these are conserved in other catalytically active AAA+ ATPases; see panel (b).
### 10. Constraint analysis of empirical and MD simulated structures (SPARC)

SPARC (Search Procedures for Analysis of Residue Constraints) [88], like BPPS, runs in various modes. These modes fall into two categories: (1) the computation of DCA and STARC S-scores for one or more structures; and (2) Finding interactions within a time series set of MD simulated structures of a specific protein or protein complex. The following table lists the modes for each of these categories:

<table>
<thead>
<tr>
<th>mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank</td>
<td>Perform DCA and compute STARC S-scores</td>
</tr>
<tr>
<td>hetmer</td>
<td>Rank various protein structures based on S-scores.</td>
</tr>
<tr>
<td>simu</td>
<td>Compute S-scores for interacting heteromeric subunits (need fasta seqs with NCBI taxids)</td>
</tr>
<tr>
<td>dist</td>
<td>Find interactions within a time series set of MD simulated structures.</td>
</tr>
<tr>
<td>correl</td>
<td>Residue-to-residue or residue-to-ligand heavy atom contact distances</td>
</tr>
<tr>
<td>sc2sc</td>
<td>Show contingency tables for correlated residue interactions and report interactions that form and dissociate during the simulation</td>
</tr>
<tr>
<td>sc2bb</td>
<td>Residue sidechain hydrogen bond interactions</td>
</tr>
<tr>
<td>sc2sb</td>
<td>Run both sc2sc and sc2bb modes; also reports CH-pi and aromatic-aromatic interactions</td>
</tr>
<tr>
<td>bb2bb</td>
<td>Find changes in backbone c=O to backbone n-h distances</td>
</tr>
</tbody>
</table>

**SPARC rank mode.** In the rank mode, SPARC takes as input an MSA in cma format and the path to a directory containing coordinate files for aligned sequences of known structure; it makes associations based on sequence/structural identifiers. Specifically, structural file names must use the syntax `<pdbid>_H.pdb` where `<pdbid>` is the lowercase version of the NCBI pdb identifier (e.g., '1abc' for pdb identifier 1ABC_A). The results for a rank search of death domain structures are shown in Table 10.1. Fig. 10.1 shows the locations of the highest scoring direct couplings at homodimeric interfaces.

Table 10.1. SPARC ranking of pyrin-related death domain structures by S-score. Eighteen proteins of known structure were identified among 3,572 pyrin domain aligned sequences, five of which are shown. Search parameters: \( r = 4.0 \) Å; \( m = 5 \). See Table 10.2 for parameter definitions. A colon between two chain designations (e.g., A:C) indicates that \( S \) was computed using, for each residue pair, the shorter of the internal versus the homodimeric 3D distances (e.g., the A-to-A versus the A-to-C residue distances).

<table>
<thead>
<tr>
<th>pdbid</th>
<th>chain(s)</th>
<th>S</th>
<th>L</th>
<th>D</th>
<th>X</th>
<th>d</th>
<th>F</th>
<th>( \Delta S )</th>
<th>resolution</th>
<th>method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6ncv</td>
<td>A:C</td>
<td>42.0</td>
<td>1977</td>
<td>111</td>
<td>141</td>
<td>60</td>
<td>1.9</td>
<td>12.9</td>
<td>3.7 Å</td>
<td>cryo-EM</td>
<td>NLRP6 filament</td>
</tr>
<tr>
<td>6ncv</td>
<td>A:B</td>
<td>41.5</td>
<td>1977</td>
<td>111</td>
<td>141</td>
<td>60</td>
<td>1.9</td>
<td>12.4</td>
<td>3.7 Å</td>
<td>cryo-EM</td>
<td>NLRP6 filament</td>
</tr>
<tr>
<td>6ncv</td>
<td>A:H</td>
<td>37.1</td>
<td>1977</td>
<td>107</td>
<td>141</td>
<td>55</td>
<td>1.9</td>
<td>8.0</td>
<td>3.7 Å</td>
<td>cryo-EM</td>
<td>NLRP6 filament</td>
</tr>
<tr>
<td>6ncv</td>
<td>A:Q</td>
<td>36.6</td>
<td>1977</td>
<td>107</td>
<td>154</td>
<td>57</td>
<td>2.0</td>
<td>7.6</td>
<td>3.7 Å</td>
<td>cryo-EM</td>
<td>NLRP6 filament</td>
</tr>
<tr>
<td>2n1f</td>
<td>A:B</td>
<td>35.2</td>
<td>1977</td>
<td>98</td>
<td>141</td>
<td>53</td>
<td>1.8</td>
<td>11.8</td>
<td>4.0 Å</td>
<td>cryo-EM</td>
<td>ASC filament</td>
</tr>
<tr>
<td>6ncv</td>
<td>A</td>
<td>29.1</td>
<td>1977</td>
<td>92</td>
<td>141</td>
<td>46</td>
<td>1.9</td>
<td></td>
<td>3.7 Å</td>
<td>cryo-EM</td>
<td>NLRP6 filament</td>
</tr>
<tr>
<td>2n1f</td>
<td>A:G</td>
<td>28.2</td>
<td>1977</td>
<td>87</td>
<td>181</td>
<td>50</td>
<td>2.3</td>
<td>4.8</td>
<td>4.0 Å</td>
<td>cryo-EM</td>
<td>ASC filament</td>
</tr>
<tr>
<td>6ncv</td>
<td>A:R</td>
<td>27.7</td>
<td>1977</td>
<td>102</td>
<td>141</td>
<td>47</td>
<td>1.9</td>
<td>-1.4</td>
<td>3.7 Å</td>
<td>cryo-EM</td>
<td>NLRP6 filament</td>
</tr>
<tr>
<td>2n1f</td>
<td>A</td>
<td>23.4</td>
<td>1977</td>
<td>80</td>
<td>181</td>
<td>44</td>
<td>2.3</td>
<td></td>
<td>4.0 Å</td>
<td>cryo-EM</td>
<td>ASC filament</td>
</tr>
<tr>
<td>2n1f</td>
<td>A:H</td>
<td>22.7</td>
<td>1977</td>
<td>88</td>
<td>141</td>
<td>41</td>
<td>1.8</td>
<td>-0.7</td>
<td>4.0 Å</td>
<td>cryo-EM</td>
<td>ASC filament</td>
</tr>
<tr>
<td>4ewi</td>
<td>A</td>
<td>19.4</td>
<td>2045</td>
<td>103</td>
<td>151</td>
<td>42</td>
<td>1.9</td>
<td></td>
<td>2.28 Å</td>
<td>X-ray</td>
<td>NLRP4</td>
</tr>
<tr>
<td>3qf2</td>
<td>A</td>
<td>18.9</td>
<td>2045</td>
<td>101</td>
<td>187</td>
<td>45</td>
<td>2.4</td>
<td></td>
<td>1.7 Å</td>
<td>X-ray</td>
<td>NALP3</td>
</tr>
<tr>
<td>4ewi</td>
<td>A:B</td>
<td>18.7</td>
<td>2045</td>
<td>107</td>
<td>179</td>
<td>45</td>
<td>2.3</td>
<td>-0.6</td>
<td>2.28 Å</td>
<td>X-ray</td>
<td>NLRP4</td>
</tr>
<tr>
<td>5h7n</td>
<td>A:B</td>
<td>18.2</td>
<td>2042</td>
<td>100</td>
<td>185</td>
<td>44</td>
<td>2.4</td>
<td>0.2</td>
<td>1.85 Å</td>
<td>X-ray</td>
<td>NLRP12</td>
</tr>
<tr>
<td>5h7n</td>
<td>A</td>
<td>18.0</td>
<td>2042</td>
<td>97</td>
<td>185</td>
<td>43</td>
<td>2.4</td>
<td></td>
<td>1.85 Å</td>
<td>X-ray</td>
<td>NLRP12</td>
</tr>
<tr>
<td>3qf2</td>
<td>A:B</td>
<td>18.0</td>
<td>2045</td>
<td>105</td>
<td>187</td>
<td>45</td>
<td>2.4</td>
<td>-0.9</td>
<td>1.7 Å</td>
<td>X-ray</td>
<td>NALP3</td>
</tr>
</tbody>
</table>
Table 10.2. List of variables defined for STARC S-scores.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L )</td>
<td>Total number of MSA column pairs used</td>
</tr>
<tr>
<td>( r )</td>
<td>Maximum 3D distance used to define contacting residue pairs (default: 4 Å)</td>
</tr>
<tr>
<td>( D )</td>
<td>Number of contacting pairs, i.e. distinguished elements</td>
</tr>
<tr>
<td>( X )</td>
<td>Optimum cut point (as defined by STARC) for partitioning an array of length ( L )</td>
</tr>
<tr>
<td>( d )</td>
<td>Number of left-distinguished elements, i.e. contacting pairs to the left of the cut point ( X ) (inclusive)</td>
</tr>
<tr>
<td>( m )</td>
<td>Minimum sequence separation between residue pairs in query protein of known structure</td>
</tr>
<tr>
<td>( \ell )</td>
<td>The length of the input MSA</td>
</tr>
<tr>
<td>( F )</td>
<td>( F = X + \ell ) indicates how spread-out the value of ( X ) is relative to the MSA length</td>
</tr>
<tr>
<td>( S )</td>
<td>(-\log_{10} P), where ( P ) corresponds to the estimated probability after correcting for multiple tests</td>
</tr>
<tr>
<td>( \Delta S )</td>
<td>Change in the value of ( S ) upon the inclusion of interactions between homomeric subunit interface(s)</td>
</tr>
</tbody>
</table>

Figure 10.1. SPARC rank analysis of pyrin-relate death domain (DD) proteins. A. Table of the 10 top residue pairs for the cryo-EM structure of the NLRP6 PYD filament (pdb id: 6ncv [89]) based on sub-sampling of aligned pyrin-related sequences. SPARC robustly ranked residue pairs based on the number of times they were among the top DC-scoring (i.e., having the top average product corrected Frobenius norms) for 100 input MSA sub-samplings with replacement. Each sub-MSA sampled consisted of 500 sequences randomly drawn from among the 3,572 sequences in the input MSA. Seven of the 10 highest ranked pairs (those shown in black font) correspond to interactions that include contacts between adjacent death domains—suggesting that these contacts are functionally important. B. Image of the NLRP6 PYD filament cryo-EM structure. The 12 pairs that interact in trans, among the 30 highest ranked pairs, are indicated by red rods. Subunits adjacent to the A chain are colored, whereas non-adjacent subunits are shown in light gray. C. Image of the NALP3 PYD crystal structure (pdb id: 3qf2 [90]). For this structure, SPARC computes a negative value for \( \Delta S \) suggesting that the interaction lacks biological relevance and thus may be a crystallographic artifact.
Analyzing heteromeric complexes. SPARC’s hetmer mode characterizes constraints associated with a heteromeric complex for which the user provides an MSA for each of two components. From each MSA, SPARC creates a subalignment consisting of the one sequence from each species that is most closely related to the corresponding ‘gold standard’ component within a given Cryo-EM or crystal structure of the complex. Finally, it computes S-scores for trans interactions between the two subunits, as illustrated in Fig. 10.2. Of course, some components may be absent for some species. To help identify such cases, SPARC outputs, for each component, a histogram of the pairwise scores between each of the candidates and its gold standard sequence—with scores for true orthologs tending to follow a unimodal distribution that is approximately normal. A published study [88] reports strong DC-constraints at several heterodimeric enzyme interfaces, indicative of selective pressures maintaining the residue couplings.

Dynamic analysis of constraints. SPARC aids the interpretation of residue constraints by characterizing them dynamically rather than merely within static structures. Unlike standard analyses of molecular dynamics (MD) simulation data [91-97], this can reveal residue interactions and allosteric couplings associated with otherwise overlooked constraints.

We illustrate this process for the bacterial enhancer binding protein (bEBP) NtrC1 from Aquifex aeolicus: bEBPs activate transcription by remodeling RNA polymerase (RNAP) containing the sigma factor σ^54 [98,99]. Simulations based on various NtrC1 ATP/ADP-bound states reveal alternative interactions involving residues distinctive of bEBP ATPases. For instance, when ATP is bound at the A:B interface and ADP at other interfaces, the AAA+ R-finger R299 interacts with the γ-phosphate of ATP (Fig. 10.3, right), whereas at the adjacent F:A interface, R299 forms a salt bridge with the catalytic base E239 of subunit A, pulling it away from the γ-phosphate of ATP to presumably inhibit ATP hydrolysis (Fig. 10.4, left). However, this salt bridge is disrupted when the nucleotide binding site of subunit F is vacant (Fig. 10.4, right). Together, these interactions may prevent ATP hydrolysis at the A:B interface until ADP is expelled from the F:A interface, perhaps thereby facilitating sequential hydrolysis around the hexameric ring. These interactions, which are absent in the corresponding crystal structure [100], were stably maintained during these 1 μs simulations.
**Correlated motion & allostERIC coupling.** SPARC can identify correlated interactions among constrained residues, as we illustrate here for NtrC1, which forms either a heptameric or a hexameric complex. Partial ATP occupancy causes the heptameric closed ring of NtrC1 to rearrange into a hexameric split ring that drives both ATP hydrolysis and the interaction of RNAP with σ^{54} [100]. During an MD simulation of the heptamer (pdb id: 3m0e [101]) formation and dissociation of R293-D239-trans and R293-D295-cis interactions are highly correlated: When one salt bridge is formed, the other often dissociates, which may help mediate the heptamer-to-hexamer transition. An MD simulation of the hexamer (pdb id: 4ly6 [100]) suggests that allosteric coupling of an R201:E246 cis-to-trans switch to ATP-hydrolysis ([Fig. 10.5](#)) may mediate RNAP-σ^{54} remodeling.

![Fig. 10.5](#). Potential allosteric coupling of ATP hydrolysis to movement of the α2 and α3 helices, which are linked to the σ^{54}-binding L1 and L2 loops—as observed during a 1 µs MD simulation of the bEBP hexamer. **A.** A E246-R201 cis-to-trans switch associated with bEBP-residues, K250 and E205 (red), and with the two highest DC-scoring pairs, E246-R201 and A197-A249 (orange). **B.** The trans state is associated with formation of catalytically favorable interactions among sensor 1 residues, the catalytic base, and the ATP-γ-phosphate. **C.** Restructuring upon ATP hydrolysis of the α3 helix and of the σ^{54}-binding L2 loop.
11. Auxiliary programs

Analyses based on our major programs, namely MAPGAPS, BPPS, SIPRIS, DARC, SPARC, and eCOMPASS, are facilitated by the auxiliary programs listed in Table 11.1. Several of these are described in more detail below. The usage statement for each program provides further information.

Table 11.1. List of auxiliary programs that augment our main analyses.

<table>
<thead>
<tr>
<th>program</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addphylum</td>
<td>Adds taxonomic information to a file of fasta sequences; requires NCBI nr and taxdump files.</td>
</tr>
<tr>
<td>bpps2vsi</td>
<td>Creates a vsi (visualize structural interactions) file from files generated by bpps (see vsi2pml below).</td>
</tr>
<tr>
<td>convert_msa</td>
<td>Converts an MSA from fasta to cma format and vice versa; also converts an MSA from cma into rtf format.</td>
</tr>
<tr>
<td>edit_cma</td>
<td>Performs editing operations on a cma-formatted MSA file.</td>
</tr>
<tr>
<td>edit_hpt</td>
<td>Performs editing operations on a bpps hyperpartition (*.hpt) file.</td>
</tr>
<tr>
<td>get_pdb</td>
<td>Automatically retrieves and adds hydrogen atoms to pdb coordinate files corresponding to pdb_identifiers within a cma-formatted MSA or a fasta sequence file.</td>
</tr>
<tr>
<td>getPDB</td>
<td>Retrieves pdb coordinate files given a list of pdb_identifiers.</td>
</tr>
<tr>
<td>matchcma</td>
<td>Reports matches to user specified pattern residues within a set of bpps subgroup MSAs.</td>
</tr>
<tr>
<td>purgemsa</td>
<td>Reduces sequence redundancy given a cma-formatted MSA or concatenated MSAs.</td>
</tr>
<tr>
<td>tree2hpt</td>
<td>Converts a tree in Newick format into a hyperpartition that can serve as input for bpps in H mode.</td>
</tr>
<tr>
<td>tweakcma</td>
<td>Performs a wide variety of operations on a cma-formatted MSA.</td>
</tr>
<tr>
<td>tweakPDB</td>
<td>Performs a wide variety of operations on a pdb structural coordinate file.</td>
</tr>
<tr>
<td>vsi2pml</td>
<td>Creates PyMOL scripts (<em>.pml) or session (</em>.pse) files from a vsi file.</td>
</tr>
</tbody>
</table>

Retrieving structural coordinates. Structural coordinate files can be retrieved from the PDB using the program get_pdb, which performs the following steps: It first retrieves NCBI pdbaa identifiers from sequences within either an input MSA or fasta file. Next, it retrieves corresponding structural coordinate files from the PDB and then runs the reduce program to add modeled hydrogen atoms. (Adding modeled hydrogen atoms aids identification of geometrically-accurate hydrogen bond interactions.) The names of final coordinate files take the form <pdb_id>_H.pdb, where <pdb_id> corresponds to the pdb identifier in lower case. This process requires both the fasta file pdbaa, which needs to be formatted using the command “makeblastdb -in $FASTADIR/pdbaa -input_type fasta -dbtype prot -parse_seqids”, and the programs blastdbcmd and makeblastdb, which may be downloaded from the NCBI via anonymous ftp at ftp.ncbi.nlm.nih.gov. The blastdbcmd program must be on your path with the environmental variable $FASTADIR set to the path to the directory containing the pdbaa file. You also need to put the script batch_download.sh (available at: www.rcsb.org) on your path and to set the environmental variable 'REDUCE_PRGM' to the reduce program, which is available at https://github.com/rlabduke/reduce.

Examining and editing structural coordinate files. The tweakPDB program takes as input a structural coordinate file in pdb format or (for a few options) a file listing the paths to multiple coordinate files. Its many different options allow the user to extract information or to make changes to the input file. For example, such information includes: covalent and hydrogen bond angles and distances; residue interactions with other residues or with substrate; residue buried surface areas; and amino acid sequences corresponding to protein chains. Possible changes to the input file include: converting MSE (selenomethionine) residues to MET or HIE, HID, HIP to HIS; identifying; eliminating water molecules; renaming chains; truncating chains; and renumbering residues in a chain. Such modifications may be necessary to complete an analysis.

Counting matching residues. The bpps program partitions an MSA into subgroups, which are stored as a concatenated series of sub-alignments within the output file <prefix>_new.mma (where the input alignment is named <prefix>.mma). Given this output file, the program matchcma computes the
percentage of matching residues for each subgroup given a user-provided residue pattern corresponding to column positions in the MSA. For example, the command:

```
matchcma matchcma AAA+_new.mma W4,YF8,D193,P10,E70,R77
```

with AAA+_new.mma being a set of subalignments of AAA+ domains, returns the following percentages of conservation among 64 subgroups of AAA+ proteins.

<table>
<thead>
<tr>
<th>ID</th>
<th>PATTERN</th>
<th>W4</th>
<th>YF8</th>
<th>D193</th>
<th>P10</th>
<th>E70</th>
<th>R77</th>
<th>Total_sq</th>
<th>AvePercent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RfcA</td>
<td>90.2</td>
<td>92.0</td>
<td>97.2</td>
<td>97.7</td>
<td>97.4</td>
<td>98.6</td>
<td>1726.8</td>
<td>(95.5)+</td>
</tr>
<tr>
<td>7</td>
<td>RfcC</td>
<td>87.3</td>
<td>94.2</td>
<td>96.6</td>
<td>94.8</td>
<td>96.4</td>
<td>98.6</td>
<td>1168.2</td>
<td>(94.6)+</td>
</tr>
<tr>
<td>6</td>
<td>RfcB</td>
<td>91.2</td>
<td>92.6</td>
<td>94.6</td>
<td>94.6</td>
<td>97.2</td>
<td>97.4</td>
<td>927.5</td>
<td>(94.6)+</td>
</tr>
<tr>
<td>8</td>
<td>RfcD</td>
<td>88.2</td>
<td>87.7</td>
<td>95.2</td>
<td>92.6</td>
<td>98.4</td>
<td>99.2</td>
<td>1274.2</td>
<td>(93.6)+</td>
</tr>
<tr>
<td>5</td>
<td>RfcS</td>
<td>93.5</td>
<td>92.6</td>
<td>95.0</td>
<td>96.2</td>
<td>85.5</td>
<td>87.9</td>
<td>3072.7</td>
<td>(91.8)+</td>
</tr>
<tr>
<td>14</td>
<td>MCM</td>
<td>0.2</td>
<td>3.9</td>
<td>0.5</td>
<td>0.0</td>
<td>:</td>
<td>:</td>
<td>8121.3</td>
<td>(0.8)-</td>
</tr>
<tr>
<td>15</td>
<td>DnaA</td>
<td>.</td>
<td>0.6</td>
<td>3.0</td>
<td>0.0</td>
<td>:</td>
<td>:</td>
<td>23669.8</td>
<td>(0.7)-</td>
</tr>
<tr>
<td>39</td>
<td>Dynein1</td>
<td>0.2</td>
<td>1.2</td>
<td>1.2</td>
<td>1.0</td>
<td>:</td>
<td>:</td>
<td>1846.8</td>
<td>(0.6)-</td>
</tr>
<tr>
<td>19</td>
<td>Clp</td>
<td>0.0</td>
<td>0.6</td>
<td>0.2</td>
<td>0.8</td>
<td>0.1</td>
<td>1.1</td>
<td>72440.2</td>
<td>(0.5)-</td>
</tr>
<tr>
<td>24</td>
<td>MoxR</td>
<td>0.1</td>
<td>0.9</td>
<td>0.0</td>
<td>0.0</td>
<td>:</td>
<td>:</td>
<td>57145.5</td>
<td>(0.2)-</td>
</tr>
<tr>
<td>21</td>
<td>ClpX</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>:</td>
<td>34122.7</td>
<td>(0.1)-</td>
</tr>
</tbody>
</table>

In this case, the pattern matches various replication factor C (Rfc) AAA+ subunits.

**Visualizing structural features.** Often an analysis involves dozens or hundreds of protein structures. To avoid creating too much output, several of our programs create a *.vsi* file, which stores the information required to visualize results using PyMOL. The `vsi2pml` program is used to obtain PyMOL scripts (i.e., *.pml files), which requires a separate coordinate file, or, if PyMOL is on your path, session files (i.e., *.pse), which include the coordinates. This allows the locations of and interactions among various categories of residues to be visualized using PyMOL, as illustrated in other sections of this tutorial.
12. Performing a complete sequence/structural analysis.

This section gives step-by-step instructions on how to perform a sequence/structural analysis using our approach. Details regarding the input, output and runtime options are described in the command line usage statement for each of our program.

**Compiling our source code.** The following items are used to compile our source code, which is available at [https://www.igs.umaryland.edu/labs/neuwald/software/](https://www.igs.umaryland.edu/labs/neuwald/software/)

- JSON library ([https://github.com/json-c](https://github.com/json-c)); this is required for compilation.
- CUDA ([https://developer.nvidia.com/cuda-downloads](https://developer.nvidia.com/cuda-downloads)) is required to run direct coupling analysis faster on a GPU; but this is optional.
- OpenMP for multiprocessing (though any decent compiler should already include it).
- Both bison and flex are required:
  - Download the src for e.g. bison from [http://ftp.gnu.org/gnu/bison/](http://ftp.gnu.org/gnu/bison/)
  - Download flex from [https://github.com/westes/flex/releases](https://github.com/westes/flex/releases)

CMake ([https://cmake.org/download/](https://cmake.org/download/)) is required for compilation. On many systems it should be already installed. To automatically obtain structural coordinate files and to model hydrogen atoms using our get_pdb program, two third-party programs need to be installed and on your path (see below).

**Obtaining and annotating sequence data.** The NCBI non-redundant (nr) and pdbaa fasta and taxdump files are available via anonymous ftp at ftp://ftp.ncbi.nih.gov/. You can access this site using a command line FTP program, login as “anonymous” and give your email address as your password. Once at this cite use the following commands to obtain these files:

```bash
cd blast/db/FASTA
get pdbaa.gz
get nr.gz
cd ..
cd pub/taxonomy/accession2taxid/
get pdb.accession2taxid.gz ${PROJECT_HOME}/molbio/ncbi_downloads/pdb.accession2taxid.gz
get prot.accession2taxid.gz ${PROJECT_HOME}/molbio/ncbi_downloads/prot.accession2taxid.gz
cd .
get taxdump.tar.gz ${PROJECT_HOME}/molbio/ncbi_downloads/taxdump.tar.gz
```

Move these files to appropriate directories and use `gunzip` to decompress them. You should set the environmental variables `$FASTADIR` and `$TAXDUMPDIR` to these directories so that our programs can access these files. To taxonomically annotate the fasta files, you first must extract the taxonomy files using the commands:

```bash
tar xvf taxdump.tar
grep 'scientific name' names.dmp > scientific.dmp
```

and then use the following commands to taxonomically annotate sequences in the fasta files.

```bash
cat nr | addphylum stdin > nrtx
cat pdbaa | addphylum stdin -pdb > pdbaax
```
To access curated hierarchical MSAs (hiMSAs), which are used as queries by our MAPGAPS program, again log onto the NCBI ftp site, go into the hiMSA directory (cd pub/mmdb/cdd/hiMSA/) and download the files you require.

You can speed up MAPGAPS searches by running many jobs concurrently over grid nodes, in which case you will need to first split the nrtx fasta file into subfiles using the command and then combine and merge the output (cma-formated) files using our tweakcma program with the -m option.

fasplit nrtx 250000 < nrtx

This will create many subfiles, designated as: nrtx.1, nrtx.2, nrtx.3, …, containing up to 250,000 sequences each. MAPGAPS searches these files to create input MSAs for analysis by our other programs.

**Obtaining structural coordinate (pdb) files.** This can be done automatically using our program get_pdb. The command `get_pdb <infile>`, where <infile> is a fasta file containing pdbaa (fasta) sequences, returns the corresponding structural coordinate files in pdb format and then adds modeled hydrogen atoms to the files. Two files will be created for each pdb identifier: the original pdb-formated file (e.g., pdb1abc.ent) and a smaller file that is used by our programs and that includes modeled hydrogen atoms. This requires that two third-party programs be on your path, namely: batch_download.sh, which retrieves the files, and reduce, which adds modeled hydrogen atoms. These are available, respectively, at: https://www.rcsb.org/docs/programmatic-access/batch-downloads-with-shell-script and https://github.com/rlabduke/reduce.

Our program tweakPDB can be used to further modify or analyze pdb files.

**Obtaining an input MSA for analysis by our programs.** If you have downloaded an hiMSA curated by the NCBI CDD group (see above), you can convert this into MAPGAPS-format using the cdd2mgs program. The output files can then be used to search for and multiply align sequences belonging to the superfamily modeled by the hiMSA. Alternatively, if no such hiMSA is available, you can use a cma-formated MSA (obtained using GISMO or another MSA program) to create a hiMSA using the following steps:

```
cp <infile>.cma <infile>.tpl
cat $FASTADIR/nrtx.*_map.seq > All
mapgaps <infile> All
mapgaps All_map // creates *mpa and other files
mapgaps All_map $FASTADIR/pdbaatx
mapgaps All_map $FASTADIR/pdxa.txt_A.cma
mapgaps All_map $FASTADIR/nrtx.*_A.cma > X.cma

\mv -f X.cma > Main.cma
\mv -f X.merged.match80.match80.purge90.cma > Main.cma
\mv -f X.merged.match80.match80.purge90.cma > Main.cma
\mv -f X.merged.match80.match80.purge90.cma > Main.cma
```

The resultant Main.cma and Main.hsw files are used as input to the bppps and darc programs. If the input MSA is in mfasta format, you can use convert_msa (in fa2cma mode) to convert it into cma format. Currently under development is the LAPIS (Lots of Accurately-aligned Proteins Initiated from Scratch) program, which performs the above steps automatically starting from a fasta file of sequences belonging to a given protein superfamily. To heuristically reduce sequence redundancy the faster tweakcma -
cdhit=<int> option may be used. The relative quality of MSAs may be assessed using eCOMPASS. An input MSA may be modified using the tweakcma or edit_cma programs.

**Partition the MSA into subgroups based on conserved patterns.** Running BPPS 1 globally partitions an MSA into divergent subgroups based on subgroup-specific patterns. To evaluate consistency between runs use the BPPS E mode. To expand the alignment within each partition use BPPS 2 and BPPS 3 modes, which will create a hiMSA of the superfamily. If the input MSA is huge, BPPS can take considerable time, in which case DARC may be used to create the lineage within an implicit hierarchy for a specific query sequence. DARC also runs a STARC (DCA) analysis and creates both a SIPRIS (*.sprs) input file and a *vsi file. SIPRIS identifies and computes the significance of pattern residue structural clusters. The vsi2pml program takes as input a *.vsi file to create PyMOL files to visualize the structural locations of residue constraints. For both BPPS and DARC a *.terms output file to assess the fasta define terms and taxonomic information for each partition. (Can also create a terms file using tweakcma with the -terms option.)

**Defining and refining models of protein superfamily residue constraints.** BPPS creates both a *.hpt file, describing the partitions and corresponding patterns, and a *.sma file, containing a seed sequence for each partition. BPPS also has an option (-heatmap) that creates asset of heat map, one for each subgroup showing the degree to which that subgroup’s pattern is conserved both within that subgroup and withing other subgroups. Such heat maps may indicate that the hierarchy should be further edited (possibly into a non-hierarchical ‘hyperpartition’) to model more accurately the constraints imposed on the superfamily. (The heatmap PyMOL files can reveal whether certain subgroups share constrained residues in a non-hierarchical manner.) Such editing may be performed using either our edit_hpt program and/or a text editor, such as vim. Likewise, the corresponding *.sma file may be edited. Given such *.hpt and *.sma files, the BPPS H mode may be used not only to refine a partition, but—given an input MSA of the same length as the original MSA—to also update the overall bpps model as addition sequences become available. This allows models of protein superfamilies to be maintained and refined over time without having to start all over again from scratch. Of course, high quality hiMSA models may also be used as MAPGAPS queries.

**Characterizing residue constraints dynamically.** In a cellular context, of course, proteins typically undergo conformational changes that may not be observed in available structural models. To explore the dynamic structural properties associated with BPPS-defined residues and high DC-scoring residue pairs one may perform molecular dynamics simulations on proteins of interest. We typically set up such simulations using CHARMM-GUI (https://www.charmm-gui.org/) and run the simulation using OpenMM (https://openmm.org/). The SPARC program can be used to search the set of simulated (or also empirical) structures for interactions of interest.

**Interpret results in the light of published research.** One should read the literature on proteins under investigation to help interpret results.
13. Appendix 1: Gibbs Sampler for Multi-alignment Optimization (GISMO)

**GISMO.**

**Notation and Definitions.** The following notation is used for vectors \( \mathbf{v} = (v_1, \ldots, v_n)^T \) and \( \mathbf{w} = (w_1, \ldots, w_n)^T \):

\[
|\mathbf{v}| = |v_1| + \ldots + |v_n|, \quad \mathbf{v} + \mathbf{w} = (v_1 + w_1, \ldots, v_n + w_n)^T, \quad \mathbf{v}/\mathbf{w} = (v_1/w_1, \ldots, v_n/w_n)^T, \quad \mathbf{v}^n = v_1^{n_1} \ldots v_n^{n_n},
\]

and \( \Gamma(\mathbf{v}) = \Gamma(v_1) \ldots \Gamma(v_n) \). Given \( K \) proteins, their sequences are defined by \( \mathbf{R} = (R_1^T, \ldots, R_K^T)^T \) where each vector \( R_k = (r_{k,1}, \ldots, r_{k,n_k}) \) corresponds to the \( k \)-th sequence, \( n_k \) is the \( k \)-th sequence’s length and the \( r_{k,i} \) corresponds to the \( i \)-th residue in that sequence. \( h(\cdot) \) defines a counting function where, for example, \( h(R_k) \) returns a length 20 vector of the counts for the residue types in \( R_k \).

A block-based alignment of the input sequences is defined by \( W \) columns. The set of variables defining the sequence positions for column \( j \) is defined by \( A_j = \{a_{i,j} : k,j \} \). We define \( A_{j[-k]} = A_j - \{a_{k,j}\} \) to denote the set \( A_j \) without \( a_{k,j} \). An alignment is defined by the matrix \( \mathbf{A} = (A_1, \ldots, A_w)^T \) and \( \{\mathbf{A}\} = \{a_{k,j} : k = 1, \ldots, K, j = 1, \ldots, w\} \) denotes the set of residues indices for the alignment variable \( \mathbf{A} \). We represent the collection of residues indexed by elements in a set \( C \) as \( \mathbf{R}_C \). For instance, \( \mathbf{R}_{\{a_k\}} = \{a_{k,j} : k = 1, \ldots, K; j = 1, \ldots, w\} \) represents the set of residues in the alignment defined by \( \mathbf{A} \).

**GISMO Statistical Model.** The residue frequencies observed for column \( c \) are modeled as a multinomial distribution with parameters \( \theta_c = (\theta_{1,c}, \ldots, \theta_{20,c})^T \) where \( \sum_{i=1}^{20} \theta_{i,c} = 1 \) and \( \theta_{i,c} > 0 \) for all \( i \). That is, the vector \( \Theta = (\theta_1, \ldots, \theta_w) \) defines a product multinomial model corresponding to the full alignment. The vector \( \Theta_0 \) corresponds to a background amino acid residue distribution. Hence, the complete-data likelihood function is given by

\[
\pi(\mathbf{R}|\Theta_0, \Theta, \mathbf{A}) \propto \Theta_0^{h(R)} \prod_{j=1}^{w} \left( \frac{\Theta_j}{\Theta_0} \right)^{h(R_{j|c})}
\]

where it is assumed that \( \Theta \sim D(\mathbf{B}) \) and \( \Theta_0 \sim D(\mathbf{a}) \) (where \( D \) denotes the Dirichlet distribution), and where \( \mathbf{B} = (\beta_1, \ldots, \beta_w) \) specifies the Dirichlet distribution parameters (commonly interpreted as numbers of pseudocounts) at each column position \( j \), and \( \mathbf{a} \) specifies the parameters for the background distribution. (Recall that the alignment is specified by the matrix \( \mathbf{A} = (A_1, \ldots, A_w) = (a_{k,j})_{K \times w} \) where \( a_{k,j} \) indicates the position of the \( j \)-th column, which is assumed to be present in all of the sequences.) The likelihood of \( \mathbf{A} \) with the \( \theta \)'s integrated out is

\[
\pi(\mathbf{R}|\mathbf{A}) \propto \Gamma(h(\mathbf{R}_{\{a\}}) + \alpha) \cdot \prod_{j=1}^{w} \Gamma(h(\mathbf{R}_{\{a\}}) + \beta_j).
\]

The conditional predictive probability distribution of this conserved region occurring at position \( i \) in sequence \( k \) is given by
where the $\hat{\theta}$ are the posterior means of the $\theta_i$ given the observed sequence data $R$ and the current alignment $A_{[-k]}$. This statistical model serves as the foundation for the HMM [10] used in later stages of sampling.

**Dirichlet mixture priors.** In order to capture the fact that certain biochemically or structurally similar amino acid residues are more likely to occur together we have incorporated Dirichlet Mixture priors [102,103], as refined by [104]. In order to speed up sampling, GISMO uses a 20 component mixture in the first (competitive) phase of sample, inasmuch as the goal is to merely obtain a reasonable starting alignment without overtraining the evolving HMM. After this initial phase GISMO applies a 58-component mixture.

**Down weighting for sequence redundancy.** Sequences are down weighted for redundancy using the following procedure. For each sequence $k$ a non-integer weight is computed using the method of Henikoff and Henikoff [105] as:

$$wt(k) = \sum_{j=1}^{w} \left( Nt_j \cdot N_{r,k,j} \right)^{-1}$$

where $Nt_j$ is the number of residue types at each position $j$ and where $N_{r,k,j} = \left\{ \left| r_{x,j} \right| \mid 1 \leq x \leq K \land r_{a,j} = r_{a,j} \right\}$ is the number of sequences with the same residue at position $j$ as for sequence $k$. The rational for this formulation is that if a sequence matches lots of sequences at most positions, then it should receive a lower weight than a sequence that matches few sequences at most positions. These weights are then normalized and integerized as:

$$Wt(k) = \left\lceil \frac{wt(k)}{wt_{\text{max}}} \right\rceil$$

where $wt_{\text{max}}$ corresponds to the maximum non-integer sequence weight. Because these weights depend upon the evolving alignment, they are updated after each sampling cycle.

**Inferred HMM transition probabilities.** We model the transition probabilities for the HMM (shown on the right) using a generalization of our previous formulation [10] as follows. The probability matrix for transitions from column $j$ states in the HMM is:

$$
\begin{pmatrix}
M_{j+1} & I_j & D_{j+1} \\
M_j & 1 - \epsilon_o[j] - \delta_o[j] & \epsilon_o[j] & \delta_o[j] \\
I_j & 1 - \epsilon_i[j] & \epsilon_i[j] & 0 \\
D_j & 1 - \delta_d[j] & 0 & \delta_d[j]
\end{pmatrix}
$$

where $1 \leq j \leq w$ and where M, I, and D denote match, insertion and deletion states, respectively. The probability matrix for transitions out of the start state is:

$$
\begin{pmatrix}
M_1 & D_1 \\
Start & 1 - \delta_s[0] & \delta_s[0]
\end{pmatrix}
$$

Transitions into M and I states emit a residue as specified by the $\Theta$ of our statistical model.

**Inference of transition probabilities.** For a given alignment $A$, each sequence $S_k$ is associated with a “path” through the HMM indicating its alignment against the model $\Theta$. We denote the collection of these paths by $\Lambda$ and the total number of HMM transitions of type $M \rightarrow M$, $M \rightarrow I$, $D \rightarrow D$ at position $j$ by

$$\pi(a_k = i \mid A_{[-k]}, R) \propto \prod_{j=1}^{w} \left( \frac{\hat{\theta}_j}{\theta_0} \right)^{b_{a,j}}$$
Ignoring the indexing variable $j$ for clarity, the likelihood of the transition probability parameters at each position is

$$h(\Lambda | \iota, \delta) = (1 - \iota_o \delta_o^{N_o^w})^{N_o^w} \delta_o^{N_o^w} (1 - \iota_e \delta_e^{N_e^w})^{N_e^w} \delta_e^{N_e^w}.$$ 

with independent prior distributions

$$(\iota_o, \delta_o, 1 - \iota_o \delta_o) \sim D(n_{o,m}, n_{o,d}, n_{o,m}), \quad \iota_e \sim \text{Beta}(n_{e,i}, n_{e,m}), \quad \text{and} \quad \delta_e \sim \text{Beta}(n_{e,d}, n_{e,m}),$$

where $n_{o,m}, n_{o,d}, n_{o,m}, n_{i,e}, n_{i,m}, n_{d,e}, n_{d,m}$ are corresponding prior pseudo counts. The corresponding maximum a posteriori probability (MAP) estimates for the transition probabilities at each position $j$ are computed from these observed and prior counts. These define the position specific gap penalties. The joint posterior distribution for the alignment and transition probability parameters is

$$g(\Lambda, \iota, \delta) = h(\Lambda | \iota, \delta) \times h(\Lambda | \delta, \iota) \times P(\iota, \delta),$$

where $P(\mathbf{R} | \Lambda, \Lambda)$ is a generalization of Equation (1), and where $\iota$ and $\delta$ are length $w$ vectors representing the column-specific transition probabilities with prior probability:

$$P(\mathbf{R} | \Lambda, \Lambda) \propto (|,)(1) \times (|,)(1) \times (|,)(1).$$

Given the alignment and thus the paths $\Lambda$, we have the conditional posterior distribution

$$p(\iota, \delta | \Lambda, \Lambda) \propto \prod_{j=1}^{n_w} \left[ \iota_o [j]^{N_o [j]} \cdot \delta_o [j]^{N_o [j]} \cdot (1 - \iota_o [j] \delta_o [j])^{N_o [j]} \cdot (1 - \iota_e [j] \delta_e [j])^{N_e [j]} \right].$$

Sampling on the distribution for each position $j$ is done by drawing the random variables:

$$\delta_o [j] \sim \text{Beta}(N_{o,m} [j] + n_{o,d}, N_{o,m} [j] + n_{o,m} + n_{m}),$$

$$\delta_e [j] \sim \text{Beta}(N_{e,d} [j] + n_{e,m}, N_{e,m} [j] + n_{e,m} + n_{m}),$$

$$\iota_o [j] = (1 - \delta_o [j]) \iota_o [j], \quad \text{where} \quad \iota_e [j] \sim \text{Beta}(N_{e,m} [j] + n_{e,i}, N_{e,m} [j] + n_{e,m}),$$

and

$$\iota_e [j] \sim \text{Beta}(N_{e,d} [j] + n_{e,m}, N_{e,m} [j] + n_{e,m}).$$

For computational efficiency, the $\iota$ and $\delta$ may be integrated out [10] to get

$$h(\Lambda) = \int h(\Lambda | \iota, \delta) P(\iota, \delta) d\iota d\delta.$$

This gives rise to a new posterior distribution $g(\Lambda, \Lambda) \propto P(\mathbf{R} | \Lambda, \Lambda) \times h(\Lambda)$, for which the transition probability parameters need not be fixed or updated and which allows the optimal indel penalties to be determined from the sequence data.

**Sampling algorithm.** GISMO’s MCMC sampling algorithm explores the space of possible alignments by executing Markovian transitions between alignments. This involves sampling alternative alignments of either individual sequences or groups of sequences. In either case, such sampling is done as follows: First, the sequence or sequences are removed from the alignment and the posterior parameters of the HMM are recalculated based on the retained aligned sequences and the priors. Next, emission probabilities for the twenty amino acids at each position are sampled from the posterior emission probability distributions defined by the HMM parameters; note that these sampled probabilities define a sampled HMM. Finally, the previously removed sequences are optimally realigned to the sampled HMM. We explored sampling
transition probabilities in the same way, but found little benefit of doing so; instead, the MAP estimates for
transition probabilities are used. GISMO applies simulated annealing [21] to favor convergence on an
optimal alignment in later stages of sampling. Sampling starts at a “temperature” of $T = 1$ (i.e., sample each
transition directly proportional to its actual probability $p$) and ends at $T = 0$ (i.e., always take the highest
probability transition); between these two extremes the temperature is dropped in $\Delta T = 0.1$ increments with
sampling probabilities set to $p^{\frac{1}{T}}$. Sampling iteratively through all of the sequences continues until this
fails to find a new highest probability state.
14. Appendix 2: Bayesian Partitioning with Pattern Selection (BPPS)

Bayesian Partitioning with Pattern Selection (BPPS). Given a typically very large multiple sequence alignment (MSA), denoted here as $X$, BPPS applies Markov chain Monte Carlo (MCMC) sampling to articulate a superfamily into a set of hierarchically nested partitions corresponding to a tree. Each subtree $h$, which may consist of only a single node, when attached to the root corresponds to a family, and, in general, when attached to a parent node corresponds to a child subgroup. The sampler defines each subgroup, also denoted by $h$, based on residue patterns distinguishing subgroup members from sequences assigned to other nodes in the parent subtree. For instance, a simple pattern for subgroup $h$ might consist of $\{V, I, L\}$, $\{D, E\}$, and $\{F, Y\}$ at column positions 3, 10 and 23, respectively. BPPS favors assignment of those sequences to subtree $h$ conserving a pattern that is not conserved in sequences assigned to other nodes in the parent subtree. Hence, BPPS favors assignment to each parent node those sequences conserving the parent node’s pattern but lacking each of the descendent nodes’ patterns. For a non-root node $n$ in the hierarchy this process defines a ‘contrast’ alignment (as in Fig. 6.2C,D) divided into foreground and background sequences, corresponding respectively to the subtree rooted at $n$ and to the rest of the subtree rooted at the parent of $n$. MCMC sampling is used to determine the number and arrangement of the nodes in the tree, the sequences belonging to each node, the pattern positions for each subgroup, and the conserved residues at each pattern position. The sampler favors convergence on a hierarchy where the pattern defining the partitioning for each node best distinguishes its foreground from its background. BPPS weights sequences, as in PSI-BLAST[105,106], to avoid modeling conserved patterns merely due to sequence redundancy.

For the ensuing discussion, we define the following: For vectors $v = (v_1, \ldots, v_T)^\top$ and $w = (w_1, \ldots, w_T)^\top$, $v/w = (v_1/w_1, \ldots, v_T/w_T)^\top$, $v + w = (v_1 + w_1, \ldots, v_T + w_T)^\top$, $\log v = (\log v_1, \ldots, \log v_T)^\top$, $|v|$ is the sum over vector elements, and $\{v, w\}$ denotes the inner product of $v$ and $w$ and is equivalent, as applied here, to the dot product $v \cdot w = \sum_{i=1}^{T} v_i w_i$. Given an $N$ node hierarchy $H$, we define $S$ as a vector of $N$ disjoint sets, such that $S_n$ contains the sequences assigned to node $n$, and $H$ is defined as a vector of tri-partitions of node indices $1 \leq n \leq N$, such that $H_n \equiv (H^+_h, H^-_h, H^o_h)$ specifies subtree $h$’s foreground, background and “non-participating” nodes, respectively. We define node $n = 1$ as the root and $h = 1$ as the superfamily tree (i.e., $H^+_1 = \{n \mid 1 < n \leq N\}$), for which the background set ($H^-_1 = \{0\}$) consists of a single node ($n = 0$) for unrelated sequences (denoted as $S_0$). The remaining $H^+_h$ are configured hierarchically starting from the root, such that: $H^+_h$ specifies the nodes in subtree $h$; $H^-_h$ specifies the nodes in the tree rooted at the parent node of $h$ but absent from $H^+_h$; and $H^o_h$ specifies nodes in neither $H^+_h$ nor $H^-_h$. To ensure that $H$ corresponds to a tree, we require that each $H^+_h$, other than $H^+_1$, is a proper subset of only one other $H^+_h’$ (i.e., $\forall h : h > 1 \rightarrow \exists! h’ : H^+_h \subset H^+_h’$) and that $H^-_h$ consist of nodes in $H^+_h$ that are not in $H^+_h’$ (i.e., $H^-_h = H^+_h’ - H^+_h$).

BPPS defines a prior on $H$ that depends only on $N$ and positive parameter $\nu$ and that assumes a maximum number of nodes $N_{\max}$, so that
\[ p(H) = \frac{p(N)}{a_N}, \text{ where } p(N) = \begin{cases} 
abla^{N-1} \cdot \left( (1-\nu) / (1-\nu^{N_{\text{max}}}) \right) & \text{if } \nu \neq 1 \\
abla / N_{\text{max}} & \text{if } \nu = 1 \end{cases}, \]

and where \( a_N \), the number of unlabeled, unordered rooted trees with \( N \) nodes, is defined recursively as:

\[
a_N = \begin{cases} 
1 & \text{if } N = 1 \\
\sum_{j_1 + \ldots + \left( (N-1) - j_{N-1} \right)}^{N-1} (a_k + j_k - 1) & \text{if } N > 1 
\end{cases}
\]

with \( j_k \) being the number of subtrees with \( k \) nodes [107]. Computation suggests that the growth of \( a_N \) is \( O(2.96^N) \). By default, \( N_{\text{max}} = 500 \) and \( \nu = 1 \) so that \( p(H) = (a_N \cdot N_{\text{max}})^{-1} \), which corresponds to a uniform prior where every size tree (up to \( N_{\text{max}} \)) is equally likely. Setting \( \nu > 1 \) or \( \nu < 1 \) favors hierarchies with more or fewer nodes, respectively. Note, however, that adding nodes when unjustified by the data is disfavored regardless of \( p(H) \) due to the nature of our Bayesian formulation.

Let \( \sigma_n \) be the prior for a sequence assigned to node \( n \). Given \( N \), the prior for \( S \) is then given by

\[ p(S) = \prod_{n=0}^{N} \sigma_n^{k_n}. \]

By default, we choose a prior for the rejected sequence node of \( \sigma_0 = 0.5 \) and for other nodes \( \sigma_n = (1-\sigma_0) \cdot N^{-1} \) uniformly.

Given \( H \) and \( S \), foreground pattern residue sets are denoted by \( A \), where \( A_{h,c} \neq \emptyset \) for each pattern column position \( c \) in subgroup \( h \), and \( A_{h,c} = \emptyset \) for non-pattern positions. The pattern residue sets \( A_h \) are constrained by a foreground consensus sequence for subgroup \( h \), denoted as \( y_h \), and by the requirement that the residues in each set be functionally similar, as were defined by a detailed analysis of amino acid Dirichlet mixture components [104]. For example, if a tryptophan residue occurs at position \( c \) of the consensus for \( hH \), then \( y_{h,c} = W \) and

\[ A(W) \equiv \{ \{W\}, \{W,F\}, \{W,Y\}, \{W,F,Y\} \} \text{ and } A_{h,c} \in A(W) \cup \{\emptyset\}, \]

where \( A(r) \) denotes the allowed pattern residue sets for consensus residue \( r \). We define prior probabilities for \( A \) as

\[ p(A) = \prod_{h=1}^{N} \prod_{c=1}^{C} \rho_{A_{h,c}} \text{ (product categorical distributions), where, when } A_{h,c} = \emptyset, \rho_{A_{h,c}} = q_\emptyset \]

(0.999 by default) and otherwise \( \rho_{A_{h,c}} = \kappa \cdot q_{A_{h,c}} / \|A_{h,c}\| \). Here, \( \kappa \) is a constant chosen so that the priors for pattern positions sum to \( 1 - q_\emptyset \); \( 0 < q < 1 \) (and 0.5 by default) is a tuning parameter with smaller values yielding higher aggregate priors for the class of “functional” residue sets of smaller cardinality \( |A_{h,c}| \); and \( \|A_{h,c}\| \), defined as the number of possible residue sets of cardinality \( |A_{h,c}| \), functions to distribute prior probabilities uniformly among these sets. For example, if \( y_{h,c} = W \) and \( A_{h,c} = \{W,F\} \), then \( \|A_{h,c}\| = 2 \) because, in this case, there are two possible sets of cardinality 2.
Since we are interested only in whether a residue in column \( c \) of a sequence \( s \) (i.e., \( x_{s,c} \)) is functional or non-functional, we introduce the variable \( \chi \), where \( \chi_{h,s,c} = (1,0)^T \) and \( \chi_{h,s,c} = (0,1)^T \) imply that, for subtree \( h \), \( x_{s,c} \) corresponds to a ‘functional’ and a ‘non-functional’ pseudo-residue, respectively. Since \( A_{h,c} = \emptyset \) at non-pattern columns, corresponding residues are all non-functional. Gaps are also treated as non-functional.

Given \( H, S \) and \( A \), let \( \Theta_{h,c} \) be the 2-dimensional vector specifying the observed functional and non-functional pseudo-residue background frequencies for column \( c \) of subtree \( h \), and let \( \Theta \) be the matrix of all \( \Theta_{h,c} \). Let \( \Theta^{(\alpha_h)}_{h,c} = (1-\alpha_h) \Theta_{h,c} + \alpha_h (1,0)^T \) model the foreground composition where \( 1-\alpha \) specifies the fraction of background ‘contamination’ at pattern positions in the foreground. The prior probability density for \( \alpha_h \) is defined by a beta distribution

\[
p(\alpha_h) = \frac{\Gamma(a_{h,0} + b_{h,0})}{\Gamma(a_{h,0}) \Gamma(b_{h,0})} \alpha_h^{a_{h,0}-1} (1-\alpha_h)^{b_{h,0}-1},
\]

where \( a_{h,0} \) and \( b_{h,0} \) are functional and non-functional pseudo-counts, respectively, and where by default \( a_{h,0} = b_{h,0} = 1 \). The prior probability density for \( \Theta_{h,c} \) is defined by a product Beta distribution:

\[
p(\Theta) = \prod_{h=1}^N \prod_{c=1}^C \frac{\Gamma(a+b)}{\Gamma(a) \Gamma(b)} \theta_{h,c,1}^{a-1} \theta_{h,c,2}^{b-1},
\]

where \( C \) is the number of columns in the MSA, and \( a = b = 1 \) by default.

Conditional on \( H, \chi \), and \( S \), let \( \xi_{h,c} \) denote the inferred number of functional pseudo-residues in column \( c \) of subtree \( h \) that are not due to background contamination. Then, conditional on \( \alpha_h \) and \( \Theta_{h,c} \),

\[
\xi_{h,c} | \alpha_h, \Theta_{h,c} \sim \text{Binom} \left( Nf_{h,c}, \frac{\alpha_h}{\alpha_h + (1-\alpha_h) \theta_{h,c,1}} \right),
\]

where \( \theta_{h,c,1} \) and \( Nf_{h,c} \) are, respectively, the background frequency and the total foreground number (with background contamination included) of the pattern-matching pseudo-residues in column \( c \) for subtree \( h \). Conditional on \( H, \chi, S, \xi_{h,c} \equiv \left( \xi_{h,1}, \ldots, \xi_{h,C} \right)^T \), and \( Nf_h = \sum_c Nf_{h,c} \), the posterior distribution of \( \alpha_h \) is

\[
\left[ \alpha_h | \xi_{h,c} \right] \propto \alpha_h^{a_{h,0}-1} (1-\alpha_h)^{b_{h,0}-1} - \text{Beta} \left( \xi_{h,c} + a_{h,0}, Nf_h - \xi_{h,c} + b_{h,0} \right).
\]

The conditional distribution for \( \Theta_{h,c} \) is:

\[
\Theta_{h,c} | H, S, \xi_{h,c}, \chi_{h,c} \sim \text{Beta} \left( \sum_{h \in H_2} \sum_{s \in S} \chi_{h,s,c,1} + Nf_{h,c} - \xi_{h,c}, \psi_1, \sum_{h \in H_2} \sum_{s \in S} \chi_{h,s,c,2} + \psi_2 \right),
\]

where \( \psi \equiv (a, b)^T \) specifies pseudo-counts with \( \psi = (1,1)^T \) by default.

The sampler infers \( H, S, A, \alpha, \) and \( \Theta \) from \( X \), which defines an MSA. Given these variables, the logarithm of the joint probability distribution[22] is defined as:

\[
\log P(X, H, S, A, \alpha, \Theta) = \log P(X | H, S, A, \alpha, \Theta) + \log P(H) + \log P(S) + \log P(A) + \log P(\alpha) + \log P(\Theta)
\]
where, assuming statistical independence among subtrees (but see below),

\[ \log P(X|H,S,A,a,\Theta) = \sum_{h=1}^{N} \left( \sum_{n \in H'_h} \sum_{s \in S_h} \sum_{c=1}^{C} \left( \log \theta_{h,c} \cdot x_{h,s,c} \right) + \sum_{n \in H'_h} \sum_{s \in S_h} \sum_{c=1}^{C} I_{h,i} \left( \log \theta_{h,c}^{(a)} \cdot x_{h,s,c} \right) \right) \]

and where \( I_{h,c} \equiv \begin{cases} 0, & \text{if } A_{h,c} = \emptyset \\ 1, & \text{if } A_{h,c} \neq \emptyset \end{cases} \). Note that for non-pattern positions \( \theta_{h,c} = (0,1)^{T} \) and \( x_{h,s,c} = (0,1)^{T} \) so that \( \left( \log \theta_{h,c} \cdot x_{h,c} \right) = \log 0 \cdot 0 + \log 1 \cdot 1 = 0 \); hence, these positions contribute nothing to the posterior probability. However, because the configuration of each subtree constrains to some degree the possible configurations of other subtrees above or below it in the hierarchy, our independence assumption is invalid. These constraints reduce the probabilities for some states to zero, so that the probabilities assigned to the remaining, reachable states will sum to less than 1, and our formulation is therefore conservative (i.e., computed probabilities are smaller than they should be). This also occurs due to other imposed constraints, such as placing an upper bound on the number of pattern positions or on the depth of the hierarchy, or requiring that a minimum number of sequences be assigned to each node. Nevertheless, in searching for an optimum, Equation 1 is valid as an objective function, its use here.

**BPPS sampling strategies.** Conditioned on fixed \( H \), BPPS samples over \( S \) and \( A \) by iteratively applying the following. For each sequence \( s \), let \( n \) be its assigned node and remove \( s \) from \( S_n \). Then, sample \( s \) to a new node \( n' \) with probability proportional to \( P(X,H,S,A,a,\Theta|s \in S_{n'}) \) after having updated \( \Theta \) and \( a \). Likewise, for each column position \( c \) in each subtree \( h \), remove the pattern set \( A_{h,c} \) and sample in a new pattern set \( A'_{h,c} \in A(h_{h,c}) \cup \{\emptyset\} \) with probability proportional to \( P(X,H,S,A,a,\Theta|A_{h,c} = A'_{h,c}) \) after having updated \( \chi, \Theta \) and \( a \). However, if the number of pattern positions for a given subtree \( h \) is greater than a specified maximum \( C_{\max} \) (25 by default), reduce the number down to \( C_{\max} \) by removing the lowest probability pattern positions.

The BPPS sampler is initialized by setting \( N = 1 \) with \( H_1 = \{H_1^+,H_1^-,H_1^o\} = \{\{1\},\{0\},\emptyset\} \), and assigning all sequences to the root node \( (n = 1) \) with \( C_{\max} \) pattern positions for subtree \( h = 1 \) and with background pseudo-residue frequencies at each position \( (\theta_{0,c}) \) derived from the overall residue frequencies for the entire MSA. At this stage, sampling over \( S \) merely involves iteratively assigning sequences either to the foreground \( (S_1) \) or to the background \( (S_0) \), where the background represents unrelated sequences inadvertently included in the alignment. Sampling over \( A \) generally tweaks pattern assignments slightly due to removal of unrelated sequences. This provides a good starting point to speed up convergence with essentially no risk of getting trapped in a suboptimal state. Convergence is defined by a cycle of sampling over \( S \) and \( A \) that fails to improve upon the best configuration found thus far, as defined by the log-probability (Equation 1). BPPS saves the best configuration for the final output.

After convergence with \( N = 1 \) nodes, a child node may be added to the root node, as follows. First, some of the sequences assigned to the root are reassigned to the child node by selecting a subset of sequences that are more similar to each other than they are to the remaining sequences. Selections are based on similarity to the query, when one is designated, and on similarity to an arbitrary sequence otherwise. Next, BPPS samples for a few cycles over \( S \) and \( A \), as described above, to search for a configuration that improves upon the previous hierarchy based on Equation 1. If the hierarchy fails to improve and a query has not been provided, several other candidate queries may be selected in turn until either an improved state is found or until a prespecified number of attempts are tried. If the hierarchy is improved, BPPS further enlarges and rearranges the evolving hierarchy \( H \) by adding more leaf nodes and by deleting, inserting or
moving nodes using this same basic strategy. To avoid excessively complex hierarchies, BPPS requires that each leaf node contain a minimum number of sequences (50 by default); those that do not are pruned. After convergence, the sampler applies simulated annealing[21] to ‘drop into’ a more nearly optimal configuration.

When DARC applies BPPS, it focuses on the query’s lineage within the superfamily hierarchy by first defining the query’s family based on residues that most distinguish family members from other superfamily members. Next, DARC seeks to recursively define, in a similar manner, the query’s subfamily, and other subgroups further down the query’s lineage to a prespecified maximum depth.
15. Appendix 3: Direct Coupling Analysis (DCA)

Direct Coupling Analysis (DCA). DARC performs DCA using the algorithm implemented in the CCMpred program version 0.3.2 (https://travis-ci.org/soedinglab/CCMpred)[75], which is essentially identical to the plmDCA[64] and GREMLIN[108] algorithms and which we modified to output DCA scores in PSICOV format. Our description here follows closely the one given for CCMpred[75]. The rationale behind DCA is that, over evolutionary time, mutations at a given residue position are compensated for by mutations at interacting positions to thereby maintain structural integrity. DCA works by avoiding the confounding effect of indirect correlations due, for example, to two residues both interacting with a third residue, but not with each other. DARC uses the sub-MSA defined by BPPS to compute the highest scoring directly coupled residue pairs (DC-pairs).

The CCMpred algorithm eliminates indirect interactions from an interaction network by inferring a generative model of the MSA based on a Markov Random Field (MRF). We again represent the input MSA as an $R \times C$ row × column matrix $X$, where element $X_{s,c}$ corresponds to the residue in row (i.e., sequence) $s$ and column $c$. The columns correspond to vertices of the MRF with single-residue emission potentials $\varepsilon_c(r)$ for amino acid residue $r \in \{1, \ldots, 20\}$ in column $c$; covariation between columns corresponds to edges of the MRF with pairwise emission potentials $\varepsilon_{c,d}(r_s, r_d)$ for residues $r_s$ and $r_d$ in columns $c$ and $d$, respectively. In theory, one could optimize the parameters of the MRF given the MSA using as the objective function the probability:

$$P(\varepsilon | X) = \frac{1}{Z} \prod_{s=1}^{R} \prod_{c=1}^{C} \exp \left( \sum_{d=1}^{C} \varepsilon_{c,d}(X_{s,c}, X_{s,d}) \right)$$

where $Z$ is a normalization constant to ensure that the sum over all sequences equals 1. However, because computing $P(\varepsilon | X)$ is intractable for a non-trivial MSA, the following pseudo-log-likelihood is used instead as the objective function:

$$pLL(\varepsilon | X) = \sum_{s=1}^{R} \sum_{c=1}^{C} \left[ \varepsilon_c(x_{s,c}) + \sum_{d=1}^{C} \varepsilon_{c,d}(x_{s,c}, x_{s,d}) - \log Z_{s,c} \right]$$

where $Z_{s,c} = \sum_{r=1}^{20} \exp \left[ \varepsilon_c(r) + \sum_{d=1}^{C} \varepsilon_{c,d}(r, x_{s,c}) \right]$. Because computation of the normalization constants $Z_{s,c}$ involve summing over only $C$ terms, these are much faster to compute than $Z$ for $P(\varepsilon | X)$. The gradient (the vector of partial derivatives) of this pseudo-log-likelihood is given by:

$$\frac{\partial pLL(\varepsilon | X)}{\partial \varepsilon_{c,d}(r, r')} = \sum_{s=1}^{R} \left[ \delta_{s,c,r'} - \frac{1}{Z_{s,c}} \exp \left( \sum_{i=1}^{C} \varepsilon_{i,c}(r_i, x_{s,i}) \right) \right]$$

$$= \sum_{s=1}^{R} \left[ \delta_{s,c,r'} - p(x_{s,c} = r | x_{s,1}, \ldots, x_{s,c-1}, x_{s,c+1}, \ldots, x_{s,C}, V, E) \right]$$
where $\delta_{x,y}$ is the Kronecker delta function.

In order to favor sparse solutions, we add an $L_2$ regularization term $R(\varepsilon)$ and maximize $pIL(\varepsilon|X) - R(\varepsilon)$ using the nonlinear conjugate gradient method where

$$R(\varepsilon) = \lambda_{\text{single}} \sum_{c=1}^{C} \|\varepsilon_c\|_2^2 + \lambda_{\text{pair}} \sum_{c,d=1,\neq c}^{C} \|\varepsilon_{c,d}\|_2^2,$$

where the regularization coefficients are $\lambda_{\text{single}} = 1; \lambda_{\text{pair}} = 0.2 \times (L - 1)$ [108], and where $\|\varepsilon_c\|_2^2$ and $\|\varepsilon_{c,d}\|_2^2$ are the sum of squared residuals, which measure the discrepancy between the model and the data (with smaller values indicating a tighter fit of the model to the data).

After a successful optimization, the couplings between residue positions $K_{c,d}$ are ranked by the Frobenius norms of the edge potentials $\varepsilon_{c,d}$:

$$K_{c,d} = \sqrt{\sum_{r,r'=1}^{20} \varepsilon_{c,d} (r,r')^2}$$

Lastly, an Average Product Correction[109] is applied to arrive at the final score:

$$\zeta_{c,d} = K_{c,d} \frac{K_{c,d}}{\bar{K}}$$

where "." denotes averaging over the corresponding row or column and $\bar{K}$ is the average over all matrix elements.

**Evaluating the robustness of DCA score rankings.** To determine whether different input MSAs rank DC-pairs consistently, an auxiliary subsampling routine is included as an option in DARC. For the analysis here, this routine draws from the input MSA 1,000 samples of 1,000 sequences, from each of which DCA scores are computed. Between samplings, the previously sampled sequences are replaced prior to sampling the subsequent set. The percentage of times that each residue pair was among those with the top 20, 10, 5 or 2 DC-pairs is given in an output table. Those pairs consistently selected among the 20 top scores are also output.
16. Appendix 4: Initial Cluster Analysis (ICA)

Initial Cluster Analysis (ICA). To compute $CLSP$, $3DSDC$, $3DSP$, or $DCSP$ (denoted generically here as $S$) we apply Initial Cluster Analysis[110], a statistical approach to address the following question: Consider an array of 0s and 1s of length $L$ and containing $D$ 1s. Are some or all of the 1s significantly clustered near the start of the array, and, if so, how surprising is the most significant such clustering? To make this determination, ICA applies the Minimum Description Length (MDL) principle[111], an information theoretical regularization method for finding the best hypothesis for a given set of data.

The MDL principle defines a theory $\theta$ as a probability distribution $P_\theta$ over all possible sets of data and the description length of a data set $E$ given a theory $\theta$ as $DL(E|\theta) = -\log(P_\theta(E))$. A model $M$ is a parameterized set of theories, and the description length of $E$ given $M$ is defined as $DL(E|M) = \min_{\theta \in M} DL(E|\theta)$. The MDL principle asserts that among multiple models to explain $E$, one should prefer the model $M$ that minimizes $DL(E|M) + COMP(M)$, where the description length or complexity $COMP(M)$ is the log of the number of effectively independent theories $M$ contains. For ICA, the MDL principle determines whether the hypothesis $H_1$ that the 1s cluster near the start of the sequence is better than the null hypothesis $H_0$ that the 1s and 0s occur randomly.

ICA treats $H_1$ as a single-parameter model, whose parameter $x$ describes the location of a cut at a discrete point from 1 to $L-1$ along the array, thereby dividing it into an initial segment $s_1$ of length $x$, and a terminal segment $s_2$ of length $y = L - x$. If $s_1$ contains $D_1$ 1s, and $s_2$ contains $D_2 = D - D_1$ 1s, assume that $s_1$ is generated by Bernoulli trials with maximum-likelihood probability $P_1 = D_1 / x$ for a 1, and $s_2$ is generated by Bernoulli trials with probability $P_2 = D_2 / y$ for a 1. Given a particular fixed value for $x$, the probability of $E$ is $P_x(E) = P_1^{D_1} (1-P_1)^{x-D_1} P_2^{D_2} (1-P_2)^{y-D_2} / Z$, where $Z$ is a normalization constant taken over all length $L$ sequences having $D$ 1s. Hence the description length of $E$ under $H_1$ is $DL(S|H_1) = -\log(\max_x P_x(E))$. ICA computes the complexity of $H_1$, as

$$COMP(H_1) \approx \log\left(\sqrt{D/\pi} \frac{L-1}{2}\right).$$

ICA treats $H_0$ as a model consisting of a single Bernoulli-trial theory for generating $E$, with the probability of a 1 taken as $P = D/L$, and of a 0 as $Q = 1 - P$. Hence, $DL(E|H_0) = -\log\left(P^D Q^{L-D}\right)$, which is $L$ times the entropy of the Bernoulli trial. Because $H_0$ contains only one theory, its complexity is zero. The MDL principle says that we should prefer $H_1$ to $H_0$ when $DL(E|H_1) + COMP(H_1) < DL(E|H_0)$. Treating each hypothesis as equally likely a priori, we may view the difference $\Delta$ between the two sides of this inequality as a log-odds ratio, and use the logistic function $\frac{e^\Delta}{1+e^\Delta}$ to convert this into a $p$-value (see p. 37 of [112]), from which $S = -\log_{10}(p)$ is defined.
17. Appendix 5: Structurally Interacting Pattern Residues' Inferred Significance (SIPRIS)

**SIPRIS.** SIPRIS relies on Initial Cluster Analysis (ICA), as described in Appendix 4 and which addresses the following questions: Consider a string of 0s and 1s of length L and containing D 1s. Are some or all of the 1s significantly clustered near the start of the sequence, and, if so, how surprising is the most significant such clustering? Here we focus on the statistical and information theoretical bases of ICA as applied to BPPS-SIPRIS analyses.

**BPPS-defined residue sets.** Modes 2-3 of BPPS generate a hiMSA (Fig. 6.1). For each subgroup (i.e., subtree) G within a hierarchy, BPPS defines a corresponding set of “discriminating” residues that most distinguish members of that subgroup from closely related subgroups. This set is ordered from the most to the least distinguishing residues. We assume that these residues are likely responsible for functions specific to subgroup G. Although such a set typically includes residues with well-characterized functions, our focus is on residues of unknown functional relevance. When mapped to available structures, these distinguishing residues may readily suggest plausible hypotheses; in this respect, a BPPS analysis is informative by itself. However, SIPRIS can obtain deeper insight into and corroboration of a BPPS analysis by identifying significant overlap between BPPS-defined discriminating residues and structurally defined residue sets; we term the intersection of two such sets a **BPPS-SIPRIS cluster**. SIPRIS analysis was motivated, in part, by Karlin and Zhu’s approach [113] for identifying significant clusters of residues that share physical-chemical properties.

**BPPS-SIPRIS predefined clusters.** The simplest BPPS-SIPRIS analysis is based on a specific, predefined structural cluster of n residues. This corresponds to a ball-in-urn problem, in which the BPPS-defined distinguishing residues correspond to $N_1$ red balls, the remaining residues to $N_2$ black balls, and the cluster to n balls drawn from the urn. The probability that at least $x$ of the $n$ residues are distinguishing (i.e., are “red”) is given by the cumulative hypergeometric distribution:

$$P(x, n, N_1, N_2) = \sum_{i=\min(n, N_1)}^{\max(n-x, N_2)} \binom{N_1}{i} \binom{N_2}{n-i} \binom{N_1+N_2}{n}^{-1}.$$ 

**BPPS-SIPRIS optimized-clusters.** Similar to BPPS-predefined clustering is choosing the optimal BPPS-structural cluster among various alternatives. To construct these, we start from a well-defined position in space, and sequentially add “structurally-adjacent” residues (variously defined, as described in Results) to generate a set of nested, structurally defined clusters. From this nested set, we select the structural cluster that optimally overlaps with the BPPS-defined residue set by applying the **Minimum Description Length (MDL)** principle [111], as described in the next section. Optimizing over different starting residues, or different numbers of discriminating residues, requires further $p$-value adjustment, for which we currently apply the overly-conservative Bonferroni correction to obtain an upper bound.

**The MDL principle.** To avoid overfitting BPPS-SIPRIS statistical models to observed data, we apply the **MDL** principle [111], which can be understood as formalizing Occam's Razor (“a model should not be needlessly complex”). Conceptually, this principle claims that the best among a set of alternative models is that which minimizes the description length of the model, plus the maximum-likelihood description length of the data given the model. This approach accounts for the implicit number of independent tests performed when optimizing the parameters of a model, and strikes a balance between a model's complexity and its ability to fit the data—in our case to describe biologically relevant amino acid residue patterns. More formally, a **theory** is a probability distribution over all possible sets of data, and a **model** is a
parameterized set of theories. The description length of the data $D$ given a model $M$, is then defined by $DL(D|M) \equiv -\log P(D|T)$, where $T$ is maximum-likelihood theory contained in $M$ (i.e. the theory which yields the greatest probability for $D$). The description length of the model $M$ is defined by $DL(M) \equiv \log(N)$, where $N$ is the number of the effectively distinct theories (i.e. parameter settings) $M$ accommodates [111]. The MDL principle aims to minimize $DL(D|M) + DL(M)$.

MDL applied to BPPS-SIPRIS clustering. BPPS-optimized clustering presents several mathematical challenges. Computing valid $p$-values requires adjusting for the multiple tests implicit in optimizing over starting residues and clusters. Also, this optimization itself may carry an implicit bias favoring small or large clusters, as outlined below.

We start with a null model in which discriminating residues (e.g., defined by BPPS) are distributed randomly throughout an entire sequence. Given a fixed number of discriminating residues, this model yields a uniform likelihood for all sets of data, and serves as a basis of comparison for likelihoods generated by an alternative model. This model divides the sequence into an initial segment of length $x$ (which we refer to as a cluster) having $m$ discriminating residues, and a terminal segment of length $y$ having $n$ discriminating residues. The model assumes discriminating residues are generated with different probabilities in the initial and terminal segments, and its maximum-likelihood theory assigns the likelihood

$$p = \left(\frac{m}{x}\right)^x \left(\frac{x-m}{x}\right)^{-m} \left(\frac{n}{y}\right)^y \left(\frac{y-n}{y}\right)^{-n}$$

to the data. For a particular cut-point $x$, this likelihood requires choosing the discriminating-residue probabilities $m/x$ and $n/y$ for the initial and terminal segments, and is easily normalized for the selection of these parameters. Our aim, however, is to pick the $x$ (i.e., cluster) that yields the greatest likelihood for the data. Applying the MDL principle requires calculating the effective number of independent tests $N$ implicit in choosing $x$ [110]. By treating $x$ as a continuous as opposed to a discrete parameter, we are able to calculate its Fisher information [110], and thus $N$.

One subtlety is that simply choosing the cut point $x$ yielding the greatest likelihood implicitly favors low or high values of $x$. This occurs because the Fisher information is greater at extreme values of $x$, implying that the likelihoods are more independent of one another at those values. Empirical analyses show that this bias toward large and small clusters often yields suboptimal results from a biological perspective. However, by adding an $x$-dependent correction, derived from the Fisher information, to our optimization, we may flatten the implicit prior associated with $x$ [110]. Random simulation shows that analytic $p$-values computed using our approach fall within about 20% of empirical $p$-values. We still need to adjust these $p$-values for clusters found using different starting residues. Absent a better approach, we currently apply the simple but overly conservative Bonferroni correction [114].
18. Appendix 6: Statistical Tool for Analysis of Residue Couplings (STARC)

DCA and 3D contacts concurrence scores \((3DSDC)\). The \(3DSDC\)-scores apply ICA[110] to measure, as the \(-\log_{10}(p)\), the statistical significance of the correspondence between pairwise structural interactions and DC scores. Given an array of residue pairs ordered by their DC scores, we ask how well it agrees with an alternative ordering based on 3D pairwise distances. More specifically, we seek to identify an optimal initial cluster of elements of the array (defined by a cut), as measured by a relevant \(p\)-value. We are given an array of \(L\) residue pairs ordered by their DC-scores. \(D\) of the pairs (denoted by ‘1’s) are separated within a reference 3D structure by \(\leq z\) Å (with \(z = 3.5\) Å by default) and \(L - D\) (denoted by ‘0’s) are not. We ask: what initial cluster, consisting of pairs up to and including a cut point \(X\), contains the most surprising number \(d\) of ‘1’s, and what is its probability of occurring by chance?  (We term the \(d\) 1’s in an initial cluster “left-distinguished pairs.”) For \(L = 18\) and \(D = 7\), for example, one such array is “101101000001000101”, with optimal cut point \(X = 7\) (underlined), yielding \(d = 5\). Since the pairs are ranked by pairwise distance, we might then represent our example array as “401603200000070005” with digits \(> 0\) denoting the ranks of distinguished pairs. ICA ignores these ranks when choosing the optimal \(X\), whereas we would prefer the \(d\) distinguished pairs to the left of \(X\) to have superior ranks (i.e., lower numbers) than those to the right.

To generalize ICA to exploit ranking information we incorporate a ball -in-urn model to calculate a ranking specific \(p\)-value \(P_b\). For a specific cut point \(X\) that yields \(d\) left-distinguished pairs, we imagine first coloring red, among all \(D\) distinguished pairs, those \(d\) pairs with the smallest pairwise distances; and then recording the number \(R\) that are red among the left-distinguished pairs. Ideally, all the left-distinguished pairs will outrank the remaining distinguished pairs, yielding \(R = d\), but more generally higher values of \(R\) are better; in the example of the previous paragraph, \(D = 7\), \(d = 5\) and \(R = 4\). Given the null hypothesis that rankings are random, we may then use the cumulative hypergeometric distribution to calculate the probability \(P_b\) that \(\geq R\) of the left-distinguished pairs are red:

\[
P_b = \sum_{i=R}^{d} \left( \begin{array}{c} d \\ i \\ \end{array} \right) \left( \begin{array}{c} D-d \\ d-i \\ \end{array} \right) / \left( \begin{array}{c} D \\ d \\ \end{array} \right).
\]

This corresponds to drawing \(d\) balls from an urn containing \(D\) balls, of which \(d\) are red; note that the number of balls drawn here equals the number colored red. A low value of \(P_b\) is reported for a cut with a surprising number, among its \(d\) left-distinguished pairs, having the \(d\) smallest pairwise distances.

Before it corrects for having optimized over all possible cuts, ICA can be understood as calculating a \(p\)-value \(P_a\) for finding \(d\) distinguished pairs to the left of a cut point \(X\). Because the calculation of \(P_a\) ignores ranking information, it will be independent of \(P_b\), and these two \(p\)-values may therefore be combined to yield a joint \(p\)-value \(P_j\) [115-117] using the formula

\[
P_j = P_a P_b (1 - \ln P_a P_b).
\]

Low values of \(P_j\) may arise from low values of \(P_a\), or \(P_b\), or of both. \(P_j\) can provide a statistically stronger measure of the congruence of two orderings, here derived from DC scores and 3D distances, than does \(P_a\) alone. The \(p\)-values \(P\) we report in this paper correspond to \(P_j\), after it has been corrected for optimization over the multiple cut points \(X\) considered[110].

For homomeric structures, DARC assesses the correspondence, not only between DCA scores and internal 3D-contacts alone (e.g., labeled as ‘A’ for chain A), but also between DCA scores and both internal and adjacent subunit interface 3D-contacts (e.g., labeled as ‘A:B’ for chain A and adjacent chain B). The change in \(3DSDC\) upon inclusion of interface contacts is denoted as \(\Delta S_{DC}\). High positive values for \(\Delta S_{DC}\) suggest that strong selective pressures are maintaining 3D contacts between adjacent subunits. In contrast,
negative values for $\Delta S$ suggest that subunit interactions may be functionally insignificant and perhaps due to a crystallographic artifact.

**BPPS and DCA or 3D contact concurrence scores ($\rho_{BPS}, \rho_{DC}$).** DARC provides a measure of statistical significance (based on ICA) for the concurrence between pairs of BPPS-defined residues and either the highest scoring DC pairs or the closest 3D-contacts. The overlap between BPPS and DCA[74] assessed in this way is often weak. Hence, DCA and BPPS are often complementary, so that combining both analyses often provides deeper insight into the relationship between protein structure and function.

**Pattern residue 3D-clustering significance scores ($c_{3D}$).** DARC applies ICA to estimate constraints tending to cluster BPPS-defined residues structurally[74]. The $L$ positions of the ICA array correspond to the $L$ residues within a 3D structure of the DARC query protein or of other proteins belonging to the query family. The 1s in the array correspond to a fixed number of BPPS-defined pattern residues and the 0s to the remaining residues. DARC orders array elements based on their 3D distance from a starting residue. It then determines the most significant 3D-cluster of these residues among a nested set of clusters, each centered on the starting residue. This is performed, starting with each of the BPPS-defined residues in turn, and the highest scoring 3D-cluster among these is reported along with the starting residue and the corresponding $c_{3D}$ score. The $c_{3D}$ score measures the significance of the intersection between a 3D cluster and the BPPS residue set. In addition to the strategy just described (termed “spherical expansion”), DARC allows either core expansion or hydrogen-bond-network expansion[74]. Core expansion sequentially adds the residue closest to a residue within the cluster’s “core”. This core is defined as the starting residue $R$ plus all cluster residues whose distance to their $k$th closest cluster residue is less than $R$’s distance to its $k$th closest cluster residue (with $k=7$ by default; this was selected empirically to avoid both spherical- and tentacle-shaped clusters.) In this case, the cluster typically expands less symmetrically. Hydrogen-bond-network expansion sequentially adds a residue forming the closest sidechain-to-sidechain or sidechain-to-backbone hydrogen bond with a cluster residue.

<table>
<thead>
<tr>
<th>MSA</th>
<th>PyMOL</th>
<th>mismatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>dkyellow</td>
<td>yellow1</td>
<td>yellow0</td>
</tr>
<tr>
<td>red</td>
<td>red1</td>
<td>red0</td>
</tr>
<tr>
<td>orange</td>
<td>orange1</td>
<td>orange0</td>
</tr>
<tr>
<td>magenta</td>
<td>magenta1</td>
<td>magenta0</td>
</tr>
<tr>
<td>green</td>
<td>green1</td>
<td>green0</td>
</tr>
<tr>
<td>cyan</td>
<td>cyan1</td>
<td>cyan0</td>
</tr>
<tr>
<td>blue</td>
<td>blue1</td>
<td>blue0</td>
</tr>
<tr>
<td>purple</td>
<td>purple1</td>
<td>purple0</td>
</tr>
<tr>
<td>teal</td>
<td>teal1</td>
<td>teal0</td>
</tr>
<tr>
<td>brown</td>
<td>brown1</td>
<td>brown0</td>
</tr>
<tr>
<td>gray</td>
<td>gray1</td>
<td>gray0</td>
</tr>
</tbody>
</table>
19. Appendix 7: Contrast alignments for bacterial clamp loader subunits.

**A. γ + δ' AAA+ domains:**

![Alignment diagram](image)

**B. AAA+**

![Alignment diagram](image)

**C. γ + δ AAA+**

![Alignment diagram](image)

**D. Clamp loader γ AAA+ domains:**

![Alignment diagram](image)

**E. γ + δ' AAA+**

![Alignment diagram](image)

Figure 19.1. DARC-generated alignments highlighting all residues conserved in γ and δ' clamp loader proteins and residues distinctive of the AAA+ superfamily, of the γ + δ' subgroup, and of γ but not δ'. These are shown using five versions of the...
same representative set of \( \gamma \) proteins (in panels A-E) and of \( \delta' \) proteins (in panels A-C). Residues are highlighted to indicate amino acid biochemical properties based on the following color code: red font with yellow highlight, non-polar (AVILMWFY); blue font with yellow highlight, cysteine (C); red, acidic (DE); cyan, basic (KR); magenta, polar (STNQ); green, glycine (G); blue, histidine (H); black, proline (P). Non-conserved positions in panels A and D and non-pattern residues in panels B,C and E are shown in gray font. The leftmost columns in panels B,C and E give the NCBI sequence identifiers; these are colored the same as the residue sidechains in Figure 9.1. A. Alignment highlighting all \( \gamma + \delta' \) conserved residues. Those sequences above the line within the alignment correspond to representative \( \gamma \) proteins from the (distinct) phyla denoted in the leftmost column; the first sequence corresponds to the E. coli \( \gamma \) subunit (pdb_id: 3glfB). Those sequences below the line correspond to representative \( \delta' \) proteins from distinct phyla, the first sequence of which corresponds to the E. coli \( \delta' \) subunit (pdb_id: 3glfE), which was used as the DARC query. The positions listed at the bottom correspond to the E. coli \( \gamma \) subunit. B. BPPS contrast alignment showing the same sequences as in panel (a), but highlighting only those residues most distinctive of the AAA+ superfamily. The heights of the red bars above each highlighted column estimate the selective pressure imposed on pattern residues at that position using a semi-logarithmic scale. Directly below the aligned sequences, the characteristic AAA+ residues at each position are shown and, directly below these, corresponding frequencies are given in integer tenths; a ‘7’, for example, indicates that the corresponding residue occurs in 70-80% of the 452,949 AAA+ sequences in the alignment. Below this is shown the residue positions and sequence of the E. coli \( \gamma \) subunit (with the Thr 165 residue that was mutated to Val highlighted in red), and shown below these are predicted secondary structure elements (symbol: H, helix; E, strand), helix and strand designations, and AAA+ structural motifs (red font) and putative clamp binding loops C1 and C2 (green font). Secondary structure assignments were calculated for the E. coli \( \gamma \) subunit using DSSP [PMID 6667333]. C. BPPS contrast using the same format as in panel B to highlight those residues that most distinguish \( \gamma \) and \( \delta' \) subunits from other AAA+ proteins. D. DARC-generated alignment highlighting all residues conserved in \( \gamma \). E. DARC-generated alignment highlighting residues distinguishing \( \gamma \) subunits from \( \delta' \) subunits. A few of these are conserved in other catalytically active AAA+ ATPases; see panel B.
20. References:


83. Schrodinger, LLC. (2010).


NOTES:

RtcR bEBP: CARF relaces the receiver domain.