Obtaining statistical evidence for underlying protein molecular mechanisms

Andrew F. Neuwald

Institute for Genome Sciences and Department of Biochemistry & Molecular Biology,University of Maryland School of Medicine, HSF-I, Room 134, 20 Penn Street, Baltimore, MD 21201
Tel: 410-706-6724; Fax: 410-706-1482
E-mail: aneuwald@som.umaryland.edu

1. Overview

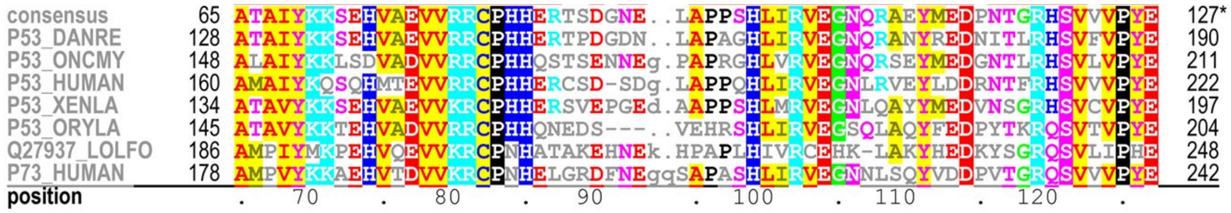
In the *Novum Organum* [2] Francis Bacon described, as the first step in applying the scientific method, the compilation of observational data, followed by the categorization of these observations and the generation of hypotheses. Our tools facilitate this process by extensively characterizing protein 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.

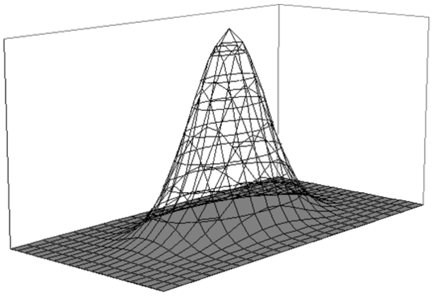
This document describes a suite of programs for identifying determinants of protein functional specificity that were developed in collaboration with statisticians Stephen F. Altschul (NCBI, NIH)(Gibbs sampler, GISMO, BPPS, STARC, SIPRIS, DARC, cCOMPASS), Jun S. Liu (Harvard)(Gibbs sampler, GISMO, BPPS),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

2. Statistics and probability in biology



```
consensus      65  ATAIYKKSEHVAEVVRRCPHHERTSDGNE..LAPPSHLIRVEGNQRAEYMEDPNTGRHSVVVPYE  127*
P53_DANRE     128  ATAIYKKSEHVAEVVRRCPHHERTPDGDN..LAPAGHLIRVEGNQRANYREDNITLRHSVFVPYE  190
P53_ONCMY     148  ALAIYKKLSDVADVVRRCPHHQSTSENNEg.PAPRGHLVRVEGNQRSEYMEDGNTLRHSVLVPYE  211
P53_HUMAN     160  AMAIYKQSQHMTEVVRRCPHHERCSD-SDg.LAPPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYE  222
P53_XENLA     134  ATAVYKKSEHVAEVVKRCPHHERSVEPGEd.AAPPSHLMRVEGNLQAYYMEDVNSGRHSVCVPYE  197
P53_ORYLA     145  ATAVYKKTEHVADVVRRCPHHQNEDS---..VEHRSHLIRVEGSQLAQYFEDPYTKRQSVTVPYE  204
Q27937_LOLFO  186  AMPIYMKPEHVQEVVKRCPNHATAKEHNEk.HPAPLHIVRCEHK-LAKYHEDKYSGRQSVLIPHE  248
P73_HUMAN     178  AMPVYKKAEHVTDVVKRCPNHELGRDFNEgqSAPASHLIRVEGNNLSQYVDDPVTGRQSVVVPYE  242
position            .   70   .   80   .   90   .   100   .   110   .   120   .
```

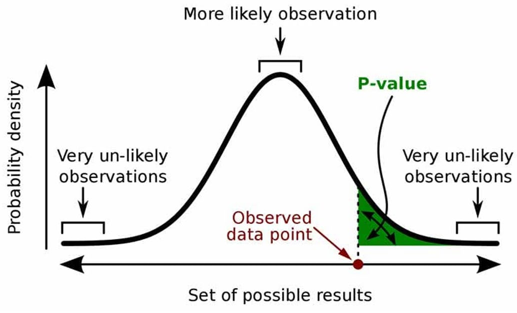
Due to the inherent variability of biological systems, a protein, such as the tumor suppressor p53 shown above, does not correspond to a specific sequence, but instead must be defined probabilistically as an evolutionary ensemble of sequences sharing a common cellular function. This ensemble can be modeled 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.



Representation of a protein probability distribution. The *x,y*-plane corresponds to a projection of the 'space' of all proteins onto 2-dimensions with similar sequences corresponding to points in the plane that are nearer to each other. The *z*-dimension plots the probability that each sequence (i.e., each point in the plane) is a p53 tumor suppressor protein. The point of highest probability may be viewed as a consensus sequence for p53.

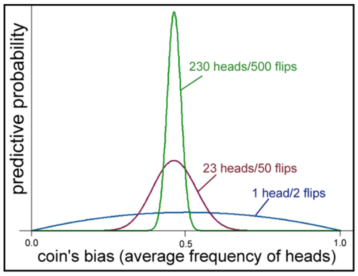
Analogy between our approach and classical genetics. Classical geneticists obtained evidence for pairs of linearly ordered genes based on patterns of inherited traits in the absence of any direct cytological or molecular data. Likewise, our approach obtains evidence for underlying proteins mechanisms 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.

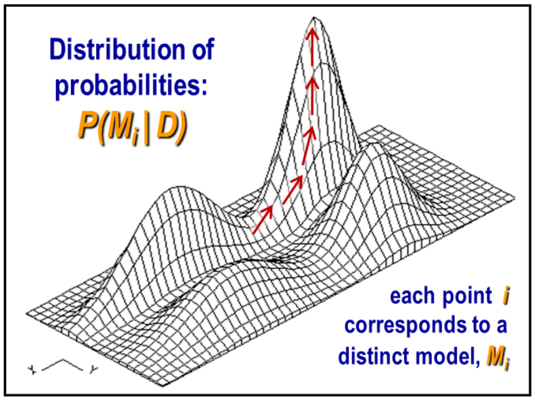


In contrast, a Bayesian assigns probabilities to hypotheses, arguing that all that we know for certain is the observed data; we can't observe the underlying model generating the data. Bayesian statistics incorporate prior probabilities and updates model probabilities as more data become available. Prior 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.

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.



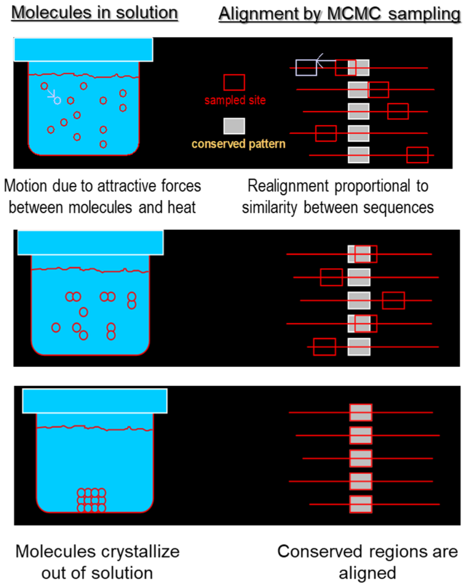
Bayesian version of the scientific method. Our tools are largely based on Bayesian **Markov chain Monte Carlo (MCMC)** sampling, an approach that considers multiple alternative models concurrently: Instead of a single hypothesis regarding a specific model (termed M_i) being consistent or inconsistent 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., $P(M_{xj} | D)$ 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.


Distribution of probabilities:
$P(M_i | D)$
each point i corresponds to a distinct model, M_i
x
y

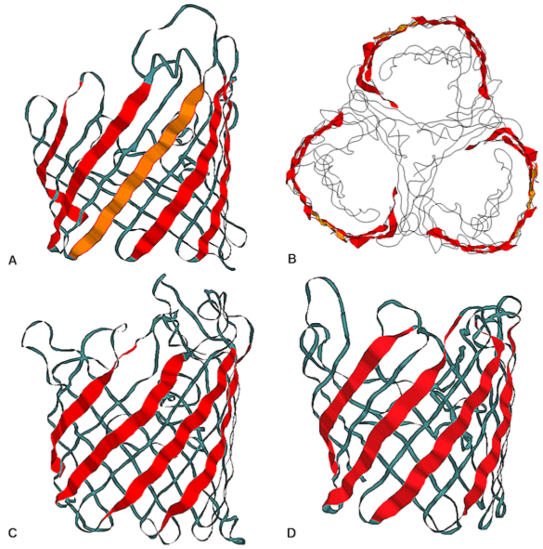
3. Multiple Sequence Alignment (the Gibbs sampler and GISMO)

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.

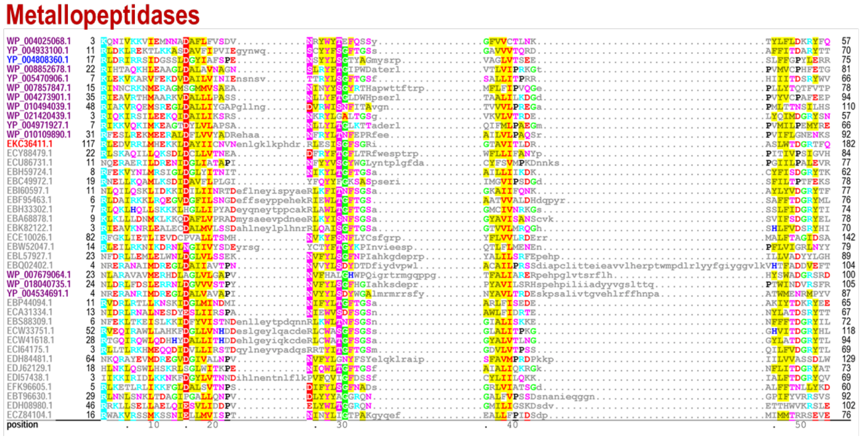

Molecules in solution
Alignment by MCMC sampling
sampled site
conserved pattern
Motion due to attractive forces between molecules and heat
Realignment proportional to similarity between sequences
Molecules crystallize out of solution
Conserved regions are aligned

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 crystalize out of solution and the conserved regions are aligned.

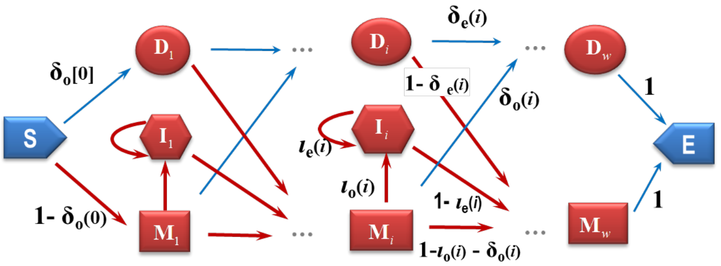


MCMC sampling and assignment uncertainty. An additional advantage of MCMC sampling is that it can provide a measure of model uncertainty as illustrated by the flipped coin example above. This involves continuing to sample among alternative models after convergence, in which case uncertain aligned blocks 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_x alternative (slightly different) HMMs having generated the input sequence data D (i.e., it computes $P(M_{i,j} \mid D)$ 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).

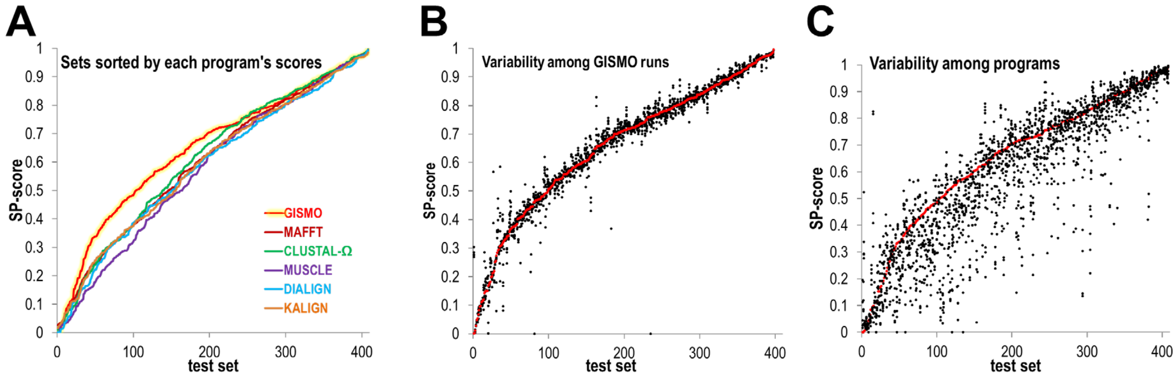
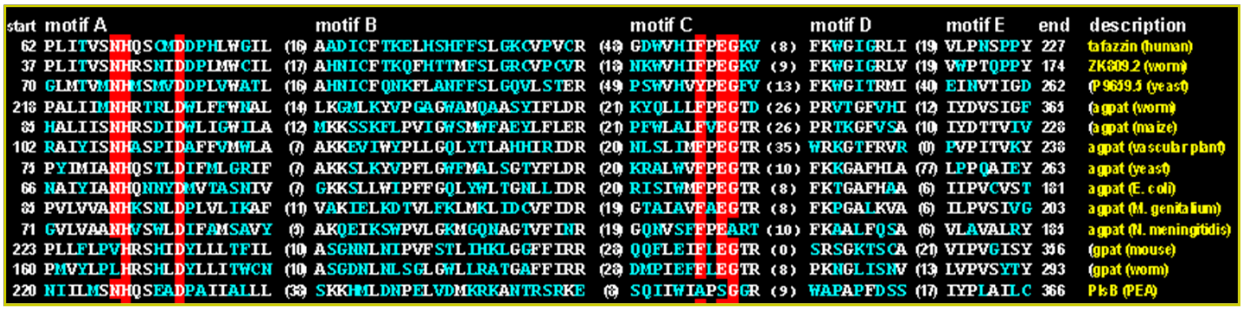


Figure 3.1. Alignment quality among various MSA programs based on their SP-scores, which vary from 0 (no correctly aligned sequence pairs) to 1 (all pairs aligned correctly). **A**. The sorted (lowest to highest) SP-scores obtained by six MSA programs. **B**. Run-to-run variability in SP-scores over 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.

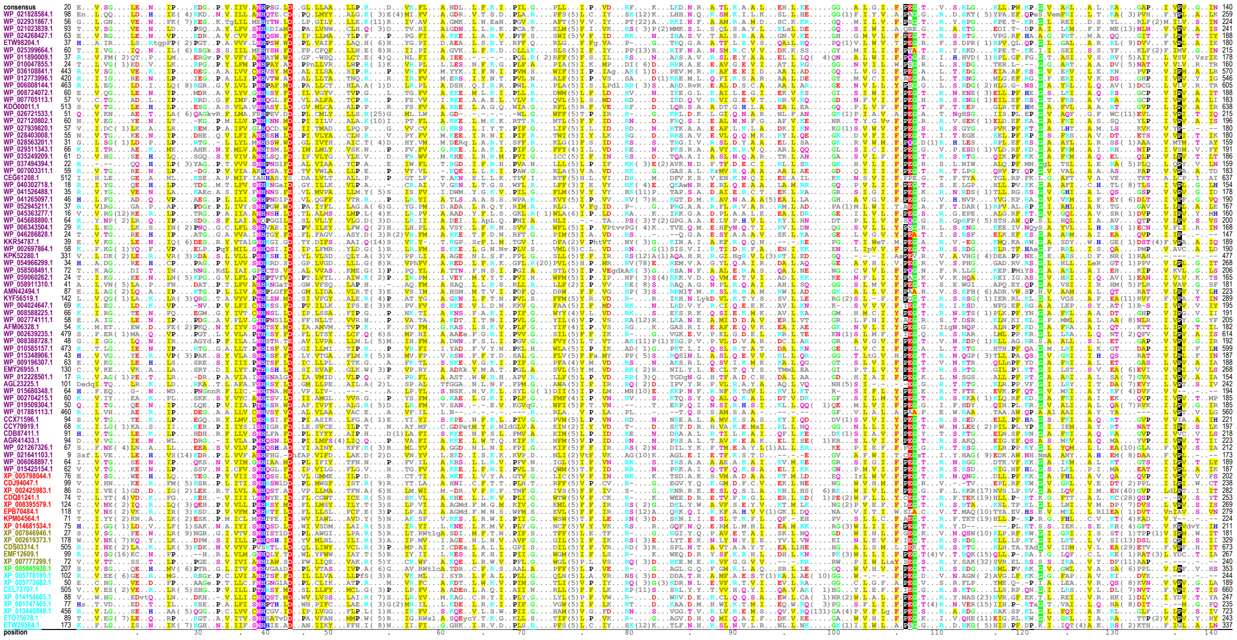
Version 2 of GISMO can run a user-specified number of threads in parallel, which allows for a more thorough exploration of the posterior probability distribution over possible HMMs. Rather than speeding up the program, this is designed to improve MSA accuracy by improving its ability to find a more nearly 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:

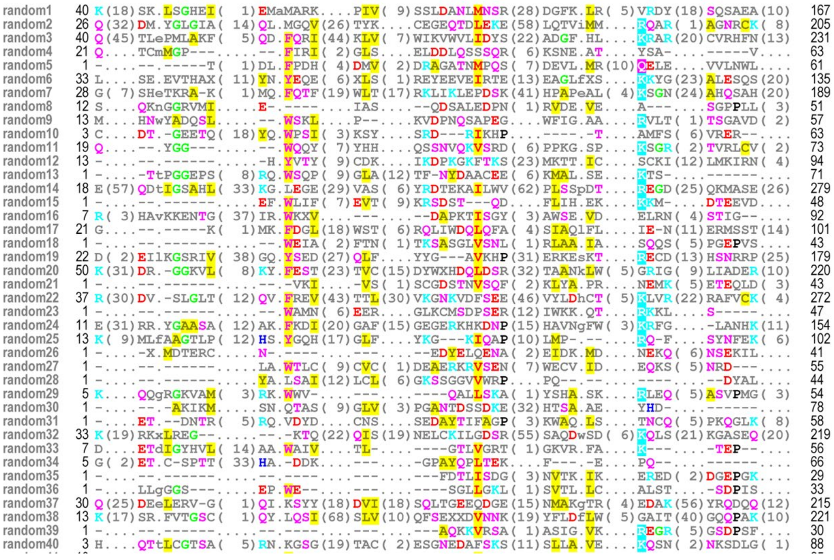


start	motif A		motif B		motif C		motif D		motif E	end	description
62	PLITVSNHQSCMDDPHLWGIL	(16)	AADICFTKELHSHFFSLGKCVPVCR	(48)	GDWVHIFPEGKV	(8)	FKWGIGRLI	(19)	VLPNSPPY	227	tafazzin (human)
37	PLITVSNHRSNIDDPLMWCIL	(17)	AHNICFTKQFHTTMFSLGRCVPCVR	(13)	NKWVHIFPEGKV	(9)	FKWGIGRLV	(19)	VWPTQPPY	174	ZK809.2 (worm)
70	GLMTVMNHMSMVDDPLVWATL	(16)	AHNICFQNKFLANFFSLGQVLSTER	(49)	PSWVHVYPEGFV	(13)	FKWGITRMI	(40)	EINVTIGD	262	(P9659.5 (yeast)
218	PALIIMNHRTRLDWLFFWNAL	(14)	LKGMLKYVPGAGWAMQAASYIFLDR	(21)	KYQLLLFPEGTD	(26)	PRVTGFVHI	(12)	IYDVSIGF	365	(agpat (worm)
85	HALIISNHRSDIDWLIGWILA	(12)	MKKSSKFLPVIGWSMWFAEYLFLER	(21)	PFWLALFVEGTR	(26)	PRTKGFVSA	(10)	IYDTTVIV	228	(agpat (maize)
102	RAIYISNHASPIDAFFVMWLA	(7)	AKKEVIWYPLLGQLYTLAHHIRIDR	(20)	NLSLIMFPEGTR	(35)	WRKGTFRVR	(0)	PVPITVKY	238	agpat (vascular plant)
75	PYIMIANHQSTLDIFMLGRIF	(7)	AKKSLKYVPFLGWFMALSGTYFLDR	(20)	KRALWVFPEGTR	(10)	FKKGAFHLA	(77)	LPPQAIEY	263	agpat (yeast)
66	NAIYIANHQNNYDMVTASNIV	(7)	GKKSLLWIPFFGQLYWLTGNLLIDR	(20)	RISIWMFPEGTR	(8)	FKTGAFHAA	(6)	IIPVCVST	181	agpat (E. coli)
85	PVLVVANHKSNLDPLVLIKAF	(11)	VAKIELKDTVLFKLMKLIDCVFIDR	(19)	GTAIAVFAEGTR	(8)	FKPGALKVA	(6)	ILPVSIVG	203	agpat (M. genitalium)
71	GVLVAANHVSWLDIFAMSAVY	(3)	AKQEIKSWPVLGKMGQNAGTVFINR	(19)	GQNVSFFPEART	(10)	FKAALFQSA	(6)	VLAVALRY	185	agpat (N. meningitidis)
223	PLLFLPVHRSHIDYLLLTFIL	(10)	ASGNNLNIPVFSTLIHKLGGFFIRR	(28)	QQFLEIFLEGTR	(0)	SRSGKTSCA	(21)	VIPVGISY	356	(gpat (mouse)
160	PMVYLPLHRSHLDYLLITWCN	(10)	ASGDNLNLSGLGWLLRATGAFFIRR	(28)	DMPIEFFLEGTR	(8)	PKNGLISNV	(13)	LVPVSYTY	293	(gpat (worm)
220	NIILMSNHQSEADPAIIALLL	(33)	SKKHMLDNPELVDMKRKANTRSRKE	(8)	SQIIWIAPSGGR	(9)	WAPAPFDSS	(17)	IYPLAILC	366	PlsB (PEA)

Among the sequences found was tafazzin, which is associated with Barth syndrome, an inherited cardiomypathic 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:



```
random1   40 K(18)SK.LSGHEI(  1)EMaMARK....PIV( 9)SSLDANLMNSR(28)DGFK.LR( 5)VRDY(18)SQSAEA(10)   167
random2   26 Q(32)DM.YGLGIA( 14)QL.MGQV(26)TYK....CEGEQTDLEKE(58)LQTViMM....RQAR( 1)AGNRCK( 8)   205
random3   40 Q(45)TLePMLAKF(  5)QD.FQRI(44)KLV( 7)WIKVWVLIDYS(22)ADGF.HL....KRAR(20)CVRHFN(13)   231
random4   21 Q....TCmMGP---......--.FIRI( 2)GLS....ELDDLQSSSQR( 6)KSNE.AT....YSA-....-----V....   63
random5    1 -.....--.-----T(  1)DL.FPDH( 4)DMV( 2)DRAGATNMPQS( 7)DEVL.MR(10)QELE....VVLNWL....   61
random6   33 L....SE.EVTHAX( 11)YN.YEQE( 6)XLS( 1)REYEEVEIRTE(13)EAGLfXS....KKYG(23)ALESQS(20)   135
random7   28 G( 7)SHeTKRA-K(  1)MQ.FQTF(19)WLT(17)RKLIKLEPDSK(41)HPAPeAL( 4)KSGN(24)AHQSAH(20)   189
random8   12 S....QKnGGRVMI.....E-.----....IAS....--QDSALEDPN( 1)RVDE.VE....A---....SGPPLL( 3)   51
random9   13 M....HNwYADQSL......--.WSKL....P--....KVDPNQSAPEG....WFIG.AA....RVLT( 1)TSGAVD( 2)   57
random10   3 C....DT.-GEETQ( 18)YQ.WPSI( 3)KSY....SRD---RIKHP....----.-T....AMFS( 6)VRER--....   63
random11  19 Q.....--.YGG---......--.WQQY( 7)YHH....QSSNVQKVSRD( 6)PPKG.SP....KSGR( 2)TVRLCV( 2)   73
random12  13 -.....--.-------......-H.YVTY( 9)CDK....IKDPKGKFTKS(23)MKTT.IC....SCKI(12)LMKIRN( 4)   94
random13   1 -.....-TtPGGEPS(  8)RQ.WSQP( 9)GLA(12)TF-NYDAACEE( 6)KMAL.SE....KTS-....------....   71
random14  18 E(57)QDtIGSAHL( 33)KG.LEGE(29)VAS( 6)YRDTEKAILWV(62)PLSSpDT....REGD(25)QKMASE(26)   279
random15   1 -.....--.-------.....EF.WLIF( 7)EVT( 9)KRSDST---QD....FLIH.EK....KKM-....DTEEVD....   48
random16   7 R( 3)HAvKKENTG( 37)IR.WKXV....---....--DAPKTISGY( 3)AWSE.VD....ELRN( 4)STIG--....   92
random17  21 G....--.-----K(  1)MK.FDGL(18)WST( 6)RQLIWDQLQFA( 4)SIAQlFL....IE-N(11)ERMSST(14)   101
random18   1 -.....--.-------......--.WEIA( 2)FTN( 1)TKSASGLVSNL( 1)RLAA.IA....SQQS( 5)PGEPVS....   43
random19  22 D( 2)EIlKGSRIV( 38)GQ.YSED(27)QLF....YYG---AVKHP(31)ERKEsKT....RECD(13)HSNRRP(25)   179
random20  50 K(31)DR.-GGKVL(  8)KY.FEST(23)TVC(15)DYWXHDQLDSR(32)TAANkLW( 5)GRIG( 9)LIADER(10)   220
random21   1 -.....--.-------......--.-VKI....-VS( 1)SCGDSTNVSQF( 2)KLYA.PR....NEMK( 5)ETEQLD( 3)   43
random22  37 R(30)DV.-SLGLT( 12)QV.FREV(43)TTL(30)VKGNKVDFSEE(46)VYLDhCT( 5)KLVR(22)RAFVCK( 4)   272
random23   1 -.....--.-------......--.WAMN( 6)EER....GLKCMSDPSER(12)IWKK.QT....RKKL....S-----....   47
random24  11 E(31)RR.YGAASA( 12)AK.FKDI(20)GAF(15)GEGERKHKDNP(15)HAVNgFW( 3)KRFG....-LANHK(11)   154
random25  13 K( 9)MLfAAGTLP( 12)HS.YGQH(17)GLF....YKG---KIQAP(10)LMP-.--....RQ-F....SYNFEK( 6)   102
random26   1 -.....-X.MDTERC.....N-.----....---....--EDYELQENA( 2)EIDK.MD....NEKQ( 6)NSEKIL....   41
random27   1 -.....--.-------.....LA.WTLC( 9)CVC( 1)DEAERKRVSEN( 7)WECV.ID....EQKS( 5)NRD---....   55
random28   1 -.....--.-------.....YA.LSAI(12)LCL( 6)GKSSGGVVWRP....PQ--.--....----....--DYAL....   44
random29   5 K....QQgRGKVAM(  3)RK.WWV-....---....----QALLSKA( 1)YSHA.SK....RLEQ( 5)ASVPMG( 3)   54
random30   1 -.....--.-AKIKM.....SN.QTAS( 9)GLV( 3)PGANTDSSDKE(32)HTSA.AE....YHD-....------....   78
random31   1 -....ET.--DNTR(  5)RQ.VDYD....CNS....SEDAYTIFAGP( 3)KWAQ.LS....TNCQ( 5)PKQGLK( 8)   58
random32  33 K(19)RKxLREG--......--.-KTQ(22)QIS(19)NELCKILGDSR(55)SAQDwSD( 6)KQLS(21)KGASEQ(20)   219
random33   7 D....ETdIGYHVL( 14)AA.WAIV....TL-....----GTLVGRT( 1)GKVR.FA....K---....-TEP--....   56
random34   5 G( 2)ET.C-SPTT( 33)HA.-DDK....---....-GPAYQPLTEK....F---.-E....PQ--....------....   66
random35   1 -.....--.-------......--.----....---....---FDTLISDG( 3)NVTK.IK....ERED( 2)DGEPGK....   29
random36   1 -....LLgGGS---.....EP.WE--....---....----SGLLGKL( 1)SVTL.LC....ALST....-SDPIS....   33
random37  30 Q(25)DEeLERV-G(  1)QI.KSYY(18)DVI(18)SQLTGEEQDGE(15)NMAKgTR( 4)EDAK(56)YRQDQQ(12)   215
random38  13 K(17)SR.FVTGSC(  1)QY.LQSI(68)SLV(10)QFSEXXDVNNK(19)YFLDfLW( 5)GAIT(40)GQQPAK(10)   221
random39   1 -.....--.-------......--.----....---....---AQKKVRSA( 1)ASIG.VK....REGR( 5)GSDPSF....   30
random40   3 H....QTtLCGTSA(  5)RN.KGSG(19)TAC( 2)ESGNEDAFSKS(11)SLLA.VE....KQSN( 2)NKSDLG( 1)   88
```

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?

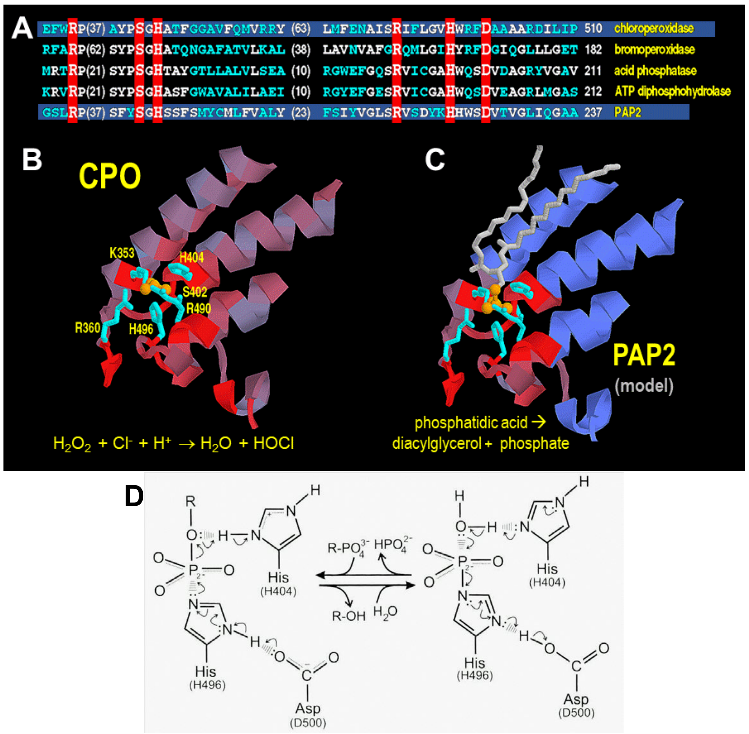

A
EFWRP(37) AYPSGHATFGGAVFQMVRRY (63) LMFENAISRIFLGVHWRFDAAAARDILIP 510 chloroperoxidase
RFARP(62) SYPSGHATQNGAFATVLKAL (38) LAVNVAFGRQMLGIHYRFDGIQGLLLGET 182 bromoperoxidase
MRTRP(21) SYPSGHTAYGTLLALVLSEA (10) RGWEFGQSRVICGAHWQSDVDAGRYVGAV 211 acid phosphatase
KRVRP(21) SYPSGHASFGWAVALILAEI (10) RGYEFGESRVICGAHWQSDVEAGRLMGAS 212 ATP diphosphohydrolase
GSLRP(37) SFYSGHSSFSMYCMLFVALY (23) FSIYVGLSRVSDYKHHWSDVTVGLIQGAA 237 PAP2
B
CPO
K353
H404
S402
R490
R360
H496
H_2O_2 + Cl^- + H^+ → H_2O + HOCl
C
PAP2
(model)
phosphatidic acid →
diacylglycerol + phosphate
D
R
O
H
N
N
H
O
P
O
O
2-
N
N
H
O
O
C
His
(H404)
His
(H496)
Asp
(D500)
$R\text{-}PO_4^{3-}$ HPO_4^{2-}
R-OH H_2O
H
O
H
N
N
H
O
P
O
O
2-
N
N
H
O
O
C
His
(H404)
His
(H496)
Asp
(D500)

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.

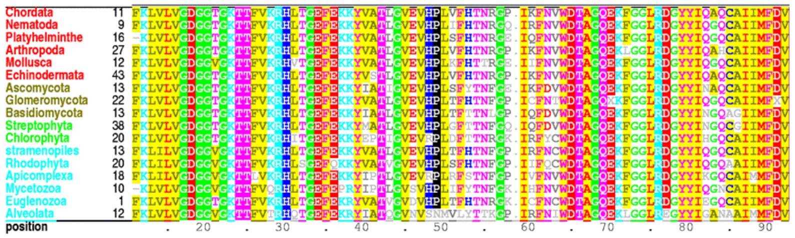
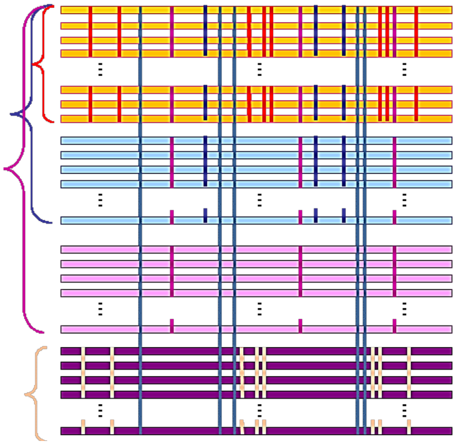

Chordata 11 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYVATLGVEVHPLVFHTNRGP.IKFNVWDTAGQEKFGGLRDGYYIQAQCAIIMFDV
Nematoda 9 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYVATLGVEVHPLIFHTNRGQ.IRFNVWDTAGQEKFGGLRDGYYIQGQCAIIMFDV
Platyhelminthe 16 -KLVLVGDGGTGKTTFVKRHLTGEFEKKYVATLGVEVHPLSFHTNRGE.IQFNVWDTAGQEKFGGLRDGYYIQGQCAIIMFDV
Arthropoda 27 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYVATLGVEVHPLVFHTNRGP.IKFNVWDTAGQEKLGGLRDGYYIQAHCAIIMFDV
Mollusca 12 FKLVLVGDGGVGKTTFVKRHVTGEFEKKYVATLGVEVHPLKFHTTRGE.IIFNVWDTAGQEKFGGLRDGYYIQGQCAIIMFDV
Echinodermata 43 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYVSTLGVEVHPLVFHTNRGP.IKFNVWDTAGQEKFGGLRDGYYIQAQCAIIMFDV
Ascomycota 13 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYIATIGVEVHPLSFYTNFGE.IKFDVWDTAGQEKFGGLRDGYYINAQCAIIMFDV
Glomeromycota 22 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYVATLGVEVHPLTFHTNFGP.ICFNTWDTAGQXKFGGLRDGYYIQGQCAIIMFXV
Basidiomycota 13 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYIATLGVEVHPLTFTTNLGP.IQFDVWDTAGQEKFGGLRDGYYINGQAGIIMFDV
Streptophyta 38 FKLVLVGDGGTGKTTFVKRHLTGEFEKKYMATLGVEVHPLGFTTNFGQ.IQFDVWDTAGQEKFGGLRDGYYINGQCGIIMFDV
Chlorophyta 20 FKLVLVGDGGTGKTTFVKRHITGEFEKKYEPTIGVEVRPLDFTTNRGK.IRFYCWDTAGQEKFGGLRDGYYIHGQCAIIMFDV
stramenopiles 13 FKLILVGDGGVGKTTFVKRHLTGEFEKKYVATLGVEVHPLTFHTNFGP.IRFNCWDTAGQEKFGGLRDGYYIQGQCAIIMFDV
Rhodophyta 20 FKLILVGDGGVGKTTFVKRHLSGEFQKKYVATVGVEVHPLSFHTNFGP.IIFQCWDTAGQEKFGGLRDGYYIQGQAAIIMFDV
Apicomplexa 18 FKLILVGDGGVGKTTLVKRHLTGEFEKKYIPTLGVEVRPLRFSTNFGP.IVFNVWDTAGQEKFGGLRDGYYIKGQCAIMMFDV
Mycetozoa 10 -KLVLVGDGGVGKTTFVQRHLTGEFEPRYIPTLGVSVHPLIFYTNFGK.IHFNVWDTAGQEKFGGLRDGYYIQGNCAIIMFDV
Euglenozoa 1 FKLVLVGDGGTGKTTFVKRHLTGEFEKRYVATVGVDVHPLTFHTNRGK.ICFNCWDTAGQEKFGGLRDGYYIEGQCAIIMFDV
Alveolata 12 FKLVLVGDGGVGKTTFVTRHQTGEFEKRYIATQGVNVSNMVLYTTKGP.IRFNIWDTAGQEKLGGLREGYYIGANAAIMMFDV
position . 20 . 30 . 40 . 50 . 60 . 70 . 80 . 90

Fig. 4.2. Ran family sequence alignment. The leftmost column specifies each sequence's phylum; these are colored by eukaryotic 'kingdoms', as follows: metazoans, red; fungi, dark yellow; plants, green; protozoans, cyan.

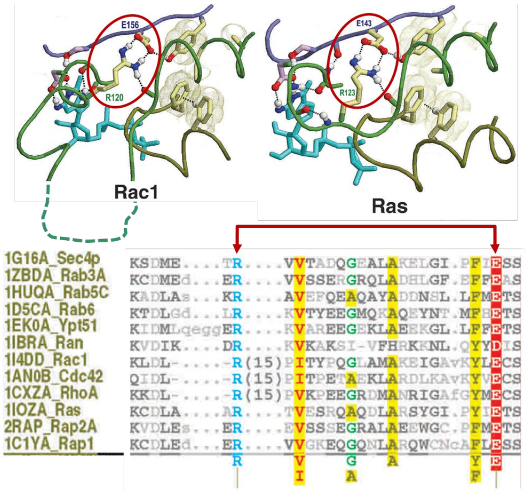
Regarding this second question, consider Ran GTPase, which is essential for the translocation of RNA and proteins through the nuclear pore complex and that shares a catalytic core with other P-loop GTPase. A sequence alignment of Ran proteins reveals a very high degree of conservation across diverse 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.

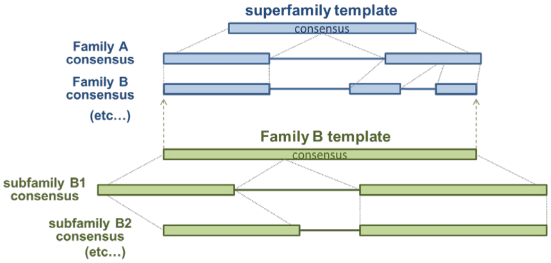


Likewise, within an alignment of all P-loop GTPases, there are residues conserved within each protein family as well as subgroup conserved residues that span multiple families; this is illustrated schematically in the figure on the right, where similarly colored horizontal bars correspond to families, 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.


E156
R120
E143
R123
Rac1
Ras
1G16A_Sec4p KSDME....TR....VVTADQGEALAKELGI.PFIESS
1ZBDA_Rab3A KCDMEd...ER....VVSSERGRQLADHLGF.EFFEAS
1HUQA_Rab5C KADLAs...KR....AVEFQEAQAYADDNSL.LFMETS
1D5CA_Rab6 KTDLGd...LR....KVTYEEGMQKAQEYNT.MFHETS
1EK0A_Ypt51 KIDMLqeggER....KVAREEGEKLAEEKGL.LFFETS
1IBRA_Ran KVDIK....DR....KVKAKSI-VFHRKKNL.QYYDIS
1I4DD_Rac1 KLDL-.....-R(15)PITYPQGLAMAKEIGAvKYLECS
1AN0B_Cdc42 QIDL-.....-R(15)PITPETAEKLARDLKAvKYVECS
1CXZA_RhoA KKDL-.....-R(15)PVKPEEGRDMANRIGAfGYMECS
1IOZA_Ras KCDLA....AR....TVESRQAQDLARSYGI.PYIETS
2RAP_Rap2A KVDLEs...ER....EVSSSEGRALAEEWGC.PFMETS
1C1YA_Rap1 KCDLEd...ER....VVGKEQGQNLARQWCNcAFLESS
R V G A Y E
I A F



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-formated 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 facilities 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:

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

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.acession2taxid.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 $\boldsymbol{k} \in \{A, B, V, M, F, E\}$ 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].

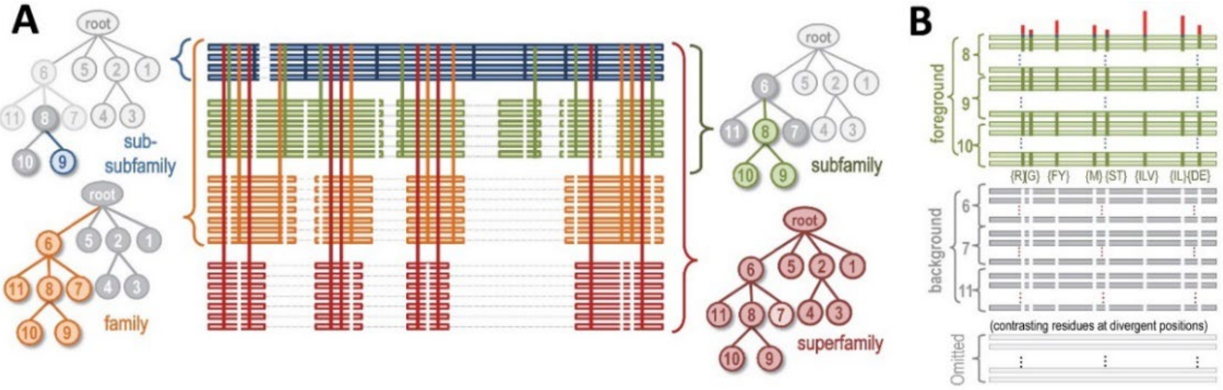
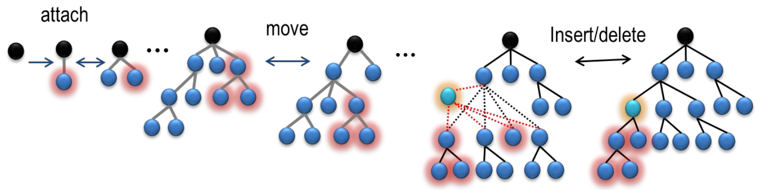


Figure 6.1. Basic features of a protein hiMSA. **A.** Schematic diagram of that part of an hiMSA corresponding to the lineage from the root node to leaf node 9. There is one such diagram for each leaf node in the hierarchy. Horizontal lines represent aligned sequences and are color-coded by level in 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:

Mode	Description
1	Initial hierarchical partitioning of MSA into subgroups
2	Create a hiMSA as in **Fig. 6.1** using a mode 1 checkpoint file
3	Create contrast alignments for a specified node’s lineage
A	Run modes 1-3 using default options with optional mapping of pattern residues to structures
Q	Perform a query-centric run in mode 1
E	Evaluate the consistency between two BPPS-generated hiMSAs
H	Run BPPS with a user curated hyperpartition (which need not correspond to a tree) and seed MSAs

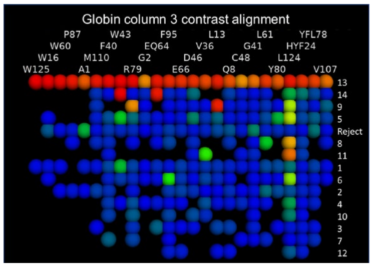


BPPS mode '1' is used when starting from an input MSA. In this mode, BPPS explores the 'space' of domain hierarchies by attaching, moving, inserting, and deleting nodes corresponding to functionally divergent subgroups. Each such hierarchy corresponds to a tree, but for reasons explained in the next 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 nats, in **nats per sequence (nps)** and in **nats 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 sigificance. Common user-modifiable BPPS parameters are the following:

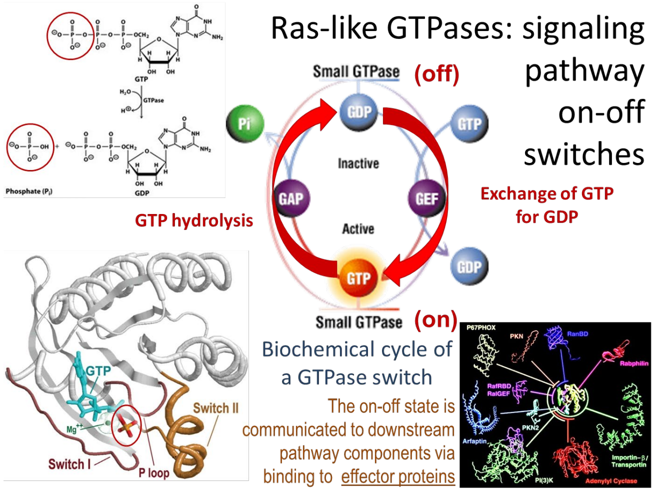
```
============== HyperPartition: ==============
      _Category_
Set:  1  2  3  4  5  6  7  8  9 10 11 12 13 14
  1:  +  -  -  -           -        -     -  -  1.Set1 (290)*
  2:  +  +  -  -           -        -     -  -  2.Set14 (77)
  3:  +  -  +  -           -        -     -  -  3.Set13 (59)
  4:  +  -  -  +  -  -  -  -        -     -  -  4.Set6 (317)*
  5:  +  -  -  +  +  -  -  -        -     -  -  _5.Set12 (75)
  6:  +  -  -  +  -  +  -  -        -     -  -  _6.Set11 (51)
  7:  +  -  -  +  -  -  +  -        -     -  -  _7.Set10 (112)
  8:  +  -  -  -           +  -  -  -     -  -  8.Set5 (337)*
  9:  +  -  -  -           +  +  -  -     -  -  _9.Set9 (203)
 10:  +  -  -  -           +  -  +  -     -  -  _10.Set8 (98)
 11:  +  -  -  -           -        +  -  -  -  11.Set4 (514)*
 12:  +  -  -  -           -        +  +  -  -  _12.Set7 (230)
 13:  +  -  -  -           -        -     +  -  13.Set3 (587)
 14:  +  -  -  -           -        -     -  +  14.Set2 (1442)
 15:  -                                         Rejected (609)
 =============================================
 1: 4392 2277  seqs (8223.4; 1.9 nps; 6.4 npws)(19 cols)
 2: 77 4315  seqs (809.021; 10.5 nps; 28.1 npws)(25 cols)
 3: 59 4333  seqs (1191.19; 20.2 nps; 49.5 npws)(25 cols)
 4: 555 3837  seqs (2431.73; 4.4 nps; 14.8 npws)(23 cols)
 5: 75 480  seqs (125.919; 1.7 nps; 8.6 npws)(19 cols)
 6: 51 504  seqs (226.983; 4.5 nps; 13.7 npws)(18 cols)
 7: 112 443  seqs (323.13; 2.9 nps; 10.3 npws)(20 cols)
 8: 638 3754  seqs (3443.37; 5.4 nps; 16.2 npws)(25 cols)
 9: 203 435  seqs (567.624; 2.8 nps; 10.2 npws)(24 cols)
10: 98 540  seqs (570.063; 5.8 nps; 18.8 npws)(25 cols)
11: 744 3648  seqs (4655.15; 6.3 nps; 25.9 npws)(25 cols)
12: 230 514  seqs (661.147; 2.9 nps; 14.0 npws)(25 cols)
13: 587 3805  seqs (4448.66; 7.6 nps; 29.5 npws)(25 cols)
14: 1442 2950  seqs (4569.38; 3.2 nps; 11.0 npws)(25 cols)
 ====== Total LPR = 30579.3 (300.0 K) (0/14 failed) ======
```

Option	Description
-heatmap	generate heatmaps showing, for each pattern, the degree of conservation among hierarchy nodes
-ichk	save intermediate checkpoint files during the run
-maxcol=<int>	set the maximum number of pattern positions per node to <int> (default: 25)
-minsize=<int>	set the minimum set size to <int> sequences (default: 50)
-minnats=<real>	set the minimum required nats per weighted sequence to <real> (default: 5.0)
-maxdepth=<int>	set the maximum tree depth to <int> (default: 2)
-run=P	Create additional output files (requires a checkpoint file)
-seed=<int>	Set the seed for the random number generator to <int>

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^{++} 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.



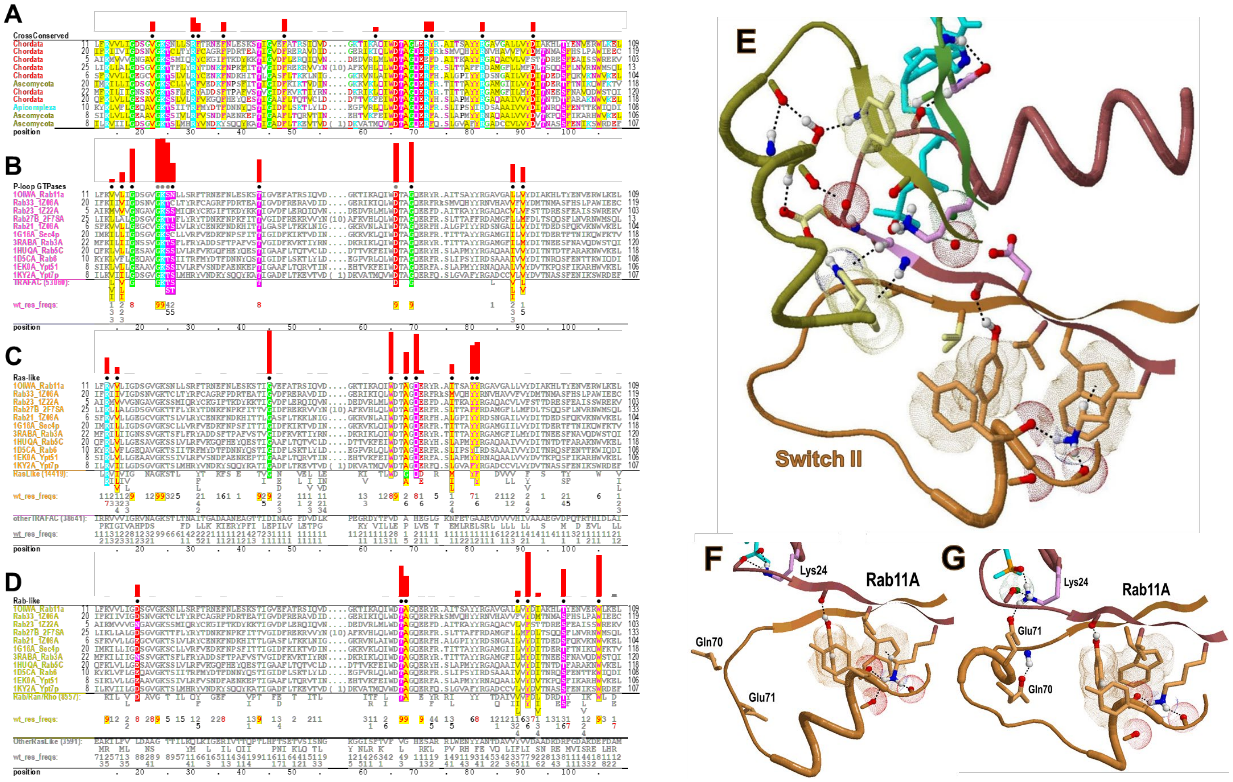
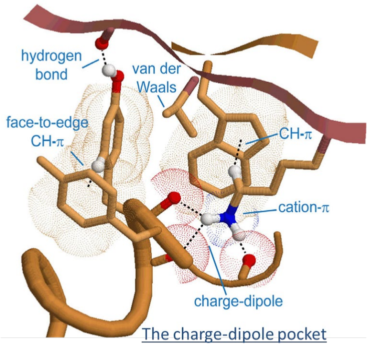


Figure 6.2. BPPS-1 analysis of Ras-like GTPases. **A-D**. Four contrast alignments highlighting pattern residues that most distinguish the subgroups to which Rab GTPases belong [33,34]. **A.** Alignment highlighting all residues that are conserved among the 11 aligned sequences representative of Rab-like 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 Mg^{++} 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 conformations (**F**, **G**) [38] 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.



The charge-dipole pocket

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 swII-CT 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}.

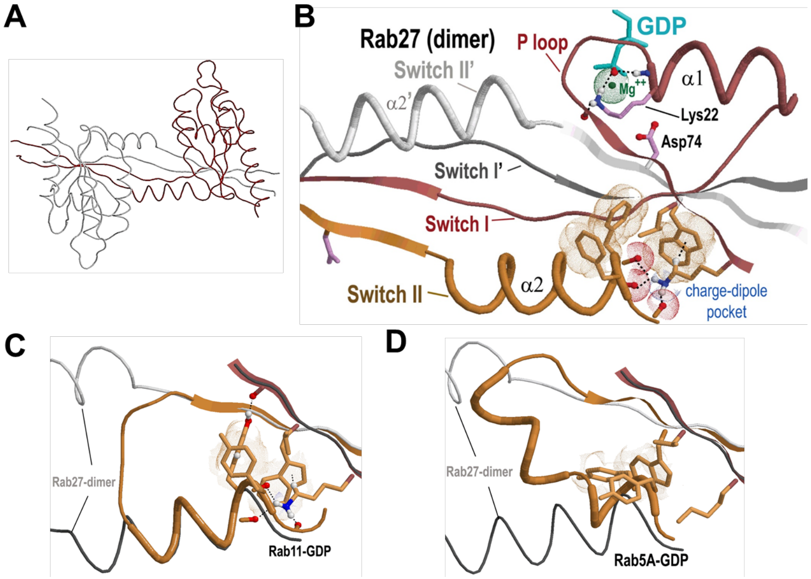
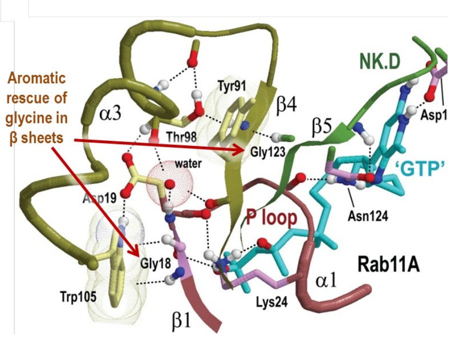
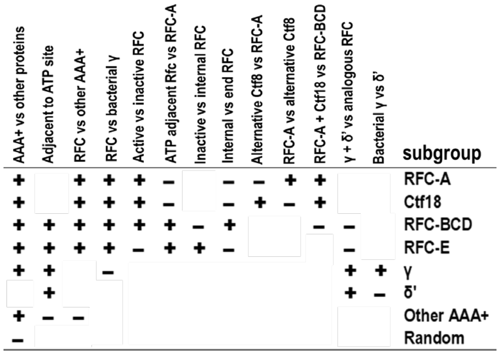


Figure 6.3. Insight from an unusual homodimeric configuration of Rab27. **A.** In the Rab27 homodimer {ref}, the switch I, switch II and inter-switch regions of each subunit are exchanged and protrude out like an antenna. **B.** In this configuration both subunits form a charge-dipole pocket. **C.** In the monomeric Rab11 GTPase formation of the charge-dipole pocket is also associated with an outward-directed switch II helix. **D.** In the monomeric Rab5 GTPase disruption of the charge-dipole pocket is associated with disruption of the outward-directed switch II helix. This correlation is seen for various Ras-like GTPases, including Arf, Arl, Sar, and Gα GTPases {ref}.



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).



AAA+ vs other proteins	Adjacent to ATP site	RFC vs other AAA+	RFC vs bacterial γ	Active vs inactive RFC	ATP adjacent Rfc vs RFC-A	Inactive vs internal RFC	Internal vs end RFC	Alternative Ctf8 vs RFC-A	RFC-A vs alternative Ctf8	RFC-A + Ctf18 vs RFC-BCD	γ + δ' vs analogous RFC	Bacterial γ vs δ'	subgroup
+		+	+	+	–		–	–	+	+			RFC-A
+		+	+	+	–		–	+	–	+			Ctf18
+	+	+	+	+	+	–	+			–	–		RFC-BCD
+	+	+	+	–	+	+	–				–		RFC-E
+	+		–								+	+	γ
	+										+	–	δ'
+	–	–											Other AAA+
–													Random

Hyperpartition for characterizing non-hierarchical BPPS constraints imposed on DNA clamp loader AAA+ domains.

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 subtrees 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**.

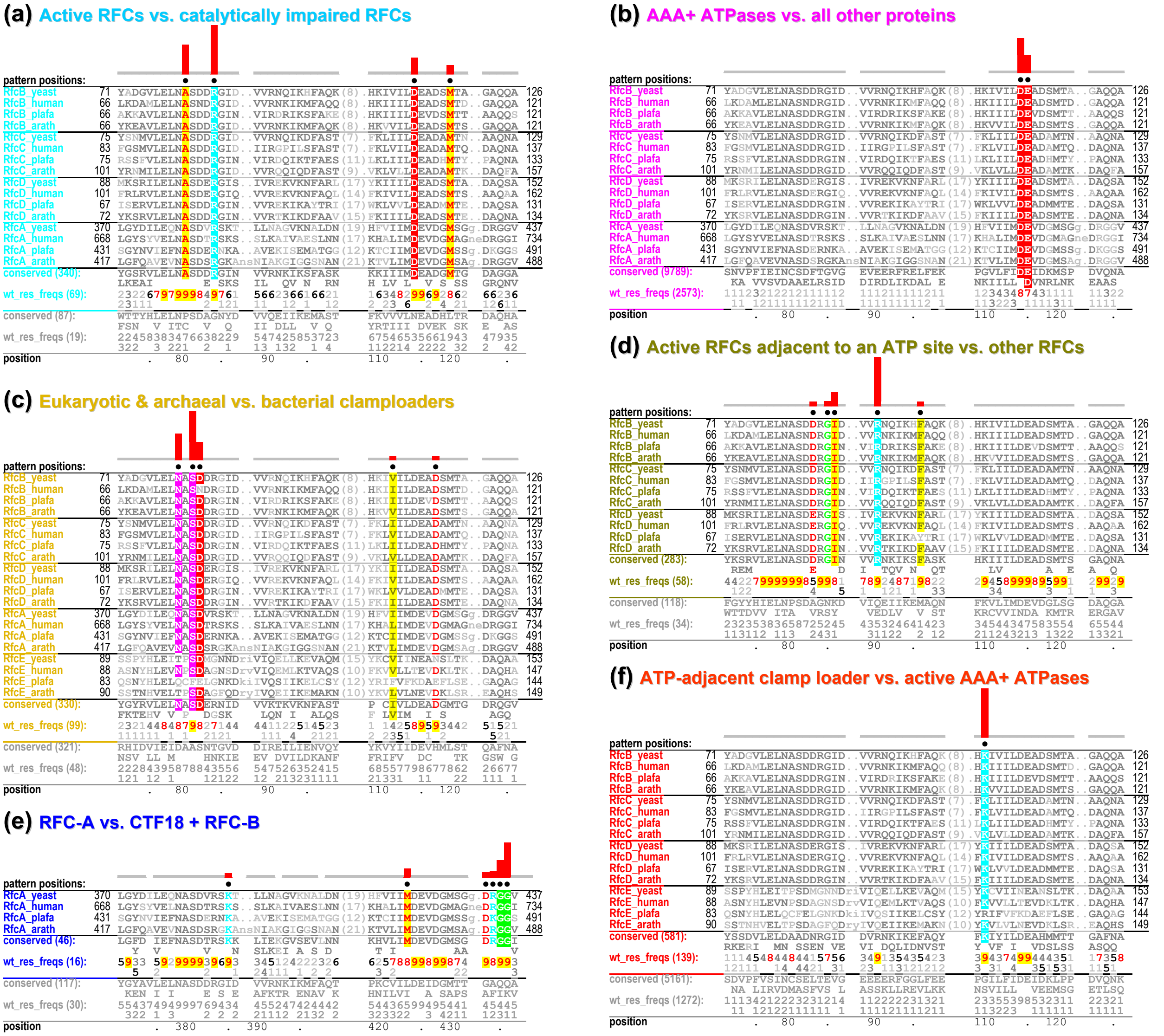
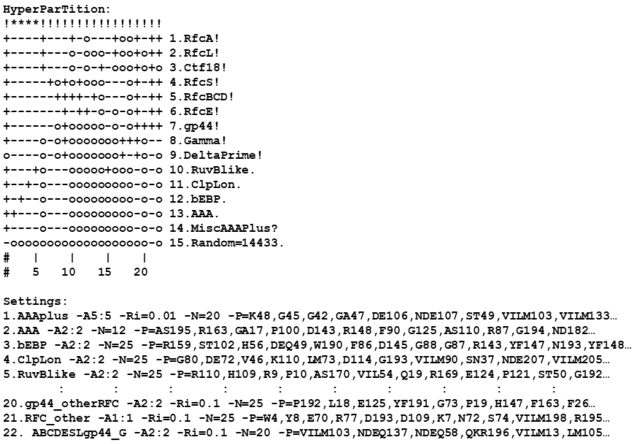

(a) Active RFCs vs. catalytically impaired RFCs
(b) AAA+ ATPases vs. all other proteins
(c) Eukaryotic & archaeal vs. bacterial clamploaders
(d) Active RFCs adjacent to an ATP site vs. other RFCs
(e) RFC-A vs. CTF18 + RFC-B
(f) ATP-adjacent clamp loader vs. active AAA+ ATPases
pattern positions:
conserved
wt_res_freqs
position

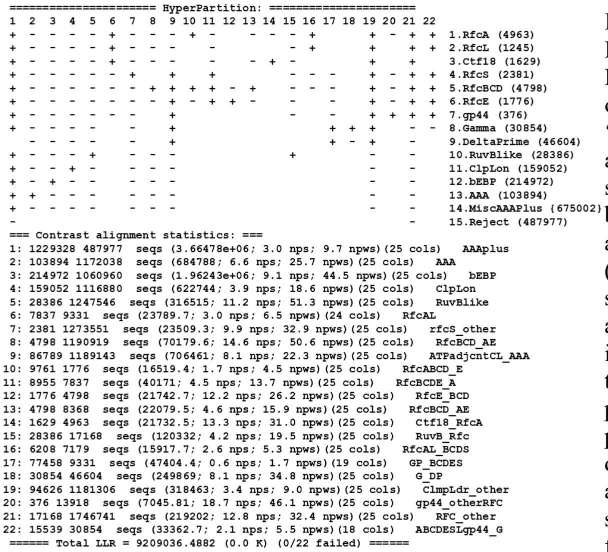
Fig. 6.4. BPPS H-mode generated contrast alignments showing sequence constraints distinctive of various types of replication factor C (RFC) clamp-loader subunits [44]. The sequences shown span a functionally crucial region of RFC subunits, within which most of the strongest constraints occur. Each alignment 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. (Background pattern residues and frequencies are not shown.) **(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 '!' and '*' 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.



```
HyperParTition:
!****!!!!!!!!!!!!!!!!!
+----+---+-o---+oo+-++ 1.RfcA!
+----+---o-ooo-+oo+o++ 2.RfcL!
+----+---o-o-+-ooo+o+o 3.Ctf18!
+-----+o+o+ooo---o+-++ 4.RfcS!
+------++++-+o---o+-++ 5.RfcBCD!
+-------+-++-o-o-o+-++ 6.RfcE!
+------o+ooooo-o-o++++ 7.gp44!
+----o-o+ooooooo+++o-- 8.Gamma!
o----o-o+ooooooo+-+o-o 9.DeltaPrime!
+---+o---ooooo+ooo-o-o 10.RuvBlike.
+--+-o---ooooooooo-o-o 11.ClpLon.
+-+--o---ooooooooo-o-o 12.bEBP.
++---o---ooooooooo-o-o 13.AAA.
+----o---ooooooooo-o-o 14.MiscAAAPlus?
-ooooooooooooooooooo-o 15.Random=14433.
#   |    |    |    |
#   5   10   15   20

Settings:
1.AAAplus -A5:5 -Ri=0.01 -N=20 -P=K48,G45,G42,GA47,DE106,NDE107,ST49,VILM103,VILM133…
2.AAA -A2:2 -N=12 -P=AS195,R163,GA17,P100,D143,R148,F90,G125,AS110,R87,G194,ND182…
3.bEBP -A2:2 -N=25 -P=R159,ST102,H56,DEQ49,W190,F86,D145,G88,G87,R143,YF147,N193,YF148…
4.ClpLon -A2:2 -N=25 -P=G80,DE72,V46,K110,LM73,D114,G193,VILM90,SN37,NDE207,VILM205…
5.RuvBlike -A2:2 -N=25 -P=R110,H109,R9,P10,AS170,VIL54,Q19,R169,E124,P121,ST50,G192…
        :       :       :       :       :       :       :       :       :       :
20.gp44_otherRFC -A2:2 -Ri=0.1 -N=25 -P=P192,L18,E125,YF191,G73,P19,H147,F163,F26…
21.RFC_other -A1:1 -Ri=0.1 -N=25 -P=W4,Y8,E70,R77,D193,D109,K7,N72,S74,VILM198,R195…
22. ABCDESLgp44_G -A2:2 -Ri=0.1 -N=20 -P=VILM103,NDEQ137,NDEQ58,QKR196,VILM13,LM105…
```



```
===================== HyperPartition: =====================
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
+  -  -  -  -  +  -  -  -  +  -     -  -  -  +        +  -  +  +  1.RfcA (4963)
+  -  -  -  -  +  -  -  -     -           -  +        +     +  +  2.RfcL (1245)
+  -  -  -  -  +  -  -  -     -     -  +  -           +     +     3.Ctf18 (1629)
+  -  -  -  -  -  +     +     +           -  -  -     +  -  +  +  4.RfcS (2381)
+  -  -  -  -  -  -  +  +  +  +  -  +     -  -  -     +  -  +  +  5.RfcBCD (4798)
+  -  -  -  -  -  -  -  +  -  +  +  -     -     -     +  -  +  +  6.RfcE (1776)
+  -  -  -  -  -  -     +                 -     -     +  +  +  +  7.gp44 (376)
+  -  -  -  -     -     +                       +  +  +     -  -  8.Gamma (30854)
   -  -  -  -     -     +                       +  -  +     -     9.DeltaPrime (46604)
+  -  -  -  +     -  -  -                 +           -     -     10.RuvBlike (28386)
+  -  -  +  -     -  -  -                             -     -     11.ClpLon (159052)
+  -  +  -  -     -  -  -                             -     -     12.bEBP (214972)
+  +  -  -  -     -  -  -                             -     -     13.AAA (103894)
+  -  -  -  -     -  -  -                             -     -     14.MiscAAAPlus {675002}
-                                                            -     15.Reject (487977)
=== Contrast alignment statistics: ===
 1: 1229328 487977  seqs (3.66478e+06; 3.0 nps; 9.7 npws)(25 cols)   AAAplus
 2: 103894 1172038  seqs (684788; 6.6 nps; 25.7 npws)(25 cols)   AAA
 3: 214972 1060960  seqs (1.96243e+06; 9.1 nps; 44.5 npws)(25 cols)   bEBP
 4: 159052 1116880  seqs (622744; 3.9 nps; 18.6 npws)(25 cols)   ClpLon
 5: 28386 1247546  seqs (316515; 11.2 nps; 51.3 npws)(25 cols)   RuvBlike
 6: 7837 9331  seqs (23789.7; 3.0 nps; 6.5 npws)(24 cols)   RfcAL
 7: 2381 1273551  seqs (23509.3; 9.9 nps; 32.9 npws)(25 cols)   rfcS_other
 8: 4798 1190919  seqs (70179.6; 14.6 nps; 50.6 npws)(25 cols)   RfcBCD_AE
 9: 86789 1189143  seqs (706461; 8.1 nps; 22.3 npws)(25 cols)   ATPadjcntCL_AAA
10: 9761 1776  seqs (16519.4; 1.7 nps; 4.5 npws)(25 cols)   RfcABCD_E
11: 8955 7837  seqs (40171; 4.5 nps; 13.7 npws)(25 cols)   RfcBCDE_A
12: 1776 4798  seqs (21742.7; 12.2 nps; 26.2 npws)(25 cols)   RfcE_BCD
13: 4798 8368  seqs (22079.5; 4.6 nps; 15.9 npws)(25 cols)   RfcBCD_AE
14: 1629 4963  seqs (21732.5; 13.3 nps; 31.0 npws)(25 cols)   Ctf18_RfcA
15: 28386 17168  seqs (120332; 4.2 nps; 19.5 npws)(25 cols)   RuvB_Rfc
16: 6208 7179  seqs (15917.7; 2.6 nps; 5.3 npws)(25 cols)   RfcAL_BCDS
17: 77458 9331  seqs (47404.4; 0.6 nps; 1.7 npws)(19 cols)   GP_BCDES
18: 30854 46604  seqs (249869; 8.1 nps; 34.8 npws)(25 cols)   G_DP
19: 94626 1181306  seqs (318463; 3.4 nps; 9.0 npws)(25 cols)   ClmpLdr_other
20: 376 13918  seqs (7045.81; 18.7 nps; 46.1 npws)(25 cols)   gp44_otherRFC
21: 17168 1746741  seqs (219202; 12.8 nps; 32.4 npws)(25 cols)   RFC_other
22: 15539 30854  seqs (33362.7; 2.1 nps; 5.5 npws)(18 cols)   ABCDESLgp44_G
 ====== Total LLR = 9209036.4882 (0.0 K) (0/22 failed) ======
```

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 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 ' ' 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**.

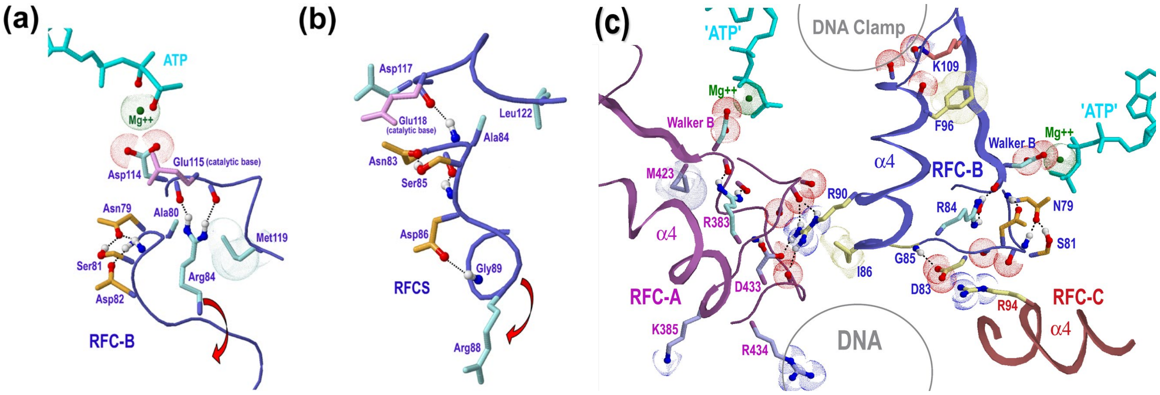


Figure 6.7. Structural features associated with the six categories of RFC constraints shown in **Fig. 6.4**. The central hole of the RFC complex, through which DNA presumably is thread, is located at the bottom of each panel. **(a)** Conformation of RFC-B when bound to ATP and the clamp [5]. In all four active RFC ATPases, the arginine corresponding to Arg84 in RFC-B forms hydrogen

bonds with 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 Auxilliary 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.

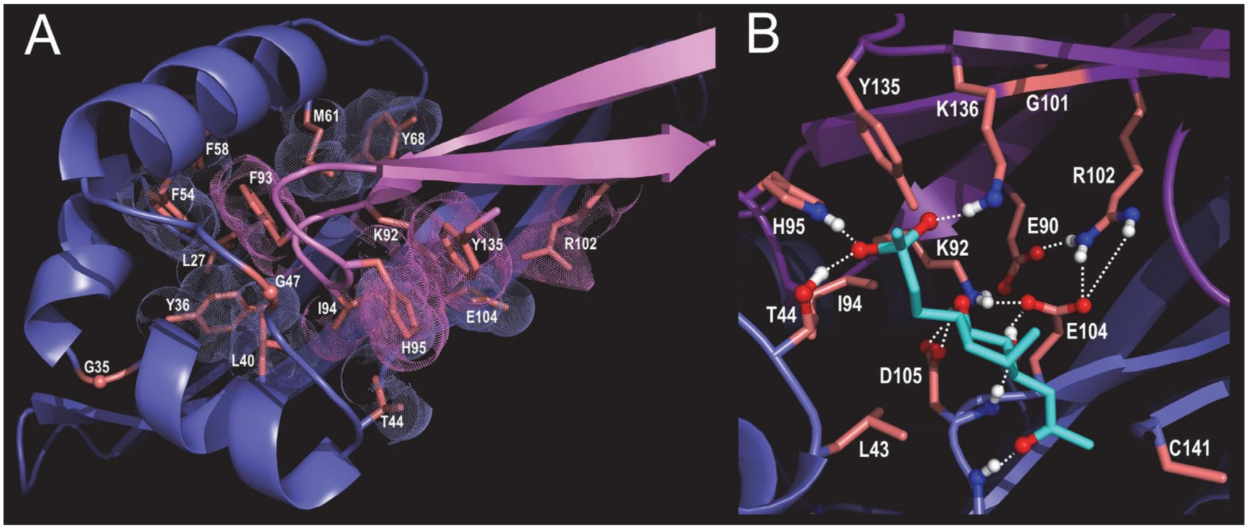


Figure 7.1. SIPRIS [46,47] estimates the statistical significance of all but two of the 25 residues most distinctive of the glucosamine-6-phosphate *N*-acetyltransferase (Gna1) family; these either (**A**) occur at the interface between homodimeric subunits (blue & pink cartoon traces) ($\boldsymbol{p} = \mathbf{8.5 \times 10^{-7}}$) or (**B**) interact with substrate (shown in cyan) ($\boldsymbol{p} = \mathbf{6.8 \times 10^{-5}}$) (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 (R^3) GTPases in **Fig. 7.2**.

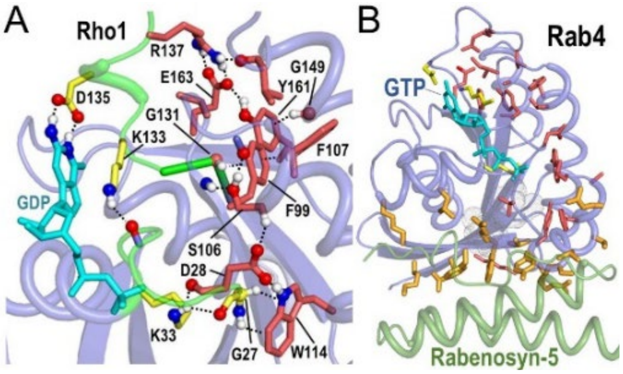


Fig. 7.2. SIPRIS defined GTPase clusters. **A**. Rab/Rho/Ran (R^3) hydrogen-bond network (red sidechains) in Rho1 (pdb_id: 3refB [49]; $\boldsymbol{p} = \mathbf{6.2 \times 10^{-5}}$). This includes a salt bridge (R137-E163) and CH-π interactions (G27-W114; G131-F99) hypothesized to modulate nucleotide 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 GTPases. **B**. A Rab4 R^3 hydrogen bond network (red sidechains) ($\boldsymbol{p} = \mathbf{2.6 \times 10^{-6}}$) and a Rab-specific cluster (orange sidechains) ($\boldsymbol{p} = \mathbf{2.9 \times 10^{-8}}$) contacting the Rabenosyn-5 effector (pdb_id: 1z0kA [50]).

A BPPS-SIPRIS comparative analysis of exonuclease-endonuclease phosphatases (EEPs) [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'.

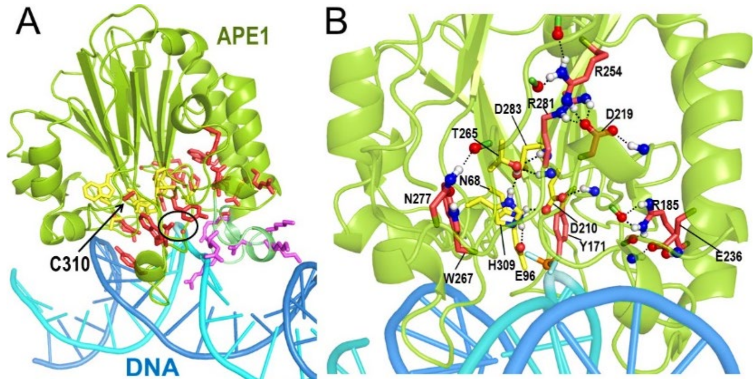


Fig. 7.3. BPPS-SIPRIS analysis of APE1 endonuclease bound to DNA with an abasic site analog (pdb_id: 5dfi) [52]. APE1 incises an abasic site phosphodiester backbone in DNA. A proposed mechanism involves superfamily-conserved active site residues forming hydrogen bonds with abasic site phosphate group oxygen atoms [53]. **A**. A cluster of EEP superfamily residues (yellow sidechains; $\boldsymbol{p = 5.2\times10^{-6}}$) and a hydrogen bond network of exoIII-AP-endo family residues (red sidechains; $\boldsymbol{p = 1.6\times10^{-6}}$), 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 exoIII-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.

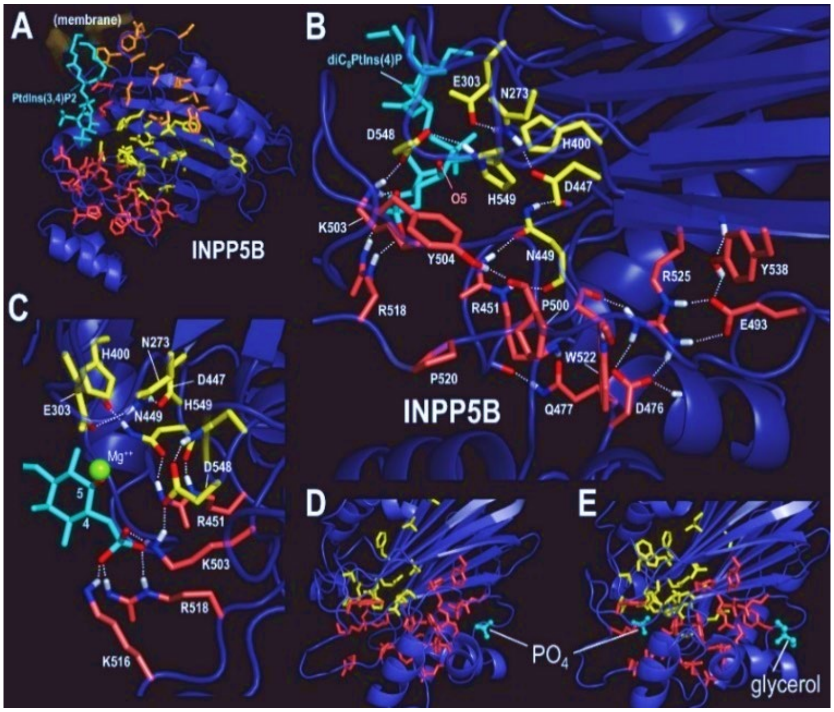
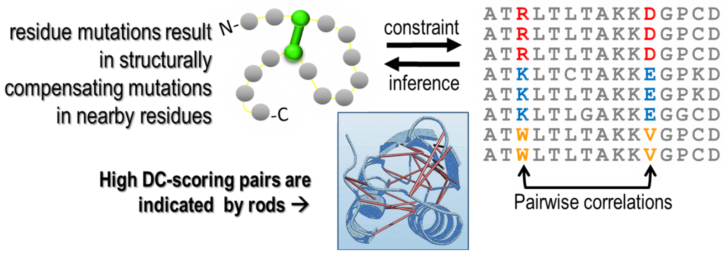


Figure 7.4. BPPS-SIPRIS analysis of the inositol polyphosphate 5-phosphatase INPP5B. Color scheme: EEP superfamily, INPP5 family and INPP5B subfamily residue sidechains: *yellow*, *red,* and *orange*, respectively; ligands, *cyan*; hydrogen bond atoms, *CPK coloring*. **A,B**. An INPP5 hydrogen bond network ($\boldsymbol{p = 1.1\times10^{-7}}$) 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 ($\boldsymbol{p = 3.2\times10^{-9}}$). **A**. INPP5B bound to the reaction product, phosphatidylinositol 3,4-bisphosphate (pdb_id: 4cml)[55]. SPIRIS clustering results: EEP, $\boldsymbol{p = 5.8\times10^{-13}}$; INPP5, $\boldsymbol{p = 3.9\times10^{-7}}$; INPP5B, $\boldsymbol{p = 0.0021}$. 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**. In 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]).

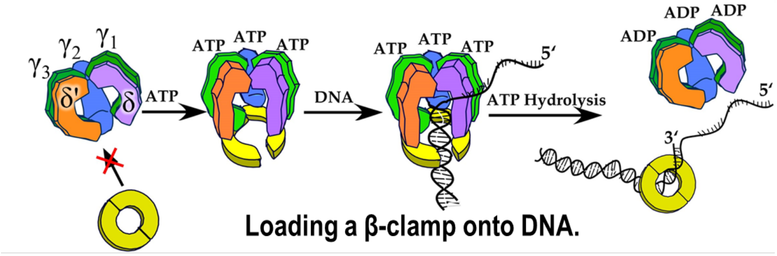
8. Correspondence between pairwise MSA correlations and structural interactions (DCA, STARC)



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 superfamily 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.



<u>The bacterial DNA clamp loader AAA+ complex</u> 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-γ_1-γ_2-γ_3-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].

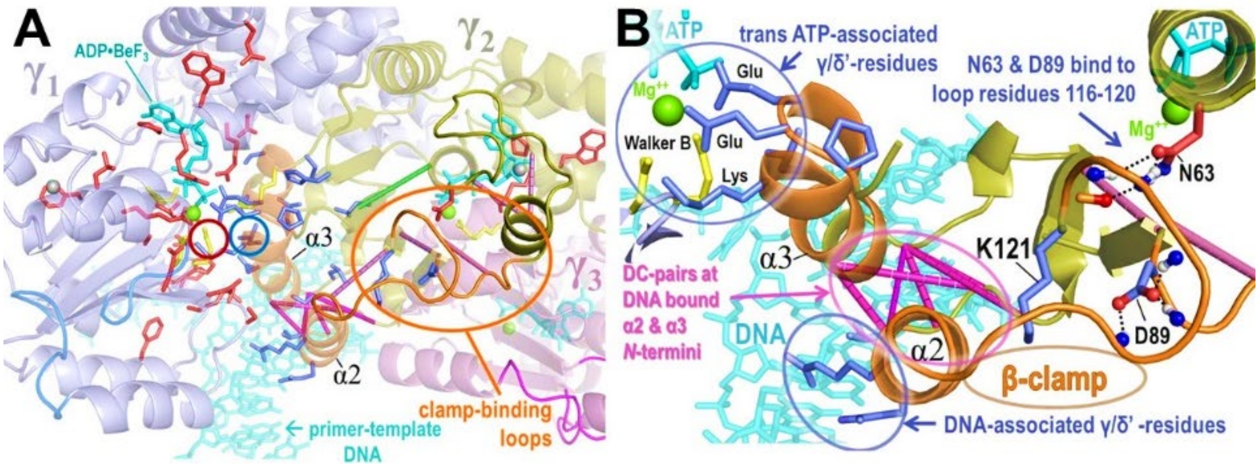

A
ADP•BeF3
γ1
γ2
α3
γ3
α2
primer-template DNA
clamp-binding loops
B
ATP
trans ATP-associated γ/δ’-residues
N63 & D89 bind to loop residues 116-120
Mg++
Glu
Walker B
Glu
Lys
ATP
Mg++
N63
K121
α3
DC-pairs at DNA bound α2 & α3 N-termini
DNA
α2
D89
β-clamp
DNA-associated γ/δ’-residues

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 γ_1/γ_2 interface and of the top γ/δ’-DC-scoring pairs linking together (within γ_2) 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; **$p=1.8\times10^{-9}$**). The γ/δ’-residues cluster around L140 (circled in blue; **$p=6.7\times10^{-10}$**). **B**. Close up of constrained residues linking the ATP, DNA, and clamp binding sites. <u>*Color scheme*</u>: γ/δ’-residues, *blue*. γ-residues, *red*. Backbones of γ_1, γ_2 and γ_3, *blue*, *dark yellow*, and *pink*, respectively. β-clamp binding loops in γ_1, γ_2, and γ_3, *marine blue*, *orange*, and *bright pink*, respectively. Helices α2, α3, and α4 of γ_2, *orange*. Walker B catalytic residues in γ_1, *yellow*. ATP analog ADP•BeF$_3$ 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 leveraging 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.

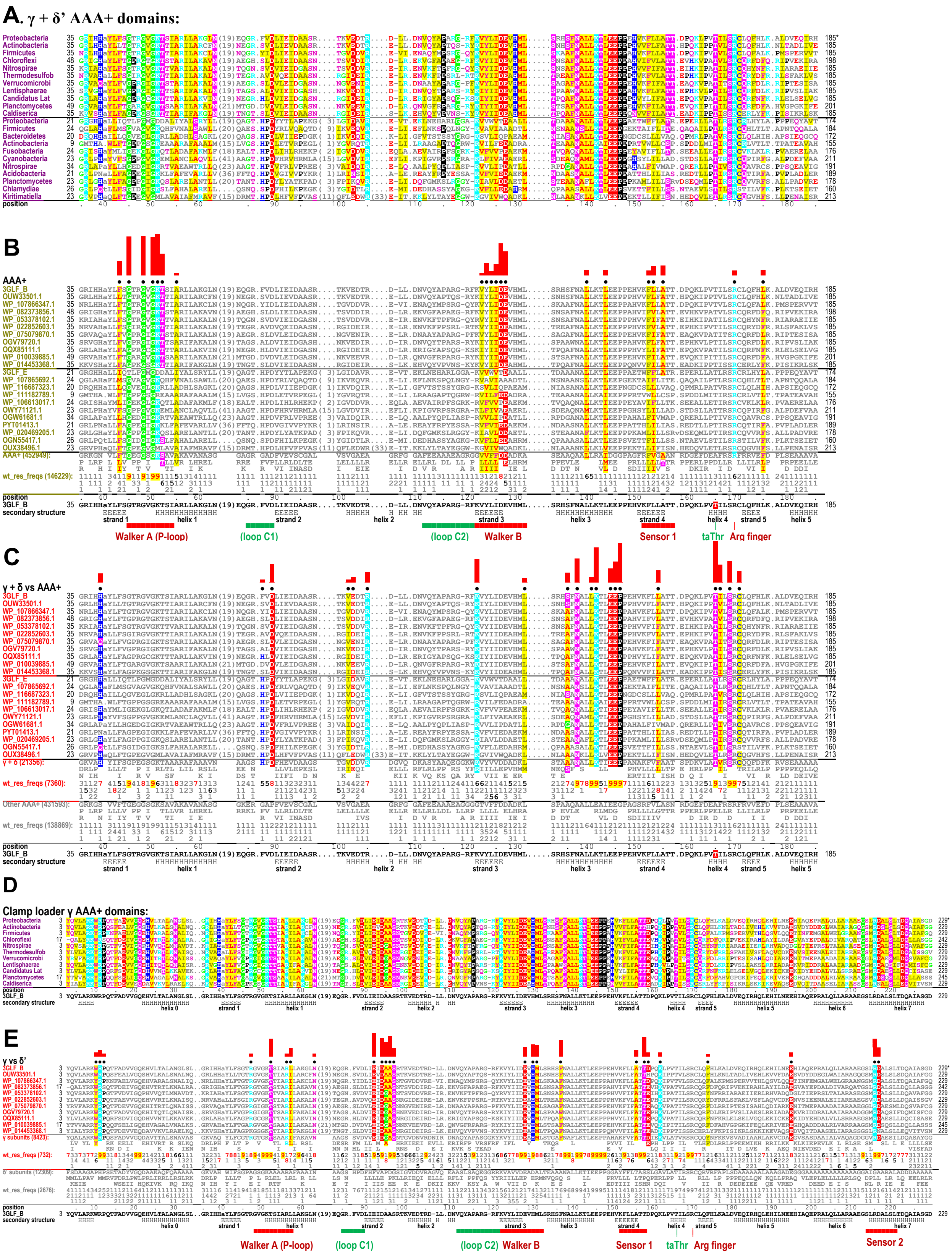


Figure 9.2. DARC-generated alignments highlighting all residues conserved in γ and δ' clamp loader proteins and residues distinctive of the AAA+ superfamily, of the γ + δ' subgroup, and of γ but not δ'. These are shown using five versions of the

same representative set of γ proteins (in panles 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 corrsponds 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:

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

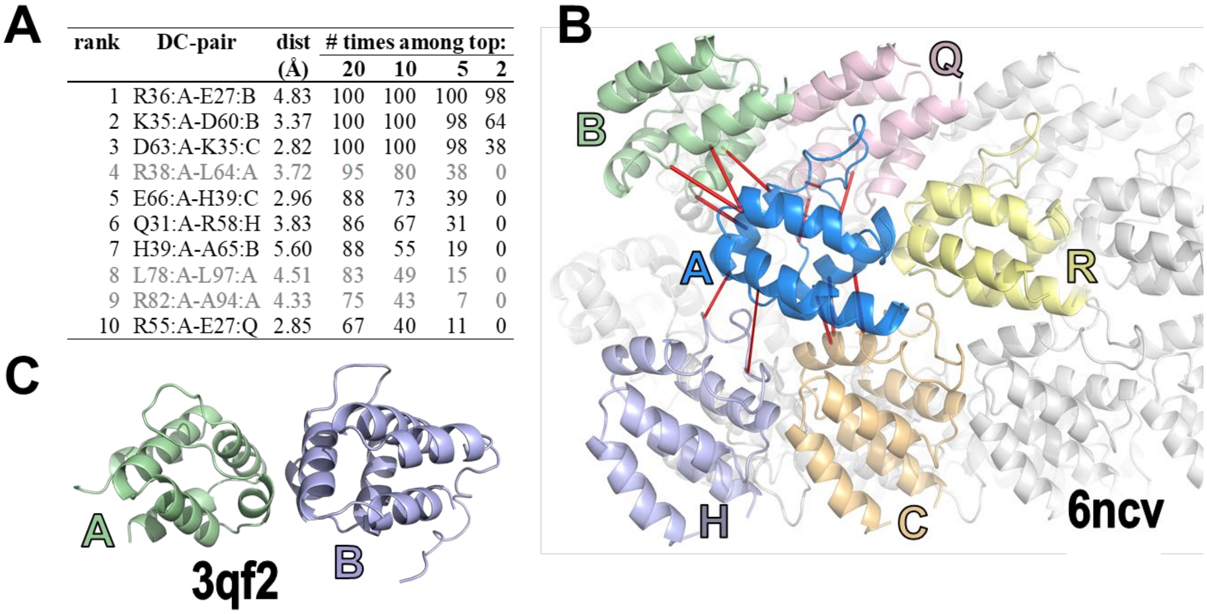
SPARC rank mode. In the rank mode, SPARC takes as input an MSA in cma format and the path to a directory containing coordinate files for aligned sequences of known structure; it makes associations based on sequence/structural identifiers. Specifically, structural file names must use the syntax <pdbid>_H.pdb 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).

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

Table 10.2. List of variables defined for STARC *S*-scores.

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



rank	DC-pair	dist (Å)	# times among top: 20	10	5	2
1	R36:A-E27:B	4.83	100	100	100	98
2	K35:A-D60:B	3.37	100	100	98	64
3	D63:A-K35:C	2.82	100	100	98	38
4	R38:A-L64:A	3.72	95	80	38	0
5	E66:A-H39:C	2.96	88	73	39	0
6	Q31:A-R58:H	3.83	86	67	31	0
7	H39:A-A65:B	5.60	88	55	19	0
8	L78:A-L97:A	4.51	83	49	15	0
9	R82:A-A94:A	4.33	75	43	7	0
10	R55:A-E27:Q	2.85	67	40	11	0

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 Δ*S* suggesting that the interaction lacks biological relevance and thus may be a crystallographic artifact.

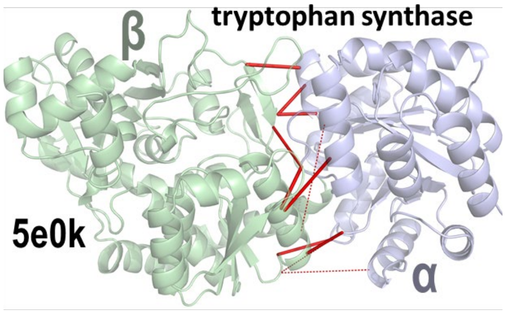


Fig. 10.2. The highest DC-scoring residue pairs across the α and β subunits of tryptophan synthase (pdb_id: 5e0k [1]) occur at the interface; $S = \log_{10}(p) = 21.1$. Solid and dashed red lines correspond to DC-pairs that are separated by ≤ 10 Å and > 10 Å, respectively.

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.

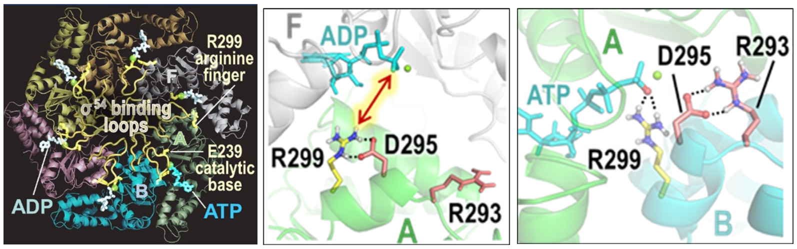


Fig. 10.3. (*left*) R299 and E239 within the bEBP hexamer bound to 1ATP + 5 ADP. (*middle*) The subunit A D295-R299 interaction may aid ADP release from subunit F. (*right*) The subunit B D295-R293 interaction frees the R-finger R299 to interact in trans with the γ-phosphate of ATP bound to subunit A.

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 bEBP-residue D295—thereby sequestering it away from ADP, which then may be more easily expelled (**Fig. 10.3, *middle***). In the same state, the bEBP-residue R293 in subunit B 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.

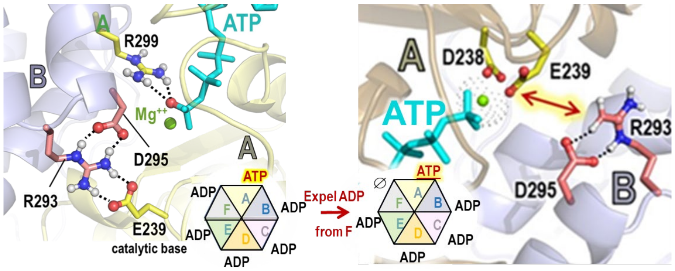


Fig. 10.4. MD conformations of R293 & E239 at the A:B interface in the 1ATP+5ADP (*left*) & 1ATP+4ADP+apo (*right*) states. The R293-E239 salt bridge may prevent ATP hydrolysis until ADP is expelled from the adjacent F subunit.

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.

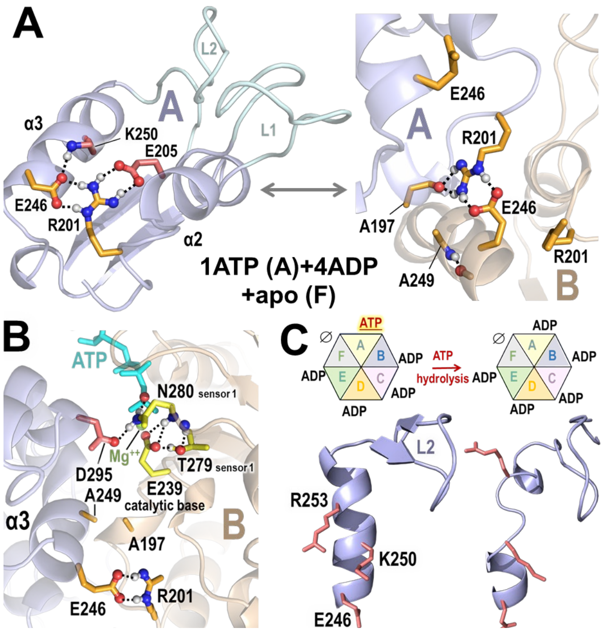


Fig. 10.5. Potential allosteric coupling of ATP hydrolysis to movement of the α2 and α3 helices, which are linked to the σ^{54}-binding L1 and L2 loops—as observed during a 1 µs MD simulation of the bEBP hexamer. **A**. A E246-R201 cis-to-trans switch associated with bEBP-residues, K250 and E205 (red), 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.

program	description
addphylum	Adds taxonomic information to a file of fasta sequences; requires NCBI nr and taxdump files.
bpps2vsi	Creates a vsi (visualize structural interactions) file from files generated by bpps (see vsi2pml below).
cdd2mgs	Converts CDD hiMSAs (available at ftp://ftp.ncbi.nih.gov/pub/mmdb/cdd/hiMSA) into MAPGAPS format.
convert_msa	Converts an MSA from fasta to cma format and vice versa; also converts an MSA from cma into rtf format.
edit_cma	Performs editing operations on a cma-formatted MSA file.
edit_hpt	Performs editing operations on a bpps hyperpartition (*.hpt) file.
get_pdb	Automatically retrieves and adds hydrogen atoms to pdb coordinate files corresponding to pdb_identifiers within a cma-formated MSA or a fasta sequence file.
getPDB	Retrieves pdb coordinate files given a list of pdb_identifiers.
matchcma	Reports matches to user specified pattern residues within a set of bpps subgroup MSAs
purgemsa	Reduces sequence redundancy given a cma-formatted MSA or concatenated MSAs.
tree2hpt	Converts a tree in Newick format into a hyperpartition that can serve as input for bpps in H mode.
tweakcma	Performs a wide variety of operations on a cma-formatted MSA.
tweakPDB	Performs a wide variety of operations on a pdb structural coordinate file.
vsi2pml	Creates PyMOL scripts (*.pml) or session (*.pse) files from a vsi file.

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.

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

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/

JSON library (https://github.com/json-c); this is required for compilation.
CUDA (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/
Download flex from https://github.com/westes/flex/releases

CMake (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:

```
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:

```
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.

```
cat nr | addphylum stdin > nrtx
cat pdbaa | addphylum stdin -pdb > pdbaatx
```

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 out 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
foreach file ($FASTADIR/nrtx.*)  mapgaps <infile> file
cat $FASTADIR/nrtx.*_map.seq > All
mapgaps <infile> All
mapgaps All_map                       // creates *mpa and other files
foreach file ($FASTADIR/nrtx.*)  mapgaps All_map file
mapgaps All_map $FASTADIR/pdbaatx
cat  $FASTADIR/pdbaatx_A.cma $FASTADIR/nrtx.*_A.cma > X.cma
tweakcma X -m                         // merges the concatenated cma files
tweakcma X.merged -mincol=0.80        // removes sequences fragments
tweakcma X.merged.match80 -U90        // reduces sequence redundancy to < 90% identity
\mv -f tweakcma X.merged.match80.purge90.cma Main.cma
twkcma Main -hsw                      // creates a Main.hsw file with sequence weights
```

The resultant Main.cma and Main.hsw files are used as input to the bpps 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 $\mathbf{v}=(v_1,...,v_n)^T$ and $\mathbf{w}=(w_1,...,w_n)^T$: $|\mathbf{v}|=|v_1|+...+|v_n|$, $\mathbf{v}+\mathbf{w}=(v_1+w_1,...,v_n+w_n)^T$, $\mathbf{v}/\mathbf{w}=(v_1/w_1,...,v_n/w_n)^T$, $\mathbf{v}^{\mathbf{w}}=v_1^{w_1}...v_n^{w_n}$, and $\Gamma(\mathbf{v})=\Gamma(v_1)...\Gamma(v_n)$. Given K proteins, their sequences are defined by $\mathbf{R}=(R_1^T,...,R_K^T)^T$ where each vector $R_k=(r_{k,1},...,r_{k,n_k})$ 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. $\mathbf{h}(\ )$ defines a counting function where, for example, $\mathbf{h}(R_k)$ 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 $A_j=\{a_{1,j},...,a_{K,j}\}$. We define $A_{j[-k]}\equiv A_j-\{a_{k,j}\}$ to denote the set A_j without $a_{k,j}$. An alignment is defined by the matrix $\mathbf{A}=(A_1,...,A_w)^T$ and $\{\mathbf{A}\}\equiv\{a_{k,j}:k=1,..,K,j=1,...,w\}$ denotes the set of residues indices for the alignment variable **A**. We represent the collection of residues indexed by elements in a set C as $\mathbf{R}_C$. For instance, $\mathbf{R}_{\{\mathbf{A}\}}=\{a_{k,j}:k=1,...,K;j=1,...w\}$ represents the set of residues in the alignment defined by **A**.

GISMO Statistical Model. The residue frequencies observed for column c are modeled as a multinomial distribution with parameters $\boldsymbol{\theta}_c=(\theta_{1,c},...,\theta_{20,c})^T$ where $\sum_{i=1}^{20}\theta_{i,c}=1$ and $\theta_{i,c}>0$ for all i. That is, the vector $\boldsymbol{\Theta}=(\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_w)$ defines a product multinomial model corresponding to the full alignment. The vector $\boldsymbol{\theta}_0$ corresponds to a background amino acid residue distribution. Hence, the complete-data likelihood function is given by

$$\pi(\mathbf{R}|\boldsymbol{\theta}_0,\boldsymbol{\Theta},\mathbf{A})\propto\boldsymbol{\theta}_0^{\mathbf{h}(\mathbf{R})}\prod_{j=1}^{w}\left(\frac{\boldsymbol{\theta}_j}{\boldsymbol{\theta}_0}\right)^{\mathbf{h}\left(\mathbf{R}_{\{A_j\}}\right)}$$

where it is assumed that $\Theta\sim D(\mathrm{B})$ and $\boldsymbol{\theta}_0\sim D(\boldsymbol{\alpha})$ (where D denotes the Dirichlet distribution), and where $\mathrm{B}=(\beta_1,...,\beta_w)$ specifies the Dirichlet distribution parameters (commonly interpreted as numbers of pseudocounts) at each column position j, and $\boldsymbol{\alpha}$ specifies the parameters for the background distribution. (Recall that the alignment is specified by the matrix $\mathbf{A}=(A_1,...,A_w)=(a_{k,j})_{K\times w}$ 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 **A** with the $\boldsymbol{\theta}$'s integrated out is

$$\pi(\mathbf{R}|\mathbf{A})\propto\Gamma\left(\mathbf{h}\left(\mathbf{R}_{\{\mathbf{A}\}^c}\right)+\boldsymbol{\alpha}\right)\cdot\prod_{j=1}^{w}\Gamma\left\{\mathbf{h}\left(\mathbf{R}_{\{\mathbf{A}_j\}}\right)+\boldsymbol{\beta}_j\right\}. \tag{1}$$

The conditional predictive probability distribution of this conserved region occurring at position i in sequence k is given by

$$\pi\left(a_k = i \middle| \mathbf{A}_{[-k]}, \mathbf{R}\right) \propto \prod_{j=1}^{w} \left(\frac{\hat{\boldsymbol{\theta}}_j}{\hat{\boldsymbol{\theta}}_0} \right)^{\mathbf{h}\left(r_{k,a_{k,j}}\right)}$$

where the $\hat{\boldsymbol{\theta}}$ are the posterior means of the $\boldsymbol{\theta}$, given the observed sequence data **R** and the current alignment $\mathbf{A}_{[-k]}$. 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:

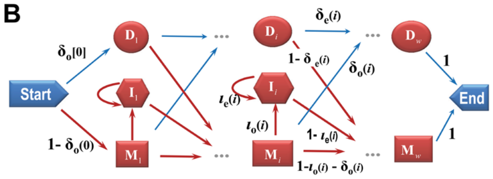
$$wt(k) = \sum_{j=1}^{w} \left(Nt_j \cdot Nr_{k,j} \right)^{-1}$$

where Nt_j is the number of residue types at each position *j* and where $Nr_{k,j} = \left| \left\{ r_{a_{x,j}} \middle| 1 \le x \le K \wedge r_{a_{x,j}} = r_{a_{k,j}} \right\} \right|$ 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:

$$Wt(k) = \left\lceil 100 \cdot Wt(k) \div wt_{\max} \right\rceil$$

where $wt_{\max}$ 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:



$$\begin{array}{cccc}
 & \mathrm{M}_{j+1} & \mathrm{I}_j & \mathrm{D}_{j+1} \\
\mathrm{M}_j & 1-\iota_o[j]-\delta_o[j] & \iota_o[j] & \delta_o[j] \\
\mathrm{I}_j & 1-\iota_e[j] & \iota_e[j] & 0 \\
\mathrm{D}_j & 1-\delta_e[j] & 0 & \delta_e[j]
\end{array}$$

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

$$\begin{array}{ccc}
 & \mathrm{M}_1 & \mathrm{D}_1 \\
\text{Start} & 1-\delta_o[0] & \delta_o[0]
\end{array}$$

Transitions into M and I states emit a residue as specified by the Θ 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 Θ. We denote the collection of these paths by Λ and the total number of HMM transitions of type M→M, M→I , … , D→D at position *j* by

$$N_{mm}[j],\ N_{mi}[j],\ N_{md}[j],\ N_{im}[j],\ N_{ii}[j],\ N_{dm}[j] \text{ and } N_{dd}[j].$$

Ignoring the indexing variable *j* for clarity, the likelihood of the transition probability parameters at each position is

$$h(\Lambda \mid \iota, \delta) = (1-\iota_o-\delta_o)^{N_{mm}}\, \iota_o^{N_{mi}}\, \delta_o^{N_{md}}\, (1-\iota_e)^{N_{im}}\, \iota_e^{N_{ii}}\, (1-\delta_e)^{N_{dm}}\, \delta_e^{N_{dd}}\,.$$

with independent prior distributions

$$(\iota_o, \delta_o, 1-\iota_o-\delta_o) \sim D(n_{mi}, n_{md}, n_{mm}),\ \ \iota_e \sim \mathrm{Beta}(n_{ii}, n_{im}),\ \text{and}\ \ \delta_e \sim \mathrm{Beta}(n_{dd}, n_{dm}),$$

where $n_{mi}, n_{md}, n_{mm}, n_{ii}, n_{im}, n_{dd}, n_{dm}$ 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

$$g(\mathbf{A}, \Lambda, \vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}}) \propto P(\mathbf{R} \mid \mathbf{A}, \Lambda) \times h(\Lambda \mid \vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}}) \times P(\vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}}),$$

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

$$P(\vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}}) = \left[D(n_{mi}, n_{md}, n_{mm}) \times \mathrm{Beta}(n_{ii}, n_{im}) \times \mathrm{Beta}(n_{dd}, n_{dm})\right]^{w}.$$

Given the alignment and thus the paths Λ, we have the conditional posterior distribution

$$p(\vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}} \mid \mathbf{A}, \Lambda) \propto \prod_{j=1}^{w}\Big[\iota_o[j]^{N_{mi}[j]+n_{mi}-1} \cdot \delta_o[j]^{N_{md}[j]+n_{md}-1} \cdot (1-\iota_o-\delta_o[j])^{N_{mm}[j]+n_{mm}-1} \times$$
$$\iota_e[j]^{N_{ii}[j]+n_{ii}-1}\, (1-\iota_e[j])^{N_{im}[j]+n_{im}-1} \cdot \delta_e[j]^{N_{dd}[j]+n_{dd}-1}\, (1-\delta_e[j])^{N_{dm}[j]+n_{dm}-1}\Big]$$

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

$$\delta_o[j] \sim \mathrm{Beta}(N_{md}[j]+n_{md}, N_{mm}[j]+N_{mi}[j]+n_{mm}+n_{mi}),$$

$$\delta_e[j] \sim \mathrm{Beta}(N_{dd}[j]+n_{dd}, N_{dm}[j]+n_{dm}),$$

$$\iota_o[j] = (1-\delta_o[j])\iota_o^{*}[j],\ \text{where}\ \iota_o^{*}[j] \sim \mathrm{Beta}(N_{mi}[j]+n_{mi}, N_{mm}[j]+n_{mm}),$$

$$\text{and}\ \ \iota_e[j] \sim \mathrm{Beta}(N_{ii}[j]+n_{ii}, N_{im}[j]+n_{im})\cdot$$

For computational efficiency, the ι and δ may be integrated out [10] to get

$$h(\Lambda) = \iint h(\Lambda \mid \vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}}) P(\vec{\boldsymbol{\iota}}, \vec{\boldsymbol{\delta}})\, d\vec{\boldsymbol{\iota}}\, d\vec{\boldsymbol{\delta}}$$
$$= \prod_{j=1}^{w}\left[\frac{\Gamma(N_{mi}[j]+n_{mi})\Gamma(N_{md}[j]+n_{md})\Gamma(N_{mm}[j]+n_{mm})\Gamma(n_{m\cdot})}{\Gamma(N_{m\cdot}[j]+n_{m\cdot})\Gamma(n_{mi})\Gamma(n_{md})\Gamma(n_{mm})}\right.$$
$$\times \frac{\Gamma(N_{ii}[j]+n_{ii})\Gamma(N_{im}[j]+n_{im})\Gamma(n_{i\cdot})}{\Gamma(N_{im}[j]+N_{ii}[j]+n_{i\cdot})\Gamma(n_{ii})\Gamma(n_{im})}$$
$$\left.\times \frac{\Gamma(N_{dd}[j]+n_{dd})\Gamma(N_{dm}[j]+n_{dm})\Gamma(n_{d\cdot})}{\Gamma(N_{dd}[j]+N_{dm}[j]+n_{d\cdot})\Gamma(n_{dd})\Gamma(n_{dm})}\right].$$

This gives rise to a new posterior distribution $g(\mathbf{A}, \Lambda) \propto P(\mathbf{R} \mid \mathbf{A}, \Lambda) \times h(\Lambda)$, 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 $p^{1/T}$. 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 $\mathbf{X}$, 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 $\{\text{V,I,L}\}$, $\{\text{D,E}\}$, and $\{\text{F,Y}\}$ 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 $\mathbf{v}=(v_1,...,v_\ell)^T$ and $\mathbf{w}=(w_1,...,w_\ell)^T$, $\mathbf{v}/\mathbf{w}=(v_1/w_1,...,v_\ell/w_\ell)^T$, $\mathbf{v}+\mathbf{w}=(v_1+w_1,...,v_\ell+w_\ell)^T$, $\log\mathbf{v}=(\log v_1,...,\log v_\ell)^T$, $|\mathbf{v}|$ is the sum over vector elements, and $\langle\mathbf{v},\mathbf{w}\rangle$ denotes the inner product of **v** and **w** and is equivalent, as applied here, to the dot product $\mathbf{v}\cdot\mathbf{w}=\sum_{i=1}^{\ell}v_i w_i$. Given an *N* node hierarchy $\mathbf{H}$, we define **S** as a vector of *N* disjoint sets, such that $\mathbf{S}_n$ contains the sequences assigned to node *n*, and $\mathbf{H}$ is defined as a vector of tri-partitions of node indices $1\le n\le N$, such that $\mathbf{H}_h\equiv\langle H_h^+,H_h^-,H_h^o\rangle$ 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., $H_1^+=\{n\,|\,1<n\le N\}$), for which the background set ($H_1^-=\{0\}$) consists of a single node (*n* = 0) for unrelated sequences (denoted as $\mathbf{S}_0$). The remaining $\mathbf{H}_h$ 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_h^o specifies nodes in neither H_h^+ nor H_h^-. To ensure that $\mathbf{H}$ 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^+ $\left(\text{i.e., } H_h^-=H_{h'}^+-H_h^+\right)$.

BPPS defines a prior on **H** that depends only on *N* and positive parameter *v* and that assumes a maximum number of nodes $N_{\max}$, so that

$$p(\mathbf{H}) = \frac{p(N)}{a_N}, \text{ where } p(N) = \begin{cases} \nu^{N-1} \cdot (1-\nu)/(1-\nu^{N_{\max}}) & \text{if} \quad \nu \neq 1 \\ 1/N_{\max} & \text{if} \quad \nu = 1 \end{cases},$$

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

$$a_N = \begin{cases} 1 & \text{if} \quad N = 1 \\ \displaystyle\sum_{\substack{j_1 + 2j_2 + \ldots + (N-1)j_{N-1} \\ = N-1}} \prod_{k=1}^{N-1} \binom{a_k + j_k - 1}{j_k} & \text{if} \quad N > 1 \end{cases}$$

with j_k being the number of subtrees with *k* nodes [107]. Computation suggests that the growth of a_N is $\mathrm{O}(2.96^N)$. By default, $N_{\max} = 500$ and $\nu = 1$ so that $p(\mathbf{H}) = (a_N N_{\max})^{-1}$, which corresponds to a uniform prior where every size tree (up to $N_{\max}$) 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 $p(\mathbf{H})$ due to the nature of our Bayesian formulation.

Let σ_n be the prior for a sequence assigned to node *n*. Given *N*, the prior for **S** is then given by $p(\mathbf{S}) = \prod_{n=0}^{N} \sigma_n^{|S_n|}$. 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 $\mathbf{A}$, where $A_{h,c} \neq \varnothing$ for each pattern column position *c* in subgroup *h*, and $A_{h,c} = \varnothing$ for non-pattern positions. The pattern residue sets $\mathbf{A}_h$ are constrained by a foreground consensus sequence for subgroup *h* , denoted as $\mathbf{y}_h$, 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 H_h^+, then $y_{h,c} = \mathrm{W}$ and

$$\mathbb{A}(\mathrm{W}) \equiv \{\{\mathrm{W}\}, \{\mathrm{W,F}\}, \{\mathrm{W,Y}\}, \{\mathrm{W,F,Y}\}\} \text{ and } A_{h,c} \in \mathbb{A}(\mathrm{W}) \cup \{\varnothing\},$$

where $\mathbb{A}(r)$ denotes the allowed pattern residue sets for consensus residue *r*. We define prior probabilities for $\mathbf{A}$ as $p(\mathbf{A}) = \prod_{h=1}^{N} \prod_{c=1}^{C} \rho_{A_{h,c}}$ (product categorical distributions), where, when $A_{h,c} = \varnothing$, $\rho_{A_{h,c}} = q_\varnothing$ (0.999 by default) and otherwise $\rho_{A_{h,c}} = \kappa \cdot \dfrac{q^{|A_{h,c}|}}{\|A_{h,c}\|}$. Here, κ is a constant chosen so that the priors for pattern positions sum to $1 - q_\varnothing$; $0 < q < 1$ (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 $y_{h,c} = \mathrm{W}$ and $A_{h,c} = \{\mathrm{W,F}\}$, 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 χ, where $\boldsymbol{\chi}_{h,s,c} = \left(1,0\right)^{\mathrm{T}}$ and $\boldsymbol{\chi}_{h,s,c} = \left(0,1\right)^{\mathrm{T}}$ imply that, for subtree *h*, $x_{s,c}$ corresponds to a 'functional' and a 'non-functional' pseudo-residue, respectively. Since $A_{h,c} = \varnothing$ at non-pattern columns, corresponding residues are all non-functional. Gaps are also treated as non-functional.

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

$$p\left(\alpha_h\right) = \frac{\Gamma\left(a_{h,0}+b_{h,0}\right)}{\Gamma\left(a_{h,0}\right)\Gamma\left(b_{h,0}\right)} \alpha_h^{\;a_{h,0}-1} \left(1-\alpha_h\right)^{b_{h,0}-1},$$

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 $\boldsymbol{\theta}_{h,c}$ is defined by a product Beta distribution:

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

where *C* is the number of columns in the MSA, and $a = b = 1$ by default.

Conditional on **H**, $\boldsymbol{\chi}$ 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 α_h and $\boldsymbol{\theta}_{h,c}$,

$$\xi_{h.c} \left| \alpha_h, \boldsymbol{\theta}_h \right. \sim \mathrm{Binom}\left(Nf_{h,c}, \frac{\alpha_h}{\alpha_h + \left(1-\alpha_h\right)\theta_{h,c,1}} \right),$$

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**, $\boldsymbol{\chi}$, **S**, $\boldsymbol{\xi}_h \equiv \left(\xi_{h,1},\ldots,\xi_{h,C}\right)^{\mathrm{T}}$, and $Nf_{h.} = \sum_c Nf_{hc}$, the posterior distribution of α_h is

$$\left[\alpha_h \left| \boldsymbol{\xi}_h \right.\right] \propto \alpha_h^{\left|\boldsymbol{\xi}_h\right| + a_{h,0}-1} \left(1-\alpha_h\right)^{Nf_{h.} - \left|\xi_h\right| + b_{h,0}-1} \sim \mathrm{Beta}\left(\left|\boldsymbol{\xi}_h\right| + a_{h,0}, Nf_h - \left|\boldsymbol{\xi}_h\right| + b_{h,0}\right).$$

The conditional distribution for $\boldsymbol{\theta}_{h,c}$ is:

$$\boldsymbol{\theta}_{h,c} \left| \mathbf{H}, \mathbf{S}, \boldsymbol{\xi}_h, \boldsymbol{\chi}_h \right. \sim \mathrm{Beta}\left(\sum_{n \in H_h^-} \sum_{s \in S_n} \chi_{h,s,c,1} + Nf_{h,c} - \xi_{h,c} + \psi_1, \sum_{n \in H_h^- \cup H_h^+} \sum_{s \in S_n} \chi_{h,s,c,2} + \psi_2 \right),$$

where $\boldsymbol{\psi} \equiv \left(a,b\right)^{\mathrm{T}}$ specifies pseudo-counts with $\boldsymbol{\psi} = \left(1,1\right)^{\mathrm{T}}$ by default.

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

$$\begin{aligned} \log P\left(\mathbf{X},\mathbf{H},\mathbf{S},\mathbf{A},\boldsymbol{\alpha},\boldsymbol{\Theta}\right) = {} & \log P\left(\mathbf{X}\left|\mathbf{H},\mathbf{S},\mathbf{A},\boldsymbol{\alpha},\boldsymbol{\Theta}\right.\right) + \log p\left(\mathbf{H}\right) + \log p\left(\mathbf{S}\right) \\ & + \log p\left(\mathbf{A}\right) + \log p\left(\boldsymbol{\alpha}\right) + \log p\left(\boldsymbol{\Theta}\right) \end{aligned} \quad (1)$$

where, assuming statistical independence among subtrees (but see below),

$$\log P\left(\mathbf{X}\middle|\mathbf{H},\mathbf{S},\mathbf{A},\boldsymbol{\alpha},\boldsymbol{\Theta}\right)=\sum_{h=1}^{N}\left(\sum_{n\in H_h^+\cup H_h^-}\sum_{s\in S_n}\sum_{c=1}^{C}\left\langle\log\boldsymbol{\theta}_{h,c},\boldsymbol{\chi}_{h,s,c}\right\rangle+\sum_{n\in H_h^+}\sum_{s\in S_n}\sum_{c=1}^{C}\boldsymbol{I}_{A_{h,j}}\left\langle\log\frac{\boldsymbol{\theta}_{h,c}^{(\alpha_h)}}{\boldsymbol{\theta}_{h,c}},\boldsymbol{\chi}_{h,s,c}\right\rangle\right) \qquad (2)$$

and where $\boldsymbol{I}_{A_{h,c}}\equiv\begin{cases}0, & \text{if} \quad A_{h,c}=\varnothing\\ 1, & \text{if} \quad A_{h,c}\neq\varnothing\end{cases}$. Note that for non-pattern positions $\boldsymbol{\theta}_{h,c}=(0,1)^{\mathrm{T}}$ and $\chi_{h,s,c}=(0,1)^{\mathrm{T}}$ so that $\left\langle\log\boldsymbol{\theta}_{h,c},x_{sc}\right\rangle=\log 0\cdot 0+\log 1\cdot 1=0$; 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 $P\left(\mathbf{X},\mathbf{H},\mathbf{S},\mathbf{A},\boldsymbol{\alpha},\boldsymbol{\Theta}\middle|s\in S_{n'}\right)$ after having updated $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$. Likewise, for each column position *c* in each subtree *h*, remove the pattern set $A_{h,c}$ and sample in a new pattern set $A'_{h,c}\in\mathbb{A}(\mathrm{y}_{h,c})\cup\{\varnothing\}$ with probability proportional to $P\left(\mathbf{X},\mathbf{H},\mathbf{S},\mathbf{A},\boldsymbol{\alpha},\boldsymbol{\Theta}\middle|A_{h,c}=A'_{h,c}\right)$ after having updated $\boldsymbol{\chi}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$. 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 $\mathbf{H}_1=\left\langle H_h^+=\{1\},H_h^-=\{0\},H_h^o=\varnothing\right\rangle$, 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 (the $\boldsymbol{\theta}_{0,c}$) 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* row × *C* 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 $\varepsilon_{c,d}(r_c, r_d)$ for residues r_c 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:

$$P(\boldsymbol{\varepsilon}|\mathbf{X}) = \frac{1}{Z}\prod_{s=1}^{R}\prod_{c=1}^{C}\left[\exp\left(\varepsilon_c(x_{s,c}) + \sum_{\substack{d=1\\d\neq c}}^{C}\varepsilon_{c,d}(x_{s,c}, x_{s,d})\right)\right]$$

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

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

where $Z_{s,c} = \sum_{r=1}^{20}\exp\left[\varepsilon_c(r) + \sum_{\substack{d=1\\d\neq c}}^{C}\varepsilon_{c,d}(r, x_{s,c})\right]$.

Because computation of the normalization constants $Z_{s,c}$ involve summing over only *C* terms, these are much faster to compute than *Z* for $P(\boldsymbol{\varepsilon}|\mathbf{X})$. The gradient (the vector of partial derivatives) of this pseudo-log-likelihood is given by:

$$\frac{\partial plL(\boldsymbol{\varepsilon}|\mathbf{X})}{\partial\varepsilon_{c,d}(r,r')} = \sum_{s=1}^{R}\Bigg\{\delta_{x_{s,d},r'}\Bigg[\delta_{x_{s,c},r} - \frac{1}{Z_{s,c}}\exp\Bigg(\varepsilon_c(r) + \sum_{\substack{i=1\\i\neq c}}^{C}\varepsilon_{i,c}(r, x_{s,i})\Bigg)\Bigg]\Bigg]\Bigg\}$$

$$= \sum_{s=1}^{R}\left[\delta_{x_{s,d},r'}\right]\left[\delta_{x_{s,c},r} - p\left(x_{s,c} = r\,\middle|\,(x_{s,1},\ldots,x_{s,c-1},x_{s,c+1},\ldots,x_{s,C},V,E)\right)\right]$$

where $\delta_{x,y}$ is the Kronecker delta function.

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

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

where the regularization coefficients are $\lambda_{\text{single}} = 1$; $\lambda_{\text{pair}} = 0.2 \times (L - 1)$ [108], and where $\left\| V_c \right\|_2^2$ and $\left\| E_{c,d} \right\|_2^2$ 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 $\varepsilon_{c,d}$:

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

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

$$\zeta_{c,d} = K_{c,d} - \frac{K_{c,\cdot} K_{\cdot,d}}{K_{\cdot,\cdot}}$$

where "." denotes averaging over the corresponding row or column and $K_{\cdot,\cdot}$ 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 $_{CL}S_P$, $_{3D}S_{DC}$, $_{3D}S_P$, or $_{DC}S_P$ (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 θ as a probability distribution P_θ over all possible sets of data and the description length of a data set *E* given a theory θ as $\mathrm{DL}(E|\theta) = -\log(P_\theta(E))$. A model $\mathcal{M}$ is a parameterized set of theories, and the description length of *E* given $\mathcal{M}$ is defined as $\mathrm{DL}(E|\mathcal{M}) = \min_{\theta\in\mathfrak{M}} \mathrm{DL}(E|\theta)$. The MDL principle asserts that among multiple models to explain *E*, one should prefer the model $\mathcal{M}$ that minimizes $\mathrm{DL}(E|\mathcal{M}) + \mathrm{COMP}(\mathcal{M})$, where the description length or complexity $\mathrm{COMP}(\mathcal{M})$ is the log of the number of effectively independent theories $\mathcal{M}$ contains. For ICA, the MDL principle determines whether the hypothesis $\mathcal{H}_1$ that the 1s cluster near the start of the sequence is better than the null hypothesis $\mathcal{H}_0$ that the 1s and 0s occur randomly.

ICA treats $\mathcal{H}_1$ 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 $P_x(E) = P_1^{D1}(1-P_1)^{x-D_1} P_2^{D_2}(1-P_2)^{y-D_2}/Z$, where *Z* is a normalization constant taken over all length *L* sequences having *D* 1s. Hence the description length of *E* under $\mathcal{H}_1$ is $\mathrm{DL}(S|\mathcal{H}_1) = -\log(\max_x P_x(E))$. ICA computes the complexity of $\mathcal{H}_1$, as

$$\mathrm{COMP}(\mathcal{H}_1) \approx \log\left(\sqrt{D/\pi}\,\frac{L-1}{2}\right).$$

ICA treats $\mathcal{H}_0$ 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, $\mathrm{DL}(E|\mathcal{H}_0) = -\log(P^D Q^{L-D})$, which is *L* times the entropy of the Bernoulli trial. Because $\mathcal{H}_0$ contains only one theory, its complexity is zero. The MDL principle says that we should prefer $\mathcal{H}_1$ to $\mathcal{H}_0$ when $\mathrm{DL}(E|\mathcal{H}_1)$ + $\mathrm{COMP}(\mathcal{H}_1)$ < $\mathrm{DL}(E|\mathcal{H}_0)$. Treating each hypothesis as equally likely *a priori*, we may view the difference Δ 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:

$$P(x,n,N_1,N_2)=\left[\sum_{i=\max(x,n-N_2)}^{\min(n,N_1)}\binom{N_1}{i}\binom{N_2}{n-i}\right]\div\binom{N_1+N_2}{n}.$$

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*) ≡ -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*) ≡ 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 $p = (m/x)^m ((x-m)/x)^{x-m} (n/y)^n ((y-n)/y)^{y-n}$ 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 ($_{3D}S_{DC}$). The $_{3D}S_{DC}$-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 ≤ *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* 1s in an initial cluster "left-distinguished pairs.") For *L* = 18 and *D* = 7, for example, one such array is "101101$\underline{1}$00000010001", with optimal cut point *X* = 7 (underlined), yielding *d* = 5. Since the pairs are ranked by pairwise distance, we might then represent our example array as "401603$\underline{2}$00000070005" 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 ≥ *R* of the left-distinguished pairs are red:

$$P_b = \left[\sum_{i=R}^{d}\binom{d}{i}\binom{D-d}{d-i}\right] \div \binom{D}{d}.$$

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

$$P_J = P_a P_b \left(1 - \ln P_a P_b\right).$$

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 $_{3D}S_{DC}$ upon inclusion of interface contacts is denoted as ΔS_{DC}. High positive values for ΔS_{DC} suggest that strong selective pressures are maintaining 3D contacts between adjacent subunits. In contrast, negative values for ΔS suggest that subunit interactions may be functionally insignificant and perhaps due to a crystallographic artifact.

BPPS and DCA or 3D contact concurrence scores (${}_{DC}S_P$, ${}_{3D}S_P$). 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 (${}_{CL}S_P$). 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 ${}_{CL}S_P$-score. The ${}_{CL}S_P$ 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.

MSA	PyMOL	
text	match	mismatch
dkyellow	yellow1	yellow0
red	red1	red0
orange	orange1	orange0
magenta	magenta1	magenta0
green	green1	green0
cyan	cyan1	cyan0
blue	blue1	blue0
purple	purple1	purple0
teal	teal1	teal0
brown	brown1	brown0
gray	gray1	gray0

19.Appendix 7: Contrast alignments for bacterial clamp loader subunits.

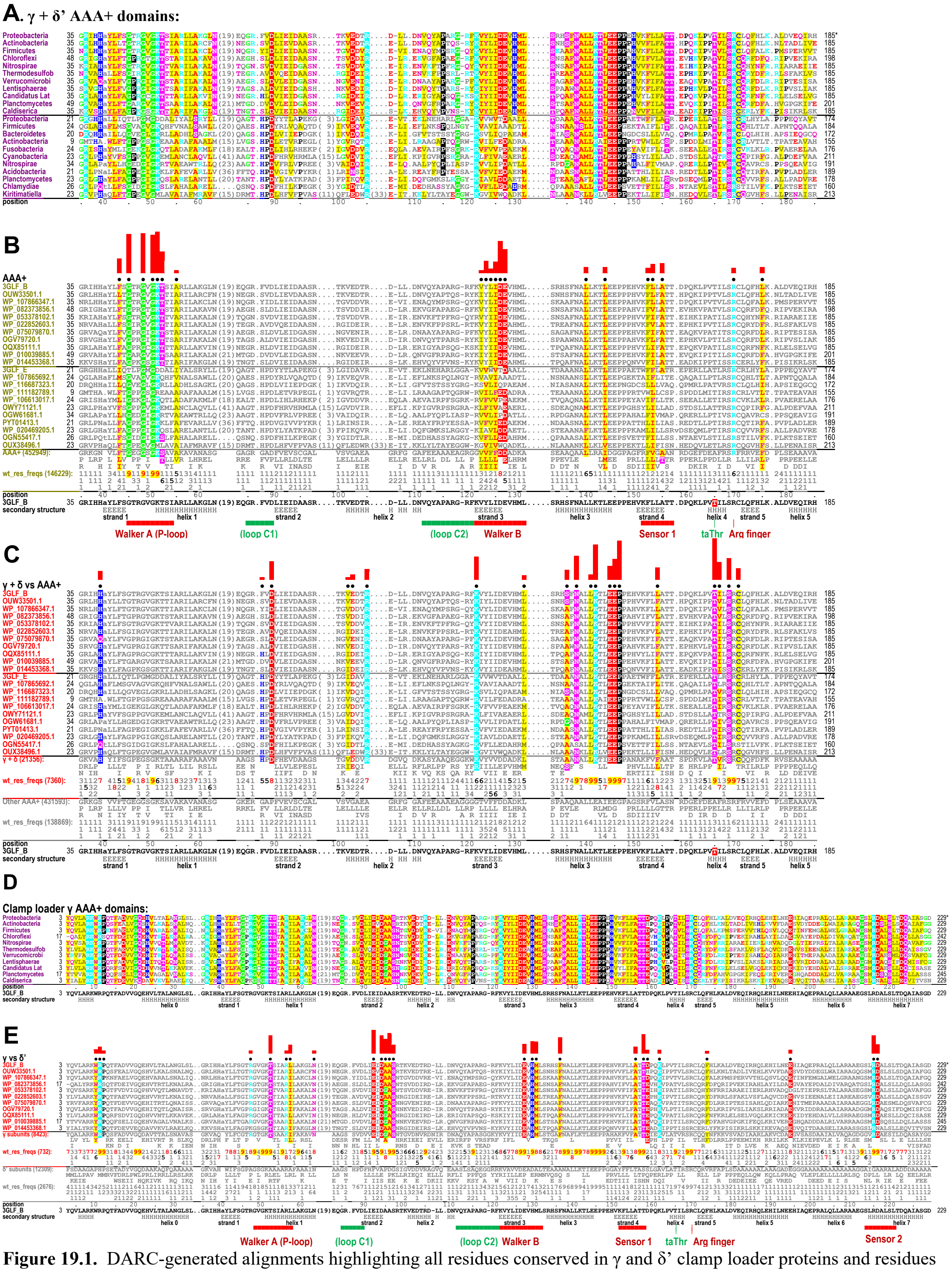


Figure 19.1. DARC-generated alignments highlighting all residues conserved in γ and δ' clamp loader proteins and residues distinctive of the AAA+ superfamily, of the γ + δ' subgroup, and of γ but not δ'. These are shown using five versions of the

same representative set of γ proteins (in panels A-E) and of δ' 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 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 corrsponds 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.

20. References:

1. Buller, A.R., Brinkmann-Chen, S., Romney, D.K., Herger, M., Murciano-Calles, J. and Arnold, F.H. (2015) Directed evolution of the tryptophan synthase beta-subunit for stand-alone function recapitulates allosteric activation. *Proc Natl Acad Sci U S A*, **112**, 14599-14604.
2. Bacon, F. and Fowler, T. (1878) *Novum organum*. Clarendon Press.
3. Neuwald, A.F. and Poleksic, A. (2000) PSI-BLAST searches using hidden markov models of structural repeats: prediction of an unusual sliding DNA clamp and of beta-propellers in UV-damaged DNA-binding protein. *Nucleic Acids Res*, **28**, 3570-3580.
4. Neuwald, A.F. and Hirano, T. (2000) HEAT repeats associated with condensins, cohesins, and other complexes involved in chromosome-related functions. *Genome Research*, **10**, 1445-1452.
5. Li, T., Chen, X., Garbutt, K.C., Zhou, P. and Zheng, N. (2006) Structure of DDB1 in complex with a paramyxovirus V protein: viral hijack of a propeller cluster in ubiquitin ligase. *Cell*, **124**, 105-117.
6. Ono, T., Losada, A., Hirano, M., Myers, M.P., Neuwald, A.F. and Hirano, T. (2003) Differential contributions of condensin I and condensin II to mitotic chromosome architecture in vertebrate cells. *Cell*, **115**, 109-121.
7. Neuwald, A.F. and Green, P. (1994) Detecting patterns in protein sequences. *J Mol Biol*, **239**, 698-712.
8. Li, W. and Godzik, A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658-1659.
9. Neuwald, A.F. and Altschul, S.F. (2016) Bayesian Top-Down Protein Sequence Alignment with Inferred Position-Specific Gap Penalties. *PLoS Comput Biol*, **12**, e1004936.
10. Neuwald, A.F. and Liu, J.S. (2004) Gapped alignment of protein sequence motifs through Monte Carlo optimization of a hidden Markov model. *BMC Bioinformatics*, **5**, 157.
11. Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. *J Mol Biol*, **147**, 195-197.
12. Neuwald, A.F., Kolaczkowski, B.D. and Altschul, S.F. (2021) eCOMPASS: evaluative comparison of multiple protein alignments by statistical score. *Bioinformatics*.
13. Neuwald, A.F. (2009) Rapid detection, classification and accurate alignment of up to a million or more related protein sequences *Bioinformatics*, **25**, 1869-1875.
14. Marchler-Bauer, A., Derbyshire, M.K., Gonzales, N.R., Lu, S., Chitsaz, F., Geer, L.Y., Geer, R.C., He, J., Gwadz, M., Hurwitz, D.I. *et al.* (2015) CDD: NCBI's conserved domain database. *Nucleic Acids Res*, **43**, D222-226.
15. Neuwald, A.F., Lanczycki, C.J., Hodges, T.K. and Marchler-Bauer, A. (2020) Obtaining extremely large and accurate protein multiple sequence alignments from curated hierarchical alignments. *Database (Oxford)*, **2020**.
16. Sayers, E.W., Barrett, T., Benson, D.A., Bolton, E., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., Dicuccio, M., Federhen, S. *et al.* (2012) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, **40**, D13-25.
17. Federhen, S. (2012) The NCBI Taxonomy database. *Nucleic Acids Res*, **40**, D136-143.
18. Neuwald, A.F. (2011) Surveying the manifold divergence of an entire protein class for statistical clues to underlying biochemical mechanisms. *Statistical applications in genetics and molecular biology*, **10**, 36.
19. Neuwald, A.F., Kannan, N., Poleksic, A., Hata, N. and Liu, J.S. (2003) Ran's C-terminal, basic patch and nucleotide exchange mechanisms in light of a canonical structure for Rab, Rho, Ras and Ran GTPases. *Genome Res*, **13**, 673-692.
20. Liu, J.S. (2008) *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York.
21. Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) Optimization by simulated annealing. *Science*, **220**, 671-680.

22. Neuwald, A.F. (2014) A Bayesian sampler for optimization of protein domain hierarchies. *Journal of Computational Biology*.
23. Neuwald, A.F. (2014) Protein domain hierarchy Gibbs sampling strategies. *Statistical applications in genetics and molecular biology*, **13**, 497-517.
24. Kannan, N., Haste, N., Taylor, S.S. and Neuwald, A.F. (2007) The hallmark of AGC kinase functional divergence is its C-terminal tail, a cis-acting regulatory module. *Proc Natl Acad Sci U S A*, **104**, 1272-1277.
25. Kannan, N. and Neuwald, A.F. (2004) Evolutionary constraints associated with functional specificity of the CMGC protein kinases MAPK, CDK, GSK, SRPK, DYRK, and CK2{alpha}. *Protein science : a publication of the Protein Society*, **13**, 2059-2077.
26. Kannan, N. and Neuwald, A.F. (2005) Did protein kinase regulatory mechanisms evolve through elaboration of a simple structural component? *J Mol Biol*, **351**, 956-972.
27. Kannan, N., Neuwald, A.F. and Taylor, S.S. (2008) Analogous regulatory sites within the alphaC-beta4 loop regions of ZAP-70 tyrosine kinase and AGC kinases. *Biochim Biophys Acta*, **1784**, 27-32.
28. Kannan, N., Wu, J., Anand, G.S., Yooseph, S., Neuwald, A.F., Venter, J.C. and Taylor, S.S. (2007) Evolution of allostery in the cyclic nucleotide binding module. *Genome Biol*, **8**, R264.
29. Neuwald, A.F. (2003) Evolutionary clues to DNA polymerase III beta clamp structural mechanisms. *Nucleic Acids Res*, **31**, 4503-4516.
30. Neuwald, A.F. (2005) Evolutionary clues to eukaryotic DNA clamp-loading mechanisms: analysis of the functional constraints imposed on replication factor C AAA+ ATPases. *Nucleic Acids Res*, **33**, 3614-3628.
31. Neuwald, A.F. (2006) Hypothesis: bacterial clamp loader ATPase activation through DNA-dependent repositioning of the catalytic base and of a trans-acting catalytic threonine. *Nucleic Acids Res*, **34**, 5280-5290.
32. Neuwald, A.F. (2007) Gα-Gβγ dissociation may be due to retraction of a buried lysine and disruption of an aromatic cluster by a GTP-sensing Arg-Trp pair. *Protein Science*, **16**, 2570-2577.
33. Neuwald, A.F. (2009) The glycine brace: a component of Rab, Rho, and Ran GTPases associated with hinge regions of guanine- and phosphate-binding loops. *BMC Struct Biol*, **9**, 11.
34. Neuwald, A.F. (2009) The charge-dipole pocket: a defining feature of signaling pathway GTPase on/off switches. *J Mol Biol*, **390**, 142-153.
35. Hall, A. (ed.) (2000) *GTPases*. Oxford University Press.
36. Li, G. and Zhang, X.C. (2004) GTP hydrolysis mechanism of Ras-like GTPases. *J Mol Biol*, **340**, 921-932.
37. Pasqualato, S., Senic-Matuglia, F., Renault, L., Goud, B., Salamero, J. and Cherfils, J. (2004) The structural GDP/GTP cycle of Rab11 reveals a novel interface involved in the dynamics of recycling endosomes. *J Biol Chem*, **279**, 11480-11488.
38. Zhu, G., Zhai, P., Liu, J., Terzyan, S., Li, G. and Zhang, X.C. (2004) Structural basis of Rab5-Rabaptin5 interaction in endocytosis. *Nat Struct Mol Biol*, **11**, 975-983.
39. Vetter, I.R. and Wittinghofer, A. (2001) The guanine nucleotide-binding switch in three dimensions. *Science*, **294**, 1299-1304.
40. Thomas, C., Fricke, I., Scrima, A., Berken, A. and Wittinghofer, A. (2007) Structural evidence for a common intermediate in small G protein-GEF reactions. *Mol Cell*, **25**, 141-149.
41. Gasper, R., Thomas, C., Ahmadian, M.R. and Wittinghofer, A. (2008) The role of the conserved switch II glutamate in guanine nucleotide exchange factor-mediated nucleotide exchange of GTP-binding proteins. *J Mol Biol*, **379**, 51-63.
42. Merkel, J.S. and Regan, L. (1998) Aromatic rescue of glycine in beta sheets. *Fold Des*, **3**, 449-455.
43. Renault, L., Kuhlmann, J., Henkel, A. and Wittinghofer, A. (2001) Structural basis for guanine nucleotide exchange on Ran by the regulator of chromosome condensation (RCC1). *Cell*, **105**, 245-255.

44. Neuwald, A.F. (2006) Bayesian shadows of molecular mechanisms cast in the light of evolution. *Trends Biochem Sciences*, **31**, 374-382.
45. Naiki, T., Kondo, T., Nakada, D., Matsumoto, K. and Sugimoto, K. (2001) Chl12 (Ctf18) forms a novel replication factor C-related complex and functions redundantly with Rad24 in the DNA replication checkpoint pathway. *Mol Cell Biol*, **21**, 5838-5845.
46. Neuwald, A.F. (2016) Gleaning structural and functional information from correlations in protein multiple sequence alignments. *Curr Opin Struct Biol*, **38**, 1-8.
47. Neuwald, A.F. and Altschul, S.F. (2016) Inference of Functionally-Relevant N-acetyltransferase Residues Based on Statistical Correlations. *PLoS Comput Biol*, **12**, e1005294.
48. Dorfmueller, H.C., Fang, W., Rao, F.V., Blair, D.E., Attrill, H. and van Aalten, D.M. (2012) Structural and biochemical characterization of a trapped coenzyme A adduct of Caenorhabditis elegans glucosamine-6-phosphate N-acetyltransferase 1. *Acta Crystallogr D Biol Crystallogr*, **68**, 1019-1029.
49. Bosch, D.E., Wittchen, E.S., Qiu, C., Burridge, K. and Siderovski, D.P. (2011) Unique structural and nucleotide exchange features of the Rho1 GTPase of Entamoeba histolytica. *J Biol Chem*, **286**, 39236-39246.
50. Eathiraj, S., Pan, X., Ritacco, C. and Lambright, D.G. (2005) Structural basis of family-wide Rab GTPase recognition by rabenosyn-5. *Nature*, **436**, 415-419.
51. Neuwald, A.F., Aravind, L. and Altschul, S.F. (2018) Inferring joint sequence-structural determinants of protein functional specificity. *Elife*, **7**.
52. Freudenthal, B.D., Beard, W.A., Cuneo, M.J., Dyrkheeva, N.S. and Wilson, S.H. (2015) Capturing snapshots of APE1 processing DNA damage. *Nat Struct Mol Biol*, **22**, 924-931.
53. Mol, C.D., Izumi, T., Mitra, S. and Tainer, J.A. (2000) DNA-bound structures and mutants reveal abasic DNA binding by APE1 and DNA repair coordination [corrected]. *Nature*, **403**, 451-456.
54. Qu, J., Liu, G.H., Huang, B. and Chen, C. (2007) Nitric oxide controls nuclear export of APE1/Ref-1 through S-nitrosation of cysteines 93 and 310. *Nucleic Acids Res*, **35**, 2522-2532.
55. Tresaugues, L., Silvander, C., Flodin, S., Welin, M., Nyman, T., Graslund, S., Hammarstrom, M., Berglund, H. and Nordlund, P. (2014) Structural basis for phosphoinositide substrate recognition, catalysis, and membrane interactions in human inositol polyphosphate 5-phosphatases. *Structure*, **22**, 744-755.
56. Speed, C.J., Matzaris, M., Bird, P.I. and Mitchell, C.A. (1995) Tissue distribution and intracellular localisation of the 75-kDa inositol polyphosphate 5-phosphatase. *Eur J Biochem*, **234**, 216-224.
57. Mills, S.J., Silvander, C., Cozier, G., Tresaugues, L., Nordlund, P. and Potter, B.V. (2016) Crystal Structures of Type-II Inositol Polyphosphate 5-Phosphatase INPP5B with Synthetic Inositol Polyphosphate Surrogates Reveal New Mechanistic Insights for the Inositol 5-Phosphatase Family. *Biochemistry*, **55**, 1384-1397.
58. de Juan, D., Pazos, F. and Valencia, A. (2013) Emerging methods in protein co-evolution. *Nat Rev Genet*, **14**, 249-261.
59. Cocco, S., Monasson, R. and Weigt, M. (2013) From principal component to direct coupling analysis of coevolution in proteins: low-eigenvalue modes are needed for structure prediction. *PLoS Comput Biol*, **9**, e1003176.
60. Hayat, S., Sander, C., Marks, D.S. and Elofsson, A. (2015) All-atom 3D structure prediction of transmembrane beta-barrel proteins from sequences. *Proc Natl Acad Sci U S A*, **112**, 5413-5418.
61. Hopf, T.A., Colwell, L.J., Sheridan, R., Rost, B., Sander, C. and Marks, D.S. (2012) Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, **149**, 1607-1621.
62. Morcos, F., Hwa, T., Onuchic, J.N. and Weigt, M. (2014) Direct coupling analysis for protein contact prediction. *Methods Mol Biol*, **1137**, 55-70.
63. Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D.S., Sander, C., Zecchina, R., Onuchic, J.N., Hwa, T. and Weigt, M. (2011) Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc Natl Acad Sci U S A*, **108**, E1293-1301.

64. Ekeberg, M., Lovkvist, C., Lan, Y., Weigt, M. and Aurell, E. (2013) Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. *Phys Rev E Stat Nonlin Soft Matter Phys*, **87**, 012707.

65. Marks, D.S., Hopf, T.A. and Sander, C. (2012) Protein structure prediction from sequence variation. *Nat Biotechnol*, **30**, 1072-1080.

66. Stein, R.R., Marks, D.S. and Sander, C. (2015) Inferring Pairwise Interactions from Biological Data Using Maximum-Entropy Probability Models. *PLoS Comput Biol*, **11**, e1004182.

67. Marks, D.S., Colwell, L.J., Sheridan, R., Hopf, T.A., Pagnani, A., Zecchina, R. and Sander, C. (2011) Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, **6**, e28766.

68. Hopf, T.A., Morinaga, S., Ihara, S., Touhara, K., Marks, D.S. and Benton, R. (2015) Amino acid coevolution reveals three-dimensional structure and functional domains of insect odorant receptors. *Nat Commun*, **6**, 6077.

69. Dwyer, R.S., Ricci, D.P., Colwell, L.J., Silhavy, T.J. and Wingreen, N.S. (2013) Predicting functionally informative mutations in Escherichia coli BamA using evolutionary covariance analysis. *Genetics*, **195**, 443-455.

70. Espada, R., Parra, R.G., Mora, T., Walczak, A.M. and Ferreiro, D.U. (2015) Capturing coevolutionary signals inrepeat proteins. *BMC Bioinformatics*, **16**, 207.

71. Hopf, T.A., Scharfe, C.P., Rodrigues, J.P., Green, A.G., Kohlbacher, O., Sander, C., Bonvin, A.M. and Marks, D.S. (2014) Sequence co-evolution gives 3D contacts and structures of protein complexes. *Elife*, **3**.

72. dos Santos, R.N., Morcos, F., Jana, B., Andricopulo, A.D. and Onuchic, J.N. (2015) Dimeric interactions and complex formation using direct coevolutionary couplings. *Scientific reports*, **5**, 13652.

73. Cheng, R.R., Morcos, F., Levine, H. and Onuchic, J.N. (2014) Toward rationally redesigning bacterial two-component signaling systems using coevolutionary information. *Proc Natl Acad Sci U S A*, **111**, E563-571.

74. Neuwald, A.F. and Altschul, S.F. (2018) Statistical investigations of protein residue direct couplings. *PLoS Comput Biol*, **14**, e1006237.

75. Seemayer, S., Gruber, M. and Soding, J. (2014) CCMpred--fast and precise prediction of protein residue-residue contacts from correlated mutations. *Bioinformatics*, **30**, 3128-3130.

76. Jones, D.T., Buchan, D.W., Cozzetto, D. and Pontil, M. (2012) PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, **28**, 184-190.

77. Baldassi, C., Zamparo, M., Feinauer, C., Procaccini, A., Zecchina, R., Weigt, M. and Pagnani, A. (2014) Fast and accurate multivariate Gaussian modeling of protein families: predicting residue contacts and protein-interaction partners. *PLoS One*, **9**, e92721.

78. Neuwald, A.F. (2014) Evaluating, comparing and interpreting protein domain hierarchies *Journal of Computational Biology*.

79. Rose, P.W., Prlic, A., Bi, C., Bluhm, W.F., Christie, C.H., Dutta, S., Green, R.K., Goodsell, D.S., Westbrook, J.D., Woo, J. *et al.* (2015) The RCSB Protein Data Bank: views of structural biology for basic and applied research and education. *Nucleic Acids Res*, **43**, D345-356.

80. Word, J.M., Lovell, S.C., Richardson, J.S. and Richardson, D.C. (1999) Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J Mol Biol*, **285**, 1735-1747.

81. Tondnevis, F., Dudenhausen, E.E., Miller, A.M., McKenna, R., Altschul, S.F., Bloom, L.B. and Neuwald, A.F. (2020) Deep Analysis of Residue Constraints (DARC): identifying determinants of protein functional specificity. *Scientific reports*, **10**, 1691.

82. DeLano, W.L. (2002). DeLano Scientific, Palo Alto, CA, USA.

83. Schrodinger, LLC. (2010).

84. Hedglin, M., Kumar, R. and Benkovic, S.J. (2013) Replication clamps and clamp loaders. *Cold Spring Harb Perspect Biol*, **5**, a010165.

85. Kelch, B.A., Makino, D.L., O'Donnell, M. and Kuriyan, J. (2012) Clamp loader ATPases and the evolution of DNA replication machinery. *BMC Biol*, **10**, 34.

86. Indiani, C. and O'Donnell, M. (2006) The replication clamp-loading machine at work in the three domains of life. *Nat Rev Mol Cell Biol*, **7**, 751-761.

87. Simonetta, K.R., Kazmirski, S.L., Goedken, E.R., Cantor, A.J., Kelch, B.A., McNally, R., Seyedin, S.N., Makino, D.L., O'Donnell, M. and Kuriyan, J. (2009) The mechanism of ATP-dependent primer-template recognition by a clamp loader complex. *Cell*, **137**, 659-671.

88. Neuwald, A.F., Yang, H. and Tracy Nixon, B. (2022) SPARC: Structural properties associated with residue constraints. *Comput Struct Biotechnol J*, **20**, 1702-1715.

89. Shen, C., Lu, A., Xie, W.J., Ruan, J., Negro, R., Egelman, E.H., Fu, T.M. and Wu, H. (2019) Molecular mechanism for NLRP6 inflammasome assembly and activation. *Proc Natl Acad Sci U S A*, **116**, 2052-2057.

90. Bae, J.Y. and Park, H.H. (2011) Crystal structure of NALP3 protein pyrin domain (PYD) and its implications in inflammasome assembly. *J Biol Chem*, **286**, 39528-39536.

91. Wang, Y., Lamim Ribeiro, J.M. and Tiwary, P. (2020) Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Current Opinion in Structural Biology*, **61**, 139-145.

92. Hildebrand, P.W., Rose, A.S. and Tiemann, J.K.S. (2019) Bringing Molecular Dynamics Simulation Data into View. *Trends in Biochemical Sciences*, **44**, 902-913.

93. Bowerman, S. and Wereszczynski, J. (2016) In Voth, G. A. (ed.), *Methods in Enzymology*. Academic Press, Vol. 578, pp. 429-447.

94. Noé, F. and Clementi, C. (2017) Collective variables for the study of long-time kinetics from molecular trajectories: theory and methods. *Current Opinion in Structural Biology*, **43**, 141-147.

95. Jin, Y., Johannissen, L.O. and Hay, S. (2021) Predicting new protein conformations from molecular dynamics simulation conformational landscapes and machine learning. *Proteins: Structure, Function, and Bioinformatics*, **89**, 915-921.

96. Michaud-Agrawal, N., Denning, E.J., Woolf, T.B. and Beckstein, O. (2011) MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. *J Comput Chem*, **32**, 2319-2327.

97. Carrillo-Tripp, M., Alvarez-Rivera, L., Lara-Ramírez, O.I., Becerra-Toledo, F.J., Vega-Ramírez, A., Quijas-Valades, E., González-Zavala, E., González-Vázquez, J.C., García-Vieyra, J., Santoyo-Rivera, N.B. *et al.* (2018) HTMoL: full-stack solution for remote access, visualization, and analysis of molecular dynamics trajectory data. *J Comput Aided Mol Des*, **32**, 869-876.

98. Bush, M. and Dixon, R. (2012) The role of bacterial enhancer binding proteins as specialized activators of sigma54-dependent transcription. *Microbiol Mol Biol Rev*, **76**, 497-529.

99. Gao, F., Danson, A.E., Ye, F., Jovanovic, M., Buck, M. and Zhang, X. (2020) Bacterial Enhancer Binding Proteins-AAA(+) Proteins in Transcription Activation. *Biomolecules*, **10**.

100. Sysoeva, T.A., Chowdhury, S., Guo, L. and Nixon, B.T. (2013) Nucleotide-induced asymmetry within ATPase activator ring drives σ54-RNAP interaction and ATP hydrolysis. *Genes Dev*, **27**, 2500-2511.

101. Chen, B., Sysoeva, T.A., Chowdhury, S., Guo, L., De Carlo, S., Hanson, J.A., Yang, H. and Nixon, B.T. (2010) Engagement of arginine finger to ATP triggers large conformational changes in NtrC1 AAA+ ATPase for remodeling bacterial RNA polymerase. *Structure*, **18**, 1420-1430.

102. Brown, M., Hughey, R., Krogh, A., Mian, I.S., Sjolander, K. and Haussler, D. (1993) Using Dirichlet mixture priors to derive hidden Markov models for protein families. *Ismb*, **1**, 47-55.

103. Sjolander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I.S. and Haussler, D. (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput Appl Biosci*, **12**, 327-345.

104. Nguyen, V.A., Boyd-Graber, J. and Altschul, S.F. (2013) Dirichlet mixtures, the Dirichlet process, and the structure of protein space. *Journal of computational biology : a journal of computational molecular cell biology*, **20**, 1-18.

105. Henikoff, S. and Henikoff, J.G. (1994) Position-based sequence weights. *J Mol Biol*, **243**, 574-578.

106. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, **25**, 3389-3402.

107. Knuth, D.E. (1997) *The Art of Computer Programming*. 3rd ed.

108. Kamisetty, H., Ovchinnikov, S. and Baker, D. (2013) Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proc Natl Acad Sci U S A*, **110**, 15674-15679.

109. Dunn, C.W., Hejnol, A., Matus, D.Q., Pang, K., Browne, W.E., Smith, S.A., Seaver, E., Rouse, G.W., Obst, M., Edgecombe, G.D. *et al.* (2008) Broad phylogenomic sampling improves resolution of the animal tree of life. *Nature*, **452**, 745-749.

110. Altschul, S.F. and Neuwald, A.F. (2018) Initial Cluster Analysis. *Journal of computational biology : a journal of computational molecular cell biology*, **25**, 121-129.

111. Grunwald, P.D. (2007) *The minimum description length principle*. MIT Press, Boston.

112. Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998) *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK.

113. Karlin, S. and Zhu, Z.Y. (1996) Characterizations of diverse residue clusters in protein three-dimensional structures. *Proc Natl Acad Sci U S A*, **93**, 8344-8349.

114. Bonferroni, C.E. (1936) Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, **8**, 3-62.

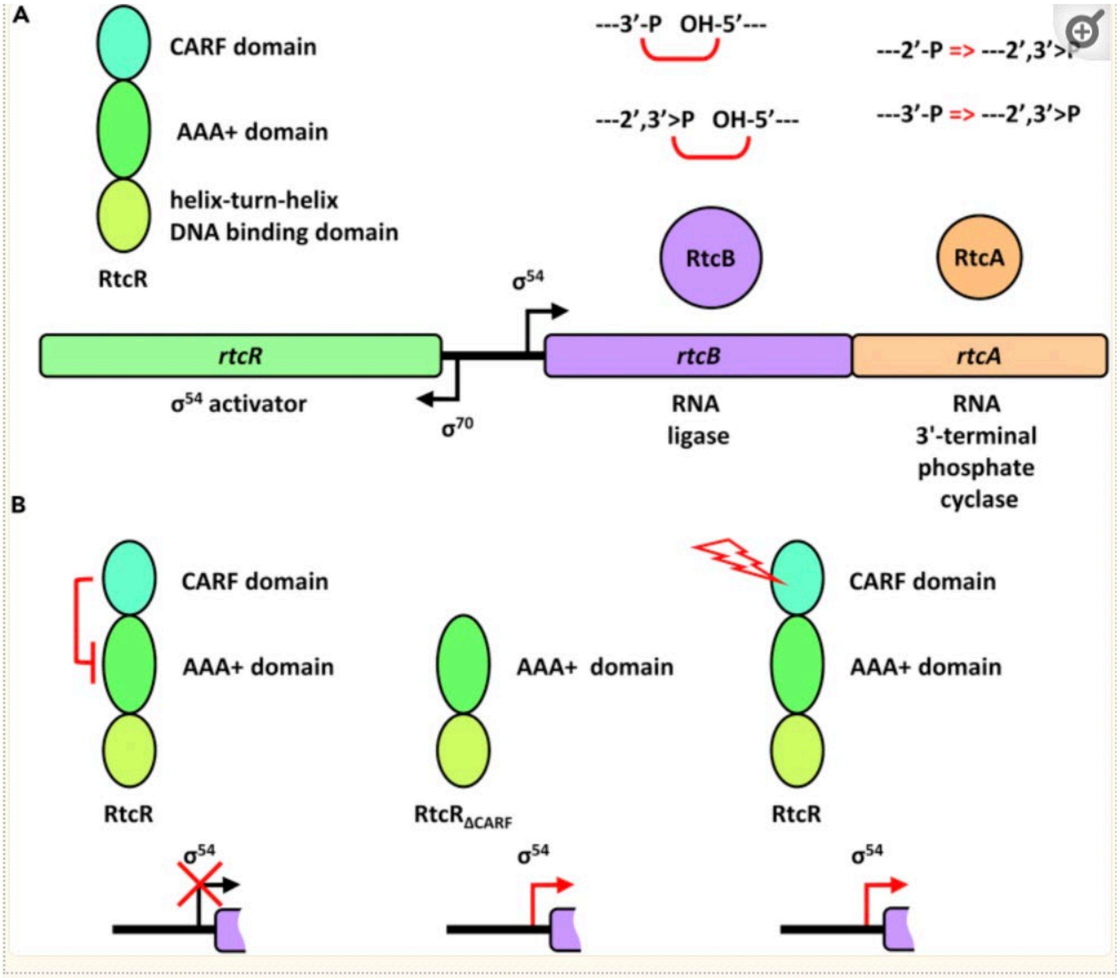
115. Bailey, T.L. and Gribskov, M. (1998) Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, **14**, 48-54.

116. Fisher, R.A. (1954) *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, Scotland.

117. Yu, Y.K., Gertz, E.M., Agarwala, R., Schaffer, A.A. and Altschul, S.F. (2006) Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. *Nucleic Acids Res*, **34**, 5966-5973.

NOTES:

RtcR bEBP: CARF relaces the receiver domain.



ure 1

ıematic representation of the Rtc system