U-Hack Med 2019: Assessing and Refining Co-evolutionary Strategies for Protein Interaction Prediction

Kimberly A. Reynolds [a]
[a] *Green Center for Systems Biology, Department of Biophysics, UT Southwestern Medical Center*
kimberly.reynolds@utsouthwestern.edu

INTRODUCTION

Proteins are linear polymers assembled from 20 amino acids, and in solution, fold into three dimensional structures than can catalyze enzymatic reactions, do mechanical work, and self assemble into larger structures. Interactions between proteins provide the foundation for complex cellular functions like metabolism, motility, and communication. Proteins can interact *physically*: by forming stable complexes, or transiently binding and sometimes modifying each other. They can also interact *functionally*: by sharing chemical substrates or regulatory mechanisms. An ability to map both physical and functional interactions between proteins, and determine which amino acids within the protein are responsible for mediating and specifying the interaction, is a critical first step to understanding how cells work and evolve. Understanding protein interactions at residue-level resolution is necessary to design precise mutagenesis experiments, engineer protein complex specificity, create new biosynthetic pathways, interpret disease causing mutations, and introduce new protein regulation.

One powerful approach for predicting protein interactions is the analysis of protein sequence evolution. Here, the basic premise is that interactions between amino acids will lead to their correlated evolution over time - changes in amino acid identity at one protein position will be compensated by changes at another. Thus, by constructing large multiple sequence alignments comprised of protein amino acid sequences from diverse extant species, and examining correlations in amino acid substitutions, we can infer interacting sites. In this hackathon, we will test and refine two approaches for protein interaction prediction by coevolution: 1) mirror tree (MT) and 2) the statistical coupling analysis (SCA). SCA aims to detect correlations in the identity of amino acid substitutions, while MT aims to detect correlations in the rate of substitutions. MT also differs from SCA in that it incorporates an explicit correction for phylogenetic noise; the goal is to remove correlations that can be explained strictly due to the tree-like structure of evolution. Our goals are to assess the performance of these methods on a gold standard test set of protein interactions, examine how sensitive the two methods are to alignment content and depth, and explore strategies for improving interaction prediction with each approach.

TEST SET DATA

We are beginning with a focused test set of 19 proteins, sampling four physical complexes, and four protein pairs that work together in metabolism (Fig. 1). We selected these proteins from a previous analysis of gene synteny, in which we used analysis of conserved chromosomal proximity to identify proteins that interact (Fig. 2). Importantly, because these proteins are proximal on the chromosome - and in many cases are in the same operon and/or share a promoter - we expect that changes in the activity of one protein will necessarily be compensated by coding sequence changes in the other, rather than by changes in the non-coding (regulatory) region upstream of the protein. Thus, these pairs represent instances where we have the strongest expectation of amino-acid sequence level co-evolution.

For each protein in our test set, we have downloaded pre-trimmed EggNOG alignments filtered to contain predicted orthologs across all Eubacteria (see also: `~/TestSet/eggNOG_aligns/readme.txt`). In these fasta-format alignments, the sequence headers have the format `>xxxxxx.yyyyy`, where `xxxxx` corresponds to the NCBI taxid (which specifies organismal identity), and `yyyy` is an organism-specific locus identifier. For example, the sequence for the *E. coli* K-12 MG1666 enzyme DHFR is designated `511145.b0048`. We use the taxids to match sequences across species (e.g. to select pairs of proteins from the same organism for analysis). The locus ids provide some information about chromo-

					EggNOG alignment				
gene	name	function	PDBID	length	# seqs	# pos	frac pos	COG	uniprot ID
cydA	cytochrome d terminal oxidase, subunit I	aerobic respiratory chain	N/A	522	1321	394	0.75	COG1271C	P0ABJ9
cydB	cytochrome d terminal oxidase, subunit II	aerobic respiratory chain	N/A	379	1138	301	0.79	COG1294C	P0ABK2
nrdA	ribonucleoside diphosphate reductase I, alpha subunit	biosynthesis of deoxyribonucleotides	2X0X	761	1697	504	0.66	COG0209F	P00452
nrdB	ribonucleoside diphosphate reductase I, beta subunit	biosynthesis of deoxyribonucleotides	1BIQ	376	1003	290	0.77	COG0208F	P69924
fliG	flagellar motor switching and energizing component	flagellar motility	3HJL	331	793	315	0.95	COG1536N	P0ABZ1
fliM	flagellar motor switching and energizing component	flagellar motility	4FHR (partial)	334	681	307	0.92	COG1868N	P06974
fliN	flagellar motor switching and energizing component	flagellar motility	1YAB	137	769	87	0.64	COG1886NU	P15070
trpA	tryptophan synthase alpha subunit	tryptophan synthesis from chorismate	1V7Y	268	1347	246	0.92	COG0159E	P0A877
trpB	tryptophan synthase beta subunit	tryptophan synthesis from chorismate	2DH5	397	1509	382	0.96	COG0133E	P0A879
folA	dihydrofolate reductase	folate metabolism	1RX2	159	1183	142	0.89	COG0262H	P0ABQ4
thyA	thymidylate synthase	folate metabolism	1BID	264	1113	253	0.96	COG0207F	P0A884
purE	N5-carboxyaminoimidazole ribonucleotide mutase	purine metabolism	1QCZ	169	1556	157	0.93	COG0041F	P0AG18
purK	N5-carboxyaminoimidazole ribonucleotide synthetase	purine metabolism	1B6R	355	1120	322	0.91	COG0026F	P09029
purC	phosphoribosylaminoimidazole-succinocarboxamide synthetase	purine metabolism	2GQR	237	1163	180	0.76	COG0152F	P0A7D7
ilvC	ketol-acid reductoisomerase NAD(P) binding	branched chain amino acid biosynthesis	1YRL	491	1391	327	0.67	COG0059EH	P05793
ilvN	acetolactate synthase I, small subunit	branched chain amino acid biosynthesis	2LVW	96	67	95	0.99	COG0440E	P0ADF8
ilvB	acetolactate synthase I, large subunit	branched chain amino acid biosynthesis	1OZF	562	3055	508	0.90	COG0028EH	P08142
glgA	glycogen synthase	glycogen biosynthesis	2QZS	477	985	432	0.91	COG0297G	P0A6U8
glgC	glucose I phophate adenyltransferase	glycogen biosynthesis	1M7X	728	598	386	0.53	COG0448G	P07762

Figure 1. Test set proteins. ilvN, shown in grey, interacts with ilvC and ilvB but has too few sequences in the EggNOG alignment for reliable analysis. For a larger, editable version of this table, see `TestSet/TestSetSummary.xlsx`

somal proximity: they provide the order in which genes appear on the chromosome (b0048 is upstream of b0049). Beneath each the header, the corresponding amino acid sequence is specified as a string of 20 letters (`ACDEFGHIKLMNPQRSTVWY`), and gaps (segments of the sequence where other species have an insertion) are indicated as '`-`'. For practical purposes, in our python code we consider the gaps as a 21st amino acid.

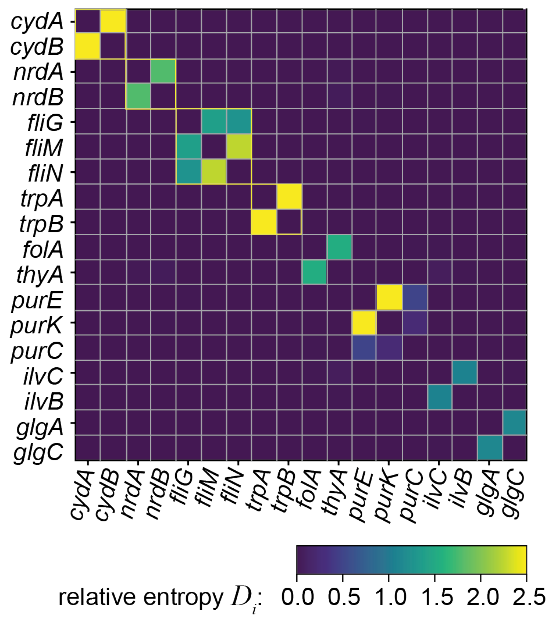


Figure 2. Synteny relationships for the test set proteins. Gene names are given along each axis. Color coding indicates the strength of the syntenic association, with higher relative entropy indicating a more significant association. Gene groups encoding physical complexes are outlined with a thin yellow box. This analysis was performed over a collection of ~ 1400 bacterial genomes.

COMPUTING ENVIRONMENT

The core SCA and mirrortree calculations are implemented in a python 2.7 module, called coevo2.py. We provide two notebooks that demonstrate usage of the module: GettingStarted.ipynb and PositionalCoevolution.ipynb. The code makes use of matplotlib, scipy, numpy, cPickle and pandas dataframes. PositionalCoevolution.ipynb involves several more computationally intensive operations, and can make use of multiprocessing via the python Pool class. One straightforward way to run the code is to request an interactive Jupyter notebook from the biohpc. This environment will assume that your notebooks are located in the directory: `~/jupyter_notebooks/`.

OBJECTIVES

During the course of the hackathon, we hope to complete four main objectives. In the first two objectives, we will be tuning key parameters of the SCA and MT algorithms, and evaluating the performance of each method as a function of this parameter variation. One relatively simple approach to evaluating performance is to construct Receiver-Operating-Characteristic (ROC) curves, which plot the true positive rate (or recall) vs. the false positive rate (or fall-out rate).

1. Evaluating the role of alignment depth and diversity

One major factor influencing the outcome of coevolutionary analyses is alignment depth (how many sequences are included) and diversity (how wide a swath of phylogeny - effectively the evolutionary tree - is sampled). While the typical assumption is that larger and more diverse alignments are better for analyses, some rapidly evolving protein features may be more readily detected from less diverse (clade-restricted) alignments. In our analysis, we will tune alignment depth and diversity in two ways. First, we will construct a series of subsampled "*E. coli* oriented" alignments,in which we retain only sequences with a defined level of similarity to the *E. coli* sequence. We will then assess how MT and SCA performance change as a function of alignment depth. Secondly, we will tune the application of sequence weights. Sequence weights are a strategy for minimizing the contribution of highly similar sequences in an alignment: during the analysis the contribution of each sequence gets down-weighted by a factor proportional to the number of sequences similar to it: $w_s = 1/N_s$, where N is the number of sequences with an identity above the threshold $1 - \delta$. We will directly change the cutoff δ, which is passed in as a parameter to the `mirrortree()` function of the `coevo2.py` module, and evaluate how this changes the performance of MT and SCA.

2. Evaluating the role of phylogenetic models in MT

In MT (but not SCA), we correct the observed correlations between proteins by partial regression against a phylogenetic model. The goal is to remove the contribution of correlations explained by phylogenetic relationships rather than shared functional constraints. Phylogenetic models can be constructed in different ways, for example, by examining the sequence similarities over a series of non-interacting housekeeping genes (proteins). The goal is to define a model which captures the "expected" sequence similarities for proteins sampled over a given set of species, in the absence of any interaction. We will compute several phylogenetic models and examine mirrortree performance for each. Phylogenetic model construction is illustrated in `GettingStarted.ipynb`. In order to produce a phylogenetic model based on housekeeping genes, we have collected alignments in `~/TestSet/eggNOG_aligns/phylogenes`. Additional sequence choices can be readily included in this set; one interesting option would be to construct a phylogenetic model over a very large number of genes, for example the entire *E. coli* proteome.

3. Quantitatively defining expectations for positive and negative interaction cases in both MT and SCA

It is critical to know what computational scores or outcomes correspond to truly non-interacting and truly interacting protein pairs. To establish this, we will examine the distribution of SCA and MT scores for: 1) non-interacting protein pairs in our test set and 2) between two halves of the same protein (which, by definition, strongly interact). We can assemble a large number of split-protein alignments for use as test cases using the function: `~/TestSet/splitAlign.py`. An interesting possible outcome of this objective would be using the score distributions for interacting and non-interacting test set pairs to establish a p-value based score for interaction prediction (e.g. what is the probability that the protein comes from the non-interacting distribution?, for a possible starting point see this paper.)

4. Understanding the positional basis for protein interaction

Both SCA and MT can be formulated to provide an amino acid resolution view of which positions co-evolve. In this objective, we will examine: 1) How is the coevolutionary signal distributed across the sequence? Is the signal carried by a few positions, or uniformly distributed? 2) Which positions contribute the most to coevolution? Are they the most evolutionarily conserved positions, or the most rapidly evolving sites? Are they structurally distributed, or focused around the protein interaction interface or active site? The first steps towards this objective, including calculating positional MT, are illustrated in `~/PositionalCoevolution.ipynb`. Structures for a few proteins of interest are located in `~/TestSet/PDBs`; and this set can be readily expanded using the information in `~/TestSet/TestSetSummary.xlsx`

Beyond these objectives, other productive possible directions include expanding the test set (with additional EggNOG alignments), and using agglomerative clustering (or k-means, or perhaps another approach) to define coevolving residue groups within each protein. These new co-evolving subunits would then form the basis for a hierarchical version of mirrortree, in which we analyze correlations among subsets of the protein sequence, cluster, and then re-analyze over the new clustered set (Fig. 3).

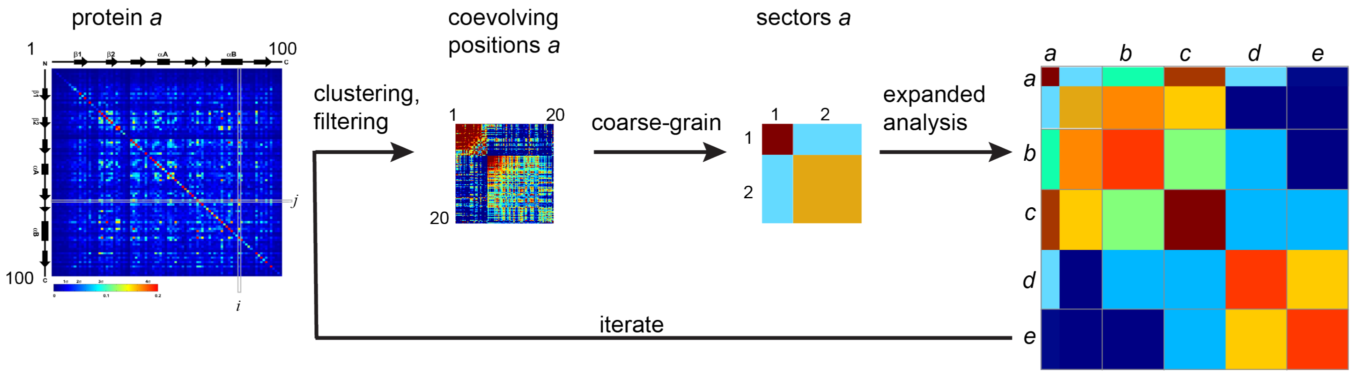


Figure 3. Hierarchical mirrortree schematic.

A BRIEF (MATHEMATICAL) PRIMER ON CO-EVOLUTIONARY SEQUENCE ANALYSIS

SCA and MT both operate on alignments of homologous protein sequences, with the expectation that the sequences in each alignment share common selective constraints on function. Below, we show two common representations of alignments in the code `coevo2.py`. (Fig. 4).

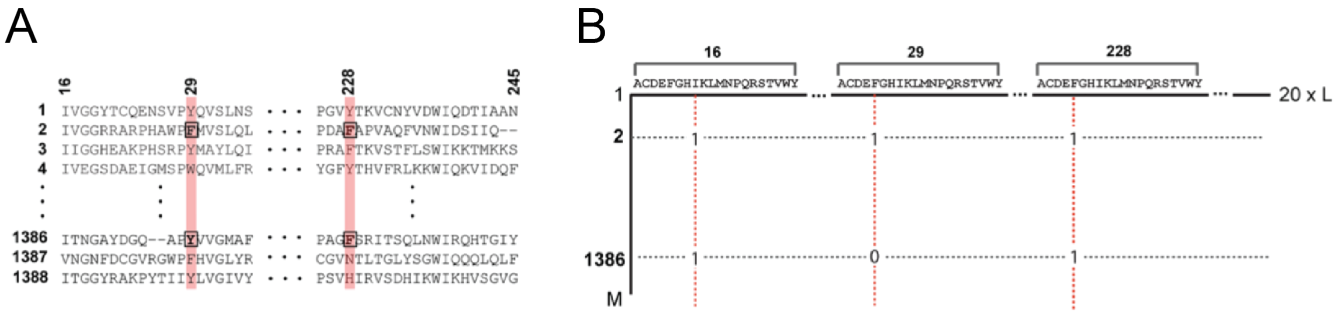
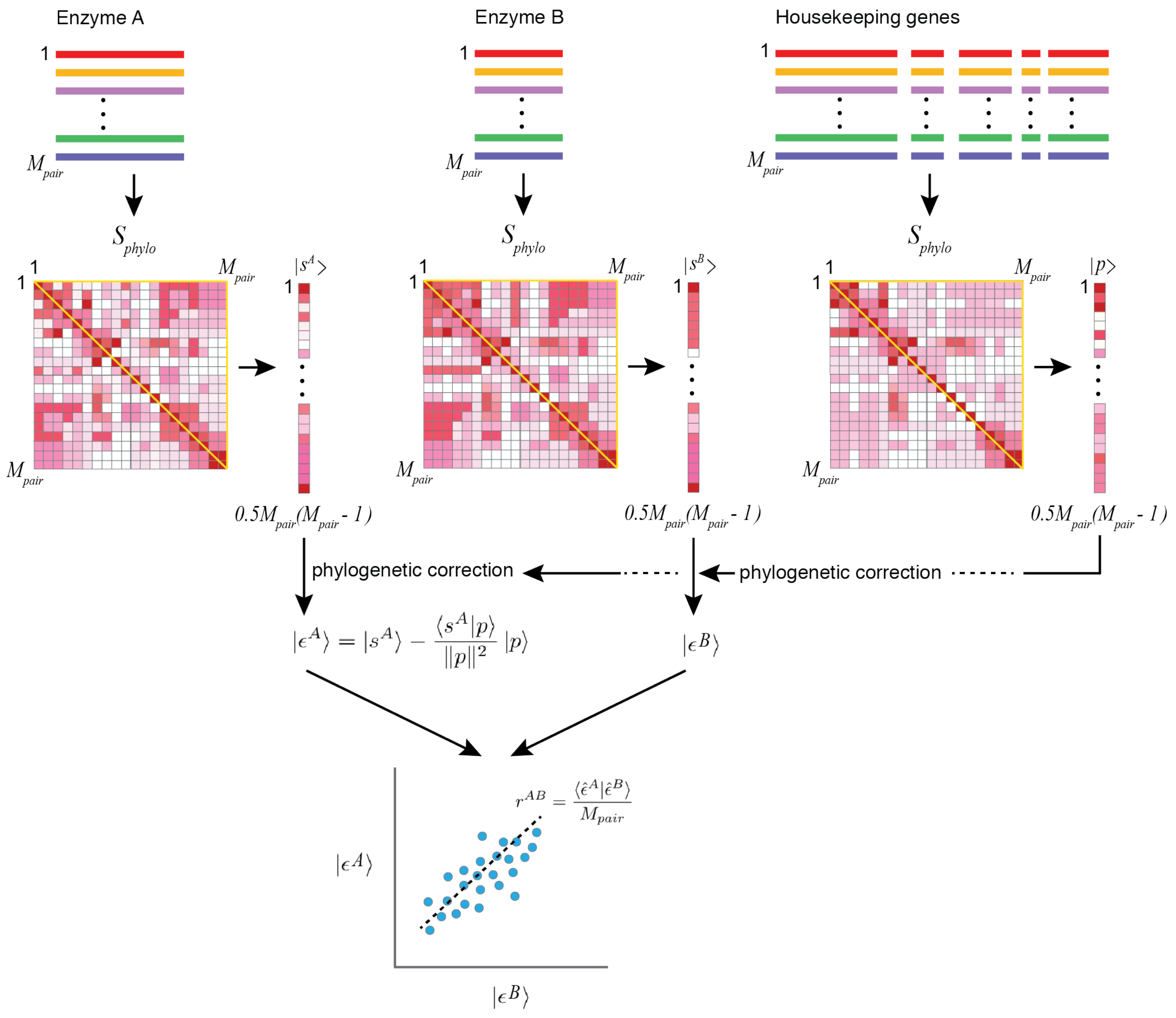

A
16
29
228
245
1 IVGGYTCQENSVPYQVSLNS ... PGVYTKVCNYVDWIQDTIAAN
2 IVGGRRARPHAWPFMVSLQL ... PDAFAPVAQFVNWIDSIIQ--
3 IIGGHEAKPHSRPYMAYLQI ... PRAFTKVSTFLSWIKKTMKKS
4 IVEGSDAEIGMSPWQVMLFR ... YGFYTHVFRLKKWIQKVIDQF
1386 ITNGAYDGQ--APYVVGMAF ... PAGFSRITSQLNWIRQHTGIY
1387 VNGNFDCGVRGWPFHVGLYR ... CGVNTLTGLYSGWIQQQLQLF
1388 ITGGYRAKPYTIIYLVGIVY ... PSVHIRVSDHIKWIKHVSGVG
B
16
29
228
ACDEFGHIKLMNPQRSTVWY
ACDEFGHIKLMNPQRSTVWY
ACDEFGHIKLMNPQRSTVWY
20 x L
1
2
1386
M
1
1
1
1
0
1

Figure 4. Multiple Sequence Alignment Representations. Figure is reproduced from Rivoire et al.. A) Text based representation. This representation can be instantiated from a fasta file using the function `read_algs()`. The alignment is composed of M sequences (rows) and L positions (columns). The text representation can be converted to a numeric format, in which each amino acid (or gap) is encoded by a number 1-21 using the function `lett2num()`. B) Binarized representation. In this format, the columns are expanded to the number of positions times the number of amino acids, $L * 21$, and a 1 or 0 indicates the amino acid identity at each site. This representation can be obtained using the function `binarize_msa()`



Enzyme A
Enzyme B
Housekeeping genes
1
M_{pair}
S_{phylo}
$|s^A>$
$|s^B>$
$|p>$
$0.5M_{pair}(M_{pair} - 1)$
phylogenetic correction
phylogenetic correction
$|\epsilon^A\rangle = |s^A\rangle - \frac{\langle s^A|p\rangle}{||p||^2}|p\rangle$
$|\epsilon^B\rangle$
$r^{AB} = \frac{\langle\hat{\epsilon}^A|\hat{\epsilon}^B\rangle}{M_{pair}}$
$|\epsilon^A\rangle$
$|\epsilon^B\rangle$

Figure 5. Calculating a MirrorTree protein interaction score.

MirrorTree (MT)

The goal of MT is to detect co-evolution by looking for correlations in the evolutionary rate of two proteins (Fig. 1). The method provides a statistical assessment of whether two protein families are undergoing similar amounts of sequence change across the same branches of a phylogenetic tree. To begin,a given protein family A is represented by an M x M sequence similarity matrix, S^A. This two-dimensional array is computed from the multiple sequence alignment (MSA), and describes the percent identity of the amino acid sequence across all pairs of species that contain it. The information in a similarity matrix provides the basis for estimating phylogenetic trees (hence the name of the algorithm), though a phylogenetic tree is not explicitly constructed in many mirrortree implementations (including ours). The upper-triangle of S^A contains $M_{pair} = 0.5M(M-1)$ total elements where each possible pair of species is represented exactly once. Since the object of MT is to compare identity change across species, this upper triangle is reshaped into a one-dimensional vector denoted $|s^A\rangle$ in bra-ket notation. The base implementation of mirror-tree computes the Pearson correlation between two such similarity vectors. We define $|\hat{s}^A\rangle = (|s^A\rangle - |\mu_{s^A}\rangle)/\sigma_{s^A}$, where μ_{s^A} and σ_{s^A}

are the mean and standard deviation of $|s^A\rangle$ respectively. The term $|\mu_{s^A}\rangle$ is used to represent a vector with the same dimensions as $|s^A\rangle$ where every element is equal to the mean. The Pearson correlation between two families A and B can then be written using the following inner product:

$$r^{AB} = \frac{\langle \hat{s}^A | \hat{s}^B \rangle}{M_{pair}} \qquad \textbf{Equation 1.}$$

By definition, this value ranges from -1 to 1 and captures the magnitude and direction of linear association between the two similarity vectors. Negative values indicate that one protein family is being conserved while the other is varying, and vice versa. This is an extreme case and not expected to occur in the base implementation of mirror-tree due to the phylogenetic relationship of the samples, but can occur when phylogeny is corrected for (see below). Positive values occur when the conservation or variation of one protein family is correlated with the other. Positive values with the highest magnitude are taken as an indicator of functional coupling.

To correct for the contribution of phylogeny to the above correlations, we modify the above equation to compute the partial correlation between s^A and s^B given an underlying dependence on phylogeny, following from (Sato et al.). As with individual protein families, phylogenetic information is encoded in an M x M similarity matrix defined across the same set of M species as the protein pair under consideration. A number of numerical models of phylogeny may be used, including a simple average sequence similarity across all analyzed protein families contained in those M species. Other examples include the sequence similarity of 16s ribosomal RNA as well as various housekeeping genes (such as the subunits of RNA polymerase). The upper-triangle of the phylogenetic similarity matrix P is reshaped into a one dimensional vector which we denote $|p\rangle$. First, we obtain the components of $|s_A\rangle$ and $|s_B\rangle$ that are orthogonal to phylogeny $|p\rangle$ by subtracting their projection on to the phylogenetic vector:

$$|\epsilon^A\rangle = |s^A\rangle - \frac{\langle s^A | p \rangle}{\|p\|^2} |p\rangle \qquad \textbf{Equation 2.}$$

The Pearson correlation between orthogonal components $|\epsilon^A\rangle$ and $|\epsilon^B\rangle$ is then computed from an inner product of their standard scores $|\hat{\epsilon}^A\rangle$ and $|\hat{\epsilon}^B\rangle$ as above. This value constitutes the partial correlation r^{ABp} between families A and B with the effect of $|p\rangle$ removed, and has been shown to yield fewer false positives than the uncorrected MT score.

Statistical Coupling Analysis (SCA)

SCA detects co-evolution at the level of individual amino acid positions by examining correlations in amino acid frequency between protein positions. A key aspect of the SCA method is that these correlations are weighted by conservation, such that correlations between more conserved sites are emphasized. Conservation of a single amino acid a at a position i is computed as the Kullback-Leibler relative entropy D_i^a given f_i^a (the frequency of a at site i), and q_i^a (the background distribution of amino acids):

$$D_i^a = f_i^a \ln \frac{f_i^a}{q^a} + (1 - f_i^a) \ln \frac{1 - f_i^a}{1 - q^a} \qquad \textbf{Equation 3.}$$

To compute correlations in conservation between pairs of positions, we weight the raw correlations in amino acid frequency by a conservation based term:

$$\tilde{C}_{ij}^{ab} = \phi_i^a \phi_j^b |f_{ij}^{ab} - f_i^a f_j^b| \text{ where } \phi_i^a = \frac{\partial D_i^a}{\partial f_i^a} = \ln \left[\frac{f_i^a (1 - q^a)}{(1 - f_i^a) q^a} \right] \qquad \textbf{Equation 4.}$$

Computed over all pairs of positions, this yields an LxLx20x20 matrix. To create a measure of coupling between positions (rather than between every amino acid at a position), the matrix is dimension reduced by taking the Frobenius norm of the 20x20 matrix for each position pair. This yields the final SCA matrix for a protein, $\tilde{C_{ij}}$.

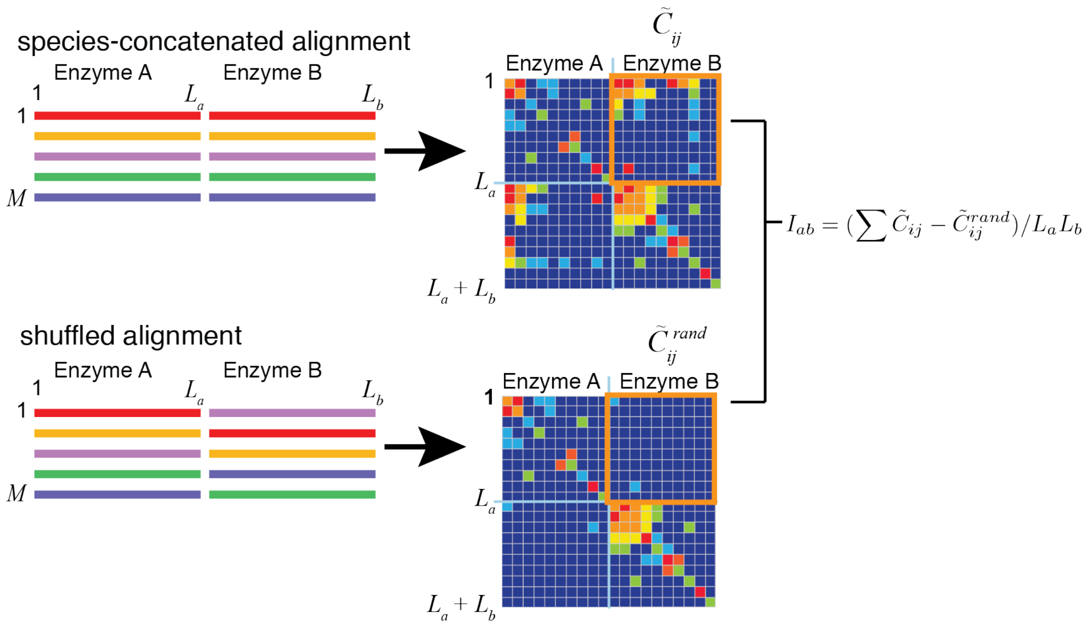


Figure 6. Calculating a SCA-based protein interaction score.

To create a SCA-based protein interaction score, we concatenate alignments for two proteins of interest by species (Fig. 6). We then compute a SCA matrix over both proteins, where off-diagonal blocks within this matrix describe inter-protein coevolution. To help identify significant inter-protein co-evolution, we compare to a "randomized" matrix, for an alignment in which the two concatenated proteins have been randomly shuffled. Given these data, we define the protein interaction score as the average across all inter-protein couplings, after subtracting the randomized matrix:

$$I_{ab} = (\sum \tilde{C}_{ij} - \tilde{C}_{ij}^{rand})/L_aL_b \qquad \textbf{Equation 5.}$$