CHAPTER TWO

Inference of Direct Residue Contacts in Two-Component Signaling

Bryan Lunt,* Hendrik Szurmant,† Andrea Procaccini,* James A. Hoch,† Terence Hwa,‡ *and* Martin Weigt*

Contents

Abstract

Since the onset of the genomic era more than 1000 bacterial genomes have been sequenced and several fold more are expected to be completed in the near future. These genome sequences supply a wealth of information that can be

* Institute for Scientific Interchange, Viale S. Severo 65, Torino, Italy
† Department of Molecular and Experimental Medicine, The Scripps Research Institute, La Jolla, California, USA
‡ Center for Theoretical Biological Physics, University of California San Diego, La Jolla, California, USA

Methods in Enzymology, Volume 471
ISSN 0076-6879, DOI: 10.1016/S0076-6879(10)71002-8

exploited by statistical methods to gain significant insights into cellular processes. In Volume 422 of *Methods in Enzymology* we described a covariance-based method, which was able to identify coevolving residue pairs between the ubiquitous bacterial two-component signal transduction proteins, the sensor kinase and the response regulator. Such residue position pairs supply interaction specificity in the light of highly amplified but structurally conserved two-component systems in a typical bacterium and are enriched with interaction surface residue pairings. In this chapter we describe an extended version of this method, termed "direct coupling analysis" (DCA), which greatly enhances the predictive power of traditional covariance analysis. DCA introduces a statistical inference step to covariance analysis, which allows to distinguish coevolution patterns introduced by direct correlations between two-residue positions, from those patterns that arise via indirect correlations, that is, correlations that are introduced by covariance with other residues in the respective proteins. This method was shown to reliably identify residue positions in spatial proximity within a protein or at the interface between two interaction partners. It is the goal of this chapter to allow an experienced programer to reproduce our techniques and results so that DCA can soon be applied to new targets.

1. Introduction

Proteins serve to execute most biochemical functions within all cellular organisms. Equally important as the individual activity of a particular protein is its ability to specifically interact with partner proteins. Cases in point are multiprotein machineries such as the ribosome, RNA polymerase holoenzyme, or the bacterial motility apparatus, the flagella. Most of the individual components of these macromolecular complexes are without use to a cell when not in contact with the other components of the complex. For this reason, protein–protein interaction interfaces are considered as potential, yet relatively unutilized drug targets (Wells and McClendon, 2007).

High-resolution X-ray structures have provided significant insights into many macromolecular protein complexes and identified their protein–protein interfaces. One of the great success stories in X-ray crystallography was the resolution of the entire ribosome (Ramakrishnan, 2008; Yusupov *et al.*, 2001). Still, not all protein complexes are as inherently stable as the above-mentioned examples (Cusick *et al.*, 2005). Indeed, there is a requirement for many protein interactions to be transient to allow a single protein to travel in cellular space and to interact with different partners, while utilizing overlapping interaction surfaces. An example, well known to the bacterial signal transduction community, is the chemotaxis response regulator protein CheY, which utilizes overlapping surfaces to interact with either the P2 domain of the kinase CheA, the C-terminal signature peptide

of the phosphatase CheZ or the N-terminal signature peptide of the flagella switch protein FliM (and FliY in *Bacillus subtilis*) (Dyer *et al.*, 2004; Szurmant *et al.*, 2003; Welch *et al.*, 1998; Zhao *et al.*, 2002; Zhu *et al.*, 1997). In general protein–protein interactions in signal transduction are expected to be transient, for above-mentioned reasons. Capturing such transient interactions in X-ray crystals has proven challenging.

The common signaling cascade utilized by the bacteria is the two-component system (Hoch, 2000). These systems transform a signal into an appropriate response via two proteins, a signal detecting sensor histidine kinase and a response regulator protein, typically a transcription factor. The message between the proteins is passed by transfer of a phosphoryl group from the kinase to the regulator. Not due to lack of effort, as of August 2009, there was no published structure of the complex of a true sensor kinase/response regulator trapped in the act of phosphotransfer. A close structural and functional homologue and the only representative crystal structure of such a complex is that of the sporulation phosphorelay proteins Spo0B with Spo0F (Zapf *et al.*, 2000). Based on structural similarity, Spo0B is an evolutionary divergent kinase, having lost the ability to autophosphorylate but having retained the ability to interact with and transfer phosphate to response regulator proteins Spo0F and Spo0A (Burbulys *et al.*, 1991; Varughese, 2002).

In the light of the obvious difficulty of capturing transient interactions by experimental means, we recently developed a covariance-based method utilizing the mutual information (MI) measure with the aim of identifying residue/residue contacts at protein/protein interfaces from sequence alone. This method, applied to two-component signaling proteins, was described in some detail in Volume 422 of *Methods in Enzymology* (White *et al.*, 2007). This chapter is an extension of the previous work featuring an additional step, which vastly improves the predictive power of covariance analysis for reasons outlined below.

Covariance-based methods have been extensively applied to gain insights into protein tertiary structure (Altschuh *et al.*, 1987; Atchley *et al.*, 2000; Göbel *et al.*, 1994; Suel *et al.*, 2003) and more recently to identifying protein interaction surfaces (Burger and van Nimwegen, 2008; Kass and Horovitz, 2002; Thattai *et al.*, 2007; White *et al.*, 2007). These methods are based on the underlying assumption that structural details of the interaction are conserved across homologous proteins, and that the residue positions at the contact surface between two protein interaction partners (or in contact within a protein fold) are constrained. Not all amino acid combinations are equally acceptable for positions in contact; the statistical properties of pairs of contact positions thus differ from arbitrarily chosen residue position pairs. To measure such constraints one needs to have a large protein sequence database of homologous paired interaction partners. In the light of the number of bacterial genomes that have been sequenced ($\sim$1000) or

whose sequence is in the works (~3500) (Liolios *et al.*, 2008), such methods are starting to become amenable for those proteins, which are encoded by the majority of bacterial genomes. Some protein systems are amplified in bacterial genomes, for example, two-component signaling systems. While these provide more statistically relevant sequence data for analysis, an added difficulty is that the actually interacting protein pairs cannot be identified merely by being found in the same genome. Instead additional information is necessary to identify, which two proteins are interaction partners. For two-component systems, this tends to be unproblematic since a large fraction of these systems are organized into operons. Hence, chromosomal adjacency is utilized to infer interaction partners.

In this context, MI measures the amount of information provided by the knowledge of the amino acid present in one position (in the first protein) about the one present in the other position (in the second protein). When applying covariance analysis to two-component signaling proteins to infer how the two proteins interact with each other during phospho-transfer, it becomes apparent that within the highest MI residue position pairs (i.e., the most constrained ones), many were found to be in close proximity in the above mentioned Spo0B/Spo0F cocrystal structure, demonstrating that covariance analysis is able to strongly enrich surface contact pairings (White *et al.*, 2007). Other high-MI residue pairings, however, were distant from the interface and involved cluster of residues connecting buried residues in the four-helix bundle core of the kinase to a highly dynamic region of the response regulator. The importance of these correlations for SK/RR recognition has been described (McLaughlin *et al.*, 2007; Szurmant *et al.*, 2008). In the light of identifying protein interaction surfaces, however, these highly correlated pairings have to be considered false positives.

A shortcoming of covariance analysis is that correlations between a pair of residue positions might arise from direct as well as indirect effects. Indirect effects can occur, for example, when a given residue has a conformational effect on the placement of residues at the protein interface. Similarly, a highly connected net of weak direct interactions will lead to inflation of covariance values due to multiple correlation chains. To reduce the effect of correlation chains we previously applied a so-called ''best-friend'' transformation (White *et al.*, 2007). Within the set of highly correlated residue positions only those pairings are considered relevant, which display the highest MI value for a particular residue positions. While such a transformation reduces the effect of correlation chains it certainly does not eliminate them, and in addition relevant information is discarded.

Covariance analysis cannot distinguish between direct and indirect correlations since it is a local measure, that is, correlations between residue positions are calculated individually without the context of the other residue

positions in the proteins of interest. To distinguish between direct and indirect correlations, each residue position pair has to be investigated in the context of all other positions in the proteins. To achieve this we developed an improved method, called direct coupling analysis (DCA), which adds a global model-inference step using message passing to covariance analysis (Weigt *et al.*, 2009). In this step, the MI values derived by covariance analysis are split into direct and indirect contributions. The approach is based on the premise that only these strong direct interactions are an indicator for correlated substitutions caused by functional residue contacts. The procedure produces a new measure termed direct information (DI), which represents the contribution to the MI that is estimated to derive due to direct correlation of two residue positions.

When, applying DCA to two-component signaling proteins, it becomes apparent that the predictive power of covariance analysis is greatly enhanced by the message-passing step. This has been described in detail (Weigt *et al.*, 2009). Here we highlight some of the results. Positions of the top 15 MI and DI pairings in structural HisKA and RR models are compared in Fig. 2.1A. Within the set of highly correlated residue positions less than 40% are found at the interaction surface between Spo0F and Spo0B, whereas the remainder of connections is distal to the interface (Fig. 2.1B and C). After applying the message-passing filter, the set of directly correlated positions involves 10 pairings that are at the interaction surface in the Spo0B/Spo0F cocrystal structure.[1] Such information is sufficient to generate high-resolution structural models of protein complexes, where the structure of the individual proteins is known, as described in the following chapter in this edition of *Methods in Enzymology*.

In the following we describe the step-by-step procedure of how to extract coupled interaction surface residue positions from databases of interacting proteins. The dataflow of the entire process is given in Fig. 2.2. The procedure is presented in two major sections. The first section, Section 2, describes how a database of interacting protein sequences is build. Genomic data is downloaded from NCBI's RefSeq database (described in Section 2.1), analyzed to create a database of predicted operons (described in Section 2.2), and searched with the HMMER package to extract and align protein domains of interest to the user (described in Section 2.3). Extracted protein domains are joined into predicted pairs (i.e., pairs of sample sequences of interacted proteins) using the Operon database (described in Section 2.4). Utilizing the assembled database, the second section, Section 3, describes how coupled residue positions are identified by MI analysis, and subsequently how direct correlation is distinguished from

[1] The other five parings involve HisKA helix α2 residues 291, 294, and 298, which are ignored here, since they cannot be reliably mapped to Spo0B, but likely represent interface contact pairings in SK/RR complexes.

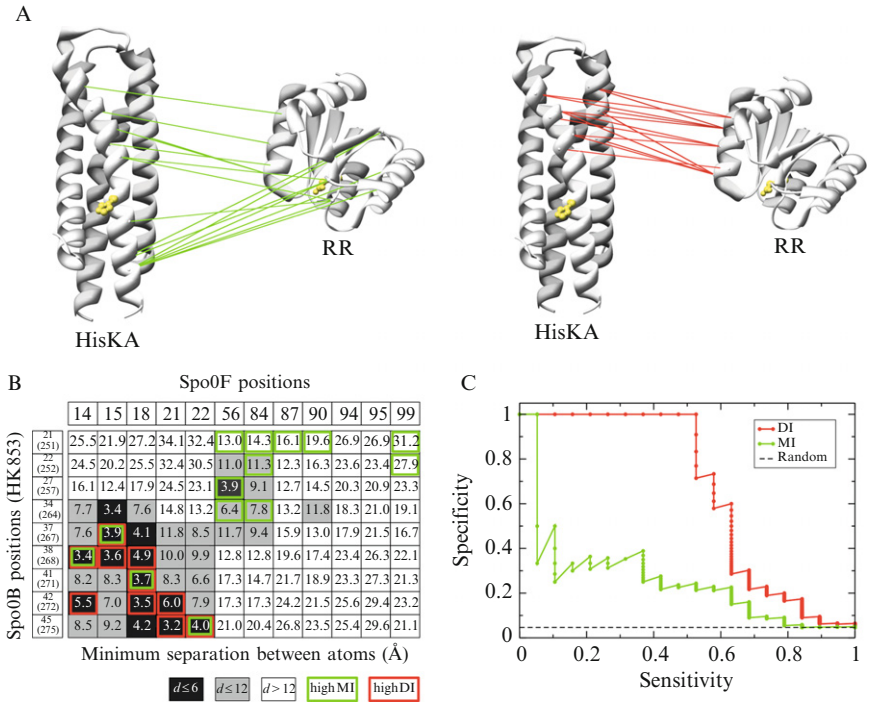

Figure 2.1 Comparison of results derived by covariance analysis with direct coupling analysis. (A) The top 15 residue pairings identified by covariance analysis (left in green) and the top 15 residue pairings identified by direct coupling analysis (DCA), which includes an additional statistical inference step are mapped on exemplary structures for the HisKA domain (HK853 from *Thermotoga maritima*: PDBID 2C2A) and RR domain (Spo0F from *B. subtilis*: PDBID 1PEY). It becomes apparent that the additional inference step increases strongly the specificity of the contact residue prediction. (B) The table shows minimal atom distances in the Spo0B/Spo0F cocrystal structure between all residues that are identified by covariance analysis or DCA. The top pairings identified by covariance analysis are framed in green and identified by DCA are framed in red. Since the Spo0B helix $\alpha 2$ is oriented different and cannot be aligned with regular HisKA helix $\alpha 2$, residue positions 291, 294, and 298 (HK853 numbering) are ignored for this analysis. Out of the 15 pairings displayed in each of the figures in Panel (A), 14 high-MI and 10 high-DI pairings do not involve these residues, and are included into the figure. (C) Comparison of specificity/sensitivity curves of covariance analysis versus DCA, where distances below 6 Å in the Spo0B/Spo0F cocrystal structure are considered as real contacts. (See Color Insert.)

indirectly coupled positions by applying a message-passing algorithm. In Section 3.1, we explain a reweighting procedure to compensate for unequal sampling of the space of possible protein sequences. The main part of the section deals with the extraction of correlation measures, in particular

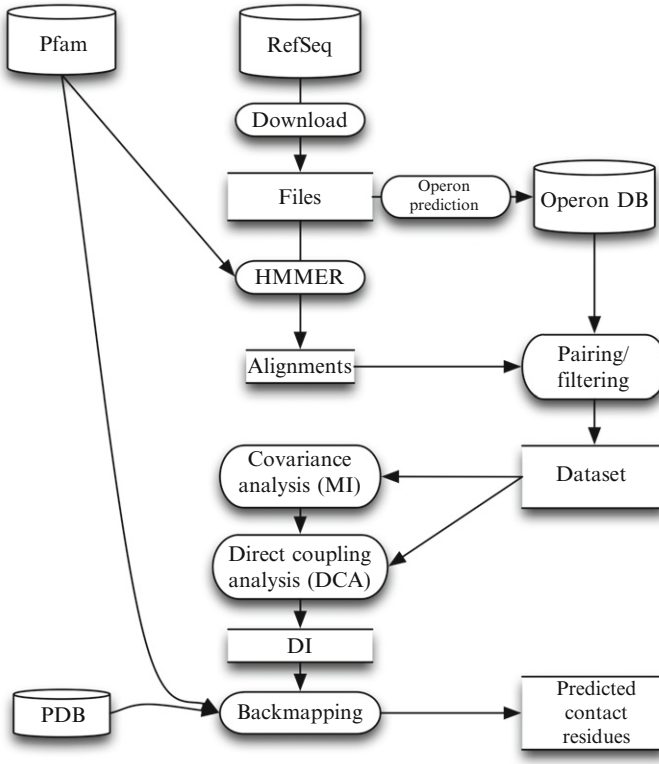

Figure 2.2 The dataflow of the entire process. Data is retrieved from public databases, analyzed to predict operons and populate the Operon database, searched with HAMMER and paired and filtered with the Operon database to create the dataset. In the "direct coupling analysis" step, the statistical correlations between columns in the protein alignments are analyzed to determine which are direct and indirect. Finally, "direct information" is used to predict interfacing residues, and this prediction is expressed on a molecular model.

MI and the novel DI, which will provide candidate pairs for interprotein residue contacts. As the derivation of the inference algorithm in the second section is given in detail in Weigt *et al.* (2009) and based on standard Belief Propagation methods described in Kschischang *et al.* (2001), Mezard and Mora (2009), and Yedidia *et al.* (2001), we will dispense with the derivation and describe only implementation in concrete terms. It is the ultimate goal that using this document, an experienced programer should be able to reproduce our techniques and results. Furthermore, it is greatly hoped that these techniques can then be extended to new problems, or added to by the reader.

2. Extraction Tools

Before it is possible to make any detailed inference about a pair of interacting protein domain families, it is necessary to generate a paired dataset of those domains in question. In the case where the domains appear on the same protein, this is relatively straightforward. However, interprotein interactions are of as much or more interest, and the generation of these datasets presents significantly more difficulty. The dataset generation phase is of crucial importance, because larger datasets will provide better estimates of the statistical properties of the domains in questions. However, if a large dataset is generated at the expense of fidelity, the estimates generated will be of little value. In this section, we present the tools and techniques for extracting the largest possible high-fidelity datasets.

2.1. Data sources

2.1.1. RefSeq

The NCBI RefSeq database (Pruitt *et al.*, 2009) provides nonredundant, curated sequence information for a variety of organisms, its bacterial database is available from ftp://ftp.ncbi.nih.gov/genomes/Bacteria. As of July 2009, it contains 905 unique taxonomy IDs for genome projects of bacteria and their substrains. This data is made available in a variety of common file-formats.

2.1.2. Pfam

The Protein Families Database (Finn *et al.*, 2008) contains HMMER (Eddy, 1998) format profile hidden Markov models (HMM(s)) for 10,340 protein domains and families. These models are instrumental in the extraction and alignment phase of dataset generation.

2.2. Operon database

To go beyond the limitations of earlier protein-pair prediction schemes based on GI number adjacency (and presumed genetic adjacency), in this work we have introduced the use of a database of predicted operons. Genes that function together are often transcribed on a single mRNA in bacteria, and in particular this is the case for many of the TCS, which are at the center of interest in this publication. We take advantage of this fact to increase the number and accuracy of our protein-pair predictions.

2.2.1. Database structure

The database is a simple relational database consisting of two necessary and two optional tables that represent the relationship between genes, predicted operons, and optionally chromosomes and genomes. The Entity Relationship diagram of the database is given in Fig. 2.3.

From the various files provided in the RefSeq directory indicated in Section 2.1, a unified list of all predicted protein and RNA coding genes can be extracted for each chromosome. This list is then broken into regions of contiguous genes with the same coding sense. Finally, each of these contiguous regions are broken into predicted operons at any intergenic region larger than a specified threshold, in our case chosen to be any distance larger than 200 base pairs (bp). Operons are predicted solely with an intergenic distance cutoff of same-sense genes inspired by (Moreno-Hagelsieb and Collado-Vides, 2002). Brouwer *et al.* (2008) concluded that Moreno-Hagelsieb and Collado-Vides' method is more effective than many more sophisticated contributions involving considerable time investment.

2.2.2. Comparison to known operons

Of 876 *Escherichia coli* operons contained in RegulonDB (Gama-Castro *et al.*, 2008) whose descriptions contain either the word "experimental" (experimentally identified) or that are predicted based on having no adjacent genes on the same strand, 576 are predicted identically with the 200 bp cutoff, 219 are joined with others, but have none of their own genes separated, and 19 are split apart, with 62 unaccounted for. Since it is more

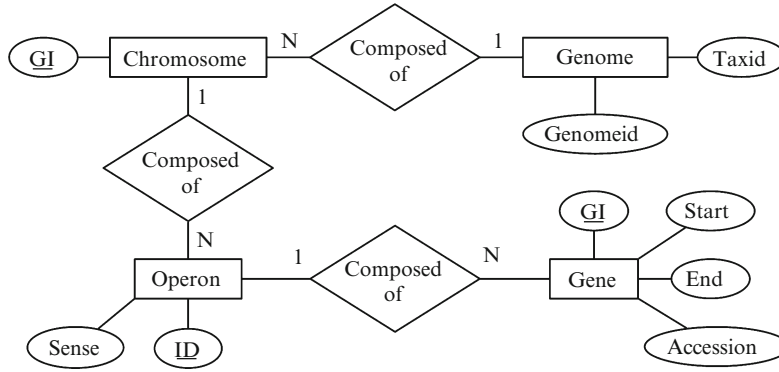

Figure 2.3 Entity relationship diagram of a simple realization of the Operon database. Genes store the starting and ending position, GI number, an Accession, and the Operon ID of the operon that contains them. Operons contain a system generated ID and the coding sense of the genes contained thereon. If the Chromosome and Genome tables are impelmented, Operons will also have a reference to the GI of the chromosome that contains them, Chromosomes make reference to the Genome that contains them. Any of these tables can be extended with other information.

important in our application to maintain the grouping of nearby genes while providing a reasonable cutoff for pairing searches, we consider identically predicted and joined operons as ''correct.'' This gives us a 91% accuracy for maintaining operon grouping on the experimentally identified *E. coli* operons.

2.3. Extraction and alignment

The HMMER suite of tools is used to create alignments to the HMMs of those domains from which one wishes to produce matched pairs. Additionally, any domain that will be used in the logical filtering step must also be extracted and aligned. For example, in addition to the HisKA and Response_reg domains, the HATPase_c domain is also extracted to be used as a filter to improve the extraction specificity of HisKA-containing proteins. All extracted domains are automatically aligned to their HMMs, and we use this alignment to join all domains into large multiple-sequence alignments (MSA).

As insert positions in the alignment will be of varying length and introduce MSA columns that are predominantly gap characters, insert positions are discarded, resulting in alignments only of those residues from match and gap states in the HMM.

It is essential at this stage that the GI number, accession, subsequence location of the match, and other data be stored for use in the next processes. In our implementation, this is stored in the description line of a FASTA-like format file, for example:

```
>gi|16131282|ref|NP_417864.1|/5-117 E=2.4e-41
  [Escherichia_coli_K12_substr_MG1655]
```

Note that it may be practical to include the *E*-value of the alignment to the HMM for further processing into this description line, as will be discussed below.

2.4. Pairing and filtering

The processes in this section describe the actual creation of protein pairs. These steps need not be executed in the order presented here, and can be mixed together into dataflows appropriate for the system under scrutiny.

2.4.1. Single protein architecture filtering

Members of each input alignment can be filtered based on the presence or absence of other domains on the same protein, for example, HisKA containing proteins without an occurrence of HATPase_c may be discarded, as can hybrid proteins containing both HisKA and Response_reg, and proteins containing multiples of either, etc.

The fraction of false positive HisKA containing proteins extracted by the HMM is drastically reduced by requiring the presence of the HATPase_c domain. No similar constraint exists for Response_reg domains (the corresponding proteins may exist as single-domain proteins). We have used the *E*-value provided by HMMer to reduce the false positive rate, including only domains with $E < 0.01$. Similarly, models from the Pfam database include various cutoff scores that have been well tuned by the Pfam team.

2.4.2. Operon pairing

All protein domains provided to the operon pairing function should have the ID of the operon that they appear on looked up in the Operon database, and be put into groups accordingly. It is important to mention that at this stage, these groupings are more flexible than being a simple pair. For example, two or more of any of the target domains may appear on one operon, or other filtering domains may or may not appear. This is handled in the next stage.

2.4.3. Operon architecture filtering

Once the domains of interest have been grouped by predicted operon, these operons can be filtered according to the presence or absence of other filter domains or even whole protein architectures appearing in the same operon. For example, in the TCS system, an operon containing multiple Response_reg domains will be discarded, whereas those with an unambiguous HisKA/Response_reg pair will be immediately added to the dataset.

2.4.4. Orphan datasets

Finally, though the handling of unpaired "Orphans" is beyond the scope of this document, it should be noted in passing that domains that could not be placed in paired proteins can be saved for use in other analyses.

2.5. Final dataset

After being extracted according to the description in the preceding sections, data takes the form of concatenated strings of length N from an alphabet consisting of the standard IUPAC amino acid codes, and an additional character ("-") to represent alignment gaps. This results in a $Q = 21$ letter alphabet. Letting M be the number of domain pairs in the dataset, this gives an MSA in the form of a $M \times N$ matrix (A_i^a), $i = 1, \ldots, N$, $a = 1, \ldots, M$ over $\{1, \ldots, Q\}$, of data from which we will compile the statistics that will be used for inference.

In the specific case of TCS, we obtained $M = 8998$ pairs. The single Pfam domains have length 87 (HisKA) resp. 117 (Response_reg), resulting in $N = 204$ letters per MSA row.

3. DCA: Direct Coupling Analysis

In this section, we present the algorithm for modeling the statistical properties of protein domain pairs. As our ultimate goal is to predict physical interprotein residue contacts in protein multimers, we will first measure the statistical correlation between residue positions in the domain-pairs using Shannon's MI (Shannon, 1948). However, while MI can help determine which residue pairs show a statistically significant correlation, MI alone cannot reveal which pairs interact directly. Significant values of MI can be the result not only of strong direct couplings, but also of multitudinous couplings through intermediate residues. Thus without some way to discriminate between the contributions to MI from direct coupling and induced coupling, we would be at a loss to identify physically interacting residue positions. In this section, we introduce a global inference method that will lead to the notion of DI, as a measure of those contributions to MI, which result only from direct interaction. The main idea of disentangling direct and indirect statistical coupling is illustrated in Fig. 2.4, and the dataflow for the full statistical analysis of our dataset is given in Fig. 2.5.

In this section, we describe our step-by-step approach to statistical analysis of the MSA. First we describe how to correct for uneven sampling effects by a simple reweighting procedure, and introduce reweighted frequency counts for single residue positions and position pairs. Based on these counts we calculate MI as a total correlation measure. The major part of this section is dedicated to disentangling direct and indirect statistical coupling: First the global statistical model is introduced together with compatibility

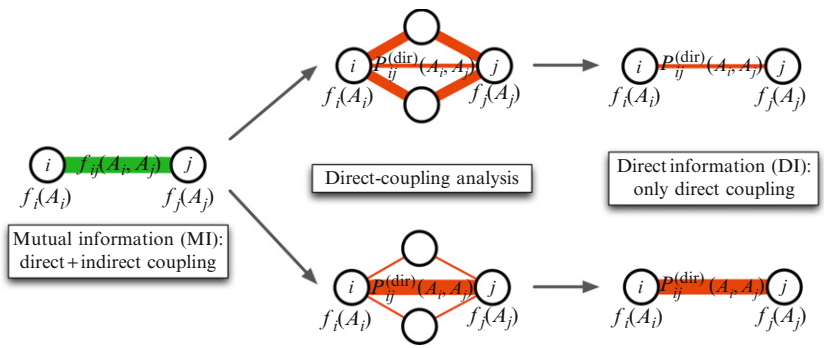

Figure 2.4 The main idea of *direct coupling analysis* (DCA). The MI between MSA columns in our dataset measures the total statistical coupling between two-residue positions in the protein domains under consideration. However, as illustrated in the figure, high MI can result from direct and indirect couplings, which are disentangled by DCA. *Direct information* (DI) measures the correlation between the two positions due to direct coupling alone, by pruning all indirect effects including intermediate positions.

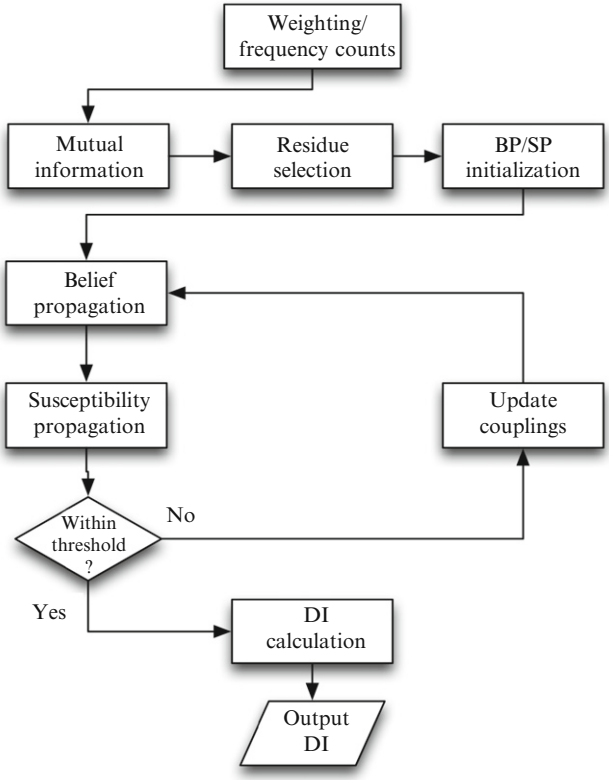

Figure 2.5 The dataflow of the *direct coupling analysis* (DCA) segment. The dataset is read and reweighted, ''mutual information'' is calculated and used to select residues for DCA, Belief Propagation and Susceptibility Propagation are used to calculate two-site marginal values, calculated values are compared to observed values and couplings are updated until overall convergence. Finally, direct information (DI) is calculated and output.

contraints to the empirical genomic data, then our approach of extracting model parameters using message passing is discussed. It results directly in a description of how to calculate DI, which is used to predict residue contacts. At the very end of this section, a backmapping procedure is discussed, which allows for translation of the results obtained in terms of MSA columns to actual representatives of the protein families.

3.1. Weighting

For a number of reasons, the dataset at this point may not represent an even sampling of the space of possible functional pairs. These reasons include the effects of phylogeny, paralogy, and oversampling of given pairs caused by

duplication within a single genome or duplicates derived from an abundance of very similar substrains of a particular bacterial species. Such pairs may represent a disproportionately large part of the dataset, and must be downweighted.

Each of the concatenated paired sequences in the dataset is taken as a single sequence. Each one is compared to all others according to a user defined distance metric. For each entry (i.e., sequence), the number of entries closer than a chosen threshold for the chosen distance metric is recorded. Thus we define the list of weights W^a for each entry a in the dataset as

$$\begin{aligned} W^a &= \frac{1}{d^a} \\ d^a &= |\{x \in \text{entries} \mid \text{dist}(x, a) \leq \text{threshold}\}| \end{aligned} \tag{2.1}$$

Note that the elements x will always include a as $\text{dist}(a, a) \equiv 0$, thus a functional pair with no similar pair receives a weight of 1, each copy of a pair that occurs twice receives a weight of 0.5 and so forth.

In our case, we chose to use the Hamming distance between the two strings, and the cutoff value of 80% sequence identity was chosen, however, for systems showing much higher conservation than the TCS system, this would not be appropriate.

3.2. Frequency counts

Both parts of the inference task are dependent solely on single and dual-site frequency counts, which are loaded from the dataset described in the previous section.

Thus we define:

$$\begin{aligned} f_i(A_i) &= \frac{1}{\lambda Q + \sum_a W^a}\left[\lambda + \sum_{a=1}^{M} \delta(A_i, A_i^a) W^a\right] \\ f_{ij}(A_i, A_j) &= \frac{1}{\lambda Q + \sum_a W^a}\left[\frac{\lambda}{Q} + \sum_{a=1}^{M} \delta(A_i, A_i^a)\delta(A_j, A_j^a) W^a\right] \\ \delta(x, y) &= \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases} \end{aligned} \tag{2.2}$$

for the frequency of occurrence of amino acid A_i in column i, and the frequency of co-occurrence of (A_i, A_j) in residue pair (i, j). These formulae contain a pseudocount $\lambda > 0$, which helps to regularize frequency counts for finite-sample effects. It prevents zero counts, which would lead to divergent couplings in the following inference task. In our implementation we choose a pseudocount of one, which becomes less and less relevant in

cases where $M \gg \lambda Q$. It is essential to note that the pseudocount terms in Eq. (2.2) are chosen such that consistency is preserved,

$$\sum_{A=1}^{Q} f_{ij}(A_i, A_j) \equiv f_i(A_i)$$

for all $i, j \in \{1,\ldots,\mathrm{N}\}$ and all $A_i \in \{1,\ldots,\mathrm{Q}\}$

The frequency counts f_i and f_{ij} are the only inputs to the calculation of MI and to DCA. Indeed, this has important implications for the efficiency of these algorithms, as an increase in the size of the dataset will give better estimates of occurrence and co-occurrence frequencies, without affecting the speed of inference.

3.3. Mutual information

One of the simplest methods to detect correlation between column couplets is Shannon's MI (Shannon, 1948):

$$\mathrm{MI}_{ij} = \sum_{A_i, A_j \in \{1,\ldots,Q\}} f_{ij}(A_i, A_j) \ln \frac{f_{ij}(A_i, A_j)}{f_i(A_i) f_j(A_j)} \tag{2.3}$$

MI measures the Kullback–Leibler divergence of the joint distribution $f_{ij}(A_i, A_j)$ and the factorized term $f_i(A_i)f_j(A_j)$. MI is the amount of information in nats that are available about the identity of amino acid A_i by knowing A_j, and *vice versa*. One nat is equivalent to $\sim$1.44 bits = 1 bit/ln 2 = 1 nat. When $f_{ij}(A_i, A_j)$ is in fact factorized, and the joint frequency shows nothing more than what would be expected of two independent random variables, MI is zero, otherwise it is positive. Mutual information MI_{ij} is calculated for every residue pair $i, j | i < j$ and these values are stored.

3.4. Global statistical modeling

As illustrated in Figs. 2.1 and 2.4, high MI can result both from direct and indirect couplings including intermediate residues. Hence some of the high-MI pairings are found at the interaction surface of the proteins of interest whereas others are not. To improve the predictive power of covariance analysis, the direct coupling effect alone needs to be estimated, which may be connected to physical interprotein contacts; it is necessary to consider not only a single residue pair at a time (as in MI), but also to model the global statistical properties of entire protein sequences. In principle, we would thus wish to construct a full joint-probability distribution for the entire concatenated protein string $P(A_1,\ldots,A_N)$, however, in order to accurately sample this distribution directly, $\Omega(21^N)$ sequence pairs would be needed. With current dataset sizes, accurate direct estimates are only

available for pairs of residues. Even triplets of residues would need substantially more than $Q^3 = 9261$ samples, which goes beyond the currently still limited number of data available from fully sequenced genomes.

Because we are only capable of directly measuring single and pairwise joint distributions, the only consistency-check that we can make on our statistical model is that it renders those same distributions when marginalized.

$$\begin{aligned} f_i(A_i) \equiv P_i(A_i) &= \sum_{A_k|k\neq i} P(A_1, \ldots, A_N) \\ f_{ij}(A_i, A_j) \equiv P_{ij}(A_i, A_j) &= \sum_{A_k|k\neq i,j} P(A_1, \ldots, A_N) \end{aligned} \tag{2.4}$$

The principle of maximum entropy (Jaynes, 1949) provides the motivation for the otherwise minimally constrained model that we shall construct. As discussed in more detail in Weigt *et al.* (2009), this principle leads to a simple form of the statistical model in terms of pairwise residue interactions and single residue biases:

$$\begin{aligned} P(A_1, \ldots, A_N) &= \frac{1}{Z} \prod_{i<j} \exp\{-e_{ij}(A_i, A_j)\} \prod_i \exp\{h_i(A_i)\} \\ Z &= \sum_{\{A_i\}} \prod_{i<j} \exp\{-e_{ij}(A_i, A_j)\} \prod_i \exp\{h_i(A_i)\} \end{aligned} \tag{2.5}$$

This distribution includes (still unknown) local biases (fields) $h_i(A_i)$ and two-residue couplings (interactions) $e_{ij}(A_i, A_j)$, which will ultimately be used to estimate the direct interactions between i and j. Z serves to guarantee the normalization $\sum_{\{A_i\}} P(A_i, \ldots, A_j) = 1$. Readers with a statistical-physics background will recognize Eqs. (2.5) as the Boltzmann–Gibbs distribution of a disordered Q-state Potts model.

The model parameters $\{h_i(A_i)\}$ and $\{e_{ij}(A_i, A_j)\}$ have to be determined such that Eqs. (2.4) are fulfiled. To do this exactly would require summation over 21^N terms and is therefore computationally infeasible. To overcome this barrier, we first have to restrict the number of residues under consideration (i.e., to reduce N), and second to use semiheuristic approaches like message passing providing beliefs for the single-residue and pair marginals of $P(A_1, \ldots, A_N)$ (i.e., to go from an exact procedure of exponential time complexity to a semiheuristic polynomial-time algorithm).

3.5. Residue selection

The main idea of selecting potentially relevant residues is that the set of position pairs i and j with considerable direct coupling is included in the set of position pairs of high total statistical coupling as measured by MI. In our model inference, we therefore include only those columns of our dataset,

which show high MI with at least one other residue in the other protein domain.

Interdomain mutual information values MI_{ij} (where $i \leq N_1 < j$, with $N_1 = 87$ denoting the length of the HisKA domain) calculated in the previous step are therefore sorted, and those residues participating in position pairs with highest MI are progressively selected until the requisite number of residues is attained.

As all subsequent calculations will only operate on this subset of residues, we will reuse the symbol N, which from now on will be taken to be the size of the subset in consideration. The value of N can be selected by the implementor based on time and hardware constraints. In our implementation, we chose to use 60 residues. Results are found to depend only weakly on this number (Weigt *et al.*, 2009).

3.6. Initialization

The following data structures will be necessary during the execution of the program, some are optional caches of derived values, which are used often; omitting them results in a loss of speed in exchange for the smaller memory-footprint.

$$e_{ij} \in \mathbb{R}_{Q\times Q} \,|\, \forall A_i : \sum_{A_j} e_{ij}(A_i, A_j) = 0 \wedge \forall A_j : \sum_{A_i} e_{ij}(A_i, A_j) = 0$$

- The statistical "residue couplings" for every residue pair $i, j | i < j$.
- Initialized to zeros.
- For all pairs, $e_{ji} \equiv e_{ij}^{\mathrm{T}}$ will also be needed, but as computer linear algebra systems (Blackford *et al.*, 2002; Jones *et al.*, 2001) provide matrix multiplication functions that accept a transposed matrix, it is unnecessary to store these values in memory. Furthermore, specifically not doing so ensures consistency.

$$G_{ij} \in \mathbb{R}_{Q\times Q} \,|\, G_{ij}(A_i, A_j) \equiv \mathrm{e}^{-e_{ij}(A_i,\, A_j)}$$

- Optional caches of the exponentiated values of e_{ij} (Boltzmann weights), as these values are used quite often, this offers a significant performance improvement.

$$P_{i\to j} \in \mathbb{R}_{Q\times 1} | \sum_{A_i} P_{i\to j}(A_i) = 1$$

- Probability vector messages for the Belief Propagation step. This represents the belief that i has that it should take on values A_i in the absence of the direct influence of j.

- These messages are initialized randomly, but normalized according to their definition.
- In pseudocode, the datastructure containing these vectors is listed as P_ij and subscripted with the source and target, for example, P_ij[x, y] for the message from x to y.

$$M_{i\to j;k} \in \mathbb{R}_{Q\times Q} | \forall A_k : \sum_{A_i} M_{i\to j;k}(A_i, A_k) = 0$$

- Susceptibility messages giving the partial derivatives $\partial P_{i\to j}(A_i)/\partial h_k(A_k)$ for the Susceptibility Propagation step.
- Initialized to zeros.
- In pseudocode, the datastructure containing these matrices is listed as M_ijk and subscripted with the source, target, and influencing field, for example, M_ijk[x, y, z] for the message from x to y with respect to variation of the field in z.

3.7. Belief Propagation

Standard Belief Propagation (BP) is an efficient method for estimating the marginal values of unobserved nodes in Markov Random Fields. In our inverse problem, the single-site marginal values are fixed to known values (to the frequency counts f_i), but the messages from this step are necessary for the later Susceptibility Propagation step.

BP acts by passing ''beliefs'' around a graph representing a Random Markov Field, from one node to another, providing to the recipient information about what values the sender would be likely to take on in the absence of the direct influence of the recipient. While BP is exact on trees, it is also possible to send messages around a loopy graph several times until these messages converge to a fixed point (i.e., until no message is updated more than a given threshold). At the end of this process, it is possible to calculate beliefs for the marginal values of every node in the graph.

Because of the inverse nature of our inference, it is possible to realize a great improvement in efficiency over standard BP: already knowing the marginal values, we wish to calculate the fields and interaction terms. While standard BP passes messages globally around a graph, in this case, solving standard BP equations with known marginal probabilities renders single edge BP equations only dependent on the two messages sent back and forth between the two nodes:

$$P_{i\to j} \sim \frac{f_i}{G_{ij} \cdot P_{j\to i}} \tag{2.6}$$

where the proportionality indicates that the message is normalized to 1 according to its definition. See Appendix for the definition of nonstandard matrix and vector operations used in this section.

Belief Propagation

```
for every residue i:
 for every residue j > i:
  until convergence:
   update message P_ij[i, j] according to Eq.(2.6)
   update message P_ij[j, i] according to Eq.(2.6)
```

3.8. Susceptibility Propagation

Because of the fully connected nature of our graph, it is a difficult problem to determine two-point distributions from the provided interaction terms. However, Mézard and Mora have made great progress in this area with the technique of Susceptibility Propagation (SP), introduced in (Mezard *et al.*, 2009). A detailed description of SP is beyond the scope of this paper, but it allows for the efficient calculation of two-point distributions.

SP is executed in a fashion more similar to traditional BP, where all messages are interdependent, and updated until all messages have converged. To obtain efficient convergence we have chosen to use a random sequential update, where the order of the nodes originating SP messages is a random permutation of the set of nodes.

As defined previously, SP messages give the partial derivatives of BP messages with regard to local fields in any position. Their update formula, as derived in the original writing (and updated to this document's matrix notation) is given as

$$M_{i\to j;k} = \mathrm{Diag}(P_{i\to j})\left[\delta(i,k)I_Q + \sum_{l\neq i,j}\frac{G_{il}M_{l\to i;k}}{G_{il}P_{l\to i}} - c_{i\to j;k}\right] \tag{2.7}$$

where $c_{i\to j;k}$ can only be determined after the first terms have been calculated, and is chosen to enforce the normalization to zero in the above definition of $M_{i\to j;k}$. To implement message update directly as given here would be extremely inefficient, but an improvement of order N can be made by caching the summation over all $l \neq i$ and individually subtracting terms j when calculating each message, and updating all messages sent by node i at once. Pseudocode for an efficient implementation of this algorithm is provided below. Please note that some operations can be made even more efficient by using such functions as "elementwise multiplication" provided in many computer linear algebra packages, and are trivial to implement if unavailable (Blackford *et al.*, 2002; Jones *et al.*, 2001).

See Appendix for the definition of nonstandard matrix and vector operations used in this section.

$$\begin{aligned} \mathrm{spA}[l,i,k] &= \frac{G_{il}M_{l\to i;k}}{G_{il}P_{l\to i}} \\ \mathrm{spB}[i,k] &= \sum_{l\neq i} \mathrm{spA}[l,i,k] \\ \mathrm{spC}[i,j,k] &= \mathrm{spB}[i,k] - \mathrm{spA}[j,i,k] \end{aligned} \tag{2.8}$$

Susceptibility Propagation

```
until all susceptibilities converge:
  for every residue i in a random order:
    for every residue k:
      for every residue l!=i:
        cache spA[l, i, k] according to Eq.(2.8)
      spB[i,k] = Σ_{l≠i} spA[l, i, k]
      for every residue j!=i:
        M_ijk[i, j, k] = spB[i, k] - spA[j, i, k]
        if i==k:
          M_ijk[i, j, k] = M_ijk[i, j, k] + I_Q
        M_ijk[i, j, k] = diag(P_ij[i, j]) * M_ijk[i, j, k]
        M_ijk[i, j,k] = M_ijk[i, j, k] -
                        P_ij[i, j]*ColumnSums
                        (M_ijk[i, j, k])
```

As before, convergence means that no message has been updated by more than a user-selected threshold.

3.9. Parameter update

3.9.1. Does the model describe empirical data within threshold?

After all SP messages have converged, it is possible to calculate the values $\partial P_i(A_i)/\partial h_j(A_j)$ according to Eq. (2.9),

$$\frac{\partial P_i}{\partial h_j} = \mathrm{Diag}(f_i)\left[\sum_{l\neq i}\frac{G_{il}M_{l\to i;j}}{G_{il}P_{l\to i}} - c_{ij}\right] \tag{2.9}$$

Pseudocode for this is analogous to that given in the SP section, above. Finally, with these values and Eq. (2.10),

$$\frac{\partial P_i}{\partial h_j} = P_{ij} - f_i f_j^{\mathrm{T}} \tag{2.10}$$

it is possible to estimate the two-residue marginal probabilities $P_{ij}(A_i, A_j)$, which will be used in this step to determine how closely the model predicts the measured two-point marginal values $f_{ij}(A_i, A_j)$.

If no marginal value $P_{ij}(A_i, A_j)$ differs from the measured marginal $f_{ij}(A_i, A_j)$ by more than the threshold value, in our implementation, gradient descent has finished, and it is now possible to calculate DI according to the subsequent section. If any marginal value differs more than this, couplings e_{ij} must be updated according to the subsequent subsection, and the BP and SP steps must be repeated.

Note that due to the heuristic character of BP on loopy graphs, the two-residue marginals are not exact. This may lead to a situation where the following coupling updates, even if derived via gradient descent in a convex optimization problem, actually increase the distance between the model-derived $P_{ij}(A_i, A_j)$ and the empirical counts $f_{ij}(A_i, A_j)$. In this case, the algorithm is required to halt and pass the best-found parameter values to the calculation of DI.

3.9.2. Update couplings

In any case where the empirical two-residue counts and the marginals calculated via SP differ by more than the threshold value, new coupling values must be chosen according to Eq. (2.11),

$$\Delta e_{ij}(A_i, A_j) = -\Delta_{\mathrm{GD}}[f_{ij}(A_i, A_j) - P_{ij}(A_i, A_j)] \tag{2.11}$$

where Δ_{GD} is the gradient descent step size. A larger Δ_{GD} will lead to a more rapid approach to the vicinity of the fixed point of the couplings, but will also tend to cause the program to overshoot and possibly to enter an infinite loop in the endgame, thus this value needs to be adjusted to the system under scrutiny.

3.10. Direct information

Once all two-residue coupling parameters have converged to a fixed point where calculated two-site marginal values, P_{ij}, match (as well as possible) the empirically observed two-site (f_{ij}) marginal values (recall that as a result of our inverted BP step, calculated single site marginals, P_i, always match observed single-site marginals, f_i), it is possible to calculate the contribution to the MI (Eq. 2.3) given only by the direct statistical interaction of two residues, introducing the joint probability that arises only from direct interaction as

$$\begin{aligned} P_{ij}^{(\mathrm{dir})} &= \frac{1}{Z_{ij}} \mathrm{Diag}(P_{i\to j})\, G_{ij} \mathrm{Diag}(P_{j\to i}) \\ Z_{ij} &= \sum_{A_i,\, A_j} \mathrm{Diag}(P_{i\to j})\, G_{ij} \mathrm{Diag}(P_{j\to i}) \end{aligned} \tag{2.12}$$

This value allows for the calculation of DI, which is analogous to MI covered earlier in this section:

$$DI_{ij} = \sum_{A_i,\ A_j} P_{ij}^{(\mathrm{dir})}(A_i, A_j) \ln \frac{P_{ij}^{(\mathrm{dir})}(A_i, A_j)}{f_i(A_i) f_j(A_j)} \tag{2.13}$$

Interdomain residue pairs are sorted according to their DI values. Large DI, that is, strong direct statistical coupling of the two residues under consideration, is taken as a predictor for a physical contact in the protein dimer. Therefore, DI is the most important output of our approach.

3.11. Backmapping

Ultimately, the overall goal of the procedure described herein is to facilitate the prediction and analysis of contact residues in interacting protein domains. This information is most easily understood, visualized, and processed when taken in the context of molecular models. While significantly less involved than the implementations of the above processes, the procedure for backmapping identified residue pairs with high DI onto molecular structures should be mentioned.

While Pfam provides this data in the tables msd_data, pdb, and pdbmap, this necessarily excludes novel domains and data newer than the newest Pfam release. In cases where possible, it is recommended to use this provided data.

Simple backmapping can be achieved by reading sequence data from molecular model files, searching, and aligning this data with HMMER. Making note of the location of the matching subsequence, it is simple to translate columns in the aligned dataset to residues in the model file.

These model files can be visualized with highlighting according to the DI values of residues, or multiple models may be visualized and manipulated simultaneously with links generated by high DI column-pairs. Finally, these linked models may become input to molecular dynamics calculations, or other simulations.

Appendix: Nonstandard Linear Algebra Functions

Because of its usefulness in clarifying equations, we will introduce the concept of matrix division by another matrix of the same size, a column vector with the same number of rows, or a row vector with the same number of columns. In the case of a matrix divisor, each position of the dividend is divided by the corresponding position of the divisor. In the case of column and row vector divisors, respectively; each column or row of

the dividend is divided by the corresponding position of the divisor. This function is provided in some computer linear algebra systems, but can be implemented readily if not available.

Elementwise operators

Matrix Division by Matrix:

$$\begin{aligned}&\mathrm{Mat}_{M\times N} \times \mathrm{Mat}_{M\times N} \rightarrow \mathrm{Mat}_{M\times N}\\ &\mathrm{Quotient}_{\alpha,\beta} = \mathrm{Dividend}_{\alpha,\beta}/\mathrm{Divisor}_{\alpha,\beta}\end{aligned}$$

Matrix Division by Row Vector:

$$\begin{aligned}&\mathrm{Mat}_{M\times N} \times \mathrm{Mat}_{1\times N} \rightarrow \mathrm{Mat}_{M\times N}\\ &\mathrm{Quotient}_{\alpha,\beta} = \mathrm{Dividend}_{\alpha,\beta}/\mathrm{Divisor}_{\beta}\end{aligned}$$

Matrix Division by Column Vector:

$$\begin{aligned}&\mathrm{Mat}_{M\times N} \times \mathrm{Mat}_{M\times 1} \rightarrow \mathrm{Mat}_{M\times N}\\ &\mathrm{Quotient}_{\alpha,\beta} = \mathrm{Dividend}_{\alpha,\beta}/\mathrm{Divisor}_{\alpha}\end{aligned}$$

ACKNOWLEDGMENTS

This work was supported by the Center for Theoretical Biological Physics (CTBP) sponsored by the NSF (Grant PHY-0822283) with additional support from NIH grant R01GM077298 (T. H.) and by NIH grant R01GM019416 (J. A. H.). B. L. and A. P. acknowledge funding from the EC via the STREP GENNETEC (“Genetic networks: emergence and complexity”).

REFERENCES

Altschuh, D., Lesk, A. M., Bloomer, A. C., and Klug, A. (1987). Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *J. Mol. Biol.* **193,** 693–707.

Atchley, W. R., Wollenberg, K. R., Fitch, W. M., Terhalle, W., and Dress, A. W. (2000). Correlations among amino acid sites in bHLH protein domains: An information theoretic analysis. *Mol. Biol. Evol.* **17,** 164–178.

Blackford, L. S., Demmel, J., Dongarra, J., Duff, I., Hammarling, S., Henry, G., Heroux, M., Kaufman, L., Lumsdaine, A., Petitet, A., Pozo, R., Remington, K., and Whaley, R. C. (2002). An updated set of Basic Linear Algebra Subprograms (BLAS). *Trans. Math. Soft.* **28**(2), 135–151, ISSN 0098-3500.

Brouwer, R. W., Kuipers, O. P., and van Hijum, S. A. (2008). The relative value of operon predictions. *Brief. Bioinform.* **9,** 367–375.

Burbulys, D., Trach, K. A., and Hoch, J. A. (1991). Initiation of sporulation in *B. subtilis* is controlled by a multicomponent phosphorelay. *Cell* **64,** 545–552.

Burger, L., and van Nimwegen, E. (2008). Accurate prediction of protein-protein interactions from sequence alignments using a Bayesian method. *Mol. Syst. Biol.* **4,** 165.

Cusick, M. E., Klitgord, N., Vidal, M., and Hill, D. E. (2005). Interactome: Gateway into systems biology. *Hum. Mol. Genet.* **14** Spec No. 2, R171–R181.

Dyer, C. M., Quillin, M. L., Campos, A., Lu, J., McEvoy, M. M., Hausrath, A. C., Westbrook, E. M., Matsumura, P., Matthews, B. W., and Dahlquist, F. W. (2004). Structure of the constitutively active double mutant CheYD13K Y106W alone and in complex with a FliM peptide. *J. Mol. Biol.* **342,** 1325–1335.

Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics* **14,** 755–763.

Finn, R. D., Tate, J., Mistry, J., Coggill, P. C., Sammut, S. J., Hotz, H. R., Ceric, G., Forslund, K., Eddy, S. R., Sonnhammer, E. L., and Bateman, A. (2008). The Pfam protein families database. *Nucleic Acids Res.* **36,** D281–D288.

Gama-Castro, S., Jimenez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Penaloza-Spinola, M. I., Contreras-Moreira, B., Segura-Salazar, J., Muniz-Rascado, L., Martinez-Flores, I., Salgado, H., Bonavides-Martinez, C., Abreu-Goodger, C., *et al.* (2008). RegulonDB (version 6.0): Gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res.* **36,** D120–D124.

Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins Struct. Funct. Genet.* **18,** 309–317.

Hoch, J. A. (2000). Two-component and phosphorelay signal transduction. *Curr. Opin. Microbiol.* **3,** 165–170.

Jaynes, E. T. (1949). Information theory and statistical mechanics. *Physiol. Rev.* **106,** 620–630.

Jones, E., Oliphant, T., Peterson, P., *et al.* (2001–2005). Open source scientific tools for Python. http://www.scipy.org/.

Kass, I., and Horovitz, A. (2002). Mapping pathways of allosteric communication in GroEL by analysis of correlated mutations. *Proteins Struct. Funct. Genet.* **48,** 611–617.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **47,** 498–519.

Liolios, K., Mavromatis, K., Tavernarakis, N., and Kyrpides, N. C. (2008). The genomes on line database (GOLD) in 2007: Status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res.* **36,** D475–D479.

McLaughlin, P. D., Bobay, B. G., Regel, E. J., Thompson, R. J., Hoch, J. A., and Cavanagh, J. (2007). Predominantly buried residues in the response regulator Spo0F influence specific sensor kinase recognition. *FEBS Lett.* **581,** 1425–1429.

Mezard, M., and Mora, T. (2009). Constraint satisfaction and neural networks: A statistical-physics perspective. *J. Physiol. Paris* **103,** 107–113.

Moreno-Hagelsieb, G., and Collado-Vides, J. (2002). A powerful non-homology method for the prediction of operons in prokaryotes. *Bioinformatics* **18**(Suppl. 1), S329–S336.

Pruitt, K. D., Tatusova, T., Klimke, W., and Maglott, D. R. (2009). NCBI Reference Sequences: Current status, policy and new initiatives. *Nucleic Acids Res.* **37,** D32–D36.

Ramakrishnan, V. (2008). What we have learned from ribosome structures. *Biochem. Soc. Trans.* **36,** 567–574.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.* **27,** 379–423.

Suel, G. M., Lockless, S. W., Wall, M. A., and Ranganathan, R. (2003). Evolutionarily conserved networks of residues mediate allosteric communication in proteins. *Nat. Struct. Biol.* **10,** 59–69.

Szurmant, H., Bunn, M. W., Cannistraro, V. J., and Ordal, G. W. (2003). *Bacillus subtilis* hydrolyzes CheY-P at the location of its action, the flagellar switch. *J. Biol. Chem.* **278,** 48611–48616.

Szurmant, H., Bobay, B. G., White, R. A., Sullivan, D. M., Thompson, R. J., Hwa, T., Hoch, J. A., and Cavanagh, J. (2008). Co-evolving motions at protein-protein interfaces of two-component signaling systems identified by covariance analysis. *Biochemistry* **47,** 7782–7784.

Thattai, M., Burak, Y., and Shraiman, B. I. (2007). The origins of specificity in polyketide synthase protein interactions. *PLoS Comput. Biol.* **3,** 1827–1835.

Varughese, K. I. (2002). Molecular recognition of bacterial phosphorelay proteins. *Curr. Opin. Microbiol.* **5,** 142–148.

Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009). Identification of direct residue contacts in protein-protein interaction by message passing. *Proc. Natl. Acad. Sci. USA* **106,** 67–72.

Welch, M., Chinardet, N., Mourey, L., Birck, C., and Samama, J. P. (1998). Structure of the CheY-binding domain of histidine kinase CheA in complex with CheY. *Nat. Struct. Biol.* **5,** 25–29.

Wells, J. A., and McClendon, C. L. (2007). Reaching for high-hanging fruit in drug discovery at protein-protein interfaces. *Nature* **450,** 1001–1009.

White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2007). Features of protein-protein interactions in two-component signaling deduced from genomic libraries. *Methods Enzymol.* **422,** 75–101.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001). Generalized belief propagation. *NIPS* **13,** 689–695.

Yusupov, M. M., Yusupova, G. Z., Baucom, A., Lieberman, K., Earnest, T. N., Cate, J. H., and Noller, H. F. (2001). Crystal structure of the ribosome at 5.5 A resolution. *Science* **292,** 883–896.

Zapf, J., Sen, U., Madhusudan, X., Hoch, J. A., and Varughese, K. I. (2000). A transient interaction between two phosphorelay proteins trapped in a crystal lattice reveals the mechanism of molecular recognition and phosphotransfer in signal transduction. *Structure* **8,** 851–862.

Zhao, R., Collins, E. J., Bourret, R. B., and Silversmith, R. E. (2002). Structure and catalytic mechanism of the *E. coli* chemotaxis phosphatase CheZ. *Nat. Struct. Biol.* **9,** 570–575.

Zhu, X., Volz, K., and Matsumura, P. (1997). The CheZ-binding surface of CheY overlaps the CheA- and FliM-binding surfaces. *J. Biol. Chem.* **272,** 23758–23764.