Large, Noisy, and Incomplete: Mathematics for Modern Biology

by

Michael Hartmann Baym

B.S., University of Illinois (2002)
A.M., University of Illinois (2003)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author . . . .
Department of Mathematics
August 3, 2009

Certified by . . . .
Bonnie Berger
Professor of Applied Mathematics
Thesis Supervisor

Accepted by . . . .
Michel X. Goemans
Chairman, Applied Mathematics Committee

Accepted by . . . .
David S. Jerison
Chairman, Department Committee on Graduate Students

Large, Noisy, and Incomplete:
Mathematics for Modern Biology

by

Michael Hartmann Baym

Submitted to the Department of Mathematics
on August 3, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In recent years there has been a great deal of new activity at the interface of biology and computation. This has largely been driven by the massive influx of data from new experimental technologies, particularly high-throughput sequencing and array-based data. These new data sources require both computational power and new mathematics to properly piece them apart. This thesis discusses two problems in this field, network reconstruction and multiple network alignment, and draws the beginnings of a connection between information theory and population genetics.

The first section addresses cellular signaling network inference. A central challenge in systems biology is the reconstruction of biological networks from high-throughput data sets, We introduce a new method based on parameterized modeling to infer signaling networks from perturbation data. We use this on Microarray data from RNAi knockout experiments to reconstruct the Rho signaling network in Drosophila.

The second section addresses information theory and population genetics. While much has been proven about population genetics, a connection with information theory has never been drawn. We show that genetic drift is naturally measured in terms of the entropy of the allele distribution. We further sketch a structural connection between the two fields.

The final section addresses multiple network alignment. With the increasing availability of large protein-protein interaction networks, the question of protein network alignment is becoming central to systems biology. We introduce a new algorithm, IsoRankN to compute a global alignment of multiple protein networks. We test this on the five known eukaryotic protein-protein interaction (PPI) networks and show that it outperforms existing techniques.

Thesis Supervisor: Bonnie Berger
Title: Professor of Applied Mathematics

Acknowledgements

First and foremost, I would like to give my heartfelt thanks to Bonnie Berger for her mentorship, guidance, and most of all her support.

I would also like to thank:

Abby, my family, and close friends for their support and tolerance.

Jon Kelner for teaching me more mathematics than probably any other person.

Chris Bakal for being a patient and helpful collaborator as I stumbled into the world of experiments.

The anonymous reviewers for both the *RECOMB* printing and the first submission of Part I to *Cell*, their comments and the resulting analysis have substantially improved the work.

The community of exceptional graduate students whose community, support and ideas were invaluable the last six years. In particular, Kenneth Kamrin, Rohit Singh, Lucy Colwell, and Nathan Palmer.

Patrice Macaluso, for everything.

The Fannie and John Hertz Foundation, whose financial and professional support has been invaluable during my graduate career, and the community of Hertz Fellows who continue to humble and challenge me, particularly Alex Wissner-Gross, X. Robert Bao, Jeff Gore, Mikhail Shapiro, and Erez Lieberman.

The ASEE/NDSEG Graduate Fellowship, for three years of generous funding.

Petsi Pies, Diesel Cafe, Darwin's, and 1369, whose coffee likely accounts for the majority of my productivity.

Previous Publications of this Work

Portions of Part I have appeared in the proceedings of RECOMB 2008 [9], co-authored with Chris Bakal, Norbert Perrimon, and Bonnie Berger. They are reprinted with the kind permission of Springer Science+Business Media.

Part III has appeared in the journal *Bioinformatics* and as part of the proceedings of ISMB 2009 [54], co-authored with Chung-Shou Liao, Kanghao Lu, Rohit Singh, and Bonnie Berger. It is reprinted with the kind permission of Oxford University Press.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we introduce a number of techniques for using modern mathematical and computational tools to address problems at the cutting edge of biological research.

In this chapter we introduce and summarize the main contributions of this thesis. The remained of this document will be divided into three main parts. The first, concerning inference of signaling networks from perturbation experiments is drawn heavily from a paper which appeared at RECOMB 2008 [9] as well as another manuscript currently in review. The second, concerning the connection between population genetics and information theory, is in preparation for journal publication. The third, concerning a method for multiple alignment of protein-protein interaction networks has recently appeared in Bioinformatics [54].

1.1 Signaling Network Inference

A central challenge in systems biology is the reconstruction of biological networks from high-throughput data sets. A particularly difficult case of this is the inference of dynamic cellular signaling networks. Within signaling networks, a common motif is that of many activators and inhibitors acting upon a small set of substrates. In the first chapter of this thesis, we present a novel technique for high-resolution inference of signaling networks from perturbation data based on parameterized modeling of biochemical rates. We also introduce a powerful new signal-processing method for reduction of batch effects in microarray data. We demonstrate the efficacy of these techniques on data from experiments we performed on the *Drosophila* Rho-signaling network, by comparing to chemilluminescent Western blot data. In comparison to existing techniques, we are able to provide significantly improved prediction of signaling networks on simulated data, and higher robustness to the noise inherent in all high-throughput experiments. While previous methods have been effective at inferring biological networks in broad statistical strokes, this work takes the further step of modeling both specific interactions and correlations in the background to increase the resolution. The generality of our techniques should allow them to be applied to a wide variety of networks.

1.2 Information Theory and Population Genetics

Population Genetics is the study of the dynamics of genetic information as it is transmitted from one generation to the next. While much has been proven about these dynamics, a connection with information theory has never been drawn. Of particular interest is the phenomenon of genetic drift, namely the loss of information inherent in the discrete resampling of a population between generations. This is generally studied in the context of neutral models, in which mutation is taken to be non-existant and no allele has a selective bias over any other. In the second part of this thesis, we show that under most commonly studied neutral models of population dynamics the total entropy of the population decreases linearly in each generation. Moreover, this linear factor is dependent only on the number of alleles present and the ratio of the effective population of the model to the effective population of the well-studied Wright-Fischer Model. This is the first result of any sort on the information dynamics of population genetics. Further we describe a deep structural connection between communication over a noisy channel and inter-generation dynamics, in the hope that this leads to a better understanding of the information theoretic nature of evolution.

1.3 Multiple Protein-protien Network Alignment

With the increasing availability of large protein-protein interaction networks, the question of protein network alignment is becoming central to systems biology. Network alignment is further delineated into two sub-problems: local alignment, to find small conserved motifs across networks, and global alignment, which attempts to find a best mapping between all nodes of the two networks. In this chapter, our aim is to improve upon existing global alignment results. Better network alignment will enable, among other things, more accurate identification of functional orthologs across species. In the third part of this thesis, we introduce IsoRankN (IsoRank-Nibble) a global multiple-network alignment tool based on spectral clustering on the induced graph of pairwise alignment scores. IsoRankN outperforms existing algorithms for global network alignment in coverage and consistency on multiple alignments of the five available eukaryotic networks. Being based on spectral methods, IsoRankN is both error-tolerant and computationally efficient.

Part I

Inference of Signaling Networks From Perturbation Data

Chapter 2

Introduction

Our goal in this section is to develop a technique for the high-throughput inference of signaling networks. We test and develop these methods on the *Drosophila* Rho-signaling network. The majority of this chapter has appeared in RECOMB 2008 [9], the remainder is contained in a paper to appear later this year, currently in review.

Biological signaling networks regulate a host of cellular processes in response to environmental cues. Due to the complexity of the networks and the lack of effective experimental and computational tools, there are still few biological signaling networks for which a systems-level, yet detailed, description is known [31]. Substantial evidence now exists that the architecture of these networks is highly complex, consisting in large part of enzymes that act as molecular switches to activate and inhibit downstream substrates via post-translational modification. These substrates are often themselves enzymes, acting in similar fashion.

In experiments, we are able to genetically inhibit or over-express the levels of activators, inhibitors and the substrates themselves, but rarely are able to directly observe the levels of active substrate in cells. Without the ability to directly observe the biochemical repercussions of inhibiting an enzyme in real-time, determining the true in vivo targets of these enzymes requires indirect observation of genetic perturbation and inference of enzyme-substrate relationships. For example, it is possible to observe downstream transcription levels which are affected in an unknown way by the level of active substrate [40].

The specific problem we address is the reconstruction of cellular signaling networks studied by perturbing components of the network, and reading the results via microarrays. We take a model-based approach to the problem of reconstructing network topology. For every pair of proteins in the network, we predict the most likely strength of interaction based on the data, and from this predict the topology of the network. This is computationally feasible as we are considering a subset of proteins for which we know the general network motif.

We demonstrate the efficacy of this approach by inferring from experiments the Rho-signaling network in *Drosophila*, in which some 40 enzymes activate and inhibit a set of approximately seven substrates. This network plays a critical role in cell adhesion and motility, and disruptions in the orthologous network in humans have been implicated in a number of different forms of cancer [68]. This structure, with many enzymes and few substrates (Fig. 2-1), is a common motif in signaling networks [2, 21].

To complicate the inference of the Rho-signaling network further, not every enzyme-substrate interaction predicted *in vitro* is reflected *in vivo* [63]. As such, we need more subtle information than is provided by current high-throughput protein-protein interaction techniques such as yeast two-hybrid screening [27, 35].

To probe this network, we have carried out and analyzed a series of knockout and overexpression experiments in the *Drosophila* S2R+ cell line. We measure the regulatory effects of these changes using DNA microarrays. It is important to note that microarrays measure the relative abundance of the gene transcript, which can be

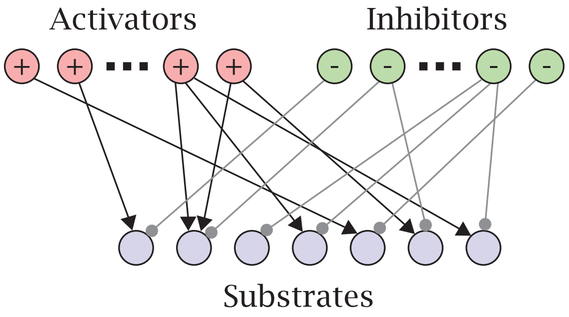


Figure 2-1: The many enzyme-few substrate motif. A triangular arrowhead represents activation, a circular arrowhead inhibition.

used as a rough proxy for the total concentration of gene product. What they do not elucidate, however, is the relative fraction of an enzyme in an active or inactive state, which is crucial to the behavior of signaling networks. To reconstruct the network from measurement, rather than directly use the microarray features corresponding to the proteins of interest, we instead use correlations in observations of the affected downstream gene products.

We take the novel approach of constructing and optimizing a detailed parameterized model, based on the biochemistry of the network we aim to reconstruct. For the first part of the network model, namely the connections of the enzymes to substrates, we know the specific rate equations for substrate activation and inhibition. By modeling the individual interactions in like manner to the well-established Michaelis-Mentin rate kinetics [62, 14, 65], we are able to construct a model of the effects of knockout experiments on the level of active substrate. Lacking prior information, we model the effect of the level of active substrate on the microarray data by a linear function. If the only source of error were uncorrelated Gaussian noise in the measurements, we could then simply fit the parameters of this model to the data to obtain a best guess at the model's topology.

However, noise and "batch effects" [50] in microarray data are a real-world complication for most inference methods, which we address in a novel way. Noise in microarrays is seemingly paradoxical. On one hand, identical samples plated onto two different microarrays will yield almost identical results [5, 51]. On the other hand, with many microarray data sets, when one simply clusters experiments by similarity of features, the strongest predictor of the results is to group by the day on which the experiment was performed. We hypothesize, in this analysis, that the batch effects in microarrays are in fact other cellular processes in the sample unrelated to the experimental state. Properly filtering the ever-present batch effects in microarray data requires more than simply considering them to be background noise. Specifically, instead of the standard approach of fitting the data to our signal and assuming noise cancels, we consider the data to be a combination of the signal we are interested in and a second, structured signal of the batch effects.

Fitting this many-parameter model with physical constraints to the actual data optimizes our prediction for the signaling network, with remarkably good results.

To test this method we have constructed random networks with structure similar to the expected biology, and used these to generate data in simulated experiments. We find that when compared to reconstructions based on other methods, we were able to obtain significantly more accurate network reconstructions. That is to say, at every specificity we obtained better sensitivity and vice-versa. The details of these other methods can be found in Sec. 5.1.

We have also reconstructed the Rho-signaling network in *Drosophila* S2R+ cells from a series of RNAi and overexpression experiments we performed. We attempted to verify our predictions with a series of chemilluminescent western blots – while the data is still preliminary, it is reasonably consistent with our predictions.

Chapter 3

Background on Signaling Network Inference

In this chapter we establish some necessary background on signaling networks. We also briefly discuss the two experimental technologies critical to the experiments, RNAi and cDNA microarrays. We conclude the chapter with a discussion of previous computational work on network inference.

3.1 Biological

3.1.1 Signaling Networks

Many proteins have an active and inactive state. Of these, a large number activate and deactivatie other proteins, often by attaching or detaching small molecules (e.g. phosphor groups) causing a change in conformation of the protein which reveals or occludes the active site. These relatively fast-acting interactions form a network that is used by the cell for most behaviors that require fast responses. In this thesis we examine the Rho signaling network, which is integral to cytoskeletal regulation, and plays a central role in cell motility.

3.1.2 RNAi

RNAi allows the *in vivo* silencing of genes in many higher organisms [61] . This is achieved by inserting a section of double-stranded RNA (dsRNA) that matches a section of a gene. By the action of a set of proteins whose actions are still not fully understood, the cell silences all transcription of that region of the DNA.

This machinery is thought to be a type of cellular immune system [36] against RNA viruses, but we can also exploit it for experimental purposes.

3.1.3 Microarrays

cDNA microarrays measure the level of a given set of RNA sequences present in the cell. The specific type of array we use is a CombiMatrix 4x2k CustomArray. Custom array technology is notable as the end-user can choose the probe sequences which are then printed onto an array by use of a modified CMOS array to electrochemically guide synthesis. Other forms of custom microarrays are achieved by ink-jet printing the nucleotides directly onto the chip. Measurements are obtained by first extracting the RNA from a population of cells, cutting it into fragments, plating it to the array where it binds with complementary sequences, fluorescent dying the bound fragments, and imaging them with a scanner.

With the custom microarray technology, we were able to choose a set of 2072 sequences of lengths between 25 and 35 corresponding to a selection of genes throughout the cell. As the microarray probe densities are not calibrated, they are not an effective measure of the relative abundances of transcripts. However, as the manufacturer claims the probe density is nearly consistent from array to array, they can be used to measure differential expression under differing experimental conditions.

Table 3.1: The strength of batch effects as measured by the mean Pearson correlation coefficient between the same or differing experiments performed on the same or differing days. The number of pairs of experiments represented is shown in parentheses.

Mean correlation (Number)	Same Experiment	Different Experiment
Same Day	0.971 (56)	0.955 (642)
Different Day	0.835 (1024)	0.840 (14154)

Noise

Noise in microarrays is seemingly paradoxical. A number of factors can confound measurements, most notably the nonlinear response curve of microarray florescence with respect to sample density. Another non-negligable source of error is intensity biases introduced by the processes of hybridization or scanning.

In general, identical samples plated onto two different microarrays side-by-side will yield almost identical results [5, 51]. However, identical samples plated on two different days produce results that diverge significantly [52]. This was notably present in our results (Table 3.1.3).

The sources of these so-called ”batch effects” are not well-understood, though it has been recently discovered that the dyes used are ozone-reactive [26, 13], and so differing ozone concentrations in the laboratory on different days will yield different results.

These batch effects are thus properly considered not to be noise in the classical sense of independent perturbation of measurements, but rather an auxiliary signal to that of the experiments.

3.2 Previous Work

A number of related techniques for inferring global patterns based on high-throughput data exist. Many of these utilize the technique of probabilistic graphical models [32, 67, 33, 66, 53]. While these techniques are effective for inferring networks in broad statistical strokes, we increase the resolution and model the rate coefficients of individual reactions. The mathematics of our methodology is in fact isomorphic to a probabilistic graphical model approach; however as our parameters correspond directly to physical quantities or coefficients, we are able to dramatically narrow our model space when compared to a more general technique such as Bayesian or Markov networks [32]. In doing so we are able to gain both greater sensitivity, specificity, and robustness to noise. A related technique, based on modeling of rate kinetics in the framework of Dynamic Bayesian Networks has been effective in modeling genetic regulatory networks [65]. Techniques from information theory, such as ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) [58, 7] and nonparameteric statistics, such as GSEA (Gene Set Enrichment Analysis) [75] have also been used to infer connections in high-throughput experiments. While not generally used for signaling network reconstruction, GSEA notably has been popular recently [50, 8], in part for its efficacy in overcoming batch effect noise.

Chapter 4

Our Approach

In order to infer the Drosophila Rho-signaling network in a more efficient manner than traditional biochemistry, we performed a number of knockout experiments, whose effect we measured with microarrays. As this is an entirely novel method for inference on new data, new mathematical techniques were required as well.

4.1 Experimental Approach

Chris Bakal, in the Perrimon Lab at Harvard Medical School performed 144 experiments, systematically knocking out (via RNAi) and over-expressing components of the Rho-signaling network in *D. melanogaster*. These were plated on to custom single-channel microarrays manufactured by Combimatrix.

For details of the experiments performed and Microarray design, see Appendix A. By the publication of this thesis, all data will have been uploaded to GEO [6].

4.2 Modeling Challenges

As experiments of this sort had not previously been performed, a number of new mathematical and computational techniques were required to infer a connections from them. The first, most obvious, difficulty is that the quantities we measure, the level of mRNA transcripts, and the quantities we care about, the amount of active-form GTPase do not have a direct or clearly known relationship. Beyond this, the measurements themselves are obscured by the high level of noise inherent to the microarray technology. Further, even with perfect measurements, the system is obscured by feedback, namely if we knock out one important protein, the cell is likely to respond by changing the amount of other proteins with redundant and overlapping function.

4.2.1 Indirect Observation

To understand the signal in our data, we construct a parameterized model of the Rho-signaling network in the hopes that knowing the expected shape of what we are looking for will help us find it.

We first illustrate our approach for a single activator-inhibitor-substrate trio before extending to the many-node case. We start by deriving the time dependence of the concentration[1] ρ of active substrate in terms of the concentrations $\bar{\rho}$ of inactive substrate, η of activator, α of inhibitor, and the base rates $\bar{\gamma}$ of activation and γ of de-activation. Fig. 4-1 depicts these kinetics. As the rate at which inactive substrate becomes active is proportional to its concentration times the rate of activation and vice-versa,

$$\frac{d\rho}{dt} = -\frac{d\bar{\rho}}{dt} = \bar{\rho}\left(\bar{\gamma} + \eta\right) - \rho\left(\gamma + \alpha\right). \tag{4.1}$$

[1] Choice of units of concentration is absorbed by scalar factors of the fit once the x_{jk} and y_{jk} coefficients are added; see Eq. 4.3.

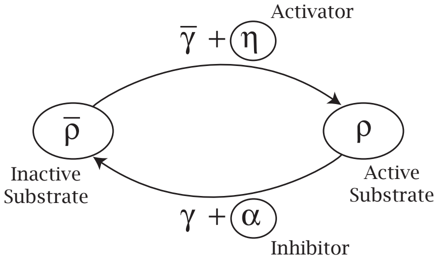


Figure 4-1: The dynamics of an activator-inhibitor-substrate trio. The circled variables are proportional to protein concentrations.

We are primarily interested in ρ, the level of active substrate, as the downstream effects of the substrate are dependent on this concentration. As the measurements are taken several days after perturbation and are an average over the expression levels of many individual cells, by ergodicity we expect to find approximately the equilibrium ($d\rho/dt = 0$) concentration of substrate.

Solving for ρ at equilibrium yields:

$$\rho = \frac{\kappa\,(\bar{\gamma} + \eta)}{\bar{\gamma} + \eta + \gamma + \alpha}. \tag{4.2}$$

where $\kappa = \rho + \bar{\rho}$ is total concentration of the substrate, approximately available from the microarray data. By choice of time units we can let $\bar{\gamma} = 1$. This result, by no coincidence, is similar to the familiar Michaelis-Mentin rate kinetics.

We now generalize the model to multiple substrates κ_k, interchangeable activators η_j with relative strength x_{kj}, and inhibitors α_j with relative strength y_{kj}. The equilibrium concentration of the level of active substrate ρ_k then becomes:

$$\rho_k = \frac{\kappa_k \left(1 + \sum_j x_{kj}\eta_j\right)}{1 + \sum_j x_{kj}\eta_j + \gamma_k + \sum_j y_{kj}\alpha_j}. \tag{4.3}$$

Lacking more detailed biological information, and aiming to avoid the introduction of unnecessary parameters, we assume a linear response from features in the microarray. Specifically, for a vector of microarray feature data $\vec{\varphi}$, we model the effect as a general linear function of the levels of active substrate, of the form $\vec{a}\vec{\rho} + \vec{r}$. Additionally we introduce a superscripted index z for those variables which vary by experiment. The level, φ_i^z, of the i^{th} feature in microarray z is in our model:

$$\varphi_i^z = \sum_k a_{ik} \left(\frac{\kappa_k^z \left(1 + \sum_j x_{kj} \eta_j^z\right)}{1 + \sum_j x_{kj} \eta_j^z + \gamma_k + \sum_j y_{kj} \alpha_j^z} \right) + r_i + \beta_i^z + \epsilon_i^z, \tag{4.4}$$

where the batch effects $\vec{\beta}$ and noise $\vec{\epsilon}$ are considered additively.

4.2.2 Noise

While much of the inconsistency intrinsic to microarray technology is dealt with by proper pre-processing of the data (see Section 6.1), there remains the problem of batch effects. As batch effects in microarrays are highly correlated, our approach is to construct a linear model of their structure. Empirically, batch effects tend to have a small number, s, of significant singular values (from empirical data $s \simeq 4$). In the singular vector basis, we can model the batch effects as a (features × s) matrix $\vec{c}$. To determine the background as a function of experiment batch, we rotate by an ($s \times$ batches) rotation matrix $\vec{u}$. Thus $\vec{c}\vec{u} = \sum_j c_{ij} u_{jd}$ is a (features × batches) matrix whose columns are the background signal by batch. Finally to extract the batch effect for a given experiment z, we multiply by the characteristic function of experiments by batches, $\vec{\chi}$, where $\chi_d^z = 1$ if experiment z happened in batch d and is 0 otherwise. Our model of batch effects is then:

$$\beta_i = \sum_{l,d} c_{il} u_{ld} \chi_d^z. \tag{4.5}$$

All together, our detailed model for experimental data based on the network, experiments, and noise becomes:

$$\varphi_i^z = \sum_k a_{ik} \left(\frac{\kappa_k^z \left(1 + \sum_j x_{kj}\eta_j^z\right)}{1 + \sum_j x_{kj}\eta_j^z + \gamma_k + \sum_j y_{kj}\alpha_j^z} \right) + r_i + \sum_{l,d} c_{il} u_{ld} \chi_d^z + \epsilon_i. \tag{4.6}$$

4.2.3 Feedback

Transcriptional feedback, i.e. changes in GAP/GEF/GTPase expression levels in response to experimental perturbation, can confound inference techniques. To take this into account, we set the expression level variables in the network model to reflect the observed expression levels for those proteins. Specifically, the parameters η_j^z, α_j^z, and κ_j^z were set to be the multiplicative difference over the mean intensity averaged over all features for that gene, (e.g. η_j^z is set to 0.5 when the observed mRNA level of GEF k in experiment z is half its mean expression). The corresponding parameters were set to 0 in experiments where RNAi experiments were performed.

While the inclusion of feedback clearly improves the model fit to data, the effect of its inclusion of result quality is unclear.

4.3 Model Parameter Fitting

Having now constructed a model of our system, we minimize the least-squares difference between the model predictions and observed data (detailed in Sec. 5.2), to obtain optimal model parameters. The resultant values of $\vec{x}$ and $\vec{y}$ predict the relative strengths of the activator-substrate interactions.

It is important to keep in mind which parameters are known and which we must fit. We know s and $\vec{\chi}$ from experiment. In lieu of detailed knowledge of the activity levels of the activator and inhibitor, we take κ_k^z, η_j^z and α_j^z to be 1 normally, 0 on those experiments for which the gene is silenced, and 2 for those in which it is overexpressed. The remaining fitting parameters of our model are $\vec{x}, \vec{y}, \vec{a}, \vec{\gamma}, \vec{r}, \vec{c}$, and $\vec{u}$.

For a vector of experimental data $\vec{d}$, we construct, as above, a model for the predicted data $\vec{\varphi}$. Fitting the model to data is done by minimizing:

$$f(\vec{x}, \vec{y}, \vec{a}, \vec{\gamma}, \vec{r}, \vec{c}, \vec{u}) = \sum_{i,z} \left(d_i^z - \varphi_i^z\right)^2 , \tag{4.7}$$

where φ_i^z is given in Eq. 4.6, subject to the constraints

$$x_{kj}, y_{kj}, \delta_k, \kappa_k \quad \geq \quad 0 \tag{4.8}$$

and the additional constraint that $\vec{u}$ is a rotation matrix. The fit with lowest objective value is the maximum likelihood predictor of the network.

To verify that we have more data than parameters, we consider a microarray with Φ features and a network model with a total of θ activators and inhibitors and σ substrates. Additionally we consider a 4-dimensional noise model for λ batches. Then for ζ experiments, we have more data than parameters precisely when:

$$\zeta > \sigma + 4 + \frac{(\theta + 3)\sigma + 4\lambda - 10}{\Phi} \tag{4.9}$$

In a realistic setting, for 26 enzymes, six substrates, with on average six experiments per batch, and assuming each experiment has at least 50 features, then we need to perform at least 14 experiments in order to have more data than parameters. As the batch effect model has substantially lower rank than the number of batches, as long as there are at least five batches, over-fitting is unlikely.

Table 4.1: Full model of signaling network experiments

6804 Independent Variables

η_j^z	(14 × 126)	GEF Perturbations
α_j^z	(13×126)	GAP Perturbations
γ_j^z	(6 × 126)	GTPase Perturbations
di_d^z	(21 × 126)	Day Indicator

261072 Independent Variables

d_j^z	(2072 × 126)	Microarray Data

23050 Parameters

$x_{k,j}$	(6 × 14)	GTPase-GEF affinities
$y_{k,j}$	(6×13)	GTPase-GAP affinities
δ_k	(6)	Base deactivation rate
EL_k	(6)	GTPase base expression
$a_{i,k}$	(2072 × 6)	GTPase-output coefficients
r_j	(2072)	Base spot levels
$c_{i,l}$	(2072 × 4)	Batch coefficients
$u_{l,d}$	(4 × 21)	Batch rotation

4.3.1 Final Model

The resultant model has 261072 observations[2] (dependent variables), 6804 experimental parameters (independent variables), and 23050 model parameters (see Table 4.3.1). While largely linear or quadratic, the nonlinearity in the activator-inhibitor-substrate trio makes the eventual model nonconvex. Thus, unfortunately, direct optimization of the model parameters is impractical, even with modern software.

[2] Of which 13155 are missing due to a mid-experiment change in array design.

Table 4.2: Reduced model of signaling network experiments

4410 Independent Variables

d_j^z	(35 × 126)	Microarray Data

643 Parameters

$x_{k,j}$	(6 × 14)	GTPase-GEF affinities
$y_{k,j}$	(6×13)	GTPase-GAP affinities
δ_k	(6)	Base deactivation rate
EL_k	(6)	GTPase base expression
$a_{i,k}$	(35 × 6)	GTPase-output coefficients
r_j	(35)	Base spot levels
$c_{i,l}$	(35 × 4)	Batch coefficients
$u_{l,d}$	(4 × 21)	Batch rotation

4.3.2 Model reduction

To make the model tractable, we must reduce it to one with fewer variables that captures the overwhelming majority of the variation in the data that the full model predicts. We use two facts to do this. First, the last step of every component of the model a, u and r are linear. Thus the model will equivalently fit any rotation of the data matrix d, i.e. minimizing $||d - \phi||_2$ and $||Ud - \phi||_2$ are equivalent. Second, the model never explicitly makes use of the fact that the 2072 observations correspond to known biological components. Thus, if the majority of the variation is on a small number of dimensions, fitting the model to those dimensions alone will capture the majority of the variation in the data as well fitting it to the entire dataset. Empirically, the noise introduced from on-chip errors (measured by computing the variance of identical features on the same chip) is greater than the error in a 35-dimensional approximation of the data. We thus fit the model to the top 35 principle components of the data matrix.

4.3.3 Optimization

Even with the reduction, finding the global minimum for model error (the maximum likelihood predictor) is not feasible, nor is it clearly desirable. To find an optimum, we use a local solver, starting at many randomly chosen points in the parameter space. Specifically, we used the commercial solver SNOPT [34], which uses sequential quadratic programming to do local optimization.

Other optimization methods and their failings are discussed in Section 6.3.3.

4.3.4 Predicted Connections

We used a consensus of local fits to predict connections, as the different local minima found were all within noise of one another's fit qualities. Specifically, as the models fits all had approximately the same residual (within 0.75%), there was no *a priori* way to choose a best fit. Local fits were started from a large number of starting points, each with a subset of the possible connections strongly present, in order to get effective sampling of the space. Of the resulting fits, those which were identical or who a strictly better linear combination were merged.

While the majority of predicted affinity parameters (x and y) were fit to be exactly 0, those which were not were taken to be connections. The fraction of parameter fits which show a connection is treated as the confidence in the predicted edge.

4.3.5 Forced Connections

A major advantage of using a parameterized model is the ability to to add further previous knowledge about the system to reduce the space of possible model fits. To this end, we created a literature-compiled set of connections which are believed to not be present in the Drosophila Rho-signaling network, and forced the corresponding parameters in the model to be zero. We then re-fit the model by the same procedure as above with this further constraint.

Chapter 5

Results

To test our model, we first tested the model on simulated data. Then we tested the predictive power of the model on real data. Finally we evaluated our predictions against a set of connections inferred from literature.

The first two sections describe work done on the first 80 experiments, the available data for [9].

5.1 Simulated Data

We have generated simulated data on randomly created networks. The density of activator-substrate and inhibitor-substrate connections was chosen to reflect what is expected in the Rho-signaling network described in Section 5.2. From this, we have generated model experiment sets consisting of one knockout twice of each of the substrates and a single knockout of each activator and inhibitor in batches in random order. To further mimic our biological data set we included at least one baseline experiment in each batch. From this model we simulated experimental data with both noise and a batch-effect signal and attempted to fit the generated data.

To test against other techniques, we applied the statistics used by GSEA and ARACNE (see Section 6.4), modified for use on our model data sets. While GSEA is not typically used for signaling network reconstruct, its general usefulness in microarray analysis necessitates the comparison. ARACNE, on the other hand, while designed for a similar situation, does not directly apply, and so needs to be modified to make a direct comparison. As a baseline, we also computed the naïve (Pearson) correlation of experimental states.

On noiseless data, with only a minimal set of experiments and batch effects of comparable size to the perturbation signal, we are able to achieve a perfect network reconstruction which was not achieved by any of the other methods we consider. On highly noisy data, we cannot reconstruct the network perfectly; however we consistently outperform the other methods in both specificity and sensitivity (Figure 5-1). Moreover, we find that while the model alone out-performs other techniques (comparably to AMI), the batch effect fit is of crucial importance. While this is clearly a biased result, as the simulated data is generated by the same model we assume in the fit, it does show that we are able to obtain a partial reconstruction even under high noise conditions. As this is a best-guess model from prior biological knowledge, the assumptions are far from unreasonable.

5.2 Predictive Power on Real Data

We used our method, discussed above, on forthcoming microarray data collected from RNAi and overexpression experiments to predict the structure of the Rho-signaling network in *Drosophila* S2R+ cells. This network consists of approximately 47 proteins, divided roughly as 7 GTPases, 20 Guanine Nucleotide Exchange Factors (GEFs) and 20 GTPase Activating Proteins (GAPs). Importantly, we have the additional information that, despite their misleading names, the GEFs serve to activate certain

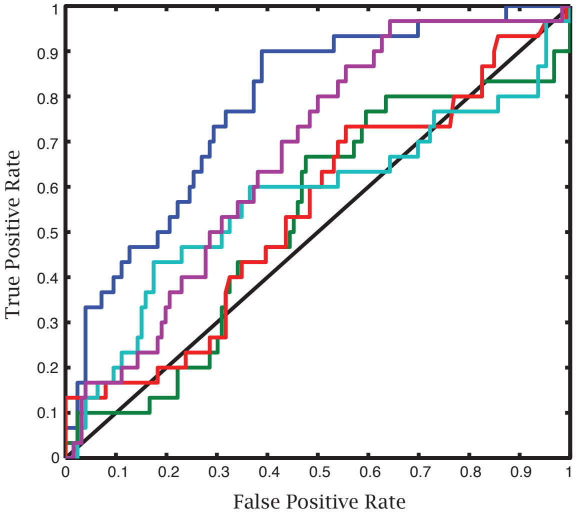


Figure 5-1: Typical ROC curve for highly noisy simulated data. Our model (dark blue) is closest to the actual network, which would be a point at $[0, 1]$. Model fitting without batch effects (purple) is also considered. The other lines represent the predictions obtained by a GSEA-derived metric (red), an ARACNE-derived metric (light blue), and naïve correlation (green). The diagonal black line is the expected performance of random guessing. This particular set of simulated data has no repeat experiments for GAPs or GEFs, a batch signal of half the intensity of the perturbations, and an approximate total signal-to-noise ratio of 1.5.

GTPases and the GAPs serve to inhibit them. The exact connections, however, are for the vast majority, unknown.

Labeled aRNA, transcribed from cDNA, was prepared from S2R+ Drosophila cells following five days incubation with dsRNA or post-transfection of overexpression constructs. The aRNA was then hybridized to CombiMatrix 4x2k CustomArrays designed to include those genes most likely to yield a regulatory effect from a perturbation to the Rho-signaling network. After standard spatial and consensus Lowess [17] normalization, we k-means clustered [57] the data into 50 pseudo-features to capture only the large-scale variation in the data.[1]

After fitting, we have computed the significance of our fit using the Akaike and Bayesian Information Criteria (AIC and BIC) [1, 70]. These measure parameter fit quality as a function of the number of parameters, with smaller numbers being better. AIC tends to under-penalize free parameters while BIC tends to over-penalize, thus we computed both. As a baseline, we computed the AIC/BIC of the null model. While a direct fit of the pseudo-features yielded a lower AIC but not BIC, an iterative re-fit and solve technique, not unlike EM, produced a significant fit by both criteria (Table 5.1, prediction in Table 5.3). This re-fitting was done by greedily resorting the groupings for meta-features based on the model fitness and refitting the model to the new meta-features. As each step strictly increases fit quality, and there are only finitely many sets of meta-features, this is naïvely guaranteed to converge in $O(n^k)$ iterations for n features and k meta-features. We find, however that the convergences generally to happens in around 5 iterations, leaving feature variance intact (an indication that this is not converging to a degenerate solution).[2]

To further test the accuracy of our model, we fit the model to four subsets of

[1] The fact there are fewer than 50 significant singular values in the data and the linearity of $a,\vec{r}$ and $\vec{\beta}$, indicates that we can not get more information from more clusters.

[2] This procedure has been replaced in current work by PCA.

Table 5.1: AIC/BIC of the null model, best naïve fit, and best fit.

Model	*Fit (f)*	*AIC*	*BIC*
Null Model ($\varphi_i^z = 0$)	0.9885	-8.389	-8.387
Best Fit	0.2342	-9.480	-8.366
Adapted Features	0.0328	-11.446	-10.332

Table 5.2: Prediction error on test data.

Test Set	*Size*	*#Unduplicated*	*Total Fit (f)*	*Test Set Fit*	*Error*
1	9	4	0.0280	0.1307	14.6%
2	17	4	0.0288	0.0632	6.10%
3c	9	0	0.0302	0.0371	3.13%
4c	9	0	0.0301	0.0517	4.06%

the 87 experiments and tested the prediction quality on the remaining experiments. The prediction error is calculated as the mean squared error of the predicted values divided by the mean standard deviation by feature. We tested on four sets: Sets 1 and 2 were chosen randomly to have nine (10.3% of experiments) and seventeen (19.5% of experiments) elements respectively, of which four of each are unduplicated experiments. Sets 3c and 4c were chosen randomly to have nine elements but were constrained not to have two elements from the same batch or experimental condition. We find that the model accurately predicts test set data (Table 5.2) for repeated experiments. Note that in Set 1, when 44% of the experiments in the test set are non-duplicated, the prediction error is significantly higher. This indicates the necessity of both the batch and network components of the model.

5.3 Model Fit on Current Data

5.3.1 Dimensionality of the reduction

Figure 5-2 shows a single model optimization for differently sized reductions of the data. While the model residual does not improve substantially beyond ten dimensions, it does show slight improvement for higher dimensions. While the ultimate model is restricted to 10 linear dimensions, these are not necessarily exactly the dimensions of largest variation in the data.

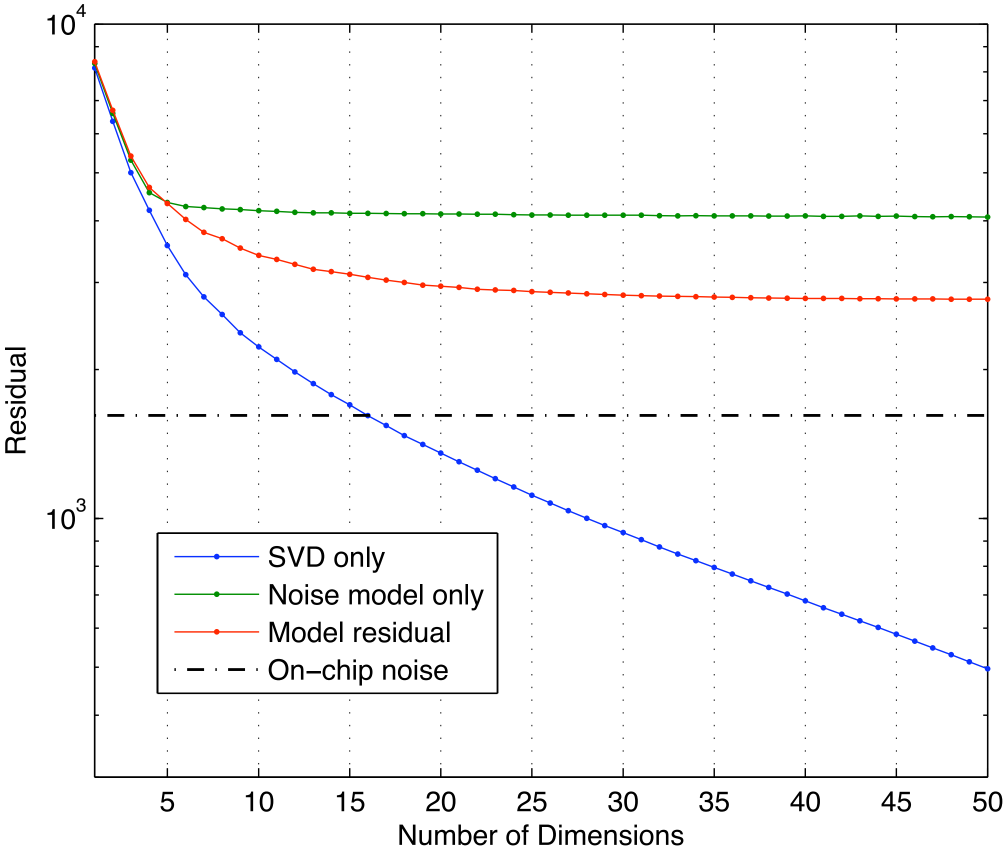


Figure 5-2: One run of the model fit at each reduction size.

5.3.2 Feedback Improves Model Fit

A lower objective value was consistently achieved with the addition of regulatory feedback. This is notable as it required no additional model parameters, and is a strong indication that the model is correctly capturing at least some aspect of the network. As seen in Figure 5-3, the chance of this happening with random feedback data is exceedingly small.

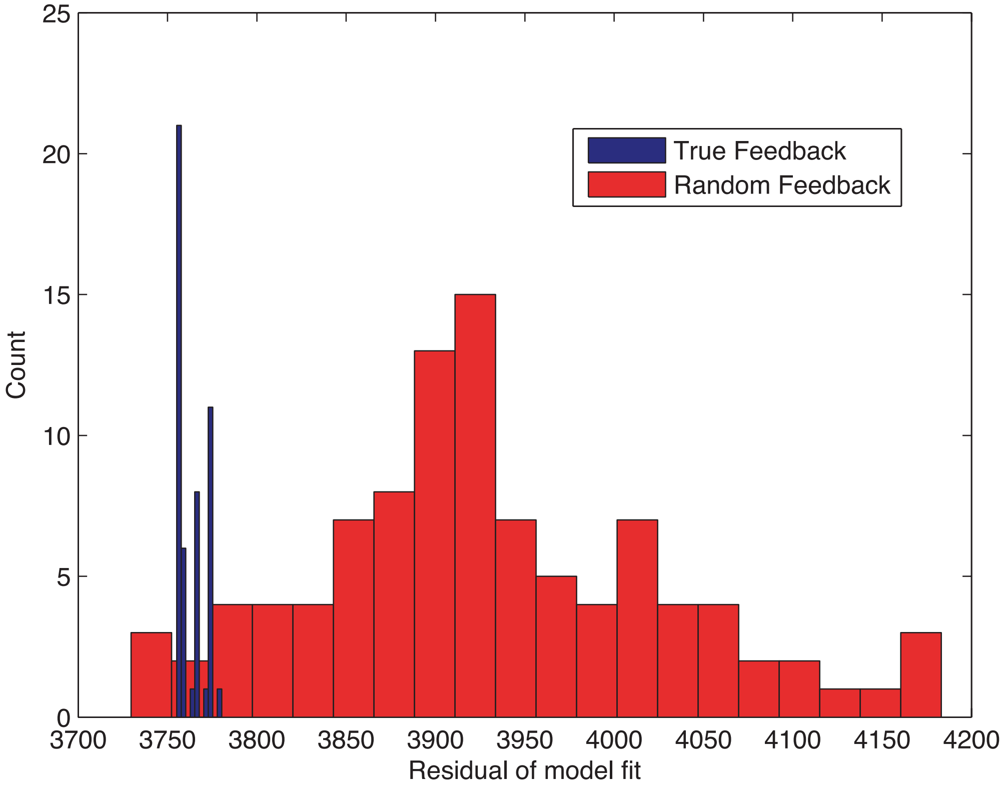


Figure 5-3: Residual of model fits for randomly permuted feedback vs true feedback. 4 of the 100 random feedback fits have a better residual than the mean of the true fits.

5.4 Predicted Connections

The final model fit predicted 66 connections, shown in Table 5.3.

5.4.1 Biochemical Validation

At the time of this writing, non-repeated validation of only 36 of the predicted connections was performed. These were performed biochemically by attaching dyes that preferentially bind to either the active or inactive forms of human Rac1 and Cdc42 and measuring their relative abundances using a quantified Western blot, in the hope that the human versions of these antibodies are similarly specific in *Drosophila*. As Western blot data is notoriously noisy, their implications about the model's predictive capacity should be treated as comparably unreliable, at least until repeat experiments are performed.

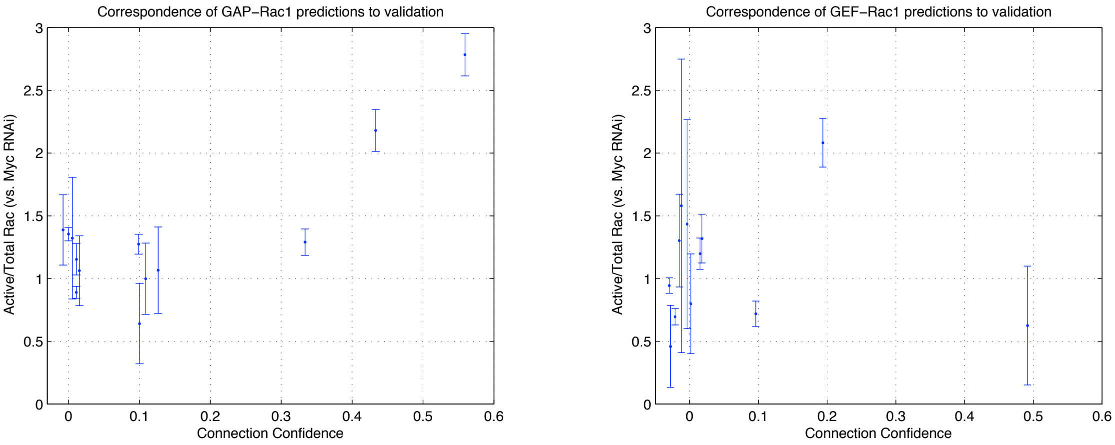
The level of active Rac1 was measured after knockout of nearly all of the GEFs and GAPs. Cdc42 was measured only for GAP knockouts.

Rac1

Our predictions lined up remarkably well with the observation of GAP-response levels in Rac1. For GEF responses, however, our predictions were no better than chance. (Figure 5-4) This is consistent with the condition in which Rac1 is normally primarily in the inactive state, i.e. removal of activators would have no effect and so would be detected by neither microarray nor Western blot, while the removal of repressors would have readily observable effects. While consistent, our data does not prove this, and independent experiments should be performed to verify this hypothesis.

Table 5.3: The predicted network. Each entry is the fraction of the nine local minima containing that connection, here taken to be a measure of the strength of prediction.

Type	Name	Rac1	Rac2	Rho1	Cdc42	RhoL	MTL
GEFs	Cdep	0	0	0	0.88889	0	0
	sif	0	0	0	0	1	0
	pbl	0.11111	0.66667	0	0	1	0
	trio	0	0	0	0	0	0
	CG3799	0.44444	0.77778	0.22222	0.22222	0.44444	0.11111
	CG10188	0	0	0	0	0	1
	CG14045	0	0.22222	0	0	0	1
	CG15611	0	0	0	0.77778	0	1
	CG30115	0	0.11111	0	0.11111	0	0
	CG30456	0	0	0	0	0	0
	RhoGEF3	0	0	1	0	0	0
	RhoGEF4	0	0	0	0	0	0
	RhoGEF64C	0.22222	0.66667	0.11111	1	1	1
	RhoGEF2	0	0	0.55556	0	0.44444	0
GAPs	p190RhoGAP	0.11111	0.33333	0	0	0	0
	RhoGAP1A	0.44444	0.11111	0.22222	0.22222	1	0
	RhoGAP5A	0	0	0	0	0	0
	RhoGAP16F	0.11111	0.55556	0	0	1	0
	RhoGAP19D	0	0	1	0.66667	0	1
	RhoGAP50C	0.55556	0.33333	0.66667	0	0	1
	RhoGAP54D	0.11111	0.22222	0	0.77778	0	0
	RhoGAP71C	0	0.11111	0	0.11111	0	0.33333
	RhoGAP84C	0.11111	0	1	1	1	1
	RhoGAP92B	0.33333	0.66667	0.11111	0	0.44444	1
	RhoGAP93B	0	0.44444	0	0	1	0
	RhoGAP100F	0	0	0.11111	0	1	0
	CdGAPr	0	0.33333	0	0.66667	0	0



(a) (b)

Figure 5-4: Rac1 predicted connections vs. validation data. The points have been jittered slightly horizontally to be distinguishable, however the two clusters in each figure with confidence around 0 or 0.1 are in fact all identical. (a) GAPs, correlation: 0.77, higher is better. (b) GEFs, correlation: -0.04, lower is better.

Cdc42

Our predicted GAP-Cdc42 connections show at best minimal correspondence with validation data. This could be genuine, or it could also be due to the fact that we used human Cdc42 antibodies to probe for Drosophila Cdc42. (Figure 5-5)

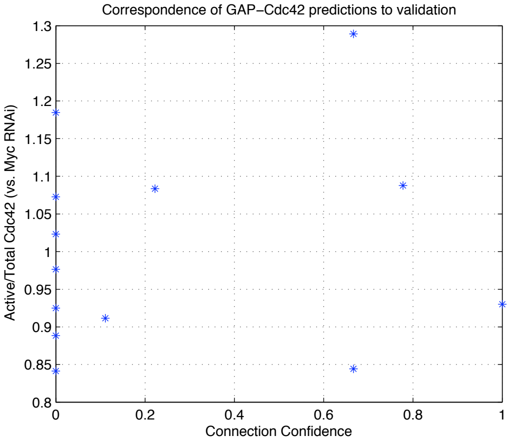


Figure 5-5: Cdc42-GAP predicted connections vs. validation data. Correlation: 0.135, higher is better.

Chapter 6

Implementation

The purpose of this chapter is to walk a reader through the implementation of every step, from the data outputted by the microarray scanner to evaluation of predicted networks. Select sections of code, primarily written in MATLAB, are included in Appendix B where noted.

6.1 Preprocessing

Experimental data arrives as a .tiff image of the microarray. (Figure 6-1.) The chemical concentrations of mRNA binding to each probe are reflected in the recorded brightnesses of the corresponding spots on the scan of the microarray. To convert this to useable data, the following steps are required:

1. Extract median intensities from the images, i.e. convert the image to a list of numbers.

2. Remove spatial artifacts from the arrays. In this case we used an iterative surface-fit method to remove spatial structure.

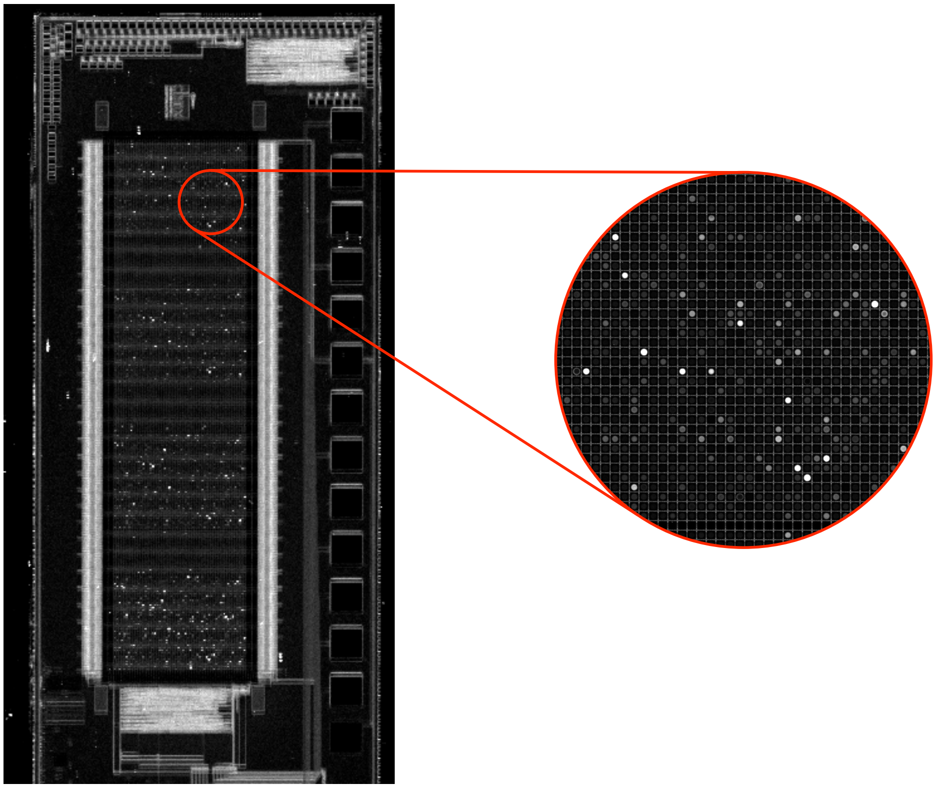


Figure 6-1: Raw Microarray Data. Each circle on the inset is a single probe.

3. Integrate different array designs. In the case of these experiments, the array design was changed after 11 experiments, yielding 44 arrays of the old design and 92 of the new design.

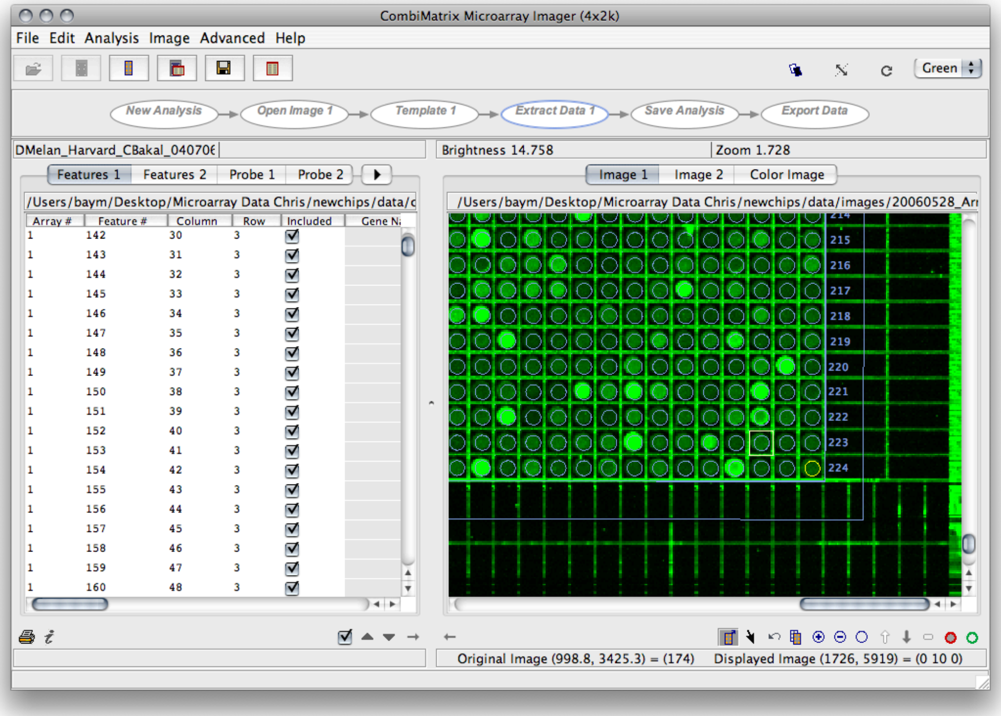
4. Normalize the values between arrays. In this case we used a consensus Loess method to produce commensurate data.

5. Removal of spurious data. Several arrays in this data set either did not hybridize properly or produced data which was clearly spurious. These were removed.

From this one obtains data amenable to mathematical analysis. With the exception of Spotting, all of this was done in a combination of PERL and MATLAB.

6.1.1 Spotting

To extract the spot intensities from the raw data, we used the Microarray Imager software (Figure 6-2) provided by CombiMatrix with the arrays. This has the advantage that it interacts natively with their XML description of the array, and so produces easily-accessed CSV files with all of the relevant information (probe sequence, label, species, mean and median intensity, etc.). The other major advantage is that the chip shape (number, size and spacing of probes) is pre-programmed, so one only needs to specify alignment and scaling.



DMelan_Harvard_CBakal_04070€

Brightness 14.758

Zoom 1.728

/Users/baym/Desktop/Microarray Data Chris/newchips/data/c

/Users/baym/Desktop/Microarray Data Chris/newchips/data/images/20060528_Ar

Array #	Feature #	Column	Row	Included	Gene Na
1	142	30	3	☑	
1	143	31	3	☑	
1	144	32	3	☑	
1	145	33	3	☑	
1	146	34	3	☑	
1	147	35	3	☑	
1	148	36	3	☑	
1	149	37	3	☑	
1	150	38	3	☑	
1	151	39	3	☑	
1	152	40	3	☑	
1	153	41	3	☑	
1	154	42	3	☑	
1	155	43	3	☑	
1	156	44	3	☑	
1	157	45	3	☑	
1	158	46	3	☑	
1	159	47	3	☑	
1	160	48	3	☑	

Original Image (998.8, 3425.3) = (174) Displayed Image (1726, 5919) = (0 10 0)

Figure 6-2: CombiMatrix's Microarray Imager software in action.

While the software nominally comes with a routine for automatically finding the array, in practice this is only occasionally effective. In practice, the most effective technique is to attempt an auto-align, then use the GUI to align the upper-left and lower-right spots on the chip. While (likely due to distortions in the scanner feed), some spots are occasionally not precisely aligned, in practice this yields almost every spot being almost entirely within the recording area.

The software at this point measures the intensities of every spot within the target circle and records the mean, median, and variance of the recorded intensities. While in practice the mean and median intensities vary only slightly, the median was chosen so as to avoid data contamination from artifacts and slightly misaligned spotting.

6.1.2 Spatial Normalization

The logic behind removal of spatial artifacts is that modulo outliers, spot intensity should not vary substantially based on location on the microarray. I.e., the spots in one area of a microarray should have the same mean intensity as those in another. More specifically, as the spots were distributed randomly across the array, any spatial structure that we do see in the array is likely due to experimental error and not a signal in the data.

Put more precisely, a smooth surface fit to the data as a function of its x- and y- coordinates should be entirely flat. To ascertain the background intensity surface, we used the `gridfit.m` tool [22], which fits a surface to the data regularized by the gradient at each point (Figure 6-3). While other tools (e.g. 2D Loess) were tried, gridfit produced indistinguishable fits considerably faster.

Specifically, to get a fit without outliers, we first fit the surface to the entire data set, then flagged all points more than 2 standard deviations from the fit surface. Removing these points the surface was re-fit. All points more than 2 standard deviations from the new surface fit were flagged, and this process was iterated until convergence.

This surface, minus its mean, was then pointwise subtracted from the corresponding spots on the array, yielding a new data set with no appreciable spatial signal.

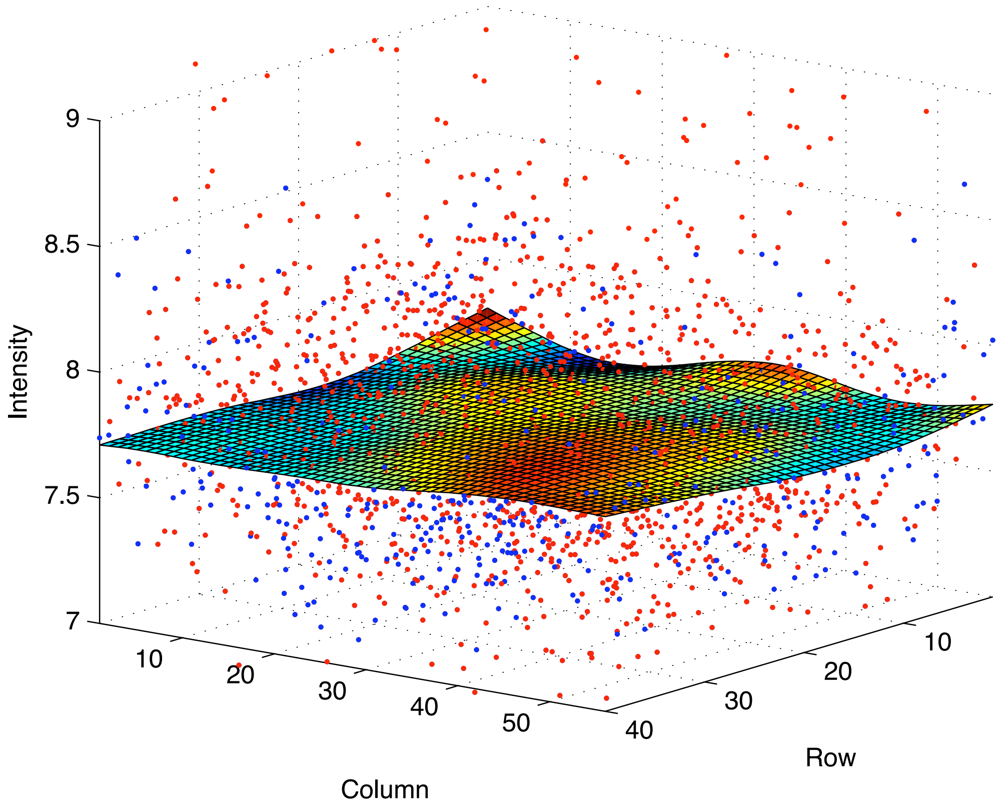


Figure 6-3: An example of a spatial artifact. The two horizontal axes are the true physical location of the spot, the vertical axis is the intensity of the spot. The resulting surface fit is shown, with outliers shown in red.

6.1.3 Integration of Different Array types

In order to better probe the genes thought most likely to vary from changes in active GTPases, after completing 44 experiments Dr. Bakal changed the design of the microarrays. The result of this was that a number of spots which were previously present were no longer, a number of new spots were present, and many of the repeated spots changed in multiplicity. To deal with this, all spots were averaged by probe sequence, producing 2072 distinct probes (1934 in the old set, 1985 in the new). Thus, the resulting data matrix is 281792 features over 136 experiments, with 14076 missing values. For the principal components reduction of dataset size, all features which did not appear in all arrays were thrown out, leaving 1847 probes, for a total of 251192 measurements.

6.1.4 Removal of Spurious Data

The two experiments with technical errors, as well as those with similarly low correlation to the mean, the ConA experiments, and an experiment in which bsk (not a g-protein) was knocked out, were removed from the dataset, to filter out both known and suspected gross experimental errors. See Figure 6-4. Notably, every removed experiment had a duplicate of itself within the kept dataset, implying that the large deviations are likely a result of error and not experimental effect.

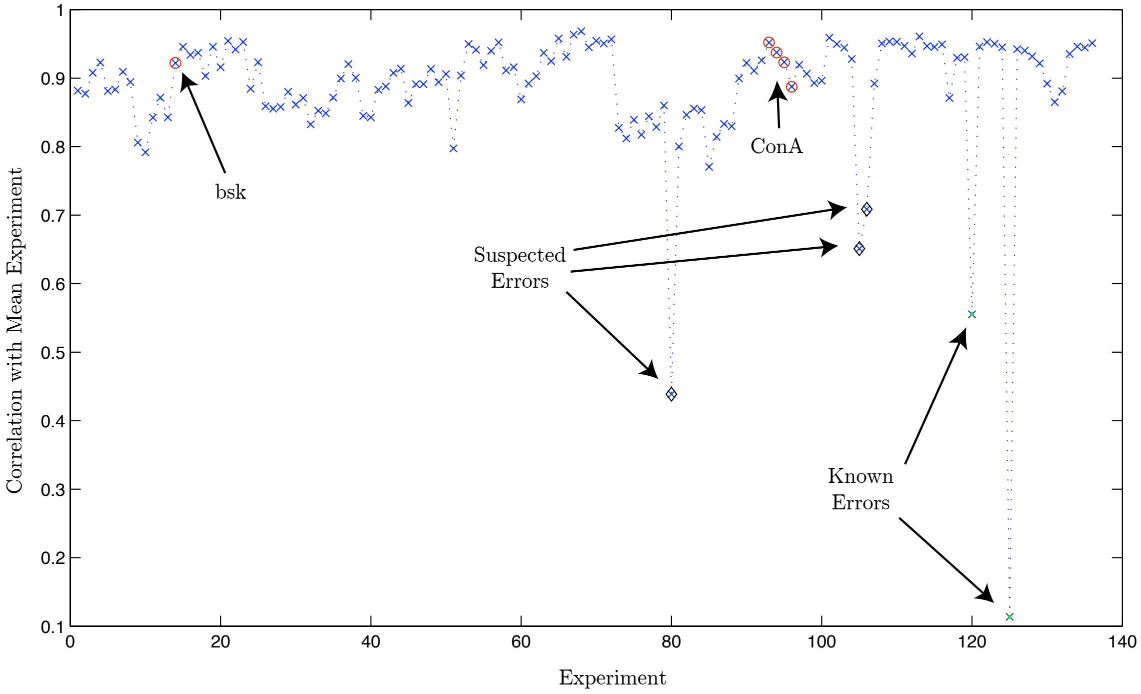


Figure 6-4: The mean correlation of experiments to the mean. Data points removed from the analysis are highlighted.

6.1.5 Inter-array Normalization

A further problem with microarray data is that different experiments (even of the same biological state) yield different background distributions of spot intensities. In general, either quantile [11] or Loess (Locally Weighted Scatterplot Smoothing) [18, 19, 81] normalization is used to deal with this (Figure 6-5), on the assumption that while the levels of various spots on the microarray should change with experiments, the overall shape of the distribution of spot intensities should not.

For these experiments, we Loess-normalize all of the microarrays to the feature-wise mean of all arrays. The geometric and arithmetic means in the case are spot-wise identical to four significant digits, so the choice of which mean to use is irrelevant. For each array, we compute the Loess regression of its intensity levels as a function of the difference from consensus intensity levels. If the two arrays have the same intensity distribution, the Loess fit should be precisely zero. If not, the fit curve is subtracted from from the corresponding features in the experiment, resulting in normalized data.[1]

Specifically, Loess works by a local center-weighted quadratic fit to windows of the data. For our experiments, we used the MATLAB Bioinformatics Toolbox [59] implementation of Loess normalization (`malowess`) with span 0.4 and order 2.

6.2 Model Creation

6.2.1 PCA reduction

We reduced the dataset to 35 features per experiment by way of an SVD. The number 35 was chosen empirically to capture the variation of the model to within an approx-

[1] This results in all experiments having the same mean and variance. On linearly-rescaled data, once every experiment is translated by the mean and rescaled by the standard deviation of the consensus, the resulting transform is precisely the Z-score normalization.

imation of the on-chip noise, computed by examining repeat experiments. (Figure 5-2).

6.2.2 AMPL Implementation

In order to use SNOPT [34], the model required translation to AMPL [30], a commonly-used modeling language. This was created in a straightforward manner via a script to convert the model minimization problem into AMPL form. While a number of isomorphic phrasings of the problem were tried, the following was the most effective:

```
minimize objvar:
     sum {i in I} ( sum {z in 1..126} ((  (phi[i,z] - sum {k in K}
     (a[i,k]*(kappa[k,z]*(1+sum {j in JE} (x[k,j]*eta[j,z]))/
     (1+sum {j in JE} (x[k,j]*eta[j,z]) + g[k] +
     sum {j in J} (y[k,j]*alpha[j,z])))) - r[i] -
      sum {l in L} (sum {d in D} (c[i,l]*u[l,d]*chi[d,z])))   )^2));
```

See Appendix B for an almost-full example (the input vectors and initial conditions are suppressed for space).

6.3 Model Fitting

6.3.1 Regularization

In order to for the model fit to converge to a minimum, a regularization term was added to the model objective function. Thus we were in fact minimizing the residual plus 0.01 times the L_1 norm of x, y, and γ, i.e.

```
minimize objvar:
```

```
sum {i in I} ( sum {z in 1..126} ((  (phi[i,z] - sum {k in K}
(a[i,k]* (kappa[k,z]*(1+sum {j in JE} (x[k,j]*eta[j,z]))/
(1+sum {j in JE} (x[k,j]*eta[j,z]) + g[k] +
sum {j in J} (y[k,j]*alpha[j,z])))) - r[i] -
sum {l in L} (sum {d in D} (c[i,l]*u[l,d]*chi[d,z])))   )^2))
+0.01*sum{k in K}(sum{j in JE}(x[k,j])+sum{j in J}(y[k,j])+g[k]);
```

The minima found with this showed precisely the same topologies as the unregularized fits were at after 100,000 iterations.

6.3.2 Consensus of Found Minima

Over 100 fits of the model to data found only 9 minima. These were all within the level of noise on the chip of one another, thus there was no *a priori* way to decide between them. As the average of the nine minima, while having a higher residual, still was within the noise threshold, we used it as a predicted model, and used a voting scheme to score the confidence of our predictions. Thus the fraction of minima in which a connection occurs is taken to be the confidence in its existence.

6.3.3 Other Methods

Other methods of model minimization were tried, but were not as effective as SNOPT, including the builtin routine in MATLAB, Newton-Raphson, and Genetic Algorithms.

MATLAB builtin

While MATLAB's built-in routine, `fmincon` is passable for small models, when the number of parameters exceeds approximately 700, it becomes inefficient to the point of being unusable for this purpose.

Newton-Raphson

For a problem of this size, Newton-Raphson proved infeasible. Each step took prohibitively long, and did not converge nearly as quickly as SNOPT. After 3 days of computation, Newton-Raphson had not found a local minimum, whereas SNOPT usually takes under 5 minutes.

Genetic Algorithms

Apropos a question at RECOMB 2008, a genetic algorithm was written to optimize the fit of model parameters. Despite trying a large number of crossover, mutation, and selection strategies, the genetic algorithm was unable to find a solution with a residual within an order of magnitude of the one found by SNOPT.

6.4 Other Techniques for Inference

6.4.1 Naïve Correlation

This starts with the assumption that if a GAP deactivates a given GTPase, the differential change to the expression profiles will be more strongly correlated than if not. Conversely it assumes that if a GEF activates a GTPase, their expression profiles will be anti-correlated.

Subtracting the same-day background experiment from each experiment, and averaging the knockouts of each protein, the Pearson correlation coefficient for proteins x and y (datasets d^x and d^y respectively) is computed as

$$S := \left\langle \frac{\left(d_i^x - \mu_{d^x}\right)\left(d_i^x - \mu_{d^y}\right)}{\sigma_{d^x}\sigma_{d^y}} \right\rangle . \tag{6.1}$$

An arbitrary cutoff distinguishes connections from non-connections. For the purpose of this work, all cutoffs were tested (Figure 5-1).

6.4.2 GSEA

GSEA starts by constructing, for each experimental condition, two subsets ("gene sets") of the features, one positive and one negative, which are used as indicators of the condition. To test whether a specific state is represented in a new experiment, the Kolmogorov-Smirnov enrichment score of those subsets in the new data is calculated (for details, see [75]). If the positive set is positively enriched and the negative set negatively enriched, the test state is said to be represented in the data. Likewise if the reverse occurs, the state is said to be negatively represented. If both are positively or negatively enriched, GSEA does not make a prediction. We are able to apply GSEA by computing positive and negative gene sets based on perturbation data for the substrates and then testing for enrichment in each of states in which we perturb an activator or inhibitor.

6.4.3 ARACNE

ARACNE, on the other hand, begins by computing the kernel-smoothed approximate mutual information (AMI) of every pair of features (for details, see [58]). In order to remove transitive effects, for every trio of features A, B, C, the pair with the smallest mutual information is marked to not be an edge. The remaining set of all unmarked edges is then a prediction of the network. As already discussed, we do not have features in our experiment that correspond directly to the levels we wish to measure. However, treating each experimental state as a feature, we are able to apply the AMI metric to obtain the relative efficacies of the activator and inhibitor perturbation experiments as predictors of the substrate perturbations. We know from the outset that the network we are trying to predict has no induced triangles, and so ARACNE would not remove any of the edges. However, the relative strengths of these predictions yield a predicted network topology.

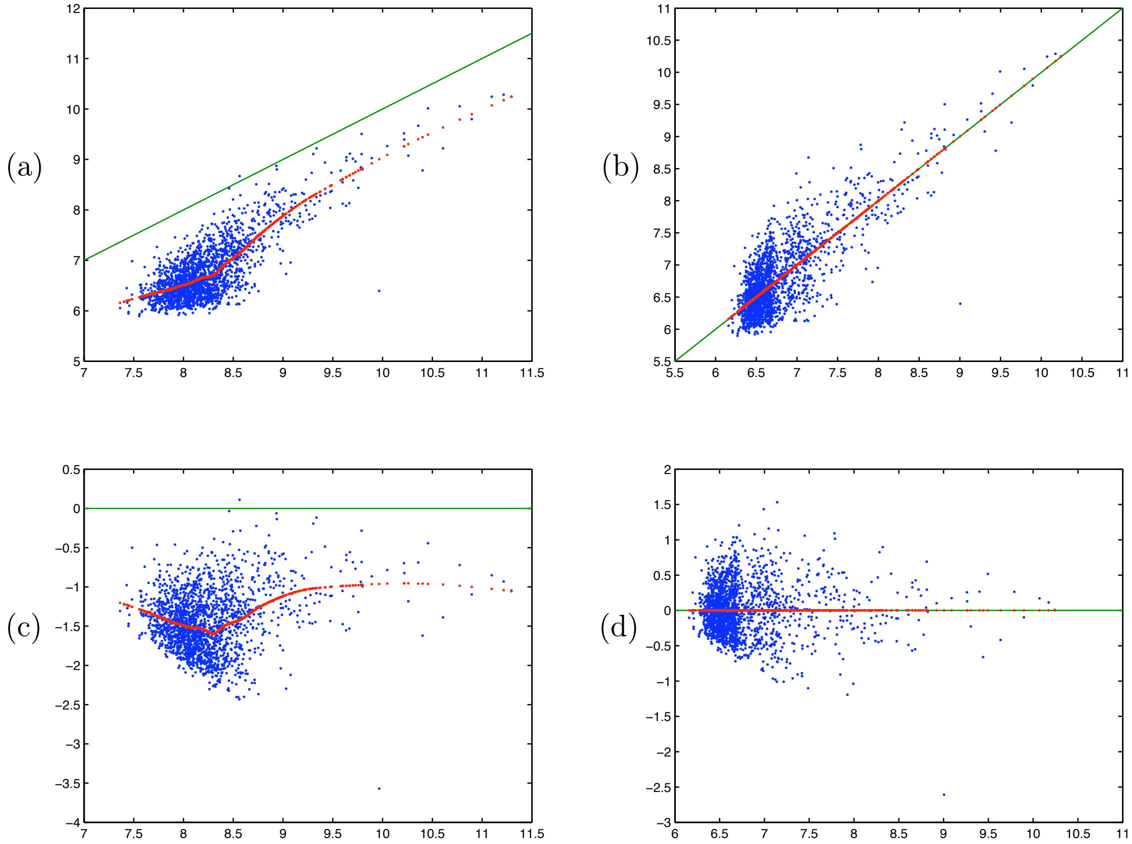


Figure 6-5: Loess Normalization – all axes are log arbitrary intensity units. In all figures the blue scatter is the data, the green is the ideal line (i.e. matching the mean), and the red is the loess fit (a) An experiment vs. the mean intensity across experiments. (b) The same experiment as (a), after subtraction of the loess fit. (c) The same experiment as (a) vs. the difference between the experimental value and the mean. (d) The same experiment vs. the difference between its values and the mean after Loess normalization.

Part II

Information Theory and Population Genetics

Chapter 7

Introduction

In this section we attempt to draw the beginnings of a deep structural connection between population genetics and information theory. We show that the natural quantification of genetic drift is in terms of the entropy of the population, as it n expectation decreases by a constant each generation, barring mutation and selection. Of interesting historical note is that Claude Shannon's MIT thesis was on the topic of population genetics [72], though Shannon himself does not appear to have studied the connection further.

Population genetics is the study of the genetic makeup of a population as it evolves over time. While these dynamics have been studied since the early 20th century, there remain many unsolved problems in the field.

In Part II of this thesis we address genetic drift. Drift here is the accumulation of sampling artifacts, arising from the fact that every generation has a quantized reproduction procedure. A particularly useful setting in which to study genetic drift is the *neutral model*, i.e. the case where there is no mutation and all alleles are selected for equally. In large populations, drift tends to be relatively unimportant, however in isolated groups, or if the population experiences a bottleneck, the effects of drift can be quite dramatic.

We address the quantification of genetic drift under neutral conditions. While the time scales of mutation and selection are well understood, drift has eluded a natural quantification.

The idea of studying biology in an information theoretic context has been in the air for some time. It has been used to study the information an entity receives about its environment[49] as well as to study message transmission between biological entities[15]. The idea of measuring genetic diversity in terms of information theory has been proposed[15], though it has not been studied in the context of canonical models of population genetics.

We analyze the expected change in Shannon entropy, $-\sum p_i \log p_i$, of a population under neutral conditions. To do this we consider both the classically studied Wright-Fischer Model, the more recent Moran Model, and the continuous diffusion approximation of population dynamics. While genetics has been previously considered in terms of entropy of an individual's genome [82], the entropy we here consider is that of a genome across a population.

We discover that in the neutral case of all three models, total entropy (entropy times population) in expectation decreases linearly. Moreover, the speed with which it decreases is dependent only on the number of present alleles and a constant relating the effective population of the model to the actual population. This suggests entropy is a natural measure of the diversity of a population with which to quantify drift.

This result has a clear discontinuity at the loss of an allele, and as such become less accurate as it approaches this point. We find, however, that a modified measure of entropy, $-\sum(1-p_i)\log(1-p_i)$, decreases in a similar linear fashion without the same discontinuity. The significance of this alternate entropy measure is not understood.

This suggests a deep connection between information theory and population genetics, namely that the the dynamics of genetic diversity in a population can be seen as the transmission of a coded message through a noisy channel.

Chapter 8

Background on Population Genetics and Information Theory

In this Chapter we lay out the necessary background on Population Genetics and Information Theory insofar as they pertain to the results of this section. For a more thorough treatment of population genetics and information theory, we refer the reader to the excellent books by Ewens [25] and Cover and Thomas [20] respectively.

8.1 Population Genetics

Evolutionary dynamics has been analyzed in a number of models, both discrete and continuous. Of the discrete time models, the Wright-Fischer and Moran are the most studied. Both are discrete-time Markov processes, in which, at each time step, randomly chosen members of the population die and reproduce. While neither is an exact model of actual dynamics, the hope is that these models capture the critical dynamics in evolutionary processes.

In all models we consider a population of size N, consisting of k^j alleles at time j, for which x_i^j is the number of members of the population with allele i and compute x_i^{j+1} at each step. A particularly useful and interesting case of both models is the *neutral* case, in which there is no mutation and selection is unbiased[46], i.e. the case in which in expectation $\mathbb{E}[x^{j+1}] = x^j$.

Note in both of these cases that the absorbing states are precisely of the form $(1, 0, \ldots, 0)$ up to permutation.

8.1.1 Wright-Fischer Model

In each generation of the Wright-Fischer model, the population is replaced by an exactly equal number of new members, each of whom is assigned a random member (with repeats) of the previous generation with whom it is exactly identical[28, 79]. (see Figure 8-1). This mapping can also be seen exactly as a multinomial resampling of the previous generation's allelic distribution.[1]

In the neutral case, for a population $x^t = (x_1^t, x_2^t, \ldots, x_k^t)$,

$$\begin{aligned} \mathbb{P}\left(\left(x_1^{t+1}, \ldots, x_k^{t+1}\right) = (y_1, \ldots, y_k)\right) &= \frac{n!}{y_1! \cdots y_k!} \left(\frac{x_1^t}{N}\right)^{y_1} \cdots \left(\frac{x_k^t}{N}\right)^{y_k} \qquad (8.1)\\ &= \frac{n!}{y_1! \cdots y_k!} \left(x_1^t\right)^{y_1} \cdots \left(x_1^t\right)^{y_k} N^{-N}, \end{aligned}$$

and critically,

$$\mathbb{P}\left(x_i^{t+1} = m\right) = \binom{N}{x_i^{t+1}} \left(\frac{x_i^t}{N}\right)^m \left(1 - \frac{x_i^t}{N}\right)^{N-m} . \qquad (8.2)$$

A well-known result that the expected fixation time, i.e. the time to homogeneity, of a two-allele haploid population is $2N(-x \log x - (1-x)\log(1-x))$ generations [46, 78, 24, 16] is, in fact, a consequence of a more general result we develop later in

[1] For the sake of uniformity of notation we consider the haploid case; however the results for a haploid population of size N apply exactly to a diploid population of size N/2.

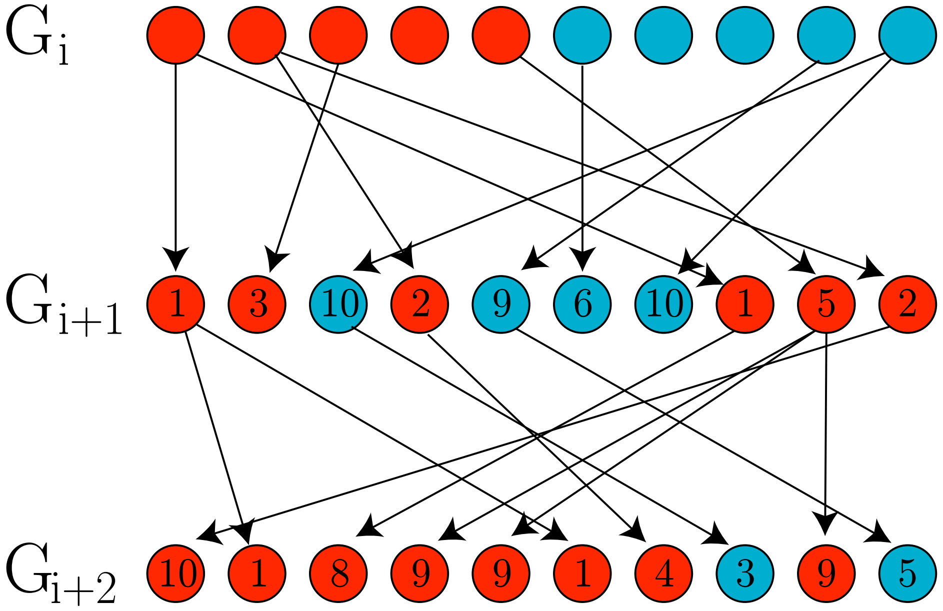


Figure 8-1: Three generations of the Wright-Fischer Model, showing the rapid effects of drift on a small population. The numbers inscribed in the individuals in generations $i+1$ and $i+2$ indicate the randomly chosen ancestor.

this chapter.

8.1.2 Moran Model

In the Moran Model, in each iteration two random members of the population are chosen (with replacement), one of which is removed and replaced with the other. Thus, in the neutral case:

$$\mathbb{P}\left(\left(x_1^{t+1},\ldots,x_k^{t+1}\right)=(y_1,\ldots,y_k)\right) \quad = \quad \begin{cases} \left(\frac{x_i^t}{N}\right)\left(\frac{x_j^t}{N}\right) & \text{if } y_i = x_i^t+1, y_j = x_j^t-1, \\ & \quad \forall(l\neq i,j), y_l = x_l^t \\ \sum_i \left(\frac{x_i^t}{N}\right)^2 & \text{if } \forall i, y_i = x_i^t \\ 0 & \text{otherwise.} \end{cases}$$

Thus in the first case, a member with allelotype i is chosen to replace allele j, and in the second a member of some allelotype is chosen to replace itself, resulting in no change in the population. While generally considered even less realistic than the Wright-Fischer Model, this model is, in general, far more mathematically tractable. It is more readily generalizable to structured populations (e.g. those where not all individuals can replace all others). A recent, useful generalization of the Moran process has been to evolution on arbitrary directed graphs [55].

8.1.3 Diffusion Model

Both studied cases fall under the broader category of models which are well-approximated by diffusion [46]. Diffusion models describe the probability density $f(x;t)$ of finding a configuration x at time t. In the neutral case, the evolution of $f(x;t)$ is given by:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\sum_{i=1}^{k-1}\frac{\partial^2 f}{\partial x_i^2}\left\{x_i\left(1-x_i\right)\right\} - \frac{1}{2}\sum_{i,j<k}\frac{\partial^2 f}{\partial x_i \partial x_j}\left\{x_i x_j\right\} \tag{8.3}$$

where the time unit is N generations.

8.2 Information Entropy

Entropy is a measure of the uncertainty of a random variable. For a discrete random variable X with values $\{x_1, \ldots, x_n\}$ the entropy of X is defined as:

$$S(X) := -\sum_i p\left(x_i\right) \log p\left(x_i\right) \tag{8.4}$$

Treating the alleles seen in a population as representative of the output of a random variable, we can define the entropy of a population of size N with k allelotypes of sizes $\{x_1, \ldots, x_k\}$ as:

$$S(x) := -\sum_i \frac{x_i}{N} \log\left(\frac{x_i}{N}\right) \tag{8.5}$$

While the population is not a random variable (though reproduction is a random process), this provides a natural measure of the diversity of a population.

Chapter 9

Neutral Model Results

The neutral model is when selection and mutation are both zero. In this case, only genetic drift is important, allowing its dynamics to be studied in isolation. In the Wright-Fischer Model and Moran Model we find total entropy decreases linearly as a function of the effective population and the number of nonzero alleles. Moreover, we show that for any diffusion-approximable model this result holds.

Thus, in most reasonable models of population genetics, in absence of mutation and selection, $T(x)$, the total entropy of the population, decreases linearly with time, dependent only on the number of possible alleles at time t.

9.1 The k-allele Wright-Fischer Model

In this model, one assumes that in each generation every member of a population chooses a random parent from the previous generation and as identical to it. Whereas the classical Wright-Fischer Model addresses strictly the case of a two-allele system, the result follow as a specific case. This amounts to multinomial sampling of the parent generation's allele distribution, thus in a given generation:

$$\begin{aligned} \langle \Delta x \rangle &= 0 & (9.1) \\ \mu_2 = \sigma_i^2 = \left\langle \Delta x_i^2 \right\rangle &= x_i \left(1 - \frac{x_i}{N}\right) & (9.2) \end{aligned}$$

This allows us to compute $\langle \Delta T(x) \rangle = N \langle S(x + \Delta x) \rangle$, the expected change in entropy in a generation. We do this by way of a second-order approximation, yielding:

$$\langle \Delta T(x) \rangle \approx NS(x) - \frac{k-1}{2}. \tag{9.3}$$

For details of the derivation, see Appendix C.2.1.

9.2 The Moran Model

In each time step of the Moran Model, one member of the population is chosen at random for duplication and one for death. In the event that these two are the same, the population is unchanged. The natural definition of a "generation" in this case is N time steps, as this yields an expected number of breeding events per population member of 1.

In this case for $n \in \mathbb{N}$,

$$\left\langle \Delta x_i^{2n-1} \right\rangle = \mathbb{P}\left(\Delta x_i = 1\right) - \mathbb{P}\left(\Delta x_i = -1\right) = p_i(1 - p_i) - \sum_{j \neq i} p_j p_i = 0 \tag{9.4}$$

and

$$\left\langle \Delta x_i^{2n} \right\rangle = \mathbb{P}\left(\Delta x_i = 1\right) + \mathbb{P}\left(\Delta x_i = -1\right) = p_i(1 - p_i) - \sum_{j \neq i} p_j p_i = 2\frac{x_i}{N}(1 - \frac{x_i}{N}) \tag{9.5}$$

and so in the Moran Model:

$$\langle \Delta T(x) \rangle = -\frac{k-1}{N} + O\left(\sum \frac{1}{x_i^2}\right) \tag{9.6}$$

For details of the derivation, see Appendix C.2.2.

9.3 Diffusion Approximation

Recall that both the Wright-Fischer Model and Moran Model fall under the class of diffusion-approximable models, in which, in the neutral case:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\sum_{i=1}^{k-1} \frac{\partial^2 f}{\partial x_i^2}\left\{x_i\left(1-x_i\right)\right\} - \frac{1}{2}\sum_{i,j<k} \frac{\partial^2 f}{\partial x_i \partial x_j}\left\{x_i x_j\right\} \tag{9.7}$$

Taking $S(t)$ in this case[1] to be the expectation over x, i.e.:

$$S(t) \equiv \int_{\sum x_i \leq 1} S(x) f(x;t) dx \tag{9.8}$$

we find:

$$\frac{\partial S(t)}{\partial t} = -\frac{k-1}{2} \tag{9.9}$$

in the case where $x_i = 0 \Rightarrow f(x;t) = 0$, i.e. when the probability of having already lost an allele is 0. In general, we let $k(x)$ be the number of non-zero alleles at x and find:

[1] Note that we are interested in the expected entropy of the distribution allelotypes in the population, not the differential entropy of the distribution of distributions, $-\int_{\sum x_i \leq 1} f(x;t) \log f(x;t) dx$.

$$\frac{\partial S(t)}{\partial t} \;=\; \int_{\sum x_i \leq 1} \frac{1-k(x)}{2} f(x;t)dx. \tag{9.10}$$

As we've normalized our time unit to N generations, this implies that in any diffusion-approximable model (Moran and Wright-Fischer included) with effective population N_e:

$$\left\langle \Delta T_e(x^{j+1}) \right\rangle \approx -\frac{k-1}{2}. \tag{9.11}$$

9.4 $-\sum(1-p_i)\log(1-p_i)$ Analysis

Interestingly, when we consider the functional

$$Z(x) \equiv \sum_i -(1-p_i)\log(1-p_i) = -\sum\left(1-\frac{x_i}{N}\right)\log\left(1-\frac{x_i}{N}\right) \tag{9.12}$$

the results of the section 9.1 come out more cleanly. Specifically:

$$\langle N_e \Delta Z(x) \rangle \approx -\frac{1}{2} \tag{9.13}$$

The significance of this functional remains unclear, though we hypothesize it is related to the population-sum constraint ($\sum_i x_i = N$). This also provides an alternate derivation of the result that the expected time to fixation of a k allele population is $-2N_e\sum_i(1-p_i)\log(1-p_i)$.

Chapter 10

The Analogy to Information Theory

The results presented earlier in this chapter are strong hints that the informatic structure of population dynamics merits further study. Beyond these results, a clear analogy between the structure of population genetics and classical information theory problems exists. In this chapter we lay out some of the similarities in the hopes that this provides a basis for future study of the believed deep connection between information theory and population genetics.

10.1 Structural Similarities

There exists a striking analogy between population genetics and information theory, (Fig. 10-1). It is our hope that by making this structure explicit, information theoretic results can be used to further illuminate our understanding of the flow of genetic information in populations.

In essence, the population genotype encodes the phenotype, which is transmitted

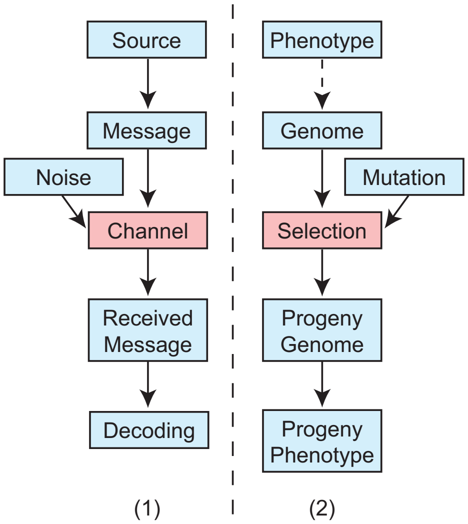


Figure 10-1: The general structure of the connection between information theory and population genetics.

to the next generation. This signal is subject to resampling (selection, drift) as well as noise (mutation), and it is the coding scheme from genome to phenome which dictates the effects of this process on phenotypic variation.

Part III

Multiple Network Alignment

Chapter 11

Introduction

Almost every biological process is mediated by a network of molecular interactions. A few examples of these include: genetic regulatory networks, signaling networks, metabolic networks, and protein-protein interaction networks. The structure of these networks is becoming increasingly well known, especially with the advent of high-throughput methods for network inference [77, 41, 48]. As with the genome, there is significant conservation of network structure between organisms [60, 83]. Thus, knowledge about the topology of a network in one organism can yield insights about not only the networks of similar organisms, but the function of their components. A problem with accurate cross-species comparison of such networks is that the known networks, however, are both incomplete and inaccurate [37, 38].

The specific problem we address is that of global alignment of multiple *Protein-Protein Interaction (PPI)* networks. A PPI network is an undirected collection of pairwise interactions on a set of proteins, where an edge represents interaction between two proteins. Given a pair of PPI networks, and a list of pairwise sequence similarities between proteins in the two networks, the problem is to find an optimal mapping between the nodes of the two networks that best represents conserved biological function. We distinguish such global network alignment from local alignment where the goal is to find multiple network *motifs*, i.e. independent regions of localized network similarity. In the multiple global network alignment case, with k networks, the problem is extended to finding clusters of proteins across the networks such that these clusters best represent conserved biological function.

This search for such an alignment is motivated by the intuition that evolution of genes happens within the context of the larger cellular system they are part of. Global network alignment can be interpreted as an evolutionary analysis done at this systems level rather than in a piecemeal, local fashion. Once a global network alignment has been estimated, we can analyze it to gather more localized, granular insights, e.g., estimating functional orthology across species.

Alignment of multiple networks poses two key problems. The first is that the computational complexity (i.e., the number of possible alignments) grows exponentially in the number of networks. The second is that the genomes corresponding to the various networks being aligned may vary widely in size (say, because of differing degrees of gene duplication). A multiple network alignment algorithm must thus efficiently identify a biologically-appropriate mapping between the genes.

Here we introduce IsoRankN (IsoRank-Nibble), which takes the approach of deriving pairwise alignment scores between every pair of networks, using the original IsoRank methodology [73, see Box 12-1]; then finds alignment clusters based on these scores. To find clusters, we use a spectral partitioning method that is both efficient and automatically adjusts to the wide variation in sizes of the species-specific networks. The algorithm is similar to the recently developed `PageRank-Nibble` algorithm [3], which approximated the *Personalized PageRank* vector. A PageRank vector (i.e., one that describes a ranking of graph nodes for, say, search) is called a *Personalized PageRank* vector if, given a particular graph node, its preference scores are concentrated on a small set of vertices, the set being tailored to the given node. This notion of vertex-specific rankings is applied in IsoRankN to find dense, clique-like clusters of proteins when computing the global alignment of multiple PPI networks.

We tested IsoRankN on the five known eukaryotic PPI networks, i.e. Human, Mouse, Fly, Worm, and Yeast. Much of the related previous work has focused on local network alignment; hence, a direct comparative evaluation of our results was difficult. As a gold standard alignment does not yet exist, we instead evaluate our alignment method on a variety of indirect criteria, including number of clusters predicted, within-cluster consistency, and GO/KEGG enrichment [4, 43]. In order to measure within-cluster consistency, we introduce a novel metric based on the entropy of the GO/KEGG annotations of predicted clusters. We believe that the characteristic of a correct global network alignment would be to preserve the relative functions of various network parts; this can be well-measured by the various GO enrichment analyses described above.

A number of related techniques for PPI network alignment exist. Most notably, these include NetworkBLAST-M [42], Græmlin 2.0 [29] and IsoRank [73], though a number of other techniques exist as well [10, 23, 44, 45, 47]. NetworkBLAST-M computes a local alignment by greedily finding regions of high local conservation based on inferred phylogeny. Græmlin 2.0, by contrast, computes a global alignment by training how to infer networks from phylogenetic relationships on a known set of alignments, then optimizing the learned objective function on the set of all networks.

IsoRank uses spectral graph theory to first find pairwise alignment scores across all pairs of networks, the details of which are provided later (Box 12-1); these pairwise scores, computed by spectral clustering on the product graph, work well in capturing both the topological similarity as well sequence similarity between nodes of the networks. However, to find multiple network alignments, IsoRank uses these scores in a time-intensive greedy algorithm. Instead, IsoRankN uses a different method of spectral clustering on the induced graph of pairwise alignment scores. The new approach provides significant advantages, not only over the original IsoRank but also over other methods.

To test IsoRankN, we show that on the PPI networks from five different eukaryotic species, IsoRankN produces an alignment with a larger number of aligned proteins, higher within-cluster consistency, and higher biological similarity than existing methods, as measured by GO/KEGG enrichment using GO TermFinder [12]. While other techniques for measuring GO enrichment exist [71, 69], they did not apply directly to the context in which we work. Additionally, IsoRankN does not require training and does not rely on induced phylogeny; thus it is not sensitive to errors in the phylogenetic tree. While this is not a significant problem with eukaryotes, inference of accurate bacterial phylogeny has proven far more difficult.

Contributions. We introduce the IsoRankN algorithm which uses an approach similar to the `PageRank-Nibble` algorithm to align multiple PPI networks. In so doing, we bring a novel spectral clustering method to the bioinformatics community. We use IsoRankN to align the known eukaryotic PPI networks and find that it efficiently produces higher-fidelity alignments than existing global multiple-alignment algorithms.

Chapter 12

IsoRank N

12.1 Methods

12.1.1 Functional Similarity Graph

The central idea of IsoRankN is to build a multiple network alignment by local partitioning of the graph of pairwise functional similarity scores. Specifically, given k PPI networks, $G_1, G_2, \ldots, G_k$, we first compute the functional similarity scores of every pair of cross-species proteins $(v_i, v_j) \in (G_l, G_m)$. This is done using the original IsoRank algorithm (see Box 12-1), but without the final step of greedily selecting an alignment. The scores generated by IsoRank have the advantage of being highly noise tolerant, a result of using a spectral approach.

The result is a *functional similarity graph*, a weighted complete k-partite graph on the k sets of proteins, where each edge is weighted by its functional similarity score. If the PPI networks were complete and exact, the multiple alignment problem would simply be to find maximally weighted cliques. As the networks are not, we introduce the *star-spread* method to find highly similar near-cliques, which yields a multiple alignment. In addition, in contrast to the *seed-path extension* method used

Box 12-1. The Original IsoRank Algorithm.

IsoRank works on the principle that if two nodes of different networks are aligned, then their neighbors should be aligned as well. In lieu of sequence similarity information, the functional similarity score R_{ij} between vertex v_i and v_j is the set of positive scores which satisfies:

$$R_{ij} = \sum_{\substack{v_u \in N(v_i) \\ v_w \in N(v_j)}} \frac{1}{|N(v_u)||N(v_w)|} R_{uw},$$

where $N(v_i)$ is the neighborhood of v_i within its own network. This can also be viewed as the steady-state distribution of a random walk on the direct product of the two networks.

To integrate a vector of sequence homologies, E, IsoRank takes a parameterized average between the network-topological similarity and the known sequence homology. It uses the power method to find the unique positive R satisfying

$$R = \alpha A R + (1 - \alpha) E, \text{ with } 0 \leq \alpha \leq 1,$$

where

$$A_{ij,uw} = \begin{cases} \frac{1}{|N(v_u)||N(v_w)|}, & v_u \in N(v_i), v_w \in N(v_j), \\ 0, & \text{otherwise.} \end{cases}$$

Given the resulting vector of pairwise functional similarity scores, R, a discrete network alignment is then greedily generated.

by NetworkBLAST-M, our method is similar to the *star aligned* approach in multiple sequence alignment introduced by [56] and CLUSTAL W [76].

12.1.2 Star Spread

We first compute, for every protein v in a chosen species, every neighbor connected to v by an edge with weight greater than a threshold; this is the *star*, S_v of the protein (see Figure 12-1(a)). We greedily order the proteins in v by the total weight of S_v and for each find the subset $S_v^* \subset S_v$ such that S_v^* is a highly weighted neighborhood of v (see Figure 12-1(b)). This is done using a spectral local graph partitioning algorithm

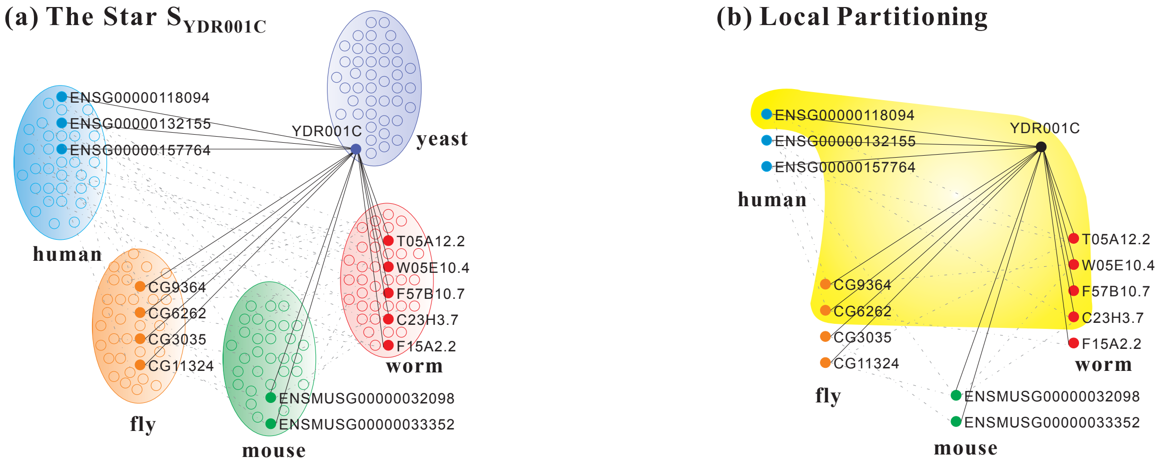


Figure 12-1: An example of star spread on the five known eukaryotic networks. (a) S_{YDR001C}, the set of all neighbors of YDR001C with a similarity bounded by a threshold $\beta = 0.01$. The illustration emphasizes the key idea of star spread, that the neighborhood of a single protein, YDR001C, has many high-weight neighbors in other networks, each of which are connected to others with varying weights. As the data are noise, we seek a highly weighted subset of this neighborhood, as opposed to a clique. (b) The shaded area is the resulting conserved interaction cluster S^*_{YDR001C}, containing YDR001C, as generated by our local graph partition algorithm.

with approximate *Personalized PageRank vectors*, similar to the `PageRank-Nibble` algorithm. The resulting S^*_v represents a *functionally conserved interaction cluster*, a set of network-aligned proteins. This is repeated for every protein in all species not already assigned to an S^*_v, yielding assignments for all vertices. While it is not clear exactly how the order of vertex choice affects the results, this ordering performs better empirically than others we have tried, including random ordering. The ordering of species is discussed below.

12.1.3 Spectral Partitioning

The main algorithmic challenge in obtaining functionally conserved interaction clusters S^*_v is uncertainty introduced by the incomplete and inaccurate PPI network data. Thus instead of finding a maximally weighted clique containing v, we find a low-conductance set containing v.

The *conductance*, $\Phi(S)$, of a subset S of a graph is the ratio of the size of the edge cut to separate S to the number of edges in the larger of the two remaining sets, providing a very natural measure of "clusterness" of a subset of vertices. Formally, $\Phi(S) = \frac{\sigma(S)}{\min\{vol(S), 2m - vol(S)\}}$, where $\sigma(S) = |\{(v_x, v_y); v_x \in S, v_y \notin S\}|$, $vol(S) = \sum_i deg(v_i)$, and m is the number of edges in S.

[3] showed that a low-conductance set containing v can be computed efficiently via the personalized PageRank vector of v. A *personalized PageRank vector* $Pr(\gamma, v)$ is the stationary distribution of the random walk on S_v in which at every step, with probability γ, the walk "teleports" back to v, and otherwise performs a lazy random walk with transition probabilities proportional to R, the vector of pairwise interaction scores (i.e. with probability 1/2, the walk does not move). Thus in this case, a personalized PageRank vector is the unique solution to:

$$Pr(\gamma, v) = \gamma\chi_v + (1-\gamma)Pr(\gamma, v)W, \tag{12.1}$$

where $\gamma \in (0, 1]$, $\chi_v(x) = \delta_{x,v}$ is the indicator vector of v, $W = \frac{1}{2}(I + D^{-1}R)$ is the lazy random walk transition matrix, and D is the diagonal of column-sums of R. For the purposes of this chapter, we instead use an efficient approximation $p \approx Pr(\gamma, v)$, the details of which can be found in [3].

To compute the minimal conductance cut, we consider the sets $T_j^p = \left\{ v_i \middle| \frac{p(v_i)}{\sum_k R_{ik}} \geq \frac{p(v_j)}{\sum_k R_{jk}} \right\}$, or those vertices which contain at least as much of the mass of p, normalized by R. As in [3], we then find the set S_v^* as:

$$S_v^* = \min_j \Phi\left(T_j^p\right). \tag{12.2}$$

12.1.4 Star Merging

While highly efficient, the star spread method has the limitation of not assigning other members of the original network to the neighborhood S_v, and so S_v^* by necessity does not contain any other proteins in the same network as v, even if it is appropriate to do so. To get around this, we introduce a procedure for merging stars, by looking at the neighbors of the neighbors of v. For two stars, $S_{v_1}^*$ and $S_{v_2}^*$, where v_1 and v_2 are in the same PPI network, if every member of $S_{v_1}^* \setminus \{v_1\}$ has v_2 as a neighbor and vice-versa, we merge $S_{v_1}^*$ and $S_{v_2}^*$.

Table 12.1: Comparative consistency on the five eukaryotic networks

	IsoRankN	IsoRank	Græmlin$_{1K}$	Græmlin$_{2K}$	NB-M
Mean Entropy	**0.274**	0.685	0.857	0.552	0.907
Mean Normalized Entropy	**0.179**	0.359	0.451	0.357	0.554
Exact cluster ratio$^\sharp$	**0.380**	0.253	0.306	0.355	0.291
Exact protein ratio†	**0.261**	0.165	0.159	0.248	0.142

Mean entropy and mean normalized entropy of predicted clusters. Note that the boldface numbers represent the best performance with respect to each measure.
$^\sharp$The fraction of predicted clusters which are *exact.*, i.e. all contained proteins have the same KEGG or GO group ID.
†The fraction of proteins in exact clusters.

12.1.5 The IsoRankN Algorithm

Given k PPI networks $G_1, G_2, \ldots, G_k$, and a threshold β, IsoRankN proceeds as follows:

1. Run the original IsoRank on every pair of networks to obtain scores R_{ij} on all edges of the functional similarity graph.

2. For every protein v, compute the star
   $S_v = \{v_j \in N(v) | w(v, v_j) \geq \beta \max(w(v, v_j))\}$, where $N(v)$ is the neighborhood of v in the functional similarity graph.

3. Pick an arbitrary remaining PPI network G_ℓ and order the proteins $v \in G_\ell$ by the sum of edge weights in the induced graph on S_v. In order, excluding proteins already assigned to clusters, spectrally partition S_v to obtain S_v^*.

4. Merge every pair of clusters $S_{v_1}^*$ and $S_{v_2}^*$ in which $\forall v_i \in S_{v_2}^* \setminus \{v_2\}, w(v_1, v_i) \geq \beta \max_j (w(v_1, v_j))$ and $\forall v_j \in S_{v_1}^* \setminus \{v_1\}, w(v_2, v_j) \geq \beta \max_j (w(v_2, v_j))$.

5. Repeat steps 3 and 4 until all proteins are assigned to a cluster.

12.2 Results

Experimental datasets. We tested IsoRankN on five eukaryotic PPI networks: *H. sapiens* (Human), *M. musculus* (Mouse), *D. melanogaster* (Fly), *C. elegans* (Worm), and *S. cerevisiae* (Yeast). IsoRankN requires two forms of data as input: PPI networks and sequence similarity scores. The PPI networks were constructed by combining data from the DIP [80], BioGRID [74], and HPRD [64] databases. In total, these five networks contained 87,737 proteins and 98,945 known interactions. The sequence similarity scores of pairs of proteins were the BLAST Bit-values of the sequences as retrieved from Ensembl [39]. We evaluated the biological relevance of our results against two gene ontology databases, GO [4] and KEGG [43]. For this chapter, we set $\alpha = 0.6$ and $\beta = 0.01$, and use Human, Mouse, Fly, Worm, Yeast as the order of species that are at the center of the star-spread. We further investigated other species permutations as discussed later.

Testing. In the results that follow, we have aimed to evaluate our method along two key dimensions: coverage and consistency. Coverage is the set of genes for which our algorithm makes non-trivial predictions. It is thus a proxy for sensitivity; a higher coverage would be desirable in that it suggests our algorithm can explain a larger amount of data. The other dimension, consistency, measures the functional uniformity of genes in each cluster. The intuition here is that each cluster should correspond to a set of genes with the same function; higher consistency is better. This measure serves as a proxy for the specificity of our method.

There currently exists no gold standard for network alignment quality, so in order to evaluate the predictions of IsoRankN we tested two properties of its predictions that we expect an optimal prediction to have. First we tested within-cluster consistency of GO/KEGG annotation, on the reasoning that predicted orthologs in an orthology should likely have similar function. Second, we tested coverage, on the reasoning that an ideal alignment should assign most proteins to a cluster. As local alignment may have ambiguous, inconsistent or overlapping clusters, we primarily compare IsoRankN to IsoRank and Græmlin 2.0. We also compare to local aligners (such as NetworkBLAST-M), however, these will have lower coverage as they only consider conserved modules.

12.2.1 Functional assignment

We tested IsoRankN as compared to IsoRank, Græmlin 2.0 and NetworkBLAST-M on the five available eukaryotic networks and found that it outperformed the other methods in terms of number of clusters predicted, within-cluster consistency, and GO/KEGG enrichment.

Græmlin 2.0 requires a training set to learn the parameters of its scoring function. As in [29], we train Græmlin 2.0 on training sets of multiple sizes. The versions of Græmlin 2.0 trained on 1000 and 2000 KEGG clusters are denoted Græmlin$_{1K}$ and Græmlin$_{2K}$ respectively. We additionally attempted to train Græmlin 2.0 on 4000 clusters, but have not included the data, as it showed strong evidence of over-fitting.

Consistency. We first measured the consistency of the predicted network alignment by computing the mean entropy of the predicted clusters. The entropy of a given cluster S_v^* is:

$$H(S_v^*) = H(p_1, p_2, \ldots, p_d) = -\sum_{i=1}^{d} p_i \log p_i, \tag{12.3}$$

where p_i is the fraction of S_v^* with GO or KEGG group ID i. We also computed the mean entropy normalized by cluster size; *i.e.,* $\bar{H}(S_v^*) = \frac{1}{\log d} H(S_v^*)$. Thus a cluster has lower entropy if its GO and KEGG annotations are more within-cluster consistent. While a cluster with one element would have entropy zero, this is to be expected, as such a cluster is perfectly consistent with itself.

IsoRankN's predicted clusters have much lower entropy than IsoRank, Græmlin 2.0, and NetworkBLAST-M (see Table 12.1). *I.e.,* the clusters obtained by IsoRankN have higher consistency of annotation. For the purpose of this measure, proteins without a GO or KEGG group ID were withheld.

We additionally measure as in [29] the fraction of clusters which are *exact*, *i.e.* those in which all proteins have the same GO or KEGG ID. For GO annotation, we restrict to the deepest categories, removing questions of multiplicity and specificity of annotations. We find that IsoRankN predicts significantly more exact clusters than existing techniques, and that a higher fraction of the predicted clusters are exact (see Table 12.1). We note that only 60-70% of the proteins in any of the aligned networks have an assigned GO or KEGG ID, comparable to the fraction of all known proteins included in GO or KEGG. Additionally the relative performance under either consistency measure does not change when restricted to GO or KEGG individually.

Coverage. We first measure coverage by the number of clusters containing proteins from k species. We find that for $k \geq 3$, IsoRankN predicts more clusters with more

Table 12.2: Number of clusters/proteins predicted containing exactly k species. The kth row contains, for each program, the number of predicted clusters for covering exactly k species and number of constituent proteins in those clusters. Note that the boldface numbers represent the best performance with respect to each row. NetworkBLAST-M is not included, as it always outputs $k = 5$ species in each cluster. *All clusters obtained by IsoRankN contain at least two species.

# of species (k)	IsoRankN	IsoRank	Græmlin$_{1K}$	Græmlin$_{2K}$
1	-/-*	155/402	1418 /**4001**	**1521**/2910
2	3844/8739	**6499/20580**	1354/ 4650	2034/5899
3	**4022/13533**	3036/13391	947/5414	1116/5072
4	**2926/13991**	2446/15422	529/5371	310/2067
5	**2056/12715**	773/9744	58/1467	11/78
Total	12848/48978	**12909/59539**	4306/20903	4992/16026

proteins (see Table 12.2) than other methods. Thus, as it has higher consistency, it is likely that IsoRankN is detecting more distant multiple network homology. For $k = 2$, IsoRank has greater coverage; however this is likely due to IsoRankN having a strict threshold for edge inclusion. Note that as a result of the star-spread approach, all clusters obtained by IsoRankN contain at least two species. Thus IsoRankN does not find paralogs within a species without there existing at least one homolog in another species. Of the 87737 total proteins, IsoRankN is able to find network homologs for 48978 (55.8%), more than any technique but IsoRank. When restricted to clusters containing at least three species, *i.e.* the multiple-alignment case, IsoRankN predicts the most clusters.

We further measure as in [42] coverage by the enrichment of predicted groups with respect to known ontology as derived from GO and KEGG. We find that IsoRankN enriches more GO and KEGG categories in every species, with a lower overall p-value (computed by GO TermFinder [12]), than any other technique (see Table 12.3).

Ordering. While we chose a particular order of genomes in the multiple alignment to report our general results, we also include results on different orderings of genomes and

Table 12.3: Comparative GO/KEGG enrichment performance

Species	IsoRankN	IsoRank	Græmlin$_{1K}$	Græmlin$_{2K}$	NB-M$^\sharp$
Total	**712/2490**	537/1760	296/772	432/1010	107/261
p-value*	**1.28 e-90**	1.31 e-68	5.47 e-38	6.87 e-54	2.19 e-14
Human	**632/2200**	478/1551	194/545	272/811	66/182
Mouse	**605/2124**	383/1371	191/538	268/794	65/178
Fly	**574/1787**	398/924	208/533	261/771	41/135
Worm	**552/1698**	376/901	104/257	140/389	32/124
Yeast	**368/938**	257/554	208/486	137/316	45/136

The number of GO/KEGG categories enriched by each method. Note that the boldface numbers represent the best performance w.r.t. each row.

*As computed by GO TermFinder. We remark that this excludes those proteins tagged IEA (inferred from electronic annotation).

$^\sharp$NetworkBLAST-M is denoted NB-M for convenience.

demonstrate that any ordering outperforms other methods (Fig. 12-2). The particular order of genomes used above was chosen to have the minimum mean normalized entropy.

While it may appear that yeast, as the best-annotated network, should be the first network chosen in the star-spread, it is sufficiently dissimilar to the other species as to cause inaccurate network alignments on such a small set of species.

Running time. Given the weighted similarity graph, the star-spread component of IsoRankN (Section 12.1.5, steps 2-5) took under 5 minutes for the 5 eukaryotic networks above. The computation of the graph, given by the original IsoRank (Section 12.1.5, step 1), took approximately 7 hour on a single processor, though can be easily 10-way parallelized. All computations were run on a 64bit 2.4GHz Linux system with 2GB RAM.

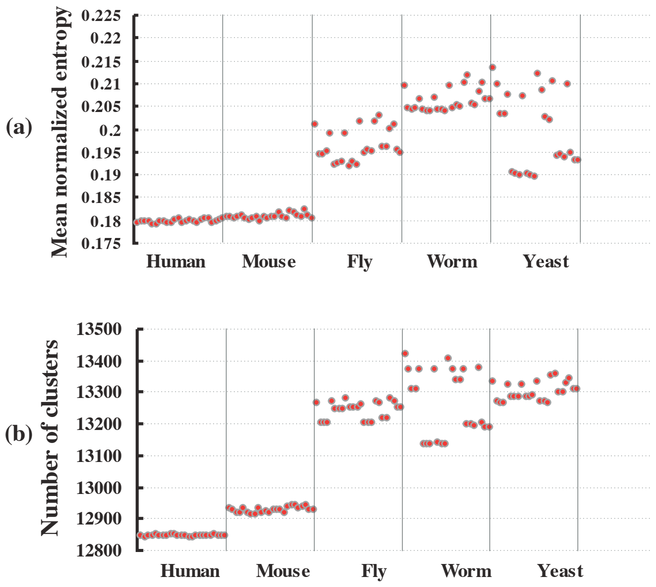


Figure 12-2: The consistency and coverage performance of IsoRankN under species permutations in the star-spread. Each dot represents one of the 120 possible permutations of the five species. (a) and (b) report the consistency and coverage of the network fit as a function of the species first at the center of the star-spread.

12.3 Conclusion

In this chapter we present an efficient method for computing multiple PPI network alignments. Based on spectral clustering on the induced graph of pairwise alignment scores, our program IsoRankN automatically handles noisy and incomplete input data. Our method differs from others in that it does not require training or phylogeny data and seeks vertex-specific rankings in the spectral clustering.

We demonstrate the effectiveness of this technique on the five available eukaryotic PPI networks. Our results suggest that IsoRankN has higher coverage and consistency compared to existing approaches, which should lead to improved functional ortholog prediction.

In future work we plan to more fully explore and evaluate the database of functional orthologs as predicted by IsoRankN. Additionally, it may be possible to modify the star-spread to account for existent gold-standard network homology data, yielding even higher fidelity multiple network alignments.

Part IV

Appendices

Appendix A

Description of Performed Experiments and Microarray Design

A.1 Microarrays

The exact probes used are available at GEO [6], accession number GPL7159. The URL of GEO is: `http://www.ncbi.nlm.nih.gov/geo/`.

A.2 Experiments

Table A.1: Experiments

Month	Day	Year	Chip	Array	Experiment
02	27	2006	A	1	S2Rp.GFPGAL4
02	27	2006	A	2	CG3799
02	27	2006	A	3	oncoCG3799.over
02	27	2006	A	4	CG3799.over
02	27	2006	B	1	S2Rp.GFPGAL4
02	27	2006	B	2	CG3799 RNAi
02	27	2006	B	3	oncoSif.over
02	27	2006	B	4	Sif.over
03	08	2006	A	1	Rho1
03	08	2006	A	2	S2Rp.GFPGAL4
03	08	2006	A	3	Rac1
03	08	2006	A	4	Rac1Rho1
04	01	2006	A	1	S2Rp.GFPGAL4
04	01	2006	A	3	oncoGEF3.over
04	01	2006	A	4	RhoGEF3
04	01	2006	B	1	RhoV14.over
04	01	2006	B	2	RacV12.over
04	01	2006	B	3	RhoGAP92B
04	01	2006	B	4	sif
04	22	2006	A	1	S2Rp.GFPGAL4
04	22	2006	A	2	RhoGAP1A
04	22	2006	A	3	RhoGAP5A
04	22	2006	A	4	RhoGAP16F
04	22	2006	B	1	S2Rp.GFPGAL4
04	22	2006	B	2	RhoGAP69C RNAi
04	22	2006	B	3	RhoGAP71C RNAi
04	22	2006	B	4	RhoGAP84C RNAi
04	24	2006	A	1	S2Rp.GFPGAL4
04	24	2006	A	2	RhoGAP1A.over
04	24	2006	A	3	oncoPbl.over
04	24	2006	A	4	RhoGEF64C.over

Month	Day	Year	Chip	Array	Experiment
05	01	2006	A	1	S2Rp.GFPGAL4
05	01	2006	A	2	sif
05	01	2006	A	3	oncoSif.over
05	01	2006	A	4	sifFL.over
06	10	2006	A	1	GFP May 26
06	10	2006	A	2	RhoGAP50C RNAi
06	10	2006	A	3	RhoGAP54D RNAi
06	10	2006	A	4	RhoGEF64C RNAi
06	13	2006	A	1	GFP May 26
06	13	2006	A	2	RhoGAP93B RNAi
06	13	2006	A	3	RhoGAP100F RNAi
06	13	2006	A	4	CG10188 (RhoGEF) RNAi
05	26	2006	A	1	GFP May 26
05	26	2006	A	2	RhoGAP50C RNAi
05	26	2006	A	3	RhoGAP54D RNAi
05	26	2006	A	4	RhoGEF64C RNAi
05	26	2006	B	1	GFP May 26
05	26	2006	B	2	MTL (GTPase) RNAi
05	26	2006	B	3	RhoBTB (GTPase) RNAi
05	26	2006	B	4	RhoL (GTPase) RNAi
05	28	2006	A	1	GFP May 26
05	28	2006	A	2	RhoGAP93B RNAi
05	28	2006	A	3	RhoGAP100F RNAi
05	28	2006	A	4	CG10188 (RhoGEF) RNAi
05	28	2006	B	1	GFP May 26
05	28	2006	B	2	CG30456 (RhoGEF) RNAi
05	28	2006	B	3	Rac F28L overexpress
05	28	2006	B	4	Rho F30L overexpress
06	21	2006	A	1	GFP June 10
06	21	2006	A	2	Cdep (RhoGEF) RNAi
06	21	2006	A	3	CG14045 (RhoGEF) RNAi
06	21	2006	A	4	CG15611 (RhoGEF)

Month	Day	Year	Chip	Array	Experiment
06	22	2006	A	1	GFP June 10
06	22	2006	A	2	CG30115 (RhoGEF) RNAi
06	22	2006	A	3	p190RhoGAP RNAi
06	22	2006	A	4	pbl (RhoGEF) RNAi
06	22	2006	B	1	GFP June 10
06	22	2006	B	2	RhoGAP19D RNAi
06	22	2006	B	3	RhoGEF4 RNAi
06	22	2006	B	4	trio (RhoGEF) RNAi
12	18	2006	A	1	GFP December 6
12	18	2006	A	2	Cdc42 RNAi G8_A 06/12/06
12	18	2006	A	3	Cdc42 RNAi G8_B 06/12/06
12	18	2006	A	4	Cdc42 RNAi D21_A 06/12/06
12	18	2006	B	1	***GFP December 10
12	18	2006	B	2	Rho1 RNAi ****06/12/06
12	18	2006	B	3	Cdc42 RNAi D21 10/12/06
04	11	2007	A	1	GRP April 11 2007
04	11	2007	A	2	Rac1 RNAi 04/11/2007
04	11	2007	A	3	Rac1 RNAi 04/11/2007
04	11	2007	A	4	Rho1 RNAi 04/11/2007
04	11	2007	B	1	GRP April 11 2007
04	11	2007	B	2	Rho1 RNAi 04/11/2007
04	11	2007	B	3	Rho1 RNAi 04/11/2007
04	11	2007	B	4	Rac1 RNAi 04/11/2007
06	21	2007	A	1	GFP May 12 2007
06	21	2007	A	2	RhoL (GTPase) RNAi 05/12/2007
06	21	2007	A	3	RhoBTB (GTPase) RNAi 05/12/2007
06	21	2007	A	4	RhoGAP1A RNAi 05/12/2007
06	28	2007	A	1	GFP May 12 2007
06	28	2007	A	2	RhoL (GTPase) RNAi 05/12/2007
06	28	2007	A	3	RhoBTB (GTPase) RNAi 05/12/2007
06	28	2007	A	4	RhoGAP1A RNAi 05/12/2007
06	28	2007	B	1	GFP June 25 2007
06	28	2007	B	2	RhoF30L (ACTIVATED RHO)
06	28	2007	B	3	Cdc42Y32A (ACTIVATED CDC42)
06	28	2007	B	4	RacF28L (ACTIVATED Rac)

Month	Day	Year	Chip	Array	Experiment
08	01	2007	A	3	CG3799 RNAi 07/13/2007
08	01	2007	A	4	CdGAPr (GAP) RNAi 07/13/2007
08	01	2007	B	1	CdGAPr (GAP) RNAi 07/13/2007
08	01	2007	B	2	CG3799 RNAi 07/13/2007
08	01	2007	B	3	MTL (GTPase) RNAi 07/13/2007
08	01	2007	B	4	GFP July 13 2007
08	14	2007	A	1	p190RhoGAP
08	14	2007	A	2	RhoGEF4
08	14	2007	A	3	RhoGEF2
08	14	2007	A	4	GFP August 7 2007
08	14	2007	B	1	pbl (RhoGEF)
08	14	2007	B	2	sif (RhoGEF)
08	14	2007	B	3	RacGAP50C (RhoGAP)
08	24	2007	A	1	GFP August 17 2007
08	24	2007	A	2	Rac 1 (”Amplicon #1” Rac1/Rac2)
08	24	2007	A	3	Rac 1 (”Amplicon #2” Rac1 only)
08	24	2007	A	4	Rac 2 (no Rac1)
09	07	2007	A	2	Rac 1 (”Amplicon #2” Rac1 only)
09	07	2007	A	3	Rac 2 (no Rac1) *AUG18_07 sample
09	07	2007	A	4	Rac 1 (”Amplicon #1” Rac1/Rac2)
09	07	2007	B	1	GFP August 7 2007
09	07	2007	B	2	pbl (RhoGEF) *AUG 7_07 sample
09	07	2007	B	3	sif (RhoGEF) *AUG7_07 sample
09	07	2007	B	4	RacGAP50C (RhoGAP) *AUG7_07 sample
09	18	2007	A	1	RhoGAP1A RNAi 05/12/2007
09	18	2007	A	2	RhoL (GTPase) RNAi 05/12/2007
09	18	2007	A	3	RhoBTB (GTPase) RNAi 05/12/2007
09	18	2007	A	4	GFP May 12 2007

Appendix B

Selected Code for Network Inference

B.1 AMPL model code

Below is the AMPL code with data and starting values suppressed as indicated. Complete code is available upon request.

```
set I := 1 .. 35;     #index of features
set K := 1 .. 6;     #index of gtpases
set J := 1 .. 13;     #index of gaps
set JE:= 1 .. 14;     #index of gefs
set L := 1 .. 4;     #index of batch dims
set D := 1 .. 21;     #index of batches

# Input data (dependent variables)
param eta {JE,1..126};  #gef expression levels
param alpha {J,1..126};  #gap expression levels
param chi {D,1..126};  #batch-experiment indicator

param kappa {K,1..126};  #gtpase expression levels

param phi {I,1..126} ; #observed data



var a {I,K};       #A matrix

var c {I,L};       #feature by batchdims

var u {L,D};       #batchdims by batch Rotation

var r {I};         #feature base level

var x {K,JE} >= 0;  #gef-gtpase connections

var y {K,J} >= 0;  #gap-gtpase connections

var g {K} >= 0;    #base deactivation rates



minimize objvar:

     sum {i in I} ( sum {z in 1..126} ((  (phi[i,z] - sum {k in K}

     (a[i,k]*(kappa[k,z]*(1+sum {j in JE} (x[k,j]*eta[j,z]))/

     (1+sum {j in JE} (x[k,j]*eta[j,z]) + g[k] + sum {j in J}

     (y[k,j]*alpha[j,z]))))- r[i]  - sum {l in L} (sum {d in D}

     (c[i,l]*u[l,d]*chi[d,z]))   )^2)) +0.01*sum{k in K}

      (sum{j in JE}  (x[k,j])+ sum{j in J} (y[k,j])+g[k] );



data;


param eta : #gef expression levels
```

```
[suppressed]

param alpha : #gap expression levels
[suppressed]

param chi : #batch-experiment indicator
[suppressed]

param kappa : #gtpase expression levels
[suppressed]

var x : #starting x
[suppressed]

var y : #starting y
[suppressed]

var c : #starting c
[suppressed]

var u : #starting u
[suppressed]

var a : #starting a
[suppressed]
```

Appendix C

Derivations of Population Genetic Results

C.1 Some facts we need

C.1.1 Wright-Fischer Model: Multinomial Sampling

Let $\sum x_i = N$, the current population. The next population will be $\sum f_i x_i = \sum(1 + g_i)x_i = M$.

Properties of multinomial sampling when $f_i = 1$:

$$\begin{aligned} \langle \Delta x \rangle &= 0 && \text{(C.1)} \\ \mu_2 = \sigma_i^2 = \left\langle \Delta x_i^2 \right\rangle &= x_i \left(1 - \frac{x_i}{N}\right) && \text{(C.2)} \\ \mu_3 &= x_i \left(1 - \frac{x_i}{N}\right)\left(1 - \frac{2x_i}{N}\right) && \text{(C.3)} \end{aligned}$$

When $f_i \neq 1$:

The raw moments are:

$$\mu_1' = \langle x_i + \Delta x_i \rangle = f_i x_i \tag{C.4}$$

$$\begin{aligned}\mu_2' = \left\langle (x_i + \Delta x_i)^2 \right\rangle = \left\langle x_i^2 + 2x_i \Delta x_i + \Delta x_i^2 \right\rangle \\ = x_i^2 + 2x_i(f_i - 1)x_i + \left\langle \Delta x_i^2 \right\rangle = (2f_i - 1)x_i^2 + \left\langle \Delta x_i^2 \right\rangle\end{aligned} \tag{C.5}$$

Recall that the second central moment in terms of the first two raw moments is

$$\sigma_i^2 = \mu_2 = \mu_2' - {\mu_1'}^2 = (2f_i - 1)x_i^2 + \left\langle \Delta x_i^2 \right\rangle - f_i^2 x_i^2 = \left\langle \Delta x_i^2 \right\rangle - (f_i - 1)^2 x_i^2$$

Note that this of course holds for moments about any mean and so:

$$\mu_2 = \left\langle \Delta x_i^2 \right\rangle - \langle \Delta x_i \rangle^2$$

and

$$\mu_3 = 2{\mu_1'}^3 - 3\mu_1' \mu_2' + \mu_3' = \mu_3 = 2 \langle \Delta x_i \rangle^3 - 3 \langle \Delta x_i \rangle \left\langle \Delta x_i^2 \right\rangle + \left\langle \Delta x_i^3 \right\rangle$$

Thus

$$\langle \Delta x_i \rangle = (f_i - 1)x_i \Rightarrow \sum \langle \Delta x_i \rangle = M - N$$

$$\left\langle \Delta x_i^2 \right\rangle = \sigma_i^2 + \langle \Delta x_i \rangle^2 = f_i x_i \left(1 - \frac{f_i x_i}{M} \right) + (f_i - 1)^2 x_i^2$$

$$\begin{aligned}
\left\langle \Delta x_i^3 \right\rangle &= f_i x_i \left(1 - \frac{f_i x_i}{M}\right)\left(1 - \frac{2 f_i x_i}{M}\right) - 2 \left\langle \Delta x_i \right\rangle^3 + 3 \left\langle \Delta x_i \right\rangle \left\langle \Delta x_i^2 \right\rangle \\
&= f_i x_i \left(1 - \frac{f_i x_i}{M}\right)\left(1 - \frac{2 f_i x_i}{M}\right) - 2 (f_i - 1)^3 x_i^3 \\
&\quad + 3 (f_i - 1) x_i \left(f_i x_i \left(1 - \frac{f_i x_i}{M}\right) + (f_i - 1)^2 x_i^2 \right) \\
&= f_i x_i \left(1 - \frac{f_i x_i}{M}\right)\left(1 - \frac{2 f_i x_i}{M}\right) + (f_i - 1)^3 x_i^3 \\
&\quad + 3 (f_i - 1) x_i \left(f_i x_i \left(1 - \frac{f_i x_i}{M}\right)\right) \\
&= f_i x_i \left(1 - \frac{f_i x_i}{M}\right)\left(1 - \frac{2 f_i x_i}{M} + 3 (f_i - 1) x_i\right) + (f_i - 1)^3 x_i^3
\end{aligned}$$

C.1.2 Shannon Entropy

$$S(x) = - \sum_i \frac{x_i}{N} \log\left(\frac{x_i}{N}\right)$$

$$\frac{\partial S(x)}{\partial x_i} = - \frac{1}{N}\left(\log\left(\frac{x_i}{N}\right) + 1\right)$$

$$\frac{\partial^2 S(x)}{\partial x_i \partial x_j} = - \frac{\delta_{ij}}{N x_i}$$

$$\frac{\partial^3 S(x)}{\partial x_i \partial x_j \partial x_k} = \frac{\delta_{ij}\delta_{jk}}{N x_i^2}$$

C.1.3 $-\sum (1 - p_i) \log (1 - p_i)$

$$Z(x) \equiv \sum_i - (1 - p_i) \log (1 - p_i) = - \sum \left(1 - \frac{x_i}{N}\right) \log\left(1 - \frac{x_i}{N}\right) \qquad \text{(C.6)}$$

$$\frac{\partial Z(x)}{\partial x_i} = \frac{1}{N}\left(\log\left(1 - \frac{x_i}{N}\right) + 1\right)$$

$$\frac{\partial^2 Z(x)}{\partial x_i \partial x_j} = - \frac{\delta_{ij}}{N(N - x_i)}$$

$$\frac{\partial Z(x)}{\partial p_i} = \log\left(1 - p_i\right) + 1$$

$$\frac{\partial^2 Z(x)}{\partial x_i \partial x_j} = -\frac{\delta_{ij}}{1 - p_i}$$

C.2 Derivations

C.2.1 Neutral Wright-Fischer Model

Below is the derivation for equation 9.3:

$$\begin{aligned}
\langle \Delta T(x) \rangle &= N \left\langle S(x + \Delta x) \right\rangle \qquad \text{(C.7)} \\
&= N \left\langle \sum_{j=0}^{\infty} \left[\frac{1}{j!} \left(\Delta x \cdot \nabla_{x'} \right)^j S(x') \right]_{x'=x} \right\rangle \\
&\approx N \left\langle S(x) + \sum_{x_i \neq 0} \frac{\partial S(x)}{\partial x_i} \Delta x_i + \frac{1}{2} \sum_{x_i, x_j \neq 0} \frac{\partial^2 S(x)}{\partial x_i \partial x_j} \Delta x_i \Delta x_j \right\rangle \\
&= N S(x) + N \sum_{x_i \neq 0} \frac{\partial S(x)}{\partial x_i} \left\langle \Delta x_i \right\rangle + \frac{N}{2} \sum_{x_i, x_j \neq 0} \frac{\partial^2 S(x)}{\partial x_i \partial x_j} \left\langle \Delta x_i \Delta x_j \right\rangle \\
&= N S(x) - \frac{N}{2} \sum_{x_i \neq 0} \frac{\langle \Delta x_i^2 \rangle}{N x_i} \\
&= N S(x) - \frac{N}{2} \sum \frac{x_i \left(1 - \frac{x_i}{N}\right)}{N x_i} \\
&= N S(x) - \frac{1}{2} \sum \left(1 - \frac{x_i}{N}\right) \\
&= N S(x) - \frac{k - 1}{2}
\end{aligned}$$

C.2.2 Neutral Moran Model

Below is the derivation for equation 9.6:

$$
\begin{aligned}
\langle T(x)+\Delta T(x)\rangle &= N\left\langle \sum_{j=0}^{\infty}\left[\frac{1}{j!}\left(\Delta x\cdot\nabla_{x'}\right)^{j} S(x')\right]_{x'=x}\right\rangle \\
&= NS(x) - N\sum_{k}\sum_{n=1}^{\infty}\frac{(2n-2)!\,\langle\Delta x_i^{2n}\rangle}{(2n)!Nx_i^{2n-1}} \\
&= NS(x) - \sum_{k}\sum_{n=1}^{\infty}\frac{2x_i(1-\frac{x_i}{N})}{(2n)(2n-1)Nx_i^{2n}} \\
&= NS(x) - \sum_{k}\frac{2(1-\frac{x_i}{N})}{N}\sum_{n=1}^{\infty}\frac{1}{\binom{2n}{2}x_i^{2n-2}} \\
&= NS(x) - \sum_{k}\frac{(1-\frac{x_i}{N})}{N}x_i^2\left(\left(1-\frac{1}{x_i}\right)\log\left(1-\frac{1}{x_i}\right)\right. \\
&\qquad \left. +\left(1+\frac{1}{x_i}\right)\log\left(1+\frac{1}{x_i}\right)\right) \\
&= NS(x) - \sum_{k}\frac{(1-\frac{x_i}{N})}{N}x_i\left((x_i-1)\log\left(1-\frac{1}{x_i}\right)\right. \\
&\qquad \left. +(x_i+1)\log\left(1+\frac{1}{x_i}\right)\right) \\
&= NS(x) - \sum\frac{1-\frac{x_i}{N}}{N}\left(1+O\left(\frac{1}{x_i^2}\right)\right) \\
&= NS(x) - \frac{k-1}{N} + O\left(\sum\frac{1}{x_i^2}\right)
\end{aligned}
\tag{C.8}
$$

$$
\langle\Delta S(x)\rangle \approx -\frac{k-1}{N^2}
$$

C.2.3 Neutral Diffusion Model

Recall here that

$$\frac{\partial f}{\partial t} = \frac{1}{2}\sum_{i=1}^{k-1}\frac{\partial^2 f}{\partial x_i^2}\left\{x_i\left(1-x_i\right)\right\} - \frac{1}{2}\sum_{i,j<k}\frac{\partial^2 f}{\partial x_i \partial x_j}\left\{x_i x_j\right\}$$

We use the result [25, (Ewens) 4.83, page 154], that

$$\begin{aligned}\frac{d}{dt}h(t) &= \mathbb{E}_t\left[\sum a_i\left(x_1,\ldots,x_k\right)\frac{\partial g}{\partial x_i} + \frac{1}{2}\sum b_i\left(x_1,\ldots,x_k\right)\frac{\partial^2 g}{\partial x_i^2}\right. \\ &\quad \left. + \sum\sum c_{ij}\left(x_1,\ldots,x_k\right)\frac{\partial^2 g}{\partial x_i \partial x_j}\right]\end{aligned} \tag{C.9}$$

In the neutral case,

$$\begin{aligned}a\left(x_1,\ldots,x_k\right) &= 0 \\ b_i\left(x_1,\ldots,x_k\right) &= x_i(1-x_i) \\ c_{ij}\left(x_1,\ldots,x_k\right) &= -x_i x_j\end{aligned}$$

Thus:

$$\begin{aligned}
\frac{d}{dt}S(t) &= \mathbb{E}_t\left[\sum a_i\left(x_1,\ldots,x_k\right)\frac{\partial S(x)}{\partial x_i}+\frac{1}{2}\sum b_i\left(x_1,\ldots,x_k\right)\frac{\partial^2 S(x)}{\partial x_i^2}\right. \qquad \text{(C.10)}\\
&\quad \left.+\sum\sum c_{ij}\left(x_1,\ldots,x_k\right)\frac{\partial^2 S(x)}{\partial x_i\partial x_j}\right]\\
&= \mathbb{E}_t\left[\frac{1}{2}\sum x_i(1-x_i)\frac{\partial^2 S(x)}{\partial x_i^2}-\sum\sum x_ix_j\frac{\partial^2 S(x)}{\partial x_i\partial x_j}\right]\\
&= \mathbb{E}_t\left[\frac{1}{2}\sum_{x_i\neq 0} x_i(1-x_i)\frac{-1}{x_i}\right]\\
&= \mathbb{E}_t\left[\frac{1-k(x)}{2}\right]\\
&= \int_{\sum x_i\leq 1}\frac{1-k(x)}{2}f(x;t)dx
\end{aligned}$$

Where $k(x)$ is the number of nonzero alleles at point x.

Bibliography

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] Reka Albert. Scale-free networks in cell biology. *J Cell Sci*, 118(21):4947–4957, 2005.

[3] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:475–486, 2006.

[4] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29, 2000.

[5] Marina Bakay, Rehannah Borup Yi-Wen Chen and, Po Zhao, Kanneboyina Nagaraju, and Eric P Hoffman. Sources of variability and effect of experimental approach on expression profiling data interpretation. *BMC Bioinformatics*, 3(4), 2002.

[6] Tanya Barrett, Dennis B. Troup, Stephen E. Wilhite, Pierre Ledoux, Dmitry Rudnev, Carlos Evangelista, Irene F. Kim, Alexandra Soboleva, Maxim Tomashevsky, and Ron Edgar. NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucl. Acids Res.*, 35(suppl_1):D760–765, 2007.

[7] Katia Basso, Adam A Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, 37(4):382–390, April 2005.

[8] Joseph A. Baur, Kevin J. Pearson, Nathan L. Price, Hamish A. Jamieson, Carles Lerin, Avash Kalra, Vinayakumar V. Prabhu, Joanne S. Allard, Guillermo Lopez-Lluch, Kaitlyn Lewis, Paul J. Pistell, Suresh Poosala, Kevin G. Becker, Olivier Boss, Dana Gwinn, Mingyi Wang, Sharan Ramaswamy, Kenneth W. Fishbein, Richard G. Spencer, Edward G. Lakatta, David Le Couteur, Reuben J. Shaw, Placido Navas, Pere Puigserver, Donald K. Ingram, Rafael de Cabo, and David A. Sinclair. Resveratrol improves health and survival of mice on a high-calorie diet. *Nature*, 444(7117):337–342, 2006.

[9] Michael Baym, Chris Bakal, Norbert Perrimon, and Bonnie Berger. High-resolution modeling of cellular signaling networks. In Martin Vingron and Limsoon Wong, editors, *RECOMB*, volume 4955 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2008.

[10] Johannes Berg and Michael Lässig. Cross-species analysis of biological networks by Bayesian alignment. *Proceedings of the National Academy of Sciences*, 103(29):10967–10972, 2006.

[11] B. M. Boldstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, 19(2):185–193, 2003.

[12] Elizabeth I. Boyle, Shuai Weng, Jeremy Gollub, Heng Jin, David Botstein, J. Michael Cherry, and Gavin Sherlock. GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, 2004.

[13] W. S. Branham, C. D. Melvin, T. Han, V. G. Desai, C. L. Moland, A. T. Scully, and J. C. Fuscoe. Elimination of laboratory ozone leads to a dramatic improvement in the reproducibility of microarray gene expression measurements. *BMC Biotechnol.*, 7:8, 2007.

[14] G. E. Briggs and J. B. S. Haldane. A note on the kinetics of enzyme action. *Biochem. J.*, 19:339–339, 1925.

[15] D. R. Brooks and E. O. Wiley. *Evolution as entropy : toward a unified theory of biology.* University of Chicago Press, Chicago, 1986.

[16] Samuel R. Buss and Peter Clote. Solving the fisher-wright and coalescence problems with a discrete markov chain analysis. *Advances in Applied Probability*, 36(4):1175–1197, 2004.

[17] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Stat. Assoc.*, 74:829–836, 1979.

[18] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[19] William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.

[20] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[21] Marie Csete and John Doyle. Bow ties, metabolism and disease. *Trends in Biotechnology*, 22(9):446–450, 2004.

[22] John D'Errico. *MATLAB Gridfit Tool*. `http://www.mathworks.com/matlabcentral/fileexchange/8998`.

[23] Janusz Dutkowski and Jerzy Tiuryn. Identification of functional modules from conserved ancestral protein protein interactions. *Bioinformatics*, 23(13):i149–158, 2007.

[24] W. J. Ewens. The mean time for absorption in a process of genetic type. *Journal of the Australian Mathematical Society*, 3(03):375–383, 1963.

[25] W. J. Ewens. *Mathematical Population Genetics*, volume 27 of *Interdisciplinary Applied Mathematics*. Springer Science+Business Media, New York, second edition, 2004.

[26] T. L. Fare, E. M. Coffey, H. Dai, Y. D. He, D. A. Kessler, K. A. Kilian, J. E. Koch, E. LeProust, M. J. Marton, M. R. Meyer, R. B. Stoughton, G. Y. Tokiwa, and Y. Wang. Effects of atmospheric ozone on microarray data quality. *Anal. Chem.*, 75:4672–4675, Sep 2003.

[27] Stanley Fields and Ok-Kyo Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340:245–246, July 1989.

[28] R. A. Fisher. *The Genetical Theory of Natural Selection*. Oxford University Press, 2nd edition, April 1958.

[29] Jason Flannick, Antal Novak, Chuong Do, Balaji Srinivasan, and Serafim Batzoglou. Automatic parameter learning for multiple network alignment. *Research in Computational Molecular Biology*, pages 214–231, 2008.

[30] Robert Fourer, David M. Gay, and Brian W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, November 2002.

[31] Adam Friedman and Norbert Perrimon. Genetic screening for signal transduction in the era of network biology. *Cell*, 128:225–231, 2007.

[32] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, February 2004.

[33] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *J. of Computational Biology*, 7(3-4):601–620, 2000.

[34] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[35] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. McDaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath, N. Ioime, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. DaSilva, J. Zhong, C. A. Stanyon, Jr. Finley, R. L., K. P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shimkets, M. P. McKenna, J. Chant, and J. M. Rothberg. A protein interaction map of drosophila melanogaster. *Science*, 302(5651):1727–1736, 2003.

[36] Leonid Gitlin and Raul Andino. Nucleic Acid-Based Immune System: the Antiviral Potential of Mammalian RNA Silencing. *J. Virol.*, 77(13):7159–7165, 2003.

[37] Jing-Dong J Han, Denis Dupuy, Nicolas Bertin, Michael E Cusick, and Marc Vidal. Effect of sampling on topology predictions of protein-protein interaction networks. *Nat Biotech*, 23(7):839–844, 2005.

[38] Hailiang Huang, Bruno M Jedynak, and Joel S Bader. Where have all the interactions gone? Estimating the coverage of two-hybrid protein interaction maps. *PLoS Comput Biol*, 3(11):e214, 11 2007.

[39] T. J. P. Hubbard, B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, K. Howe, N. Johnson, A. Kahari, D. Keefe, F. Kokocinski,

E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal, and E. Birney. Ensembl 2007. *Nucl. Acids Res.*, page gkl996, 2006.

[40] Timothy R. Hughes, Matthew J. Marton, Allan R. Jones, Christopher J. Roberts, Roland Stoughton, Christopher D. Armour, Holly A. Bennett, Ernest Coffey, Hongyue Dai, Yudong D. He, Matthew J. Kidd, Amy M. King, Michael R. Meyer, David Slade, Pek Y. Lum, Sergey B. Stepaniants, Daniel D. Shoemaker, Daniel Gachotte, Kalpana Chakraburtty, Julian Simon, Martin Bard, and Stephen H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[41] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8):4569–4574, 2001.

[42] Maxim Kalaev, Vineet Bafna, and Roded Sharan. Fast and accurate alignment of multiple protein networks. *Research in Computational Molecular Biology*, pages 246–256, 2008.

[43] Minoru Kanehisa and Susumu Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucl. Acids Res.*, 28(1):27–30, 2000.

[44] Brian P. Kelley, Roded Sharan, Richard M. Karp, Taylor Sittler, David E. Root, Brent R. Stockwell, and Trey Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11394–11399, 2003.

[45] Brian P. Kelley, Bingbing Yuan, Fran Lewitter, Roded Sharan, Brent R. Stockwell, and Trey Ideker. PathBLAST: a tool for alignment of protein interaction networks. *Nucl. Acids Res.*, 32(suppl_2):W83–88, 2004.

[46] Motoo Kimura. Solution of a Process of Random Genetic Drift with a Continuous Model. *Proceedings of the National Academy of Sciences of the United States of America*, 41(3):144–150, 1955.

[47] Mehmet Koyutürk, Yohan Kim, Umut Topkara, Shankar Subramaniam, Wojciech Szpankowski, and Ananth Grama. Pairwise alignment of protein interaction networks. *Journal of Computational Biology*, 13(2):182–199, 2006. PMID: 16597234.

[48] Nevan J. Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P. Tikuisis, Thanuja Punna, JosÃ©M. PeregrÃ-n Alvarez, Michael Shales, Xin Zhang, Michael Davey, Mark D. Robinson, Alberto Paccanaro, James E. Bray, Anthony Sheung, Bryan Beattie, Dawn P. Richards, Veronica Canadien, Atanas Lalev, Frank Mena, Peter Wong, Andrei Starostine, Myra M. Canete, James Vlasblom, Samuel Wu, Chris Orsi, Sean R. Collins, Shamanta Chandran, Robin Haw, Jennifer J. Rilstone, Kiran Gandi, Natalie J. Thompson, Gabe Musso, Peter St Onge, Shaun Ghanny, Mandy H. Y. Lam, Gareth Butland, Amin M. Altaf-Ul, Shigehiko Kanaya, Ali Shilatifard, Erin O'Shea, Jonathan S. Weissman, C. James Ingles, Timothy R. Hughes, John Parkinson, Mark Gerstein, Shoshana J. Wodak, Andrew Emili, and Jack F. Greenblatt. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

[49] Edo Kussell and Stanislas Leibler. Phenotypic Diversity, Population Growth, and Information in Fluctuating Environments. *Science*, 309(5743):2075–2078, 2005.

[50] Justin Lamb, Emily D. Crawford, David Peck, Joshua W. Modell, Irene C. Blat, Matthew J. Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N. Ross, Michael Reich, Haley Hieronymus, Guo Wei, Scott A. Armstrong, Stephen J. Haggarty, Paul A. Clemons, Ru Wei, Steven A. Carr, Eric S. Lander, and Todd R. Golub. The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. *Science*, 313(5795):1929–1935, 2006.

[51] Jennie E Larkin, Bryan C Frank, Haralambos Gavras, Razvan Sultana, and John Quackenbush. Independence and reproducibility across microarray platforms. *Nature Methods*, 2(5):337–344, 2005.

[52] Mei-Ling Ting Lee, Frank C. Kuo, G. A. Whitmore, and Jeffrey Sklar. Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18):9834–9839, 2000.

[53] Chen Li, Shunichi Suzuki, Qi-Wei Ge, Mitsuru Nakata, Hiroshi Matsuno, and Satoru Miyano. Structural modeling and analysis of signaling pathways based on petri nets. *J. Bioinformatics and Computational Biology*, 4(5):1119–1140, 2006.

[54] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–258, June 2009.

[55] Erez Lieberman, Christoph Hauert, and Martin A. Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, 2005.

[56] D J Lipman, S F Altschul, and J D Kececioglu. A tool for multiple sequence alignment. *Proceedings of the National Academy of Sciences of the United States of America*, 86(12):4412–4415, 1989.

[57] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathemtical Statistics and Probability*, pages 281–297, 1967.

[58] Adam A. Margolin, Kai Wang, Wei Keat Lim, Manjunath Kustagi, Ilya Nemenman, and Andrea Califano. Reverse engineering cellular networks. *Nature Protocols*, 1:662–671, June 2006.

[59] The MathWorks. *MATLAB Bioinformatics Toolbox.*
`http://www.mathworks.com/products/bioinfo/`.

[60] Lisa R. Matthews, Philippe Vaglio, Jrme Reboul, Hui Ge, Brian P. Davis, James Garrels, Sylvie Vincent, and Marc Vidal. Identification of Potential Interaction Networks Using Sequence-Based Searches for Conserved Protein-Protein Interactions or Interologs. *Genome Research*, 11(12):2120–2126, 2001.

[61] Gunter Meister and Thomas Tuschl. Mechanisms of gene silencing by double-stranded rna. *Nature*, 431(7006):343–349, 2004.

[62] L. Michaelis and M. Menten. Die kinetik der invertinwirkung. *Biochem. Z.*, 49:333–369, 1913.

[63] Frits Michiels, Gaston G. M. Habets, Jord C. Stam, Rob A. van der Kammen, and John G. Collard. A role for rac in tiaml-induced membrane ruffling and invasion. *Nature*, 375:338–340, 1995.

[64] Gopa R. Mishra, M. Suresh, K. Kumaran, N. Kannabiran, Shubha Suresh, P. Bala, K. Shivakumar, N. Anuradha, Raghunath Reddy, T. Madhan Raghavan, Shalini Menon, G. Hanumanthu, Malvika Gupta, Sapna Upendran, Shweta

Gupta, M. Mahesh, Bincy Jacob, Pinky Mathew, Pritam Chatterjee, K. S. Arun, Salil Sharma, K. N. Chandrika, Nandan Deshpande, Kshitish Palvankar, R. Raghavnath, R. Krishnakanth, Hiren Karathia, B. Rekha, Rashmi Nayak, G. Vishnupriya, H. G. Mohan Kumar, M. Nagini, G. S. Sameer Kumar, Rojan Jose, P. Deepthi, S. Sujatha Mohan, T. K. B. Gandhi, H. C. Harsha, Krishna S. Deshpande, Malabika Sarker, T. S. Keshava Prasad, and Akhilesh Pandey. Human protein reference database–2006 update. *Nucl. Acids Res.*, 34(suppl_1):D411–414, 2006.

[65] I. Nachman, A. Regev, and N. Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20(suppl. 1):i248–256, 2004.

[66] Dana Peer, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17:S214–S224, 2001.

[67] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, April 2005.

[68] Erik Sahai and Christopher J. Marshall. Rho-gtpases and cancer. *Nat Rev Cancer*, 2(2):133–142, 2002.

[69] Andreas Schlicker, Francisco Domingues, Jorg Rahnenfuhrer, and Thomas Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7(1):302, 2006.

[70] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[71] E. Segal, R. Yelensky, A. Kaushal, T. Pham, A. Regev, D. Koller, and N. Friedman. GeneXPress: A visualization and statistical analysis tool for gene expression and sequence data. In *Proceedings of the 11th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2004.

[72] C.E. Shannon. *An algebra for theoretical genetics.* Massachusetts Institute of Technology, Dept. of Mathematics, 1940.

[73] Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768, 2008.

[74] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. BioGRID: a general repository for interaction datasets. *Nucl. Acids Res.*, 34(suppl_1):D535–539, 2006.

[75] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, 102(43):15545–15550, 2005.

[76] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl. Acids Res.*, 22(22):4673–4680, 1994.

[77] Peter Uetz, Loic Giot, Gerard Cagney, Traci A. Mansfield, Richard S. Judson, James R. Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, Alia Qureshi-Emili, Ying Li, Brian Godwin, Diana Conover, Theodore Kalbfleisch, Govindan Vijayadamodar, Meijia Yang, Mark Johnston, Stanley Fields, and Jonathan M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.

[78] G. A. Watterson. Some theoretical aspects of diffusion theory in population genetics. *The Annals of Mathematical Statistics*, 33(3):939–957, 1962.

[79] Sewall Wright. The Differential Equation of the Distribution of Gene Frequencies. *Proceedings of the National Academy of Sciences of the United States of America*, 31(12):382–389, 1945.

[80] Ioannis Xenarios, Lukasz Salwinski, Xiaoqun Joyce Duan, Patrick Higney, Sul-Min Kim, and David Eisenberg. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucl. Acids Res.*, 30(1):303–305, 2002.

[81] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4), February 2002.

[82] Hubert P. Yockey. Information theory, evolution and the origin of life. *Inf. Sci.*, 141(3-4):219–225, 2002.

[83] Haiyuan Yu, Nicholas M. Luscombe, Hao Xin Lu, Xiaowei Zhu, Yu Xia, Jing-Dong J. Han, Nicolas Bertin, Sambath Chung, Marc Vidal, and Mark Gerstein. Annotation Transfer Between Genomes: Protein–Protein Interologs and Protein–DNA Regulogs. *Genome Research*, 14(6):1107–1118, 2004.