Supplementary Materials: Estimating the Tempo and Mode of Gene Family Evolution from Comparative Genomic Data

Matthew W. Hahn**
Center for Population Biology
University of California
Davis, CA 95616, USA

Tijl De Bie*
ISIS Research Group
University of Southampton
Southampton, SO17 1BJ, UK

Jason E. Stajich
Department of Molecular Genetics and Microbiology
Duke University
Durham, NC 27708, USA

Chi Nguyen
Department of Computer Science
University of California
Davis, CA 95616,USA

Nello Cristianini
Department of Statistics
University of California
Davis, CA 95616, USA

April 13, 2005

*These authors contributed equally to this work

Running title: Gene family evolution

Keywords: duplication, deletion, natural selection, expansion, birth-death

*Corresponding author: Matthew W. Hahn. 2320 Storer Hall. Center for Population Biology. University of California. Davis, CA 95616. Phone: (530)752-4253. E-mail: mwhahn@ucdavis.edu

Methods

Here we discuss the details of the methods. We first explain how probabilistic graphical models (PGM) can be of use in the study of gene family evolutions. Then, we discuss how they can be used to perform hypothesis tests for gene families given their size in a number of species for which the phylogeny is known. Lastly, we explain how the gene families have been determined for the five *Saccharomyces* genomes used in this paper.

Studying gene family evolution over a phylogeny using Probabilistic Graphical Models

A graphical model based on the phylogenetic tree

Given the BD model for the evolution of gene family size, the topology of the phylogenetic tree, the branch lengths of this tree, and the evolutionary rate parameter λ, it is possible to calculate the conditional likelihood (conditioned on root species family size $R = r$) of any tree with *fully specified* gene family sizes at all descendant species: $\Sigma = \sigma$ (where Σ is used as the set of random variables S for all species in the phylogeny; σ contains the particular assignments s to these random variables S in Σ). Indeed, this conditional likelihood can most conveniently be written in the following factorized way:

$$P(\Sigma = \sigma | R = r) = \prod_{S \in (\Sigma \setminus \mathcal{L}(R))} \left[\prod_{C \in \mathcal{C}(S)} P(C = c | S = s) \right], \tag{1}$$

where $P(C = c|S = s)$ is given by the BD model, with t the time elapsed between C and S (suppressed in the above equation for readability). We use the notation R for the random variable that is the root species' family size; $\mathcal{C}(S)$ for the set of child species of a species whose gene family size is given by

the random variable S (see Figure 2), and $\mathcal{L}(S)$ for the set of random variables corresponding to the leaf species descending from the species whose gene family size is denoted by S. (Note that no probability distribution for the root species' family size is given. This is why we can only compute the likelihood conditioned on the value r for R.)

The factorization in Eq. (1) can be made more explicit by representing the probability distribution as a tree-structured directed PGM as in Figure 1, where every node corresponds to one of the nodes in the phylogeny, and the conditional probabilities associated with the arrows are given by the BD transition probabilities. In this PGM, the root species corresponds to the *root* of the tree, the leaf species correspond to the *leaf nodes*, and a child species of another species corresponds to the *child node* of that other species' node.

We are thus able to compute the conditional likelihood of a given assignment of gene family sizes over an entire tree descending from a certain root species by simply computing the product of all BD probabilities corresponding the branches in the graphical model of Figure 1. This assumes that also the gene family sizes of the species corresponding to non-leaf nodes are specified.

Computing the conditional likelihood

However, we are generally not interested in the conditional likelihood of a *completely specified* phylogenetic tree $\Sigma = \sigma$ conditioned on $R = r$. Rather, we want to know the likelihood of observing the data for the leaf nodes $\mathcal{L}(R) = \ell(R)$ only—because this is what can be observed—conditioned on the root species' size (here, $\mathcal{L}(R)$ is the set of leaf random variables that represent the family sizes of the leaf species descending from the root species; $\ell(R)$ is a particular given assignment of these gene family sizes). This can be computed by averag-

ing over all possible assignments of unspecified internal nodes (except for the root node), a process called *marginalization* in the graphical models literature. Note that we still have to condition on the root species' family size because no probability distribution is specified for it. However, we will see later how such conditional likelihoods still allow for the computation of very good estimates of the p-values, which is our main goal here.

While the marginalization step involves averaging over a very large set of internal node assignments, namely a number exponential in the size of the tree, it is a nontrivial result of the theory of PGMs that these computations can be performed in a very efficient way, by resorting to algorithms referred to as the message-passing algorithm or the sum-product algorithm (see Jordan, in preparation; Pearl 1986, 1988). Concretely, our algorithm proceeds by recurring up to the root (i.e. until $S = R$):

$$
\begin{aligned}
&\text{For } \; S \in (\Sigma \backslash \mathcal{L}(R)) : \\
P(\mathcal{L}(S) = \ell(S) | S = i) \quad &= \quad \prod_{C \in \mathcal{C}(S)} P(\mathcal{L}(C) = \ell(C) | S = i), \\
&\text{and for } \; C \in \mathcal{C}(S) : \\
P(\mathcal{L}(C) = \ell(C) | S = i) \quad &= \quad \sum_{j} P(C = j | S = i) \cdot P(\mathcal{L}(C) = \ell(C) | C = j), \qquad (2) \\
&\text{with start conditions:} \\
&\text{for } \; S \in \mathcal{L}(R) : \\
P(\mathcal{L}(S) = \ell(S) | S = i) \quad &= \quad \begin{cases} 1 & \text{if } i \text{ is the given gene family size of node } S \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

The complexity of this algorithm is only linear in the size of the phylogenetic tree. Note that for a practical implementation of the algorithm, we need to

make the assumption that the maximal gene family size is limited (if not, the summation in Eq. (2) would contain an infinite number of terms). However, since the conditional probability $P(C = j|S = i)$ associated with the BD model drops off steeply for increasing values of j, this assumption is very reasonable for a large enough upper limit. For the data studied in this paper, an upper limit of 100 was more than sufficient.

Inferring λ

Thus far we have assumed that the parameter λ was given. However, we can learn λ from the data using Expectation Maximization (EM) (Dempster et al. 1977). Specifically, we equate λ to that value that maximizes the conditional log likelihood of the complete set of gene families in our dataset, which is the sum of the conditional log likelihoods of the individual gene families. Here, for each gene family, this log likelihood was conditioned on the root species family size that yields the largest value. (The root family sizes could actually be regarded as additional parameters to be inferred by the EM algorithm, thus motivating this approach.) As can be seen in Figure 2, the optimal value for λ is 0.002 per million years for the *Saccharomyces* phylogeny.

Testing hypotheses about gene family evolution

We have described how, conditioned on the family size of the root species, the likelihood for the family sizes of the given species can be computed. Of course in practice we do not know the actual value of the root node gene family size. To get around this problem, we could make the conservative choice to assign that value to it that leads to the largest conditional likelihood. Still, this is not sufficient to return an interpretable result for hypothesis-testing: a larger root family size will undesirably yield consistently lower likelihoods, since the

conditional probability distribution of a child node's family size is more spread out for a larger parent family size (remember that the square of the variance is proportional to the parent family size: $\mathrm{Var}(X(t)|X(0) = s) = 2s\lambda t$).

Therefore, in order to obtain interpretable results, we need to use p-values corresponding to these likelihoods, and not the likelihoods themselves. These p-values signify the probability of observing a particular assignment of leaf nodes, or an even less likely one. In the next section we describe how this is done.

P-values and conditional p-values for gene family evolution

As discussed in the Methods section of the paper, we can use the PGM to calculate p-values exactly conditioned on each value of the root node. We then choose the largest of these conditional p-values as our *supremum* p-value: an upper bound on the true p-value, which is conditioned on the true but unknown root family size. Unlike the calculation of conditional likelihoods, however, analytic calculation of conditional p-values requires a time exponential in the number of nodes in the tree. For the dataset presented here it was only borderline feasible (several days of computation time using Matlab on a pentium 3GHz processor). Therefore, as a computationally faster alternative, we propose the following approximation method.

The PGM defined by the BD model and the phylogenetic tree structure can be used to randomly generate leaf node gene family data starting from a given root family size. This can be done efficiently thanks to the tree structure of this PGM. Subsequently, for each of these random samples, the likelihood conditioned on its (known) root family size can be computed efficiently using our method described above. In a preprocessing step, we did this a large number of times for each possible value for the root family size (up to size 100). As a re-

sult, we obtained empirical estimates of the null distributions of the conditional likelihoods corresponding to leaf node assignments randomly generated from different root family sizes. Based on these empirical distributions, conditional p-values can be reliably and efficiently estimated by counting the proportion of these random samples that have a conditional likelihood lower than the one for the observed gene family profile, plus half the number of random samples that have an equal conditional likelihood.

The result from the sampling procedure is very accurate when 1000 or more samples are taken. For the *Saccharomyces* data we investigated and using 1000 samples, only four gene families with a true p-value upper bound below 0.01 were not identified with the sampling method (their p-values were estimated to be 0.011 or 0.012). None of the gene families with a true p-value upper bound larger than 0.01 wrongly rejected the BD model. When using $10,000$ samples, which was still much faster than the analytic method, all estimated p-values were in agreement with the analytically computed p-values.

To assess the tightness of the supremum p-value as an upper bound on the p-value, which is important for the sensitivity of our method, we can use the general fact that p-values of random samples from a null model are uniformly distributed on the interval $[0, 1]$ (Casella and Berger 1990). (To be exact, a slightly different definition of p-value has to be adopted here, namely: the probability of observing a likelihood that is lower, plus the probability of observing the same likelihood multiplied by a random number sampled from the uniform distribution on the unit interval. While this definition is less suitable for use in our actual method, it does allow us to assess the tightness of the upper bound on the p-value.) As we can see in Figure 3, the supremum p-values for a set of randomly generated samples is indeed very close to uniformly distributed,

only slightly biased towards larger values. This means that the upper bound is tight enough for our purpose. (If it were not tight, the distribution would be strongly biased towards larger p-values.) Therefore, we can refer to the supremum p-value as the p-value.

For the given yeast gene families, the distribution of the upper bounds on the p-values is given in Figure 4. Again, we note that their distribution is close to uniform, only slightly unbalanced, favoring larger values.

Identifying the unlikely branch

For the gene families that we have identified as unlikely under the BD model (i.e. the ones with a low p-value), we further want to identify the branch in the phylogenetic tree that is responsible for this violation. There are two ways of doing this.

The first way is by computing a p-value corresponding to the likelihood of the pair of subtrees obtained by removing a branch in the phylogenetic tree, and this once for each branch. In effect, the p-value is computed under a reduced model, where total freedom is left to the parent-child transition along that removed branch. If after removing one of the branches the p-value becomes larger than a certain threshold level, this branch may be held responsible for the low p-value of the complete tree. Indeed, since removal of that branch results in a large p-value, the remaining part of the tree cannot reject the BD model. Again, an upper bound on the p-value is computed as the maximal conditional p-value (i.e. the supremum p-value), in this case conditioned on two root values—one for each subtree—instead of on one. As above, the conditional p-values can be estimated by random sampling.

The second method is similar, allowing λ to be optimally tuned (via EM) for each branch of the tree separately and calculating the likelihood of the data under this model. If the likelihood improves significantly when allowing a branch to vary, it is probable that this is the branch responsible for the model violation. Using a likelihood ratio test to compare likelihoods between models with an extra parameter for λ on a single branch and the standard model of one λ for the whole tree, we were able to identify the unlikely branches. As expected, both methods described returned the same predicted branch in all cases investigated in this paper.

Identification of gene families in *Saccharomyces*

Genome sequence assemblies for *S. cerevisiae*, *S. paradoxus*, *S. mikatae*, *S. kudriavzevii*, and *S. bayanus* were obtained from Saccharomyces Genome Database (SGD; see Cliften et al. 2003; Kellis et al. 2003). Predicted proteins were obtained for *S. cerevisiae* from SGD and were generated for the rest of genomes using SNAP (Korf 2004) trained on *S. cerevisae* gene models. These protein sets were searched in an all-against-all fashion using SSEARCH (Pearson 1991; Smith and Waterman 1981) to generate a matrix of pairwise distances based on the normalized bits scores between all protein sequences. This matrix was input for the TRIBE-MCL algorithm (Enright et al. 2002; Van Dongen et al. 2000), that then produced a set of clusters with members from some or all of the input species. TRIBE clustering works by applying a series of expansion and contraction operations to the graph represented by the matrix until equilibrium has been reached; the result is a transformed matrix which is partitioned into individual gene clusters that serve as input into our analyses. Because all genes are clustered together, irrespective of their species of origin, the output is an objective measure of gene family sizes in each species. The assignment of

protein function for each gene was done using HMMER (Eddy 1998) and the Pfam database (Bateman et al. 2004) and parsed with the Bioperl SearchIO module (Stajich et al. 2002). These methods result in 3517 gene families that have representatives in all of the *Saccharomyces* genomes.

Supplementary figure legends

Figure 1: The graphical model associated with the *Saccharomyces* phylogeny. The root is denoted by the symbol R; a generic node is referred to as S; the set of leaf nodes of a node S (or R) is denoted as $\mathcal{L}(S)$ (or $\mathcal{L}(R)$); the set of child nodes of a node labeled S is indicated by $\mathcal{C}(S)$.

Figure 2: The log probability of the dataset as a function of the parameter λ. The optimum lies at $\lambda = 0.002$ (per million years).

Figure 3: The cumulative distribution of the upper bound on the p-values of a set of randomly generated gene families that is similar to the given dataset (one random gene family is generated for every given gene family, with root node equal to the optimal root node for that given gene family). Note that the cumulative distribution is close to linearly increasing on the interval $[0, 1]$, which corresponds to a uniform probability density function. This is indeed the theoretical distribution a p-value should follow.

Figure 4: The cumulative distribution of the supremum p-values for the gene families studied in this paper.

Additional figures

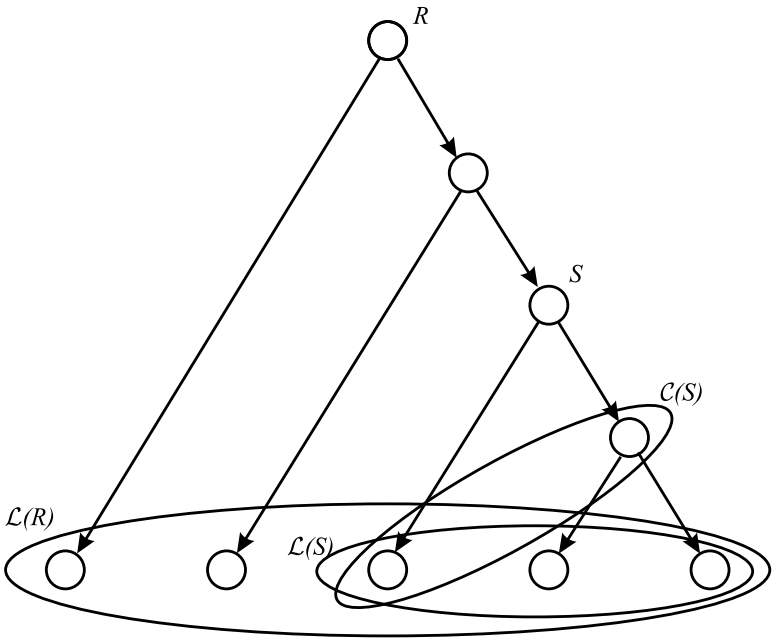

Figure 1

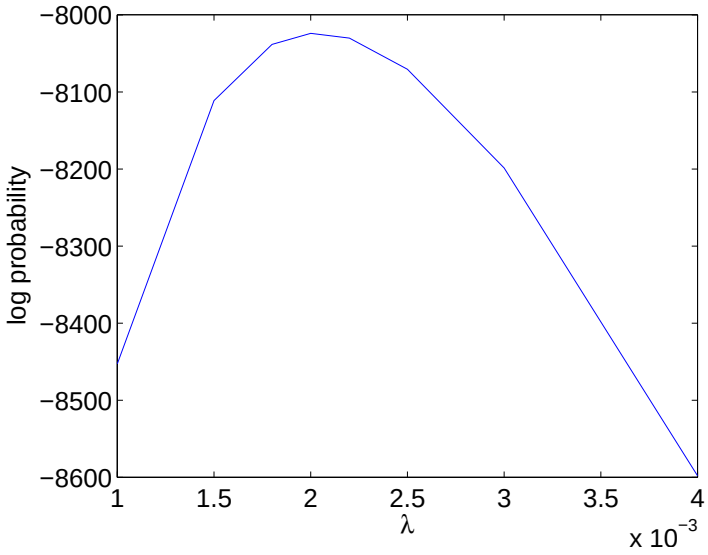
−8000
−8100
−8200
−8300
−8400
−8500
−8600
log probability
1
1.5
2
2.5
3
3.5
4
λ
x 10^−3

Figure 2

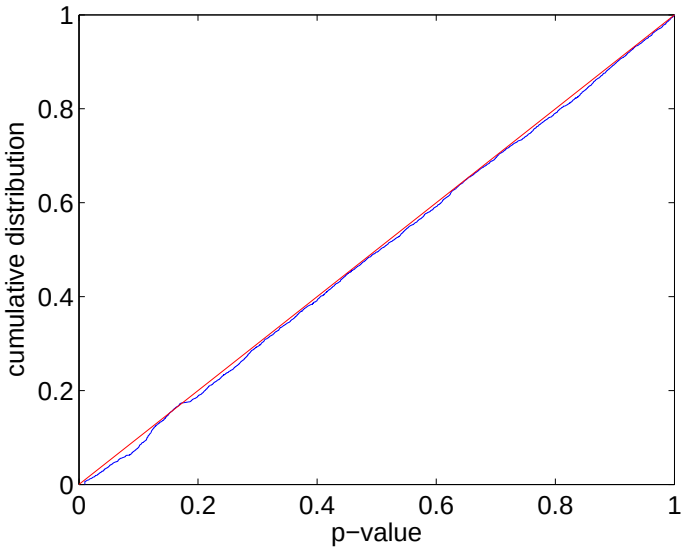

Figure 3

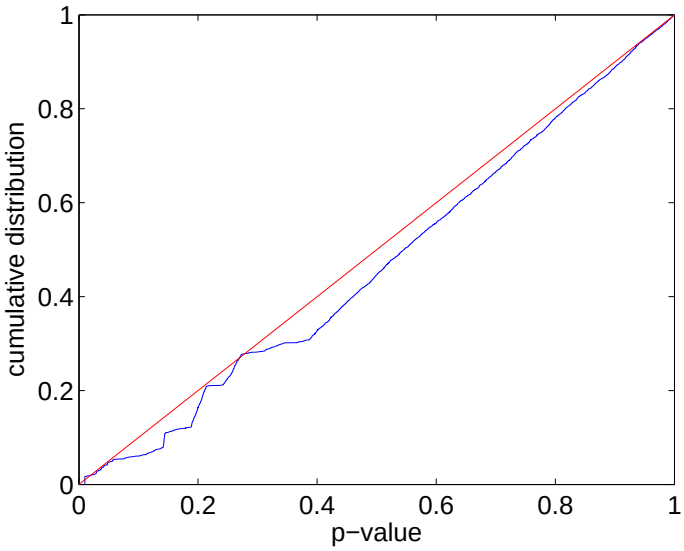

Figure 4

Additional tables

Table 1: This table shows all gene families identified as unlikely under the BD model. The first column gives the gene family name; the second column describes the gene family size among the five *Saccharomyces* species in Newick notation. The third column gives the branch that is predicted to be responsible for the overall low p-value of the family via the likelihood ratio test; Newick numbers in bold indicate the branch identified. The fourth column gives the resulting p-value after deleting the responsible branch as identified by method 1, and the last column gives likelihood ratio as computed in method 2. In three cases multiple gene families violating the BD model had the same observed family sizes; these are listed together.

Family name	Family sizes in Newick notation	Pred. branch	Method 1	Method 2
Stress response	(**15** (33 (24 (30 31))))	1	7e-3	6e6
Amino acid biosynthesis	(**3** (8 (6 (6 5))))	1	0.15	36
PGM/PMM	(**1** (3 (3 (2 1))))	1	0.09	9.3
Ribosomal L1	(1 (**4** (1 (1 1))))	2	0.66	4e3
Elongation factor	(1 (**4** (2 (1 1))))	2	0.18	46
Chaperone	(1 (**4** (2 (2 1))))	2	0.11	12
Phosphatidylinositol 4-kinase	(2 (**9** (4 (2 2))))	2	0.06	4e4
Carbamoyl-phosphate synthase	(2 (**6** (5 (3 3))))	2	0.05	20
Alpha/beta hydrolase	(2 (2 (**6** (2 2))))	3	0.77	2e3
Dihydrouridine synthase	(1 (1 (**6** (1 1))))	3	0.67	6e4
Type I phosphodiesterase	(1 (1 (**4** (1 1))))	3	0.67	4e3
Guanine nucleotide exchange factor	(2 (2 (**5** (2 3))))	3	0.25	1e3
DNA binding domain	(2 (2 (**5** (2 1))))	3	0.20	2e3
Ankyrin repeat	(1 (2 (**7** (1 1))))	3	0.19	7e4
- Unknown - Unknown	(1 (2 (**4** (1 1))))	3	0.19	82
Acetate transporter	(2 (4 (**5** (2 2))))	3	0.13	29
TruD	(1 (1 (**3** (1 2))))	3	0.11	21

Continued on next page

Table 1 – continued from previous page

Family name	Family sizes in Newick notation	Pred. branch	Method 1	Method 2
Unknown	(1 (1 (**3** (2 1))))	3	0.11	21
Flavodoxin	(2 (3 (**5** (1 1))))	3	0.11	2e3
Swi2/Snf2 ATPase	(17 (20 (**25** (18 15))))	3	0.07	5e3
GTPase-activating protein	(2 (4 (**6** (3 2))))	3	0.05	71
Maltose transport	(4 (7 (**8** (5 4))))	3	0.04	17
Trichothecene pump	(5 (5 (7 (**10** 6))))	4	0.30	6e3
RNA polymerase Rpb1	(4 (3 (5 (**7** 4))))	4	0.28	1e3
ATPase	(1 (1 (2 (**3** 1))))	4	0.13	62
MAL transcription factor	(2 (5 (4 (**7** 4))))	4	0.09	2e3
Hydroxymethylpyrimidine synthesis	(3 (5 (2 (**7** 4))))	4	0.02	2e3
Transposon	(2 (8 (15 (34 **83**))))	5	<1e-6	4e54
Ribosomal protein (60S)	(2 (1 (1 (1 **3**))))	5	0.30	1e3
eIF4E-associated protein	(1 (2 (1 (1 **3**))))	5	0.23	1e3
Hydrolase	(8 (11 (12 (11 **7**))))	5	0.17	3e3
Metal-dependent phosphohydrolases	(1 (1 (2 (1 **5**))))	5	0.12	2e4
Sortilin	(5 (4 (7 (4 **8**))))	5	0.04	1e3
Helicase	(1 (3 (3 (2 **34**))))	5	0.04	1e39
NAD kinase	(3 (1 (1 (2 **4**))))	5	0.03	50
Hydroxyisocaproate dehydrogenases	(3 (1 (2 (1 **3**))))	5	0.03	9.3
ABC transporter	(15 (18 (17 (12 **8**))))	5	0.01	4e3
Thiol oxidase	(1 (1 **(4 (2 3))**))	6	0.17	1e3
Leucine rich repeat	(4 (3 **(1 (2 1))**))	6	0.11	38
Flocculation	(10 (6 (8 **(11 14)**)))	7	0.01	85
Unknown	(7 (16 (7 **(20 17)**)))	7	2e-6	4e5
Transposon	(17 (14 (15 **(1 5)**)))	7	2e-4	6e10
Unknown	(5 (11 (14 **(4 2)**)))	7	6e-6	2e6

Continued on next page

Table 1 – continued from previous page

Family name	Family sizes in Newick notation	Pred. branch	Method 1	Method 2
HSP70 Chaperone	(13 (17 (18 **(12 13)**)))	7	0.14	23
- Transcription factor - Pol III transcription factor - Tor2p binding protein - Ribosomal SSU (40S) - Adenylate cyclase - RRM1	(1 (3 (3 **(1 1)**)))	7	0.17	37
Myosin	(5 (9 (9 **(5 5)**)))	7	0.10	76
Cation transport enzymes	(8 (10 (13 **(6 5)**)))	7	0.04	2e3
S-methyltransferase	(2 (5 (5 **(1 1)**)))	7	0.07	3e3
- PDRE transcription factor - Vacuolar membrane protein	(1 (4 (4 **(1 1)**)))	7	0.04	3e3
1,3-beta-D-glucan synthase	(3 (8 (7 **(3 3)**)))	7	0.01	5e3

References

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.L. et al. 2004. The Pfam protein families database. *Nucleic Acids Res.* **32:** Database issue D138–D141.

Casella, G. and Berger, R. 1990. *Statistical Inference*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Cliften, P., Sudarsanam, P., Desikan, A., Fulton, L., Fulton, B., Majors, J., Waterston, R., Cohen, B. A., Johnston, M. 2003. Finding functional features in Saccharomyces genomes by phylogenetic footprinting. *Science* **301:** 71–76.

Dempster, A., Laird, N., and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B.* **39:** 1–38.

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

Enright, A.J., Van Dongen, S., and Ouzounis, C.A. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30:** 1575-1584.

Jordan, M.I. *An Introduction to Graphical Models*. In preparation.

Kellis, M., Patterson, N., Endrizzi, M., Birren, B., and Lander, E. 2003. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* **423:** 241–254.

Korf, I. 2004. Gene finding in novel genomes. *BMC Bioinformatics* **5:** 59.

Pearl, J. 1986. Fusion, propagation, and structuring in belief networks. *Artif. Intell.* **29:** 241–288.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Mateo, CA.

Pearson, W.R. 1991. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics* **11:** 635–650.

Smith, T.F. and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* **147:** 195–197.

Stajich, J.E., Block, D., Boulez, K., Brenner, S.E., Chervitz, S.A., Dagdigian, C., Fuellen, G., Gilbert, J.G.R., Korf, I., Lapp, H. et al. 2002. The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.* **12:** 1611-1618.

Van Dongen, S. 2000. *A cluster algorithm for graphs.* Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.