WTFgenes: What's The Function of these genes? Static sites for model-based gene set analysis

Christopher J. Mungall and Ian H. Holmes

March 7, 2017

Supplementary Information

We first reprise the model definition as outlined in the main paper, then outline our extensions and benchmarks.

Review of model

The graphical model underpinning Bayesian TEA is sketched in Figure 1. For each of the m terms there is a boolean random variable T_j ("term j is activated"). For each of the n genes there is a directly-observed boolean random variable O_i ("gene i is observed in the gene set"), and one deterministic boolean variable H_i ("gene i is activated") defined by $H_i = 1 - \prod_{j \in G_i}(1 - T_j)$ where G_i is the set of terms associated with gene i (including directly annotated terms, as well as ancestral terms implied by transitive closure of the directly annotated terms). The probability parameters are π (term activation), α (false positive) and β (false negative), and the respective hyperparameters are $\mathbf{p} = (p_0, p_1)$, $\mathbf{a} = (a_0, a_1)$ and $\mathbf{b} = (b_0, b_1)$. The model is

$$\begin{aligned} P(T_j = 1|\pi) &= \pi \\ P(O_i = 1|H_i = 0, \alpha) &= \alpha \\ P(O_i = 1|H_i = 1, \beta) &= 1 - \beta \end{aligned}$$

with $\pi \sim \text{Beta}(\mathbf{p})$, $\alpha \sim \text{Beta}(\mathbf{a})$ and $\beta \sim \text{Beta}(\mathbf{b})$. The model of Bauer *et al.* (2010) is similar but used an *ad hoc* discretized prior for π, α and β.

Extensions and benchmarks

In developing our Bayesian TEA sampler, we introduce a collapsed version of the model in Figure 1 by integrating out the probability parameters. Let $c_p = \sum_j^m T_j$ count the number of activated terms, $c_g = \sum_i^n H_i$ the activated genes, $c_a = \sum_i^n O_i(1 - H_i)$ the false positives and $c_b = \sum_i^n O_i H_i$ the false negatives. Then

$$P(\mathbf{T}, \mathbf{O}|\mathbf{a}, \mathbf{b}, \mathbf{p}) = Z(c_p; m, \mathbf{p}) Z(c_a; n - c_g, \mathbf{a}) Z(c_b; c_g, \mathbf{b})$$

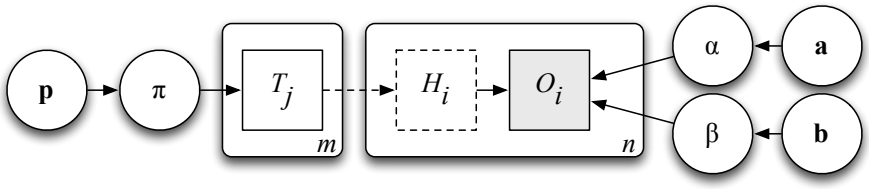

Figure 1: Model-based explanation of observed genes (O_i) using ontology terms (T_j), following Bauer *et al.* (2010). Other variables and hyperparameters are defined in the text. Circular nodes indicate continuous-valued variables or hyperparameters; square nodes indicate discrete-valued (boolean) variables. Dashed lines indicate deterministic relationships; shaded nodes indicate observations. Plates (rounded rectangles) indicate replicated subgraph structures.

where

$$Z(k; N, \mathbf{A}) = \frac{B(N - k + A_0, k + A_1)}{B(A_0, A_1)}$$

is the beta-Bernoulli distribution for k ordered successes in N trials with hyperparameters $\mathbf{A} = (A_0, A_1)$, using the beta function

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

Integrating out probability parameters improves sampling efficiency and allows for higher-dimensional models where, for example, we observe multiple gene sets and give each term its own probability π_j or each gene its own error rates (α_i, β_i). Our implementation by default uses uninformative priors with hyperparameters $\mathbf{a} = \mathbf{b} = \mathbf{p} = (1, 1)$ but this can be overridden by the user.

The MCMC sampler uses a Metropolis-Hastings kernel (Gilks *et al.*, 1996). Each proposed move perturbs some subset of the term variables. The moves include *flip*, where a single term is toggled; *step*, where any activated term and any one of its unactivated ancestors or descendants are toggled; *jump*, where any activated term and any unactivated term are toggled; and *randomize*, where all term variables are uniformly randomized. The relative rates of these moves can be set by the user.

The sampler of Bauer *et al.* (2010) implemented only the *flip* move. To test the relative efficacy of the newly-introduced moves we measured the autocorrelation of the term variables for one of the GO Project's test sets, containing 17 *S.cerevisiae* mating genes[1]. The results, shown in Figure 2, led us to set the MCMC defaults such that the *flip*, *step*, and *jump* moves are equiprobable, while *randomize* is disabled.

[1]Gene IDs: STE2, STE3, STE5, GPA1, SST2, STE11, STE50, STE20, STE4, STE18, FUS3, KSS1, PTP2, MSG5, DIG1, DIG2, STE12.

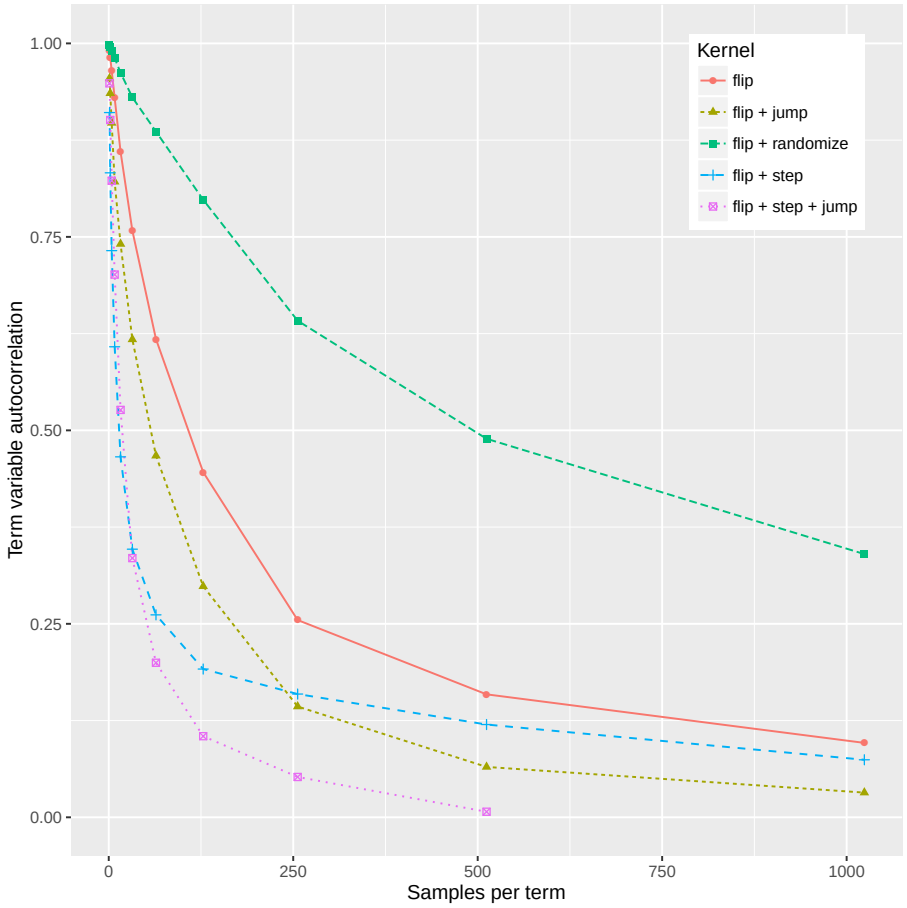

Figure 2: Autocorrelation of term variables, as a function of the number of MCMC samples, for several MCMC kernels on a set of 17 *S.cerevisiae* mating genes. A rapidly-decaying curve indicates an efficiently-mixing kernel. The kernel incorporating *flip*, *step* and *jump* moves (defined in the text) mixes most efficiently.

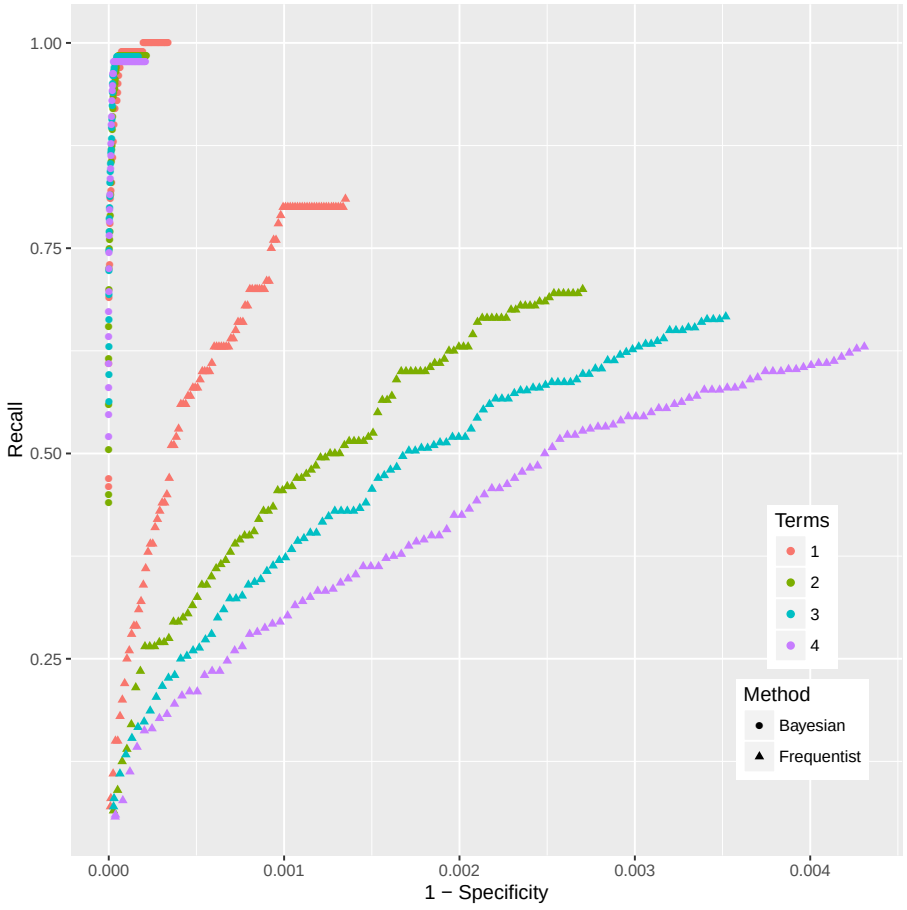

Figure 3: ROC curves for Frequentist and Bayesian TEA. The axes are scaled per term. There are 5,919 ontology terms annotated to *S.cerevisiae* genes, so (for example) a false discovery rate of 0.001 corresponds to about 6 falsely reported terms.

We have implemented both Frequentist TEA (with Bonferroni correction) and Bayesian TEA (as described above), in both C++11 and JavaScript. The JavaScript version can be run as a command-line tool using node, or via a web interface in a browser, and includes extensive unit tests. The two implementations use the same random number generator and yield numerically identical results. The C++ version is about twice as fast: a benchmark of Bayesian TEA on a late-2014 iMac (4GHz Intel Core i7), using the abovementioned 17 yeast mating genes and the relevant subset of 518 GO terms, run for 1,000 samples per term, took 37.6 seconds of user time for the C++ implementation and 79.8 seconds in JavaScript.

By contrast, the Frequentist TEA approach is almost instant. However, its weaker statistical power is apparent from Figure 3, which compares the recall *vs* specificity of Bayesian and Frequentist methods on simulated datasets. For values of N from 1 to 4, we sampled N terms from the *S.cerevisiae* subset of the Gene Ontology, and generated a corresponding set of yeast genes with false positive rate 0.1% and false negative rate 1%. The MCMC sampler was run for 100 iterations per term, and this experiment was repeated 100 times. The model-based approach has vastly superior recall to the Fisher exact test, and the difference grows with the number of terms.

References

Bauer, S., Gagneur, J., and Robinson, P. N. (2010). GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Res.*, **38**(11), 3523–3532.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, UK.