Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Application Note
OXFORD

Databases and ontologies

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

Christopher J. Mungall[1] **and Ian H. Holmes**[2,3]

[1] Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, USA.
[2] Department of Bioengineering, University of California, Berkeley, CA 94720, USA.
[3] Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, USA.

Abstract

Motivation. A common technique for interpreting experimentally-identified lists of genes is to look for enrichment of genes associated to particular Gene Ontology terms. The most common technique uses the hypergeometric distribution; more recently, a model-based approach was proposed. These approaches must typically be run using downloaded software, or on a server. **Results.** We develop a collapsed likelihood for model-based gene set analysis and present WTFgenes, an implementation of both hypergeometric and model-based approaches, that can be published as a static site with computation run in JavaScript on the user’s web browser client. Apart from hosting files, zero server resources are required: the site can (for example) be served directly from an Amazon S3 bucket. A C++11 implementation yielding identical results runs roughly twice as fast as the JavaScript version. **Availability and Implementation.** WTFgenes is available from `https://github.com/evoldoers/wtfgenes` under the BSD3 license. **Contact.** Ian Holmes `ihholmes+wtfgenes@gmail.com`. **Supplementary Information.** None.

Introduction

Term Enrichment Analysis (TEA) is a common technique for finding functional patterns, specifically over-represented ontology terms, in a set of experimentally identified genes (Boyle *et al.*, 2004). The most common approach, which we refer to as *Frequentist TEA*, is a one-tailed Fisher’s Exact Test (based on the hypergeometric distribution, which models the number of term-associations if the gene set was chosen by chance), with a suitable correction for multiple hypothesis testing. Frequentist TEA has been implemented many times on various platforms (Robinson *et al.*, 2002; Khatri *et al.*, 2002; Zeeberg *et al.*, 2003; Boyle *et al.*, 2004; Bauer *et al.*, 2008; Jiao *et al.*, 2012; Chen *et al.*, 2013).

A model-based alternative to Frequentist TEA, which more directly addresses some of the multiple testing issues (for example, by modeling the ways that an observed gene list can be broken down into complementary gene sets), is *Bayesian TEA*. In contrast to Frequentist TEA, which just rejects a null hypothesis that genes are chosen by chance, the Bayesian TEA explicitly models the alternative hypothesis that the gene set was generated from a few random ontology terms. This approach was introduced by Lu *et al.* (2008) and developed by Bauer *et al.* (2010), who demonstrated by simulation its increased statistical power when the observed gene set is indeed explained by a small number of complementary terms. It has been implemented in Java (Bauer *et al.*, 2010) and R (Bauer *et al.*, 2011), but remains significantly less well-developed than Frequentist TEA.

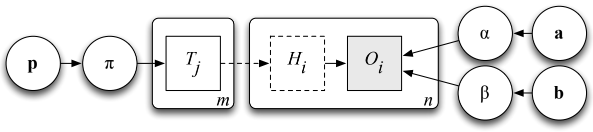

Fig. 1. Model-based explanation of observed genes (O_i) using ontology terms (T_j), following Bauer et al. (2010). Other variables and hyperparameters are defined in the text. Circular nodes indicate continuous-valued variables or hyperparameters; square nodes indicate discrete-valued (boolean) variables.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

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

Most Bayesian and Frequentist TEA implementations are designed for desktop use. Several Frequentist TEA implementations are designed for the web, such as DAVID-WS (Jiao *et al.*, 2012) and Enrichr (Chen *et al.*, 2013) which has a rich dynamic web front-end. However, web-facing Frequentist TEA implementations generally require a server-hosted back end that executes code. Further, there are no web-based Bayesian TEA implementations.

Results

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

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

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.

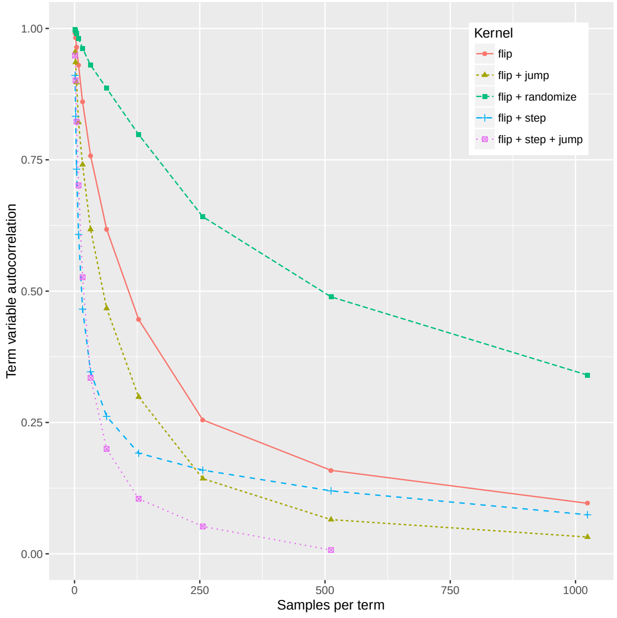

Fig. 2. Autocorrelation of term variables, as a function of the number of MCMC samples, for several MCMC kernels on a set of 17 S.cerevisiae mating genes. A rapidly-decaying curve indicates an efficiently-mixing kernel. The kernel incorporating flip, step and jump moves (defined in the text) mixes most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

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.

Our JavaScript version of Bayesian TEA, when used as a web application, includes visual feedback on the log-likelihood during a sampling run. It can be deployed as a “static site”, i.e. consisting only of static files (HTML, CSS, JSON, and JavaScript) that can be hosted via a minimal web server with no need for dynamic code execution. This has considerable advantages: static web hosting is generally much cheaper, and far more secure, than running server-hosted web applications.

An example wtfgenes static site, configured for the GO-basic ontology and GO-annotated genomes from the Gene Ontology website, can be found at `https://evoldoers.github.io/wtfgo/`

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

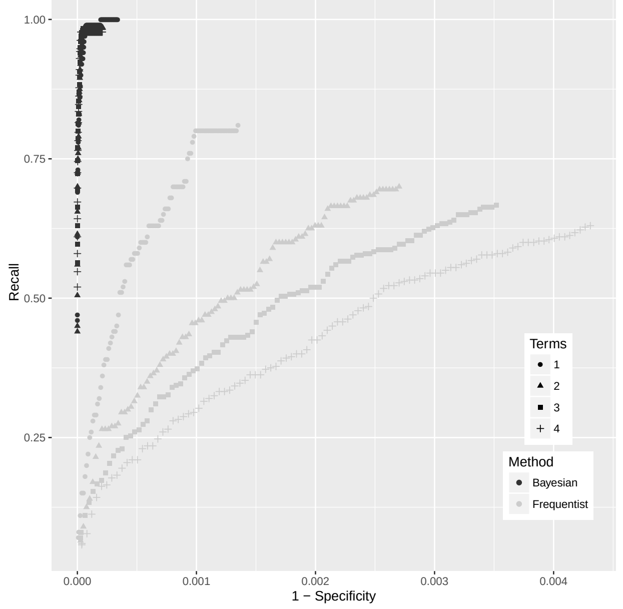

Fig. 3. ROC curves for Frequentist and Bayesian TEA. The axes are scaled per term. There are 5,919 ontology terms annotated to S.cerevisiae genes, so (for example) a false discovery rate of 0.001 corresponds to $\sim$6 falsely reported terms.

Discussion

JavaScript genome browsers such as JBrowse (Buels *et al.*, 2016) represent a broader web trend of producing static sites where possible. We have implemented such a static site approach for ontological term enrichment analysis of gene sets that offers both Bayesian and frequentist approaches. In contrast with existing Frequentist TEA services such as DAVID-WS or Enrichr, it requires no server resources and allows comparison of Bayesian and Frequentist TEA approaches.

Model-based TEA is versatile: it can readily be extended to allow for datasets that are structured temporally (Hejblum *et al.*, 2015), spatially (Lin *et al.*, 2015), or by genomic region (McLean *et al.*, 2010); to use domain-specific biological knowledge (Szczurek and Beerenwinkel, 2014); or to incorporate additional lines of evidence such as quantitative data (Kalaitzis and Lawrence, 2011). We hope our development of a collapsed likelihood, and evaluation of different MCMC kernels, will assist these efforts.

Coincidentally, Fisher's Exact Test—which we call Frequentist TEA—was originally motivated by a blind tea-tasting challenge (Fisher, 1935).

Funding

IHH was partially supported by NHGRI grant HG004483.

References

Bauer, S., Grossmann, S., Vingron, M., and Robinson, P. N. (2008). Ontologizer 2.0–a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics*, **24**(14), 1650–1651.

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.

Bauer, S., Robinson, P. N., and Gagneur, J. (2011). Model-based gene set analysis for Bioconductor. *Bioinformatics*, **27**(13), 1882–1883.

Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M., and Sherlock, G. (2004). 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.

Buels, R., Yao, E., Diesh, C. M., Hayes, R. D., Munoz-Torres, M., Helt, G., Goodstein, D. M., Elsik, C. G., Lewis, S. E., Stein, L., and Holmes, I. H. (2016). JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol.*, **17**, 66.

Chen, E. Y., Tan, C. M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G. V., Clark, N. R., and Ma'ayan, A. (2013). Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics*, **14**, 128.

Fisher, R. A. (1935). Mathematics of a lady tasting tea. In *The Design of Experiments*. Oliver and Boyd, Edinburgh.

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

Hejblum, B. P., Skinner, J., and Thiebaut, R. (2015). Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Comput. Biol.*, **11**(6), e1004310.

Jiao, X., Sherman, B. T., Huang, d. a. W., Stephens, R., Baseler, M. W., Lane, H. C., and Lempicki, R. A. (2012). DAVID-WS: a stateful web service to facilitate gene/protein list analysis. *Bioinformatics*, **28**(13), 1805–1806.

Kalaitzis, A. A. and Lawrence, N. D. (2011). A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, **12**, 180.

Khatri, P., Draghici, S., Ostermeier, G. C., and Krawetz, S. A. (2002). Profiling gene expression using onto-express. *Genomics*, **79**(2), 266–270.

Lin, Z., Sanders, S. J., Li, M., Sestan, N., State, M. W., and Zhao, H. (2015). A Markov random field-based approach to characterizing human brain development using spatial-temporal transcriptome data. *Ann Appl Stat*, **9**(1), 429–451.

Lu, Y., Rosenfeld, R., Simon, I., Nau, G. J., and Bar-Joseph, Z. (2008). A probabilistic generative model for GO enrichment analysis. *Nucleic Acids Res.*, **36**(17), e109.

McLean, C. Y., Bristor, D., Hiller, M., Clarke, S. L., Schaar, B. T., Lowe, C. B., Wenger, A. M., and Bejerano, G. (2010). GREAT improves functional interpretation of cis-regulatory regions. *Nat. Biotechnol.*, **28**(5), 495–501.

Robinson, M. D., Grigull, J., Mohammad, N., and Hughes, T. R. (2002). FunSpec: a web-based cluster interpreter for yeast. *BMC Bioinformatics*, **3**, 35.

Szczurek, E. and Beerenwinkel, N. (2014). Modeling mutual exclusivity of cancer mutations. *PLoS Comput. Biol.*, **10**(3), e1003503.

Zeeberg, B. R., Feng, W., Wang, G., Wang, M. D., Fojo, A. T., Sunshine, M., Narasimhan, S., Kane, D. W., Reinhold, W. C., Lababidi, S., Bussey, K. J., Riss, J., Barrett, J. C., and Weinstein, J. N. (2003). GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol.*, **4**(4), R28.