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

Ian H. Holmes[1]

[1] Department of Bioengineering, University of California, 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, zero server resources are required: the site can (for example) be served directly from an 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`. **Contact.** Ian Holmes `ihholmes+wtfgenes@gmail.com`.
Supplementary Information. None.

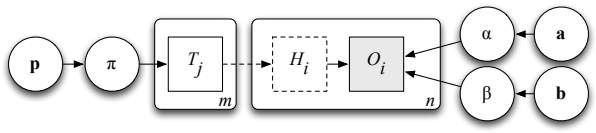

Fig. 1. Model-based explanation of observed genes (O_j) using ontology terms (T_i), following Bauer et al. (2010).

Introduction

Gene Set Enrichment Analysis (GSEA) (Subramanian *et al.*, 2005) Standard approach is a one-tailed Fisher's Exact Test (based on the hypergeometric distribution), with a correction for multiple hypothesis testing. Numerous implementations e.g. GO::TermFinder (Boyle *et al.*, 2004)

Model-based Gene Set Analysis (MGSA) (Bauer *et al.*, 2010) Bioconductor (Bauer *et al.*, 2011)

builds on earlier generative model by Lu *et al.* (2008)

The MGSA model 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 $H_i = 1 - \prod_{j \in G_i} T_j$ 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.

Most MGSA and GSEA implementations are designed for desktop use.

Several GSEA implementations are designed for web use, notably Enrichr (Chen *et al.*, 2013; Gundersen *et al.*, 2015; Kuleshov *et al.*, 2016) which has a rich dynamic web front-end. However these web-facing GSEA implementations generally require a server-hosted back end. Further, there are no web-based MGSA implementations.

Results

We sample from a collapsed version of the MGSA 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

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

is the beta-binomial distribution for k successes in N trials with pseudocounts $\mathbf{A} = (A_0, A_1)$, 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 where each proposed move perturbs some subset of the term variables. These 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.

We have implemented both GSEA (with Bonferroni correction) and MGSA (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 MGSA 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 GSEA approach is almost instant, though less statistically powerful than MGSA if the observation is indeed explained by a small number of complementary terms (Bauer *et al.*, 2010).

When run via a web browser, the JavaScript implementation 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.

Discussion

JavaScript genome browsers such as JBrowse (Buels *et al.*, 2016) are consistent with a broader web trend of producing static sites where possible

Not a direct competitor to Enrichr, which has much richer visualizations and allows user submission of gene sets, consequently requiring a server back-end

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

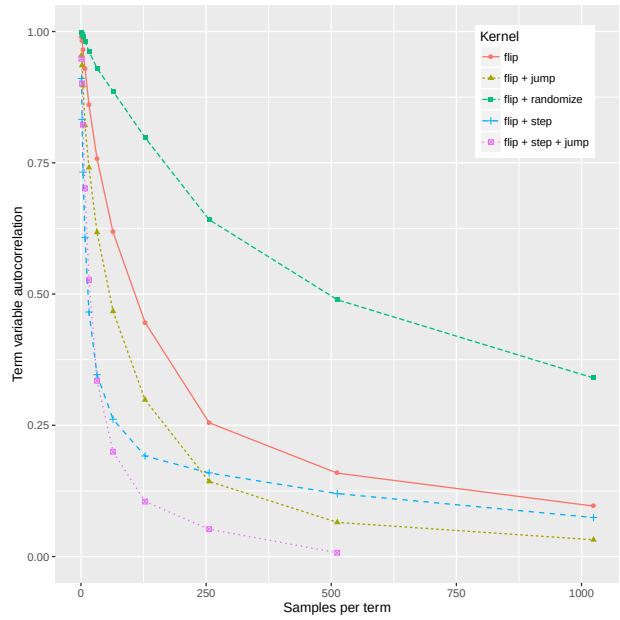

Fig. 2. Autocorrelation of term variables, as a function of the number of MCMC samples, for several MCMC kernels on a set of 17 S.cerevisiae mating genes. A rapidly-decaying curve indicates an efficiently-mixing kernel.

The model-based approach is versatile: it can readily be extended to applications that are structured temporally (Hejblum *et al.*, 2015), spatially (Lin *et al.*, 2015), by genomic region (McLean *et al.*, 2010), or using domain-specific biological knowledge (Szczurek and Beerenwinkel, 2014).

Funding

IHH was partially supported by NHGRI grant R01-HG004483.

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.

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.

Gundersen, G. W., Jones, M. R., Rouillard, A. D., Kou, Y., Monteiro, C. D., Feldmann, A. S., Hu, K. S., and Ma'ayan, A. (2015). GEO2Enrichr: browser extension and server app to extract gene sets from GEO and

analyze them for biological functions. *Bioinformatics*, **31**(18), 3060–3062.

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.

Kuleshov, M. V., Jones, M. R., Rouillard, A. D., Fernandez, N. F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S. L., Jagodnik, K. M., Lachmann, A., McDermott, M. G., Monteiro, C. D., Gundersen, G. W., and Ma'ayan, A. (2016). Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.*, **44**(W1), W90–97.

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.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.*, **102**(43), 15545–15550.

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