Check for updates

SOFTWARE TOOL ARTICLE

WTFgenes: What's The Function of these genes? Static sites for model-based gene set analysis [version 1; peer review: 2 approved with reservations]

Christopher J. Mungall¹, Ian H. Holmes^{id} 2,3¹Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA²Department of Bioengineering, University of California, Berkeley, 94720, USA³Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA**V1** First published: 04 Apr 2017, 6:423
<https://doi.org/10.12688/f1000research.11175.1>Latest published: 04 Apr 2017, 6:423
<https://doi.org/10.12688/f1000research.11175.1>

Abstract

A common technique for interpreting experimentally-identified lists of genes is to look for enrichment of genes associated with particular ontology terms. The most common test uses the hypergeometric distribution; more recently, a model-based test was proposed. These approaches must typically be run using downloaded software, or on a server. 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 Amazon S3 or GitHub Pages. A C++11 implementation yielding identical results runs roughly twice as fast as the JavaScript version. WTFgenes is available from <https://github.com/evoldoers/wtfgenes> under the BSD3 license. A demonstration for the Gene Ontology is usable at <https://evoldoers.github.io/wtfgo>.

Keywords

Gene Ontology, Graphical Model, Gene Set Enrichment Analysis

Open Peer Review

Approval Status ? ?

	1	2
version 1	?	?
04 Apr 2017	view	view
1. Cedric Simillion ^{id} , University of Bern, Bern, Switzerland		
University of Bern, Bern, Switzerland		
2. Ruth Isserlin ^{id} , University of Toronto, Toronto, Canada		

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Ian H. Holmes (ihholmes@gmail.com)

Competing interests: No competing interests were disclosed.

Grant information: IHH was partially supported by NHGRI (grant HG004483). CJM was partially supported by Office of the Director (R24-OD011883) and the Director, Office of Science, Office of Basic Energy Sciences, of the US Department of Energy (Contract No. DE-AC02-05CH11231).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2017 Mungall CJ and Holmes IH. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

How to cite this article: Mungall CJ and Holmes IH. **WTFgenes: What's The Function of these genes? Static sites for model-based gene set analysis [version 1; peer review: 2 approved with reservations]** F1000Research 2017, 6:423 <https://doi.org/10.12688/f1000research.11175.1>

First published: 04 Apr 2017, 6:423 <https://doi.org/10.12688/f1000research.11175.1>

Introduction

Term Enrichment Analysis (TEA) is a common technique for finding functional patterns, specifically overrepresented ontology terms, in a set of experimentally identified genes¹. 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^{1–8}.

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 9 and further developed by 10, who implemented model-based testing in Java and R¹¹. However, the model-based approach remains significantly less well-explored than frequentist approaches.

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 10 is similar, but uses 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-WS⁶ and Enrichr^{8,12,13}, 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 JavaScript-based Bayesian TEA implementations, and no web-facing implementations other than the Java-based Ontologizer which can be loaded via Java Web Start.

In order to further explore the model-based TEA and compare it to Frequentist TEA, and to make these investigations accessible to researchers in a way that would be easily embeddable in static websites, we developed WTFgenes, a JavaScript implementation of both approaches with (for time-sensitive applications) a parallel C++ implementation that is numerically identical.

We note in passing that Fisher's Exact Test—which we call Frequentist TEA—was originally motivated by a blind tea-tasting challenge¹⁴.

Methods

Model

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

Then

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

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

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

Figure 1. Model-based explanation of observed genes (O) using ontology terms (T), following¹⁰. 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.

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 kernel¹⁵. 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 10 implemented only the *flip* move. To test the relative efficacy of the newly-introduced moves we measured the autocorrelation of the term variables for a dataset of 17 *S.cerevisiae* genes involved in mating (The gene IDs used in this

evaluation, for purposes of reproduction, were: STE2, STE3, STE5, GPA1, SST2, STE11, STE50, STE20, STE4, STE18, FUS3, KSS1, PTP2, MSG5, DIG1, DIG2, STE12. Other representative gene sets for yeast may be obtained from the Gene Ontology website at <http://geneontology.org/experimental/enrichment-genesets/yeast/> and several of these are bundled with the example dataset in the WTFgenes repository). 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.

Implementation

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.

Operation

Our JavaScript software, when used as a web application, offers a “quick report” view using Frequentist TEA. For the

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.

slower-running, but more powerful, Bayesian TEA, the software plots the log-likelihood during an MCMC sampling run, for visual feedback. The repository includes setup scripts allowing the tool to 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>.

An earlier version of this article can be found on bioRxiv (doi: [10.1101/114785](https://doi.org/10.1101/114785)).

Results

When compiled using clang, the C++ version of WTFgenes is about twice as fast as the JavaScript version: a benchmark of

Bayesian TEA on a late-2014 iMac (4GHz Intel Core i7), using the above mentioned 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 (The full workflow for this simulation is available at <http://doi.org/10.5281/zenodo.400608>¹⁶). 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.

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.

Discussion

JavaScript genome browsers, such as JBrowse¹⁷, represent a broader web trend of producing static sites where possible, for reasons of security and performance. We have implemented such a static site generator for ontological term enrichment analysis of gene sets that offers both Bayesian and frequentist tests. In contrast with existing web services for Frequentist TEA, such as DAVID-WS or Enrichr, it requires no server resources and allows comparison of Bayesian and Frequentist approaches.

Model-based TEA is versatile: it can readily be extended to allow for datasets that are structured temporally¹⁸, spatially¹⁹, or by genomic region²⁰; to use domain-specific biological knowledge²¹; or to incorporate additional lines of evidence such as quantitative data²². We hope our development of a collapsed likelihood, and evaluation of different MCMC kernels, will assist these efforts.

Software and data availability

Latest source code: <https://github.com/evoldoers/wtfgenes>

Archived source code as at time of publication: <http://doi.org/10.5281/zenodo.400606>²³

Software license: BSD3

A demonstration for the Gene Ontology is usable at <https://evoldoers.github.io/wtfgo>.

A Makefile-driven simulation study underpinning results reported in this paper is available at <http://dx.doi.org/10.5281/zenodo.400608>¹⁶.

Author contributions

IH designed the method, wrote the software, performed the analyses, and wrote the manuscript. CM suggested the idea, consulted on the design of the software and corrected errors in the manuscript.

Competing interests

No competing interests were disclosed.

Grant information

IHH was partially supported by NHGRI (grant HG004483). CJM was partially supported by Office of the Director (R24-OD011883) and the Director, Office of Science, Office of Basic Energy Sciences, of the US Department of Energy (Contract No. DE-AC02-05CH11231).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

- Boyle EI, Weng S, Gollub J, et al.: **GO::TermFinder--open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes.** *Bioinformatics*. 2004; 20(18): 3710–3715.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Robinson MD, Grigull J, Mohammad N, et al.: **FunSpec: a web-based cluster interpreter for yeast.** *BMC Bioinformatics*. 2002; 3: 35.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Khatri P, Draghici S, Ostermeier GC, et al.: **Profiling gene expression using onto-express.** *Genomics*. 2002; 79(2): 266–270.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Zeeberg BR, Feng W, Wang G, et al.: **GoMiner: a resource for biological interpretation of genomic and proteomic data.** *Genome Biol*. 2003; 4(4): R28.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Bauer S, Grossmann S, Vingron M, et al.: **Ontologizer 2.0--a multifunctional tool for GO term enrichment analysis and data exploration.** *Bioinformatics*. 2008; 24(14): 1650–1651.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Jiao X, Sherman BT, Huang da W, et al.: **DAVID-WS: a stateful web service to facilitate gene/protein list analysis.** *Bioinformatics*. 2012; 28(13): 1805–1806.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Mi H, Muruganujan A, Casagrande JT, et al.: **Large-scale gene function analysis with the PANTHER classification system.** *Nat Protoc*. 2013; 8(8): 1551–1566.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Chen EY, Tan CM, Kou Y, et al.: **Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool.** *BMC Bioinformatics*. 2013; 14: 128.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Lu Y, Rosenfeld R, Simon I, et al.: **A probabilistic generative model for GO enrichment analysis.** *Nucleic Acids Res*. 2008; 36(17): e109.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Bauer S, Gagneur J, Robinson PN: **GOing Bayesian: model-based gene set analysis of genome-scale data.** *Nucleic Acids Res*. 2010; 38(11): 3523–3532.
[PubMed Abstract](#) | [Free Full Text](#)
- Bauer S, Robinson PN, Gagneur J: **Model-based gene set analysis for Bioconductor.** *Bioinformatics*. 2011; 27(13): 1882–1883.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Gundersen GW, Jones MR, Rouillard AD, et al.: **GEO2Enrichr: browser extension and server app to extract gene sets from GEO and analyze them for biological functions.** *Bioinformatics*. 2015; 31(18): 3060–3062.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Kuleshov MV, Jones MR, Rouillard AD, et al.: **Enrichr: a comprehensive gene set enrichment analysis web server 2016 update.** *Nucleic Acids Res*. 2016; 44(W1): W90–97.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Fisher RA: **Mathematics of a lady tasting tea.** In *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.
- Gilks WR, Richardson S, Spiegelhalter DJ: **Markov Chain Monte Carlo in Practice.** Chapman & Hall, London, UK, 1996.
[Reference Source](#)
- Holmes IH, Mungall C: **ihh/wtfgenes-paper: 0.1.0 release [Data set].** *Zenodo*. 2017.
[Data Source](#)
- Buels R, Yao E, Diesh CM, et al.: **JBrowse: a dynamic web platform for genome visualization and analysis.** *Genome Biol*. 2016; 17: 66.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Hejblum BP, Skinner J, Thiébaud R: **Time-Course Gene Set Analysis for Longitudinal Gene Expression Data.** *PLoS Comput Biol*. 2015; 11(6): e1004310.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Lin Z, Sanders SJ, Li M, et al.: **A Markov Random Field-Based Approach to Characterizing Human Brain Development Using Spatial-Temporal Transcriptome Data.** *Ann Appl Stat*. 2015; 9(1): 429–451.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- McLean CY, Bristor D, Hiller M, et al.: **GREAT improves functional interpretation of cis-regulatory regions.** *Nat Biotechnol*. 2010; 28(5): 495–501.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Szczurek E, Beerenwinkel N: **Modeling mutual exclusivity of cancer mutations.** *PLoS Comput Biol*. 2014; 10(3): e1003503.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Kalaizis AA, Lawrence ND: **A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression.** *BMC Bioinformatics*. 2011; 12: 180.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Holmes IH, Mungall C: **evoldoers/wtfgenes: 0.1.0 release [Data set].** *Zenodo*. 2017.
[Data Source](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 23 June 2017

<https://doi.org/10.5256/f1000research.12058.r23167>

© 2017 Isserlin R. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ruth Isserlin

The Donnelly Centre, University of Toronto, Toronto, ON, Canada

In the article "WTFgenes: What's The Function of these genes? Static sites for model-based gene set analysis" Mungall and Holmes introduce a java script static site implementation of a model based Bayesian method to calculate functional enrichment. Included in this is an implementation of the current standard method, fisher exact test.

In the paper the method model is very well explained but given that the authors are introducing a tool to access this model very little was discussed about how the software works. For example, the author states that other front end tools "require a server-hosted back end that executes code" but it is not clear how WTFgenes work in a way that it doesn't require a back end that executes code. I think it would be helpful to clearly outline (in a figure) the different implementation of WTFgenes, and how a user can access/set up the different parts available, required inputs and generated outputs.

Also, it would be helpful if you expand the example that are presented in the paper to something larger than 17 yeast genes. It is not discussed in the paper but I presume that there are performance limitations which is why there is both javascript and C++ versions. It would be helpful if this was stated. For example, using X number of genes would take Y time in javascript but Z using the C++ implmentation. (I am also not sure how you could switch between these two implementations. It states in the paper that the javascript version can be run by command line or via the web but it doesn't say how to run the c++ version). What is the benefit of running the javascript version by command line?

For the yeast example in the paper do you use all of Gene ontology (CC, BP, MF) or just a subset of terms?

Is there a way to output the results of the enrichment analysis so you can use the results in downstream analyses?

Minor comments:

In the paper it states that for *Frequentist* enrichment analysis you use bonferonni correction. Under the tab "Quick report" which contains these results I see a p-value. Is this the corrected p-value or nominal p-value? If it is the nominal how do we find the corrected p-value?

Some general comments/questions:

Can WTFgenes only work with gene ontology?

Given that there is no back-end web server is it easier to update the annotation that you use? It looks like it requires obo and gaf files but can it also support generic gene set files.

It might be beneficial to create a docker image of WTFgenes for easy installation of WTFgenes.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 05 May 2017

<https://doi.org/10.5256/f1000research.12058.r22056>

© 2017 Simillion C. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cedric Simillion

¹ Interfaculty Bioinformatics Unit and SIB Swiss Institute of Bioinformatics, University of Bern, Bern, Switzerland

² Department of Clinical Research, University of Bern, Bern, Switzerland

The authors present a novel "Term Enrichment Analysis" algorithm, which expands on previous work by Bauer et al. (2010). The provided implementation of the algorithm as a stand-alone web interface is very well-designed and user-friendly. The availability of a command-line implementation in C++ ensures that the method can be incorporated in diverse workflows.

I do, however, have some major criticisms about the presentation of the method in the manuscript as well as the validation method used.

Major points:

My main problem with this manuscript is that the description of the algorithm is very terse and hard to understand. In particular, the following points need clarification:

1. The algorithm model needs to be described in less mathematical terms. The present description makes it very hard for a biologist to understand the merits of the algorithm.
2. The biological meaning or impact of the mentioned hyperparameters A_0 and A_1 needs to be added.
3. The authors claim as one of the advantages of their algorithm that "*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_j, β_j)* " However, they do not mention any procedure for estimating these parameter values. A detailed example of such a procedure would greatly benefit the manuscript.
4. Related to the previous point: It seems that there are quite a few parameters in this algorithm that can be adjusted. While the implementation provided does seem to suggest sensible default values, it would be good if the authors could prove the robustness of their method by validating a test set against a range of parameter values.

The second major concern I have with this manuscript is lack of rigour and detail in the applied validation procedures.

1. It is not clear at all to me what is meant with "the autocorrelation of the term variables for a dataset". This concept needs to be explained in more detail, ideally with an example.
2. In the tuning step of the MCMC kernels, the authors used a test set of only 17 genes. Typical transcriptomics experiments yield, especially in mammals, up to thousands of differentially expressed genes. It would therefore be good to repeat this analysis with increasing test set sizes (e.g. 10 - 100 - 1000).
3. Possibly the biggest issue I have with this manuscript is that the authors compare the performance of their algorithm to that of a simple hypergeometric test, using simulated data. As several authors have already pointed out before, the hypergeometric approach is a poor strategy for doing gene set analysis¹. Validation should be against more sophisticated "frequentist" algorithms such as TopGO², PADOG³, SetRank⁴, ... as these algorithms also deal with the multiple hypothesis testing problem by considering the overlap between

different term gene sets. Ideally, a benchmarking strategy on real biological data, such as the one suggested by Tarca et al.⁵ would be used.

Minor Point:

1. Most of the literature refers to this type of analysis as "Gene Set Enrichment Analysis" GSEA. It would be good if the authors at least refer to this term as well.

References

1. Pan KH, Lih CJ, Cohen SN: Effects of threshold choice on biological conclusions reached during analysis of gene expression by DNA microarrays. *Proc Natl Acad Sci U S A*. 2005; **102** (25): 8961-5 [PubMed Abstract](#) | [Publisher Full Text](#)
2. Alexa A, Rahnenführer J, Lengauer T: Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*. 2006; **22** (13): 1600-7 [PubMed Abstract](#) | [Publisher Full Text](#)
3. Tarca AL, Draghici S, Bhatti G, Romero R: Down-weighting overlapping genes improves gene set analysis. *BMC Bioinformatics*. 2012; **13**: 136 [PubMed Abstract](#) | [Publisher Full Text](#)
4. Simillion C, Liechti R, Lischer HE, Ioannidis V, et al.: Avoiding the pitfalls of gene set enrichment analysis with SetRank. *BMC Bioinformatics*. 2017; **18** (1): 151 [PubMed Abstract](#) | [Publisher Full Text](#)
5. Tarca AL, Bhatti G, Romero R: A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity. *PLoS One*. 2013; **8** (11): e79217 [PubMed Abstract](#) | [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

No

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research