BIOINFORMATICS ORIGINAL PAPER
Vol. 26 no. 21 2010, pages 2752–2759
doi:10.1093/bioinformatics/btq511

Systems biology
Advance access publication September 8, 2010

Robust and accurate data enrichment statistics via distribution function of sum of weights

Aleksandar Stojmirović and Yi-Kuo Yu*
National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Term-enrichment analysis facilitates biological interpretation by assigning to experimentally/computationally obtained data annotation associated with terms from controlled vocabularies. This process usually involves obtaining statistical significance for each vocabulary term and using the most significant terms to describe a given set of biological entities, often associated with weights. Many existing enrichment methods require selections of (arbitrary number of) the most significant entities and/or do not account for weights of entities. Others either mandate extensive simulations to obtain statistics or assume normal weight distribution. In addition, most methods have difficulty assigning correct statistical significance to terms with few entities.

Results: Implementing the well-known Lugananni–Rice formula, we have developed a novel approach, called SaddleSum, that is free from all the aforementioned constraints and evaluated it against several existing methods. With entity weights properly taken into account, SaddleSum is internally consistent and stable with respect to the choice of number of most significant entities selected. Making few assumptions on the input data, the proposed method is universal and can thus be applied to areas beyond analysis of microarrays. Employing asymptotic approximation, SaddleSum provides a term-size-dependent score distribution function that gives rise to accurate statistical significance even for terms with few entities. As a consequence, SaddleSum enables researchers to place confidence in its significance assignments to small terms that are often biologically most specific.

Availability: Our implementation, which uses Bonferroni correction to account for multiple hypotheses testing, is available at http://www.ncbi.nlm.nih.gov/CBBresearch/qmbp/mn/enrich/.
Source code for the standalone version can be downloaded from ftp://ftp.ncbi.nlm.nih.gov/pub/qmbpmn/SaddleSum/.

Contact: yyu@ncbi.nlm.nih.gov

Supplementary information: Supplementary materials are available at *Bioinformatics* online.

Received on May 19, 2010; revised on July 30, 2010; accepted on August 31, 2010

1 INTRODUCTION

A major challenge of contemporary biology is to ascribe interpretation to high-throughput experimental or computational results, where each considered entity (gene or protein) is assigned a value. Biological information is often summarized through controlled vocabularies such as Gene Ontology (GO; Ashburner *et al.*, 2000), where each annotated term includes a list of entities. Let **w** denote a collection of values, each associated with an entity. Given **w** and a controlled vocabulary, enrichment analysis aims to retrieve the terms that by statistical inference best describe **w**, that is, the terms associated with entities with atypical values. Many enrichment analysis tools have been developed primarily to process microarray data (Huang *et al.*, 2009). In terms of biological relevance, the performance assessment of those tools is generally difficult. It requires a large, comprehensive 'gold standard' vocabulary together with a collection of **w**'s processed from experimental data, and with true/false positive terms corresponding to each **w** correctly specified. This invariably introduces some degree of circularity because the terms often come from curating experimental results. Before declaring efficacy in biological information retrieval that is non-trivial to assess, an enrichment method should pass at least the statistical accuracy and internal consistency test.

In their recent survey, Huang *et al.* (2009) list 68 distinct bioinformatic enrichment tools introduced between 2002 and 2008. Most tools share a similar workflow: given **w** obtained by suitably processing experimental data, they sequentially test each vocabulary term for enrichment to obtain its *P*-value (the likelihood of a false positive given the null hypothesis). Since many terms are tested, a multiple hypothesis correction, such as Bonferroni (Hochberg and Tamhane, 1987) or false discovery rate (FDR; Benjamini and Hochberg, 1995), is applied to *P*-value of each to obtain the final statistical significance. The results are displayed for the user in a suitable form outlining the significant terms and possibly relations between them. Note that the latter steps are largely independent from the first. To avoid confounding factors, we will focus exclusively on the original enrichment *P*-values.

Based on the statistical methods employed, the existing enrichment tools can generally be divided into two main classes. The singular enrichment analysis (SEA) class contains numerous tools that form the majority of published ones (Huang *et al.*, 2009). By ordering values in **w**, these tools require users to select a number of top-ranking entities as input and mostly use hypergeometric distribution (or equivalently Fisher's exact test) to obtain the term *P*-values. After the selection is made, SEA treats all entities equally, ignoring their value differences.

The gene set analysis (GSA) class was pioneered by the gene set enrichment analysis (GSEA) tool (Mootha *et al.*, 2003; Subramanian *et al.*, 2005). Tools from this class use all values (entire **w**) to

*To whom correspondence should be addressed.

calculate P-values and do not require preselection of entities. Some approaches (Al-Shahrour *et al.*, 2007; Blom *et al.*, 2007; Breitling *et al.*, 2004; Eden *et al.*, 2009) in this group apply hypergeometric tests to all possible selections of top-ranking entities. The final P-value is computed by combining (in a tool-specific manner) the P-values from the individual tests. Other approaches use non-parametric approaches: rank-based statistics such as Wilcoxon rank-sum (Breslin *et al.*, 2004) or Kolmogorov–Smirnov like (Backes *et al.*, 2007; Ben-Shaul *et al.*, 2005; Mootha *et al.*, 2003; Subramanian *et al.*, 2005). When weights are taken into account, such as in GSEA (Subramanian *et al.*, 2005), statistical significance must be determined from a sampled (shuffled) distribution. Unfortunately, limited by the number of shuffles that can be performed, the smallest obtainable P-value is bounded away from 0.

The final group of GSA methods computes a score for each vocabulary term as a sum of the values (henceforth used interchangeably with weights) of the m entities it annotates. In general, the score distribution $\mathrm{pdf}_m(S)$ for the experimental data is unknown. By Central Limit Theorem, when m is large, Gaussian (Kim and Volsky, 2005; Smid and Dorssers, 2004) or Student's t-distribution (Boorsma *et al.*, 2005; Luo *et al.*, 2009) can be used to approximate $\mathrm{pdf}_m(S)$. Unfortunately, when the weight distributions are skewed, the required m may be too large for practical use. Evidently, this undermines the P-value accuracy of small terms (meaning terms with few entities), which are biologically most specific.

It is generally found that, given the same vocabulary and $\mathbf{w}$, different enrichment analysis tools report diverse results. We believe this may be attributed to disagreement in P-values reported as well as that different methods have different degree of robustness (internal consistency). Instead of providing a coherent biological understanding, the array of diverse results questions the confidence of information found. Furthermore, other than microarray datasets, there exist experimental or computational results such as those from ChIP-chip (Eden *et al.*, 2007), deep sequencing (Sultan *et al.*, 2008), quantitative proteomics (Sharma *et al.*, 2009) and *in silico* network simulations (Stojmirović and Yu, 2007, 2009), that may benefit from enrichment analysis. It is thus imperative to have an enrichment method that report accurate P-values, preserves internal consistency and allows investigations of a broader range of datasets.

To achieve these goals, we have developed a novel enrichment tool, called SaddleSum, that founds on the well-known Lugananni–Rice formula (Lugannani and Rice, 1980) and derives its statistics from approximating asymptotically the distribution function of the scores used in the parametric GSA class. This allows us to obtain accurate statistics even in the cases where the distribution function generating $\mathbf{w}$ is very skewed and for terms containing few entities. The latter aspect is particularly important for obtaining biologically specific information.

2 METHODS

2.1 Mathematical foundations for SaddleSum

We distinguish two sets: the set of entities $\mathcal{N}$ of size n and the controlled vocabulary $\mathcal{V}$. Each term from $\mathcal{V}$ maps to a set $\mathcal{M} \subset \mathcal{N}$ of size $m < n$. From experimental results, we obtain a set $\mathbf{w} = \{w_j | j \in \mathcal{N}\}$ and ask how likely it is to randomly pick m entities whose sum of weights exceeds the sum $\hat{S} = \sum_{j \in \mathcal{M}} w_j$.

Assume that the weights in $\mathbf{w}$ come independently from a continuous probability space W with the density function p such that the moment generating function $\rho(t) = \int_W p(x) e^{tx} \mathrm{d}x$ exists for t in a neighborhood of 0. The density of S, sum of m weights arbitrarily sampled from $\mathbf{w}$, can be expressed by the Fourier inversion formula

$$\mathrm{pdf}_m(S) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{mK(it) - itS} \, \mathrm{d}t, \tag{1}$$

where $K(t) = \ln \rho(t)$ denotes the cumulant generating function of p. The tail probability or P-value for a score $\hat{S}$ is given by

$$\mathrm{Prob}(S \geq \hat{S}) = \int_{\hat{S}}^{\infty} \mathrm{pdf}_m(S) \mathrm{d}S. \tag{2}$$

We propose to use an asymptotic approximation to (2), which improves with increasing m and $\hat{S}$.

Daniels (1954) derived an asymptotic approximation for the density pdf_m through saddlepoint expansion of the integral (1), while the corresponding approximation to the tail probability was obtained by Lugannani and Rice (1980). Let $\phi(x) = \exp(-x^2/2)/\sqrt{2\pi}$ and $\Phi(x) = \int_x^{\infty} \phi(t) \mathrm{d}t$ denote, respectively, the density and the tail probability of Gaussian distribution. Let $\hat{\lambda}$ be a solution of the equation

$$\hat{S} = mK'(\hat{\lambda}). \tag{3}$$

Then, the leading term of the Lugananni–Rice approximation to the tail probability takes the form

$$\mathrm{Prob}(S \geq \hat{S}) = \Phi(\hat{z}) + \left(\frac{1}{\hat{y}} - \frac{1}{\hat{z}} \right) \phi(\hat{z}) + O(m^{-3/2}), \tag{4}$$

where $\hat{y} = \hat{\lambda} \sqrt{mK''(\hat{\lambda})}$ and $\hat{z} = \mathrm{sgn}(\hat{\lambda}) \sqrt{2(\hat{\lambda}\hat{S} - mK(\hat{\lambda}))}$. Appropriate summary of derivation of (4) is provided in the Supplementary Materials.

Daniels (1954) has shown that Equation (3) has a unique simple root under most conditions and that $\hat{\lambda}$ increases with $\hat{S}$, with $\hat{\lambda} = 0$ for $\hat{S} = m\langle W \rangle$ where $\langle W \rangle = \int_W x p(x) \mathrm{d}x$ is the mean of W. While the approximation (4) is uniformly valid over the whole domain of p, its components need to be rearranged for numerical computation near the mean. When $\hat{S} \gg m\langle W \rangle$, $\phi(\hat{z})/\hat{y}$ dominates and the overall error is $O(m^{-1})$ (Daniels, 1987).

SaddleSum, our implementation of Lugananni–Rice approximation for computing enrichment P-values, first solves Equation (3) for $\hat{\lambda}$ using Newton's method and then returns the P-value using (4). The derivatives of the cumulant generating function are estimated from $\mathbf{w}$: we approximate the moment generating function by $\rho(t) \approx \frac{1}{n} \sum_{j \in \mathcal{N}} e^{tw_j}$, and then $K'(t) = \rho'(t)/\rho(t)$ and $K''(t) = \rho''(t)/\rho(t) - (K'(t))^2$. Since the same $\mathbf{w}$ is used to sequentially evaluate P-values of all terms in $\mathcal{V}$, we retain previously computed $\hat{\lambda}$ values in a sorted array. This allows us, using binary search, to reject many terms with P-values greater than a given threshold without running Newton's method or to bracket the root of (3) for faster convergence. More details on the SaddleSum implementation and evaluations of its accuracy against some well-characterized distributions are in Section 2 of Supplementary Materials. When run as a term-enrichment tool, SaddleSum reports E-value for each significant term by applying Bonferroni correction to the term's P-value.

2.2 GO

The assignment of human genes to GO terms was taken from the NCBI gene2go file (ftp://ftp.ncbi.nih.gov/gene/DATA/gene2go.gz) downloaded on February 7, 2009. After assigning all genes to terms, we removed all redundant terms—if several terms mapped to the same set of genes, we kept only one such term. For our statistical experiments, we kept only the terms with no less than five mapped genes within the set of weights considered and hence the number of processed terms varied for each realization of sampling (see below).

2.3 Information flow in protein networks

ITM Probe (Stojmirović and Yu, 2009) is an implementation of the framework for exploring information flow in interaction networks (Stojmirović and Yu, 2007). Information flow is modeled through discrete time random walks with damping—at each step the walker has a certain probability of leaving the network. Although *ITM Probe* offers three modes: emitting, absorbing and channel, we only used the simplest, emitting mode, to provide examples illustrating issues of significance assignment. The emitting mode takes as input one or more network proteins, called sources, and a damping factor α. For each protein node in the network, the model outputs the expected number of visits to that node by random walks originating from the sources, thus highlighting the network neighborhoods of the sources. The damping factor determines the average number of steps taken by a random walk before termination: $\alpha = 1$ corresponds to no termination, while $\alpha = 0$ leads to no visits apart from the originating node. For our protein–protein interaction network examples, we used the set of all human physical interactions from the BioGRID (Breitkreutz *et al.*, 2008), version 2.0.54 (July 2009). The network consists of 7702 proteins and 56 400 unique interactions. Each interaction was represented by an undirected link. A link carries weight 2 if its two ends connect to the same protein and 1 otherwise.

2.4 Microarrays

From the NCBI Gene Expression Omnibus (GEO; Barrett *et al.*, 2009), we retrieved human microarray datasets with expression $\log_2$ ratios (weights) provided, resulting in 34 datasets and 136 samples in total. For each sample, when multiple weights for the same gene were present, we took their mean instead. This resulted in a $\mathbf{w}$ where each gene is assigned a unique raw weight. For evaluations, we also used another version of $\mathbf{w}$ where negative weights were set to zero. This version facilitated investigation of upregulation while keeping the downregulated genes as part of statistical background.

2.5 Evaluating accuracy of *P*-values

By definition, a *P*-value associated with a score is the probability of that score or better arising purely by chance. We tested the accuracy of reported *P*-values reported by enrichment methods via simulations on ‘decoy’ databases, which contained only terms with random gene assignments. For each term from the decoy dataset and each set of weights based on network or microarray data, we recorded the reported *P*-value and thus built an empirical distribution of *P*-values. If a method reports accurate *P*-values, the proportion of runs, which we term empirical *P*-value, reporting *P*-values smaller than or equal to a *P*-value cutoff, should be very close to that cutoff. We show the results graphically by plotting on the log–log scale the empirical *P*-value as a function of the cutoff.

For each given list of entities $\mathcal{N}$, be it from the target gene set of a microarray dataset or the set of participating human proteins in the interaction network, we produced two types of decoy databases. The first type was based on GO. We shuffled gene labels 1000 times. For each shuffle, we associated all terms from GO with the shuffled labels to retain the term dependency. This resulted in a database with $\sim 5\times 10^6$ terms (1000 shuffles times about 5000 GO terms). In the second type, each term, having the same size *m*, was obtained by sampling without replacement *m* genes from $\mathcal{N}$. The databases from this type (one for each term size considered) contained exactly 10^7 terms. The evaluation query set of 100 $\mathbf{w}$'s from interaction networks was obtained by randomly sampling 100 proteins out of 7702 and running *ITM Probe* with each protein as a single source. The weights for source proteins were not considered since they were prescribed, not resulting from simulation. Each run used $\alpha = 0.7$, without excluding any nodes from the network. For microarrays, the set of 136 samples was used. Since both query sets are of size $\sim 10^2$, the total number of $\mathbf{w}$—term matches was $\sim 10^9$.

2.6 Student's *t*-test (used by GAGE and T-profiler)

Similar to SaddleSum, *t*-test approaches are based on sum-of-weights score, but use the Student's *t*-distribution to infer *P*-values. As before, let w_j denote the weight associated with entity $j \in \mathcal{N}$, let $\mathcal{M}$ denote the set of *m* entities associated with a term from vocabulary and let $\mathcal{M}' = \mathcal{N} \setminus \mathcal{M}$. For any set $\mathcal{S} \subseteq \mathcal{N}$ of size $\mathcal{S}$, let $x_{\mathcal{S}} = \frac{1}{\mathcal{S}} \sum_{j\in\mathcal{S}} w_j$ denote the mean weight of entities in $\mathcal{S}$ and let $s^2_{\mathcal{S}} = \frac{1}{\mathcal{S}-1} \sum_{j\in\mathcal{S}} w_j - x_{\mathcal{S}})^2$ be their sample variance.

GAGE (Luo *et al.*, 2009) enrichment tool uses two sample *t*-test assuming unequal variances and equal sample sizes to compare the means over $\mathcal{N}$ and $\mathcal{M}$. The test statistic is

$$t = \frac{x_{\mathcal{M}} - x_{\mathcal{N}}}{\sqrt{s^2_{\mathcal{M}}/m + s^2_{\mathcal{N}}/m}} \tag{5}$$

and the *P*-value is obtained from the upper tail of the Student's *t*-distribution with degrees of freedom

$$\nu = (m-1)\frac{(s^2_{\mathcal{M}} + s^2_{\mathcal{N}})^2}{s^4_{\mathcal{M}} + s^4_{\mathcal{N}}}.$$

T-profiler (Boorsma *et al.*, 2005) compares the means over $\mathcal{M}$ and $\mathcal{M}'$ using two sample *t*-test assuming equal variances but unequal sample sizes. The pooled variance estimate is given by

$$s^2 = \frac{(m-1)s^2_{\mathcal{M}} + (n-m-1)s^2_{\mathcal{M}'}}{n-2},$$

and the test statistic is

$$t = \frac{x_{\mathcal{M}} - x_{\mathcal{M}'}}{s\sqrt{\frac{1}{m} + \frac{1}{n-m}}}.$$

The T-profiler *P*-value is then obtained from the tail of the Student's *t*-distribution with $\nu = n-2$ degrees of freedom.

2.7 Hypergeometric distribution

Methods based on hypergeometric distribution or equivalently, Fisher's exact test, use only rankings of weights and require selection of ‘significant’ entities prior to calculation of *P*-value. We first rank all entities according to their weights and consider the set $\mathcal{C}$ of *c* entities with largest weights. The number *c* can be fixed (say 50), correspond to a fixed percentage of the total number of weights, depend on the values of weights, or be calculated by other means. The score $\hat{S}$ for the term $\mathcal{M}$ is given by the size of the intersection, $\mathcal{C} \cap \mathcal{M}$, between $\mathcal{C}$ and $\mathcal{M}$. This is equivalent to setting $\hat{S} = \sum_{j\in\mathcal{M}} w_j$ with $w_j = 1$ for $j \in \mathcal{C}$ and 0 otherwise. The *P*-value for score $\hat{S}$ is

$$\mathrm{Prob}(S \geq \hat{S}) = \sum_{i=\hat{S}}^{\min(c,m)} \frac{\binom{m}{i}\binom{n-m}{c-i}}{\binom{n}{c}}.$$

Hence, the *P*-value measures the likelihood of score $\hat{S}$ or better over all possible ways of selecting *c* entities out of $\mathcal{N}$, with *m* entities associated with the term investigated.

In each of our *P*-value accuracy experiments, we used two variants of the hypergeometric method, one taking a fixed percentage of nodes and the other taking into account the values of weights. For microarray datasets, the fist variant took 1% of available genes (HGEM-PN1), while the second select genes with four fold change or more (HGEM-F2). In experiments based on protein networks, we took 3% of available proteins (231 entities) for the first variant (HGEM-PN3) and used the participation ratio formula to determine *c* in the second (HGEM-PR). Participation ratio (Stojmirović and Yu, 2007) is given by the formula

$$c = \frac{\left(\sum_{i\in\mathcal{N}} w_i\right)^2}{\sum_{j\in\mathcal{N}} w_j^2}.$$

We chose a smaller percentage of weights for microarray-based data (1 versus 3% for data derived for networks) because the microarray datasets generally contained measurement for more genes than the number of proteins in the network.

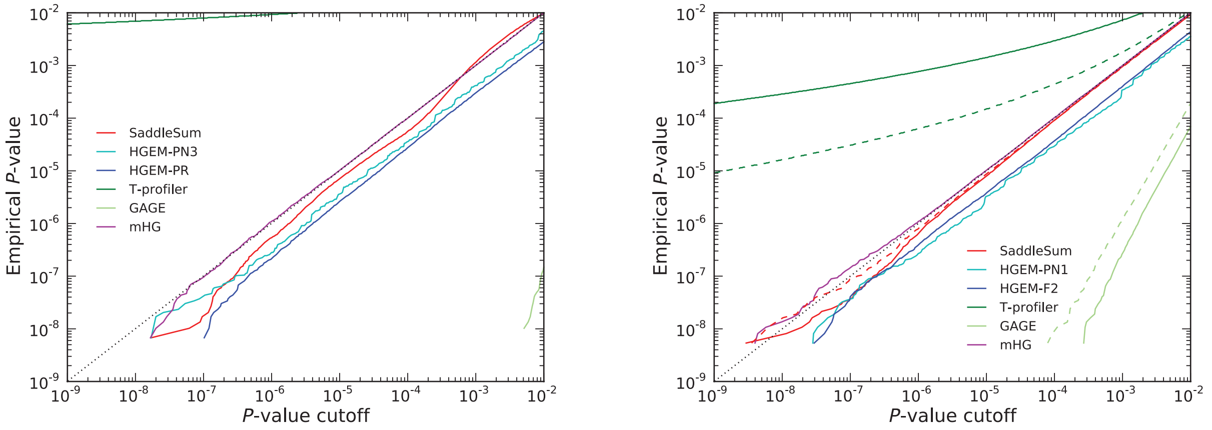

Fig. 1. Empirical P-values versus P-value cutoffs reported for investigated enrichment methods. Methods with accurate statistics have their curves follow the dotted line closely over the whole range. Each curve was constructed by aggregating the results of $\sim 10^9$ GO-based decoy term queries. Displayed on the left (right) are results using weights derived from protein network information flow simulations (microarrays). In microarray plots for SaddleSum, T-profiler and GAGE, full lines indicate the results where negative weights were set to 0, while dashed lines show the results using all weights. The reason that HGEM curves run below the theoretical line and parallel to it is that every curve is an aggregate of many curves, each of which (i) represents a single sample of weights determining parameters to be fed into hypergeometric distribution, and (ii) is a step function touching the theoretical line and dropping below it. Merging curves from many samples produces the effect seen in our plots.

2.8 mHG score

Instead of making a single, arbitrary choice of c and applying hypergeometric score, mHG method implemented in the GOrilla package (Eden *et al.*, 2009) considers all possible c's. The mHG score is defined as

$$mHG = \min_{1\leq c\leq n} \sum_{i=k}^{\min(c,m)} \frac{\binom{m}{i}\binom{n-m}{c-i}}{\binom{n}{c}},$$

where k is the number of entities annotated by the term $\mathcal{M}$ among the c top-ranked entities. The exact P-value for mHG score is then calculated by using a dynamic programming algorithm developed by Eden *et al.* (2007). For our experiments, we used an implementation in C programming language that was derived from the Java implementation used by GOrilla. The implementation uses a truncated algorithm that gives an approximate P-value with improved running speed.

2.9 Retrieval stability with respect to choice of c

To evaluate consistency of investigated methods, we compared the sets of significant terms retrieved from GO using different numbers of non-zero weights as input. For each $\mathbf{w}$, we sort in descending order the weights associated with entities. With each c selected, we kept c largest weights unchanged and set the remaining to 0 to arrive at a modified set of weights $\mathbf{w}|\mathcal{C}$. We did not totally exclude the lower weights but kept them under consideration to provide statistical background. We submitted $\mathbf{w}|\mathcal{C}$ for analysis and obtained from each statistical method a set of enriched terms ordered by their P-value. In Figure 2A and Supplementary Figure S3, we displayed the actual five most significant terms retrieved with their P-values for selected examples of weight sets. To investigate on a larger scale the retrieval stability to c changes, we computed for each method the overlap between sets of top 10 terms from two different c's for the $\mathbf{w}$ sets mentioned in 'Evaluating accuracy of P-values' and then took the average (Fig. 2B).

3 RESULTS

We compared our SaddleSum approach against the following existing methods: Fisher's exact test (HGEM; Boyle *et al.*, 2004), two sample Student's t-test with equal (T-profiler; Boorsma *et al.*, 2005) and unequal (GAGE; Luo *et al.*, 2009) variances, and mHG score (Eden *et al.*, 2007, 2009). Based on data from both microarrays and simulations of information flow in protein networks, the comparison shown here encompassed (in order of importance) evaluation of P-value accuracy, ranking stability and running time. Accurate P-value reflects the likelihood of a false identification and thus allows for comparison between terms retrieved even across experiments. Incorrect P-values therefore render ranking stability and algorithmic speed pointless. Accurate P-values without ranking stability question the robustness of biological interpretation. For pragmatic use of an enrichment method, even with accurate statistics and stability, it is still important to have reasonable speed.

3.1 Accuracy of reported P-values

The term P-value reported by an enrichment analysis method provides the likelihood for that term to be enriched within $\mathbf{w}$. To infer biological significance using statistical analysis, it is essential to have accurate P-values. We analyzed the accuracy of P-values reported by the investigated approaches through simulating $\sim 10^9$ queries and comparing their reported and empirical P-values.

Results based on querying databases with fixed term sizes are shown in Supplementary Figures S1 and S2. Shown in Figure 1 are the results for querying GO-based gene-shuffled term databases, which retain the structure of the original GO as a mixture of terms of different sizes organized as a directed acyclic graph where small terms are included in larger ones. The curves for all methods in Figure 1, therefore, resemble a mixture of curves from Supplementary Figures S1 and S2 albeit weighted toward smaller sized terms.

For weights from both network simulations and microarrays, SaddleSum as well as the methods based on Fisher's exact test (HGEM and mHG) report P-values that are acceptable (within one order of magnitude from the theoretical values). For HGEM and mHG, this is not surprising because our experiments involved shuffling entity labels and hence followed the null model of the hypergeometric distribution. On the other hand, the null model

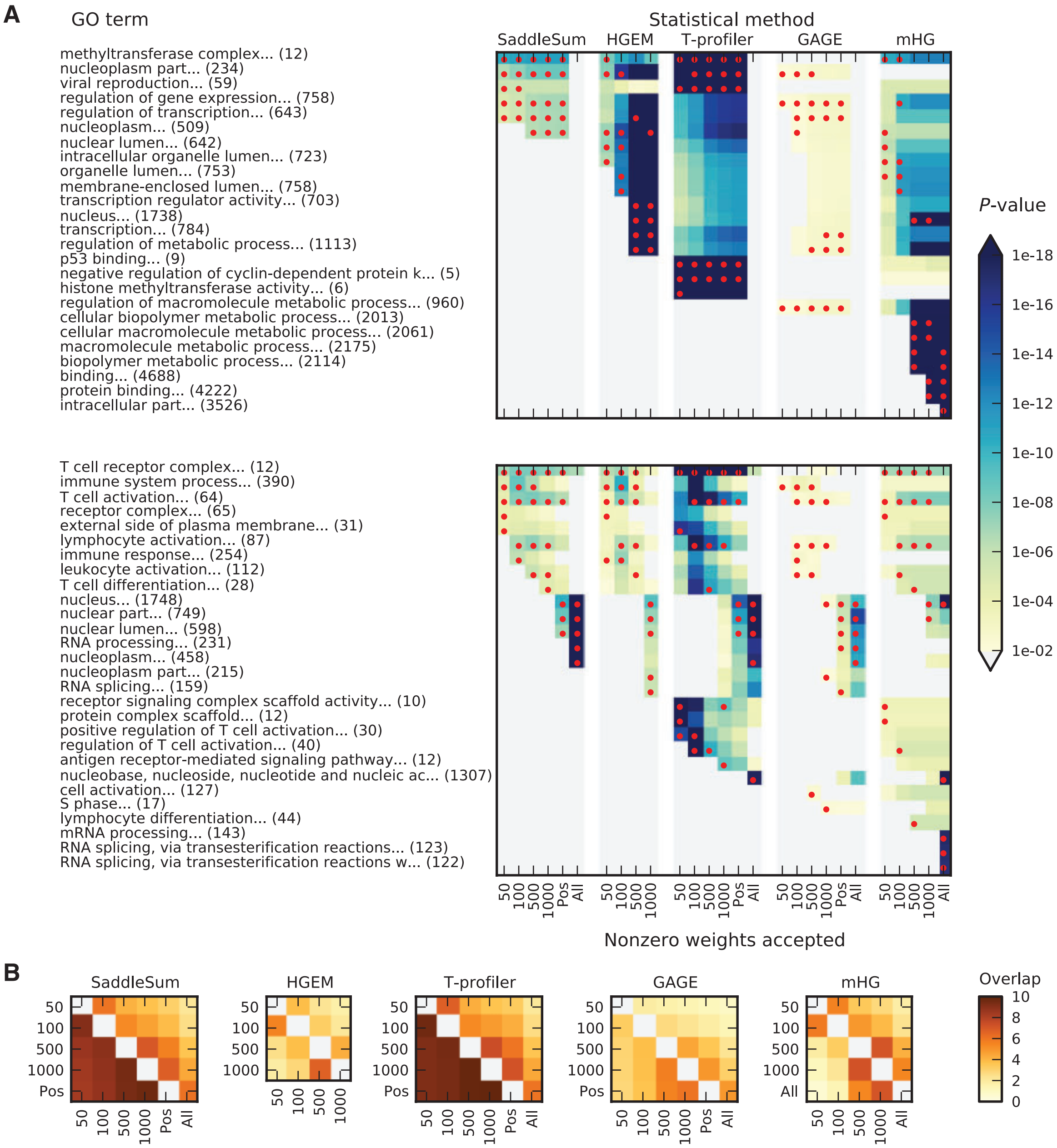

Fig. 2. *P*-value consistency and retrieval stability. (**A**) The output of *ITM Probe* emitting mode with human MLL protein (histone methyltransferase subunit) as the source (top) and the $\log_2$ ratios from the human T-cell signaling microarray GSM89756 (bottom) were processed by each of the five investigated statistical methods with varying number of weighted entities included for analysis (*All* and *Pos* include all entities; *All* uses raw weights while *Pos* sets all negative weights to 0). The *P*-values for GO terms from the union of the sets of top five hits for each method and different numbers of selected entities, are indicated by colors of the corresponding cell. Red dots show the actual top five hits for the method represented by that column. (**B**) Degree of overlap between sets of significant GO terms. Each panel corresponds to a single method with different numbers of entities used for analysis, with the results from microarray queries shown in the upper triangle and those based on network flow shown in the lower triangle. Color in each cell indicates the average pairwise overlap between the two sets of top ten entities retrieved. For example, consider the light orange colored cell (horizontally labeled by 100 and vertically labeled by 500) in the mHG panel. This indicates that on average the top ten terms retrieved by mHG using top 100 and top 500 network flow proteins share about three common terms.

of SaddleSum and the t-test methods assume weights drawn independently from some distribution (sampling with replacement). For terms with few entities ($m \leq 100$), the difference between the two null models is minimal and the P-value accuracy assessment curves for SaddleSum run as close to the theoretical line as those for HGEM methods. For $m > 100$, SaddleSum gives more conservative P-values for terms with large sums of weights (Supplementary Figs S1 and S2). In practice, this has no significant effect to biological inference. Large terms would be still selected as significant given a reasonable P-value cutoff and accurate P-values are assigned to small terms that are biologically specific.

Two-sample t-test with unequal variances as used by GAGE package reports P-values so conservative that they are often larger than 0.01 and hence not always visible in our accuracy plots. This effect persists even for m as large as 500. This might be because the number of degrees of freedom used is considerably small. In addition, its test statistic [Equation (5)] emphasizes the estimated within-term variance $s^2_{\mathcal{M}}$ that is typically larger than the overall variance $s^2_{\mathcal{N}}$.

On the other hand, T-profiler generally exaggerates (Luo *et al.*, 2009) significance because it uses the t-distribution with a large number of degrees of freedom ($n-2$). Although some small terms may appear biologically relevant (as in Fig. 2), one should not equate these exaggerated P-values with sensitivity. For microarray data, the $\log_2$ ratios are almost symmetrically distributed about 0 (Supplementary Fig. S4). The distribution of their sum is close to Gaussian. However, T-profiler still significantly exaggerates P-values for terms whose $m < 25$ (Supplementary Fig. S2). The statistical accuracy of T-profiler worsens when negative $\log_2$ ratios are set to 0. The reason for doing so is that allowing weights within each term to cancel each other may not be biologically appropriate. GO terms may cover a very general category where annotations may not always be available for more specific subterms. Subsequently, terms may get refined and new terms may emerge. In such situation, it is desirable to discover terms that have genes that are significantly upregulated even if many genes from the same term are downregulated.

3.2 Stability

P-value accuracy, although the most important criterion, measures only performance with respect to non-significant hits, that is, the likelihood of a false positive. It is also necessary to consider the quality of enrichment results in terms of the underlying biology. Testing the quality directly, as described in the introduction, is not yet feasible. Instead, we evaluated internal consistency of each method with respect to the number of top-ranked entities used for analysis. Figure 2A shows the change of P-values reported for the top five GO terms with respect to the number of selected entities using two examples with weights, respectively, from network flow simulation and microarray. Additional examples are shown in Supplementary Figure S3. Results from evaluating the overall consistency of the best 10 terms retrieved are shown in Figure 2B.

Both HGEM and mHG methods are highly sensitive to the choice of c, the number of entities deemed significant. With a small c, their sets of significant terms resemble the top terms obtained by SaddleSum, while large values of c render very small P-values for large-sized terms (often biologically non-specific). This is mainly because HGEM and mHG treat all selected significant entities as equally important without weighting down less significant entities, the collection of which may out vote the most significant ones. Hence, although mHG considers all possible c values, to obtain biologically specific interpretation, it might be necessary to either remove very large terms from the vocabulary or to impose an upper bound on c. In that respect, mHG is very similar to the original GSEA method (Mootha *et al.*, 2003), which also ignored weights. The authors of GSEA noted that the genes ranked in middle of the list had disproportionate effect to their results and produced an improved version of GSEA (Subramanian *et al.*, 2005) with weights considered.

GAGE does not show strong consistency because many P-values it reports are too conservative and fall above the 0.01 threshold we used. Consequently, the best overlap between various cutoffs is about 5 (out of 10) for network flow examples and 4 for microarray examples (Fig. 2B). T-profiler shows great internal consistency. Unfortunately, as shown in Figure 1, Supplementary Figures S1 and S2, it reports inaccurate P-values, especially for small terms. This is illustrated in the top panel of Figure 2A, where T-profiler selects as highly significant the small terms (with 5, 6 and 9 entities), which are deemed insignificant by all other methods. The same pattern can be observed in Supplementary Figure S3, although the severity is tamed for microarrays. Using weights for scoring terms, SaddleSum is also stable with respect to the choice of c but with accurate statistics.

Table 1. Running times of evaluated enrichment statistics algorithms (in seconds)

	Total running time		Average time per query	
Method	Network	Microarray	Network	Microarray
SaddleSum	558	872	0.56	0.64
HGEM	501	615	0.50	0.45
T-profiler	446	586	0.45	0.43
GAGE	499	651	0.50	0.48
mHG	2433	3407	2.43	2.51

We queried GO 10 times with each of the five examined enrichment methods using weights from 100 network simulation results and 136 microarrays (same datasets used for P-value accuracy experiments). Running times for P-value calculations on dual-core 2.8 GHz AMD Opteron 254 processors (using a single core for each run) aggregated over all samples are shown on the left, while average times per query are shown on the right. The HGEM method used 100-object cutoff.

3.3 Speed

In terms of algorithmic running time (Table 1), parametric methods relying on normal or Student's t-distribution require few computations. Methods based on hypergeometric distribution, if properly implemented, are also fast. On the other hand, non-parametric methods can take significant time if many shufflings are performed. Based on dynamic programming, mHG method can also take excessive time for large terms. SaddleSum has running time that is only slightly longer than that of parametric methods.

4 DISCUSSION

Approximating the distribution of sum of weights by saddlepoint method, our SaddleSum is able to adapt itself equally well to distributions with widely different properties. The reported P-values have accuracy comparable with that of the methods based on the hypergeometric distribution, while requiring no prior selection of the number of significant entities.

While our results show that GAGE method suffers from reduced sensitivity, it should be noted that it forms only a part of GAGE algorithm. GAGE was designed to compare two groups of microarrays (e.g. disease and control) by obtaining an overall P-value. In that scheme, the P-values we evaluated are used only for one-on-one comparisons between members of two groups. By combining one-on-one P-values (which are assumed independent), the overall P-value obtained by GAGE can become quite small.

The assumed null distribution by T-profiler (Boorsma *et al.*, 2005) is close to Gaussian. It has been commented (Luo *et al.*, 2009) that its statistics are similar to that of PAGE (Kim and Volsky, 2005), which uses Z-test. Naturally, the smallest, and likely exaggerated, P-values occur when evaluating small terms. For that reason, PAGE does not consider terms with less than 10 entities, which we included in our evaluation solely for the purpose of comparison.

Our network simulation experiments produce very different weight profiles (Supplementary Fig. S4) than that of microarrays. These weights are always positive and skewedly distributed. Even after summing many such weights, the distribution of the sum is still far from Gaussian in the tail. Therefore, T-profiler and GAGE are unable to give accurate statistics. Overall, our evaluations clearly illustrate the inadequacy, even for large terms, of assuming nearly Gaussian null distribution when the data are skewed. While Central Limit Theorem does guarantee convergence to Gaussian for large m, the convergence may not be sufficiently fast in the tail regions, which influence the statistical accuracy the most.

As presented here, SaddleSum uses given $\mathbf{w}$ both for estimating the m-dependent score distribution and for scoring each term. If a certain distribution of weights are prescribed, it is possible to adapt our algorithm to take a histogram for that distribution as input and use experimentally obtained weights for scoring only.

A possible way to improve biological relevance in retrieval is to allow for term-specific weight assignment. For example, a gene associated with a GO term can be assigned a 'NOT' qualifier to indicate explicitly that this gene product is not associated with the term considered. A way to use this information would be to change the sign of the weight for such a gene (from positive to negative or vice versa), but only when scoring the terms where the qualifier applies. Hence, potentially every term could be associated with a specific weight distribution. While all methods using weights can implement this scheme, SaddleSum is particularly suitable for it because it handles well the small terms and skewed distributions, where changing the sign for a single weight can have a considerable effect. This procedure can be generalized so that each gene in a term carries a different weight.

Several authors (Goeman and Bühlmann, 2007; Gold *et al.*, 2007; Huang *et al.*, 2009) have raised the issue of correlation between weights of entities: generally the weights of biologically related genes or proteins change together and therefore a null model assuming independence between weights may result in exaggerated P-values. In principle, a good null model is one that can bring out the difference between signal and noise. To what level of sophistication a null model should be usually is a trade-off between statistical accuracy and retrieval sensitivity. Using protein sequence comparison, for example, ungapped alignment enjoys a theoretically characterizable statistics (Karlin and Altschul, 1990) but is not as sensitive as the gapped alignment (Altschul *et al.*, 1997), where the score statistics is known only empirically because the null model allows for insertions and deletions of amino acids. Incorporating insertion and deletion into the null model made all the difference in retrieval sensitivity. This is probably because insertions/deletions do occur abundantly in natural evolution of protein sequences. The ignorance of protein sequence correlations, assumed by both ungapped and gapped alignments, does not seem to cause much harm in retrieval efficacy.

Although SaddleSum assumes weight independence and thus bears the possibility of exaggerating statistical significance of an identified term, it mitigates this issue by incorporating the entire $\mathbf{w}$ in the null distribution. It includes the entities with extreme weights that clearly represent 'signal' and not 'noise', bringing higher the tail of the score distribution and thus larger P-values. Indeed, as shown by examples in Figure 2A and Supplementary Figure S3, SaddleSum does not show unreasonably small P-values. It should also be noted that SaddleSum is designed for the simple case where a summary value is available for each entity considered—its use for analyzing complex microarray experiments with many subjects divided into several groups is beyond the scope of this article and care must be exercised when using it in this context.

SaddleSum is a versatile enrichment analysis method. Researchers are free to process appropriately their experimental data, produce a suitable $\mathbf{w}$ as input, and receive accurate term statistics from SaddleSum. Since it does not make many assumptions about the distribution of data, we foresee a number of additional applications not limited to genomics or proteomics, for example, to literature searches.

ACKNOWLEDGMENTS

This study utilized the high-performance computational capabilities of the Biowulf Linux cluster at the National Institutes of Health, Bethesda, MD. (http://biowulf.nih.gov). We thank John Wootton and David Landsman for useful comments, Roy Navon for providing us with the Java source code for the statistical algorithms of GOrilla, Weijun Luo for his help with using GAGE package and the anonymous referees for comments that helped improve the first version of this article.

Funding: Intramural Research Program of the National Library of Medicine at the National Institutes of Health.

Conflict of Interest: none declared.

REFERENCES

Al-Shahrour,F. *et al.* (2007) From genes to functional classes in the study of biological systems. *BMC Bioinformatics*, **8**, 114.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

Backes,C. *et al.* (2007) GeneTrail–advanced gene set enrichment analysis. *Nucleic Acids Res.*, **35**, W186–W192.

Barrett,T. *et al.* (2009) NCBI GEO: archive for high-throughput functional genomic data. *Nucleic Acids Res.*, **37**, D885–D890.

Ben-Shaul,Y. *et al.* (2005) Identifying subtle interrelated changes in functional gene categories using continuous measures of gene expression. *Bioinformatics*, **21**, 1129–1137.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc.*, **57**, 289–300.

Blom,E.-J. *et al.* (2007) FIVA: functional information viewer and analyzer extracting biological knowledge from transcriptome data of prokaryotes. *Bioinformatics*, **23**, 1161–1163.

Boorsma,A. *et al.* (2005) T-profiler: scoring the activity of predefined groups of genes using gene expression data. *Nucleic Acids Res*., **33**, W592–W595.

Boyle,E.I. *et al.* (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**, 3710–3715.

Breitkreutz,B. *et al.* (2008) The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res.*, **36**, D637–D640.

Breitling,R. *et al.* (2004) Iterative group analysis (iGA): a simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, **5**, 34.

Breslin,T. *et al.* (2004) Comparing functional annotation analyses with Catmap. *BMC Bioinformatics*, **5**, 193.

Daniels,H.E. (1954) Saddlepoint approximations in statistics. *Ann. Math. Stat.*, **25**, 631–650.

Daniels,H.E. (1987) Tail probability approximations. *Internat. Stat. Rev.*, **55**, 37–48.

Eden,E. *et al.* (2007) Discovering motifs in ranked lists of DNA sequences. *PLoS Comput. Biol.*, **3**, e39.

Eden,E. *et al.* (2009) GOrilla: a tool for discovery and visualization of enriched go terms in ranked gene lists. *BMC Bioinformatics*, **10**, 48.

Goeman,J. and Bühlmann,P. (2007) Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics*, **23**, 980–987.

Gold,D. *et al.* (2007) Enrichment analysis in high-throughput genomics - accounting for dependency in the NULL. *Brief. Bioinform*., **8**, 71–77.

Hochberg,Y. and Tamhane,A.C. (1987) *Multiple Comparison Procedures (Wiley Series in Probability and Statistics)*. Wiley, New York.

Huang,D.W. *et al.* (2009) Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res*., **37**, 1–13.

Karlin,S. and Altschul,S.F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl Acad. Sci. USA*, **87**, 2264–2268.

Kim,S.-Y. and Volsky,D.J. (2005) PAGE: parametric analysis of gene set enrichment. *BMC Bioinformatics*, **6**, 144.

Lugannani,R. and Rice,S. (1980) Saddle point approximation for the distribution of the sum of independent random variables. *Adv. Appl. Probab.*, **12**, 475–490.

Luo,W. *et al.* (2009) GAGE: generally applicable gene set enrichment for pathway analysis. *BMC Bioinformatics*, **10**, 161.

Mootha,V.K. *et al.* (2003) PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, **34**, 267–273.

Sharma,K. *et al.* (2009) Proteomics strategy for quantitative protein interaction profiling in cell extracts. *Nat. Methods*, **6**, 741–744.

Smid,M. and Dorssers,L.C.J. (2004) GO-Mapper: functional analysis of gene expression data using the expression level as a score to evaluate gene ontology terms. *Bioinformatics*, **20**, 2618–2625.

Stojmirović,A. and Yu,Y.-K. (2007) Information flow in interaction networks. *J. Comput. Biol.*, **14**, 1115–1143.

Stojmirović,A. and Yu,Y.-K. (2009) ITM Probe: analyzing information flow in protein networks. *Bioinformatics*, **25**, 2447–2449.

Subramanian,A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA*, **102**, 15545–15550.

Sultan,M. *et al.* (2008) A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, **321**, 956–960.