Artificial Dilution Series
Supplementary Information

Ilya Plyusnin, Liisa Holm, Petri Törönen

1 Introduction

This is the supplementary text for the main article, "Novel Comparison of Evaluation Metrics for Gene Ontology Classifiers Reveals Drastic Performance Differences". Here we compare this work to related previous research. We give in-detail descriptions of the compared evaluation metrics. We also give definitions of signal and noise in the Artificial Dilution Series (ADS) and describe the tested methods for combining semantic similarities.

We also discuss some alternative evaluation metrics from the literature. We want to underline that we had to limit the number of compared evaluation metrics and we included only the most relevant ones and their variations to comparison.

Comparing ADS with related research

To our knowledge, there has been little research on evaluation metrics with Gene Ontology. Clark and Radivojac compare seven methods by looking at the threshold position that optimizes each evaluation metric, and explore if the selected position is rational for classification purposes [1]. GO Semantic similarities [2] have been evaluated using correlation against other datasets, like sequence similarity [3], but again their performance as classifier evaluation metric has not been thoroughly tested. These tests lack the clear reference point, as the ADS noise level in our work, for comparing EvMs.

Machine learning field has a large and thorough comparison by Ferri et al. [4]. They test large number of evaluation metrics under different challenging situations with artificial datasets. Sokolova and Lapalme discuss on how measures are invariant towards the changes in the classifier results [5]. Seliya et al. and Ferri et al. compare similarities between evaluation measures [4, 6]. These comparisons, however, use mainly artificial datasets and focus on few challenging features at the time. Real data, on the other hand, consists a combination of the challenging features.

Furthermore, the classification structures, used in previous machine learning articles, differ significantly from GO. These articles discuss almost solely either binary or multinomial classification tasks, where each item cannot be correctly classified to many classes simultaneously. With GO prediction we have a task that reminds more multiple binary classification, where item can correctly belong to many classes or it might not belong to any of the available classes. In addition, GO prediction is further complicated by the correlations, created by the hierarchical structure of GO. All this creates more challenges to used EvMs.

In ADS we alter the correct gene annotation dataset by mixing a controlled proportion of lightly permuted data, *signal*, and heavily permuted data, *noise* (see fig. 1). Here signal represents true positive predictions and noise false positive predictions. We repeat this process multiple times with the same signal proportion, creating multiple datasets with the same signal level. Next, we repeat this process while we alter the signal proportion. This creates a *Dilution Series* that moves in step-wise manner from full signal data to full noise data.

Next, we add random scores from normal distribution to each prediction. This is our simulated classifier prediction score. We also generate separate smaller fully random dataset (negative dataset) with lower simulated classifier prediction scores to create a gradient in prediction accuracy over classifier scores. This represents false positive predictions that, unlike the noise above, can be excluded from the positive set with a threshold on prediction score. Finally, the set with controlled signal proportion and the negative datasets are merged and extended to parental nodes in GO structure.

The generated datasets can be used to test different evaluation metrics to see how well they separate different signal levels. We then monitor the performance of evaluation metrics analyzing rank correlation between the used

signal level and evaluation metric. Artificial Dilution Series has the following benefits:

1. We start with real-life data (with correct classifications), and propagate predictions to parental nodes with real Gene Ontology structure. This way we keep the real correlation structures and class size differences.
2. We know exactly how the different datasets should be ranked by evaluation metric. This simplifies analysis of the results.
3. Permuted datasets allow repetitive testing of evaluation metrics with large number of datasets having the same amount of error.
4. Dilution series lets us see the separation between intermediate amounts of noise. Not only difference between 0 vs 100% noise, as has been standard in the statistics and data mining research.

Points 2. and 3. have not been available in previous evaluations with real datasets [1, 3]. There analysis has usually been based on correlation between different data types and lacked repetitive tests that we present. Point 1. is difficult to replicate with fully artificial datasets, as all the features of real datasets are not always known. Furthermore, point 4 has not been used in this kind of testing before. Our results support point 4 and show that intermediate signal levels reveal more weaknesses in compared metrics than just the first and last signal level.

2 Definition of Signal and Noise in the ADS

We represent here a toy example on the definition of signal and noise in ADS system in fig. 1. Figure shows a case where gene has a one correct GO class. A signal class would be selected from the immediate neighborhood of the GO class. This is shown by a circle in our figure 1. Size of the circle is parameter in our rotation step. Noise class, on the other hand, is required to have very little overlapping tree path with the correct GO class. Therefore only the GO classes inside the box on the right would qualify as noise classes. This is the requirement that the noise classes must fulfill, when classes are swapped for noise.

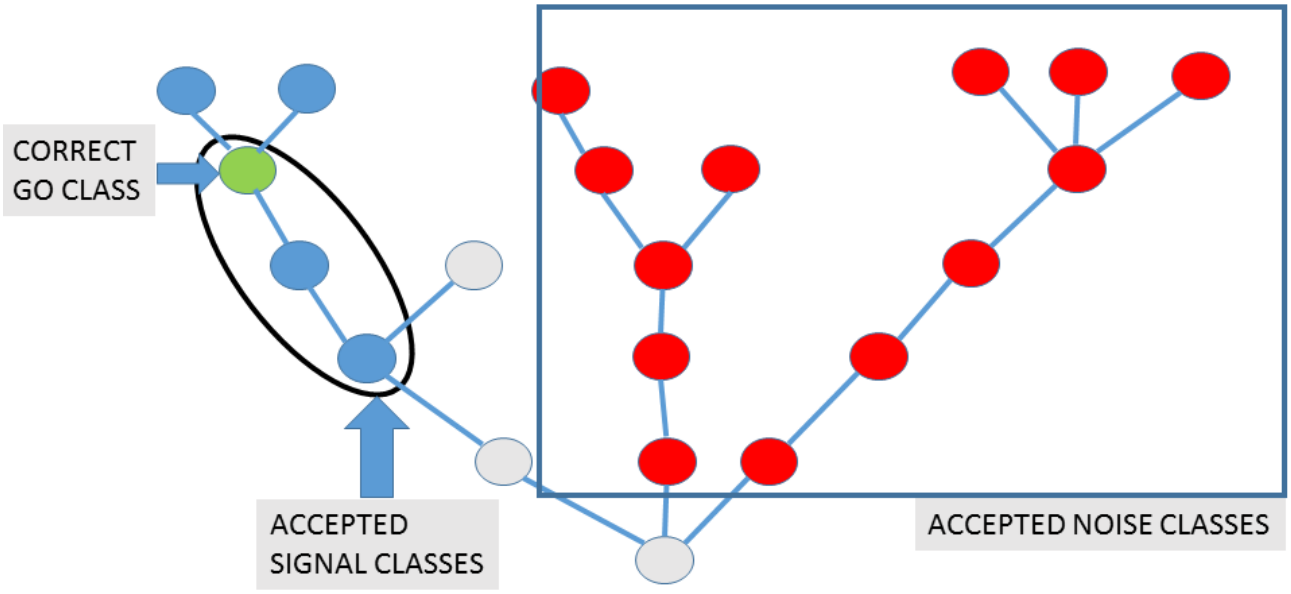

Fig 1. Signal and Noise definition Figure shows an example how signal and noise are defined in ADS. We have correct GO class, shown in green, in left branch. It's neighborhood in GO tree is shown in blue and we select either class itself or one of nearest parents as signal class. Noise class would be selected from distant branches, demonstrated with a red nodes.

Article	Method evaluated	Publ.year	Evaluation Metric
Martin et al. [7]	Gotcha	2004	Selectivity vs. p-value coverage vs. p-value
Engelhardt et al. [8]	Sifter	2005	ROC curve error percentages
Götz et al. [9]	BLAST2GO	2005	Accuracy with threshold
Friedberg [10]	AFP 2005	2006	Resnik semantics
Hawkins et al. [11]	PFP	2008	Schlicker semantics
Wass et al. [12]	ConFunc	2008	Prec-Recall curve
Chitale et al. [13]	ESG	2009	Schlicker semantics Precision and Recall values
Engelhardt et al. [14]	Sifter v.2	2011	ROC curve Prec-Recall curve
Fontana et al. [15]	ARGOT	2012	Prec-Recall curve
Minneci et al. [16]	FFPred 2.0	2013	Fmax Prec-Recall curve SimGIC
Radivojac et al. [17]	CAFA1	2013	Fmax Prec-Recall curve weighted Prec–Recall curves TC AUC
Gillis, Pavlidis [18]	CAFA1, re-eval.	2013	Prec-Recall curve, TC AUC Resnik semantics Lin semantics
Koskinen et al. [19]	PANNZER	2015	weighted Prec-Recall curve Lin semantics
Kahanda et al. [20]	CAFA2, re-eval.	2015	Fmax, TC Fmax
Jiang et al. [21]	CAFA2	2016	Fmax, Prec-Recall curve, Smin, TC AUC

Table 1. Overview of some Evaluation Metrics used in AFP literature. Table represents selected AFP method papers, AFP competition papers and re-evaluations of competitions (re-eval.) Prec-Recall refers to Precision Recall. Semantics refers to semantic similarity measure. Other abbreviations are explained under the Evaluation Metrics section in Materials and Methods. Notice the variety of metrics used.

3 Evaluation metrics tested with ADS

This chapter describes evaluation metrics for Gene Ontology (GO) classifiers (hereinafter referred as "metrics") that we tested. These included metrics previously described in the literature as well as novel modifications of these metrics. Table 1 shows a collection of currently used metrics. In total we describe here 37 metrics including variations of Jaccard correlation, Receiver Operating Characteristics (ROC) Area Under Curve (AUC), precision-recall (PR) AUC, Fmax, Smin [22], SimUI [23], SimGIC [24], Resnik [25] and Lin [26]. We also test six different methods for summarizing pairwise GO-term similarities. Supplementary table 1 presents all these metrics and their abbreviations.

3.1 General notations for metric definitions

We assume that a GO classifier outputs a set of gene annotations with scores. More technically GO classifier outputs a set of triples each consisting of three values: gene label g, GO class go and a classifier prediction score sc. We refer to this entire set of predicted gene annotations as P. A subset of P containing all annotations for gene $g = i$ is denoted P_i and a subset of P containing all annotations with GO class $go = j$ is denoted P_j. Subset of P with annotations scored greater or equal to threshold $sc \geq th$ is denoted P^{th}. Subscripts i and j and superscript th can be combined to denote subsets of P or other annotation sets with desired conditions (e.g. P_i^{th} is a subset of all annotations for gene i with scores equal or greater then th). Finally, let n denote the number of unique genes in P, $n(th)$ the number of unique genes in P^{th}, m the number of unique GO classes in P and $m(th)$ the number of unique GO classes in P^{th}.

Let's further assume that we also have a set of correct or true gene annotations T. Given T and P we can define true positive annotations, $TP = P \cap T$, false positive annotations, $FP = P \setminus T$, and false negative annotations, $FN = T \setminus P$.

For ROC AUC metrics we will also need the set of true negative annotations N. We define true negatives as the Cartesian product of the set of genes in T and the set of GO classes in GO subtracted by the correct set T:

$N = genes(T) \times classes(GO) \setminus T$

3.2 Simple metrics based on Jaccard index

Jaccard index or Jaccard similarity coefficient between predicted P and true T annotation sets at threshold th is defined as

$$Jacc(P, T, th) = \frac{|P^{th} \cap T|}{|P^{th} \cup T|} = \frac{|TP^{th}|}{|TP^{th}| + |FP^{th}| + |FN^{th}|}$$

Jaccard index is a function of threshold th. In order to convert this into a scalar metric we can take maximum value over all thresholds. Basic Jaccard formula can be applied to evaluate a set of predicted annotations P in three ways. First, we can treat all annotations $x \in P$ equally disregarding any grouping of individual annotations by shared genes or GO classes. This is our definition of unstructured Jaccard, $USJacc$

$$US\ Jacc = \max_{th} \frac{|TP^{th}|}{|TP^{th}| + |FP^{th}| + |FN^{th}|} \tag{1}$$

Second, we can calculate Jaccard index separately for each gene and take the average of these values. This metric has been proposed earlier as $SimUI$ by Gentleman [23]. To remain consistent in our terminology we refer to this metric as Gene-Centric Jaccard, $GCJacc$:

$$GC\ Jacc = \max_{th} \frac{1}{n(th)} \sum_{i=1}^{n} \frac{|TP_i^{th}|}{|TP_i^{th}| + |FP_i^{th}| + |FN_i^{th}|} \tag{2}$$

Third, we can interpret predictions P as assignment of genes to GO classes. In this case we calculate Jaccard index separately for each GO class in P. This is our definition of Term-Centric Jaccard (omitted from analysis):

$$TC\ Jacc = \max_{th} \frac{1}{m(th)} \sum_{j=1}^{m} \frac{|TP_j^{th}|}{|TP_j^{th}| + |FP_j^{th}| + |FN_j^{th}|}$$

Note that we propagate the annotations to ancestor nodes with these Jaccard methods.

3.3 Metrics based on ROC and PR curves

When evaluating a binary classifier that produces a list of scored predictions we are generally interested in estimating the level of false positive and false negative errors. Generally the number of false positives and false negatives depend on the score threshold and are in inverse relation to each other: by increasing threshold the number of false negatives decreases but the number of false positives increases. In classical ROC analysis this balance is quantified by plotting True Positive Rate (TPR) against False Positive Rate (FPR):

$$TPR(P,T,th) = \frac{|P^{th} \cap T|}{|T|} = \frac{|TP^{th}|}{|TP^{th}| + |FN^{th}|}$$

$$FPR(P,T,th) = \frac{|P^{th} \setminus T|}{|N|} = \frac{|FP^{th}|}{|FP^{th}| + |TN^{th}|}$$

Performance is quantified as area under ROC curve (ROC AUC). AUC is closely related to Mann-Whitney U-statistic and is an estimate of the probability that a binary classifier will rank an instance of the positive class higher than an instance of a negative class [27]. Let S denote a set of correct annotations, j the positive subset and k the negative subset. Let $rank(x, S)$ denote ranks assigned by the classifier to $x \in S$.

Then ROC AUC for positive class j and negative class k is:

$$AUC(rank, j, k) = \frac{1}{|j||k|}\Big(\sum_{x \in j} rank(x, S) - \frac{|j|(|j|+1)}{2}\Big)$$

AUC is a metric for binary classification. Here we consider a number of ways to extend this to a multi-class problem. Hand and Till [27] define the M metric, which is an arithmetic average of pairwise class comparisons. When applied to GO this means an arithmetic average of all pairs of GO classes in T. Ferri et all [4] define the AUNU metric, which is an arithmetic mean of one-vs-all class comparisons. This definition is similar to that used in CAFA2 competition [28] where j was set to one of GO-classes in T and k to a union class covering all genes not in j. Further, we suggest that j can be set cover all annotations in T and k to all other possible annotations i.e. the N set. These approaches lead to different AUC variants.

First, let us define positive class as the entire set of correct annotations T and negative class as all other possible annotations N. We refer to this metric as the unstructured AUC:

$$US\ AUC(rank, T, N) = \frac{1}{|T||N|}\Big(\sum_{x \in T} rank(x, T \cap N) - \frac{|T|(|T|+1)}{2}\Big) \tag{3}$$

Second, we calculate similar metric gene-wise. We refer to this metrics as the gene-centric AUC:

$$GC\ AUC(rank, T, N) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|T_i||N_i|}\Big(\sum_{x \in T_i} rank(x, T_i \cap N_i) - \frac{|T_i|(|T_i|+1)}{2}\Big) \tag{4}$$

Third variant is the arithmetic mean of one-vs-all class comparisons given in general terms by Ferri et all [4] (i.e. the AUNU metric) and applied to GO-classifiers in CAFA competitions. We refer to this as the term-centric AUC. Let $genes_j$ denote the subset of genes in T annotated with GO-class j and let n_j denote the size of that subset. Then term-centric AUC is:

$$TC\ AUC(P,T) = \frac{1}{m}\sum_{j=1}^{m} \frac{1}{n_j * (n - n_j)}\Big(\sum_{x \in genes_j} rank(x, genes) - \frac{n_j * (n_j + 1)}{2}\Big) \tag{5}$$

Another method to quantify the balance between false positives and false negatives is to plot precision against recall. Precision, $pr(th)$, is defined as the fraction of true positives from predicted and recall (syn. sensitivity), $rc(th)$, as the fraction of true positives from correct annotations:

$$pr(P,T,th) = \frac{|TP^{th}|}{|TP^{th}| + |FP^{th}|}$$

$$rc(P,T,th) = \frac{|TP^{th}|}{|TP^{th}| + |FN^{th}|}$$

Area under precision-recall curve (AUCPR) is a scalar metric that reflects classifiers overall accuracy. Several authors have argued that AUCPR is better suited for classification involving class imbalance problem (for a discussion see [29]). We note that both *US AUC* and *GC AUC* have negative class manyfold larger then the positive class and are thus susceptible to class imbalance.

In our implementation we calculated AUCPR from $pr(th)$ and $rc(th)$ curves using trapezoidal method [29]. As with ROC AUC, basic AUCPR formula can be applied to evaluated a set of predicted GO annotations in three ways. For unstructured AUCPR we treat all $x \in P$ equally disregarding any grouping of individual annotations by shared genes or GO classes. For gene-centric AUCPR we calculate pr and rc separately for each gene and calculate the arithmetic mean of these values prior to AUC calculus. For term centric AUCPR we calculate pr and rc for each GO class separately and calculate the arithmetic mean of these values prior to AUC calculus. Definitions for variations of precision, recall and AUCPR are then

$$pr_{us}(th) = \frac{|TP^{th}|}{|TP^{th}| + |FP^{th}|}$$

$$rc_{us}(th) = \frac{|TP^{th}|}{|TP^{th}| + |FN^{th}|}$$

$$US\ AUCPR(P,T) = AUC(pr_{us}(th), rc_{us}(th)) \tag{6}$$

$$pr_{gc}(th) = \frac{1}{n(th)} \sum_{i=1}^{n} \frac{|TP_i^{th}|}{|TP_i^{th}| + |FP_i^{th}|}$$

$$rc_{gc}(th) = \frac{1}{n(th)} \sum_{i=1}^{n} \frac{|TP_i^{th}|}{|TP_i^{th}| + |FN_i^{th}|}$$

$$GC\ AUCPR(P,T) = AUC(pr_{gc}(th), rc_{gc}(th)) \tag{7}$$

$$pr_{tc}(th) = \frac{1}{m(th)} \sum_{j=1}^{m} \frac{|TP_j^{th}|}{|TP_j^{th}| + |FP_j^{th}|}$$

$$rc_{tc}(th) = \frac{1}{m(th)} \sum_{j=1}^{m} \frac{|TP_j^{th}|}{|TP_j^{th}| + |FN_j^{th}|}$$

$$TC\ AUCPR(P,T) = AUC(pr_{tc}(th), rc_{tc}(th)) \tag{8}$$

F-metric is a simple metric based on precision and recall, or the harmonic mean of precision and recall at fixed threshold. It's maximum over all the thresholds, Fmax metric, is a popular implementation of it, and has been used extensively in CAFA competitions [17, 28] Note that the precision and recall values are calculated gene-wise at each threshold th and average values $pr_{gc}(th)$ and $rc_{gc}(th)$ are used:

$$Fmax = \max_{th} \frac{2 \times pr_{gc}(th) \times rc_{gc}(th)}{pr_{gc}(th) + rc_{gc}(th)} \tag{9}$$

Ferri et al. [4] considered an alternative usage for F-measure, where an average is calculated across all predicted classes. This would correspond to our Class Centric analysis. In their analysis the threshold was predefined for the analysis.

3.4 Information content

One major challenge with GO structure is the variation in the class sizes we see in the data. This causes the naive prediction of largest GO classes perform well on comparisons (see main article). This problem can be corrected by adding weights, called *Information Contents* to the class predictions that emphasize the smaller more meaningful classes. Resnik defined the information content of an individual GO class x as the negative log likelihood of x [25]. Likelihoods $p(x)$ for GO classes can be estimated from a corpus of GO annotations such as UniProt Gene Ontology Annotation database (UniProt-GOA). Thus information content of a GO class x can be defined in terms of it's frequency $f(x)$ in a given database:

$$ic(x) = log\frac{1}{p(x)} = log\frac{1}{f(x)}$$

Clark and Radivojac defined information content of GO subgraph G as a negative log likelihood of G [22]. Let $\mathcal{P}(x)$ denote the set of immediate parents of x in the GO graph. Then the likelihood of subgraph G can be factorized as a product of conditional probabilities, $p(G) = \prod_{x \in G} p(x|\mathcal{P}(x))$ [22]. Conditional probabilities $p(x|\mathcal{P}(x))$ and a quantity corresponding to $ic(x)$ can be estimated from a given database (e.g. from UniProt-GOA):

$$ic2(x) = log\frac{1}{p(x|\mathcal{P}(x))}$$

Information content of a subgraph G is then defined as a sum of $ic2(x)$ values for all $x \in G$:

$$ic(G) = log\frac{1}{p(G)} = \sum_{x \in G} ic2(x)$$

For the sake of comparison we also used a simpler definition of subgraph information content that is based on $ic(x)$ values:

$$ic'(G) = \sum_{x \in G} ic(x)$$

We note that $ic'(G)$ is a simplification that treats probabilities on individual nodes in G as independent of each other, which is clearly not the case. The motivation for using both $ic(G)$ and $ic'(G)$ is to evaluate the extent to which this simplification will effect the utility of metrics incorporating these definitions. Furthermore, the semantic similarities, used here, require simpler ic score. Whenever both definitions are used we label metrics based on $ic(G)$ with "ic2." prefix and metrics based on $ic'(G)$ with "ic." prefix. The logic here is to label metrics according to the $ic(x)$ or $ic2(x)$ values that are used in there calculus.

3.5 Metrics incorporating information content

Pesquita et al introduced an ic weighted Jaccard index [24]:

$$SimSIG(P,T) = \max_{th} \frac{1}{n} \sum_{i=1}^{n} \frac{ic(TP_i^{th})}{ic(TP_i^{th}) + ic(FP_i^{th}) + ic(FN_i^{th})} \tag{10}$$

Note that gene-wise quotients in this equation are ratios that can get similar values for genes with very different number of annotations. This can overweight genes that have few annotations and underweight genes with many annotations. To overcome this limitation we introduce a modified version that treats all annotations equally as in unstructured AUC and Jaccard metrics:

$$SimSIG2(P,T) = \max_{th} \frac{ic(TP^{th})}{ic(TP^{th}) + ic(FP^{th}) + ic(FN^{th})} \tag{11}$$

Clark and Radivojac defined remaining uncertainty, ru, as the average gene-wise information content of the FN set and misinformation, mi, as the average gene-wise information content of FP set. Based on these, they defined Smin metric (labeled here as $Smin1$).

$$ru(th) = \frac{1}{n}\sum_{i=1}^{n} ic(FN_i^{th}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{x \in FN_i^{th}} ic2(x)$$

$$mi(th) = \frac{1}{n}\sum_{i=1}^{n} ic(FP_i^{th}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{x \in FP_i^{th}} ic2(x)$$

$$Smin1(P,T) = \min_{th} \sqrt{ru(th)^2 + mi(th)^2} \tag{12}$$

Note that $Smin1$ has no gene-centric terms in it's equation. At the core of this metric is the sum of $ic(FN_i^{th})$ and $ic(FP_i^{th})$ terms across all genes. This makes $Smin1$ insensitive to distribution of misclassification errors across genes and is by this property similar to other *unstructured* metrics. To test how gene-centric terms would effect this metric we introduce a variation of $Smin1$ by changing the order of arithmetic average and Euclidean distance calculus:

$$Smin2(P,T) = \min_{th} \frac{1}{n}\sum_{i=1}^{n} \sqrt{ic(FN_i^{th})^2 + ic(FP_i^{th})^2} \tag{13}$$

While considering possible variations for $Smin$ calculus we noticed that $1/n$ terms can be moved outside the squere root. Thus, by multiplying $Smin1$ by n (the number of genes) we can define a simpler metric that has the same variance and ADS performance as $Smin1$:

$$\begin{aligned} Smin1(P,T) &= \min_{th} \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} ic(FN_i^{th})\right)^2 + \left(\frac{1}{n}\sum_{i=1}^{n} ic(FP_i^{th})\right)^2} \\ &= \frac{1}{n}\min_{th} \sqrt{ic(FN^{th})^2 + ic(FP^{th})^2} \end{aligned}$$

$$Smin3(P,T) = Smin1(P,T) * n = \min_{th} \sqrt{\left(ic(FN^{th})^2 + ic(FP^{th})^2\right)}$$

Furthermore, the CAFA organizers have considered a weighted version of F metric [28]. This uses sums of Information Content weights instead of counts of predictions. We excluded this from our current comparisons as we had to limit the number of compared metrics.

3.6 Metrics based on pairwise semantic similarities

Metrics listed to this point will only differentiate between exact matches and mismatches of predicted versus correct GO classes. However, using semantic similarity measures it is possible to define metrics that are aware of the relationships between mismatching GO classes in the ontology tree [24].

Semantic similarity metrics are generally gene-centric and have at their core pairwise comparison of predicted GO classes against correct GO classes. We implemented three core functions: *Resnik* [25], *Lin* [26] and Ancestor Jaccard index ($AJacc$). Let $x \in GO$ and $y \in GO$ be any two nodes in GO graph. And let $A(x)$ and $A(y)$ denote the sets of all ancestors in GO for nodes x and y. Then the most informative common ancestor of x and y is defined by $MICA(x,y)$ [24] (equivalent to *minimum subsummer* defined by Lord [30]):

$$MICA(x,y) = \operatorname*{argmax}_{z \in A(x) \cap A(y)} ic(z)$$

The three core semantic similarity metrics are then defined by equations:

$$Resnik(x,y) = ic(MICA(x,y))$$

$$Lin(x,y) = \frac{2 \times ic(MICA(x,y))}{ic(x) + ic(y)}$$

$$AJacc(x,y) = \frac{|A(x) \cap A(y)|}{|A(x) \cup A(y)|}$$

Now let X denote the ordered list of predicted GO classes for gene i at threshold th and Y the list of correct GO classes. Predicted GO classes are ordered using the classifiers prediction score. Let $sem(x,y)$ be any core semantic similarity function between an arbitrary pair of GO classes $x \in GO$ and $y \in GO$. Then pairwise semantic similarities for gene i at threshold th can be summarized by a similarity matrix, with rows standing for predicted and columns for correct GO classes:

$$SIM(k,l) = sem(X(k), Y(l))$$

See table 1 for an example on this.

3.7 Combining semantic similarities into a single score

To devise a scalar scoring function, we need to summarize SIM matrix into a scalar value (see table 1). We propose six ways to do this: mean value of SIM (method A), mean value of column maxima of SIM (method B), mean value of row maxima of SIM (method C), mean value of methods B and C (method D), min value of methods B and C (method E), mean value of pooled column and row maxima of SIM (method F).

$$A(X,Y,sem) = \frac{1}{|X| \times |Y|} \sum_{x \in X} \sum_{y \in Y} sem(x,y)$$

$$B(X,Y,sem) = \frac{1}{|Y|} \sum_{y \in Y} \max_{x \in X} sem(x,y)$$

$$C(X,Y,sem) = \frac{1}{|X|} \sum_{x \in X} \max_{y \in Y} sem(x,y)$$

$$D(X,Y,sem) = \frac{B(X,Y,sem) + C(X,Y,sem)}{2}$$

$$E(X,Y,sem) = \min\{B(X,Y,sem), C(X,Y,sem)\}$$

$$F(X,Y,sem) = \frac{1}{|X| + |Y|} \Big(\sum_{x \in X} \max_{y \in Y} sem(x,y) + \sum_{y \in Y} \max_{x \in X} sem(x,y) \Big)$$

Method A treats all pairwise comparisons equally and is equivalent to the all-pair arithmetic average proposed by Lord [30]. Method B is the best-match average for correct GO-classes. As such this method is sensitive to false negative errors and insensitive to false positive errors. Method C is the best-match average for predicted classes and is thus sensitive to false positive errors and insensitive to false negative errors (see discussion in the next section). Methods B and C are asymmetrical and this is corrected with method D. Best-match average methods were previously described by several authors [24, 31–33].

As B can only monitor false negatives and C only the false positives, they are expected to represent weak performance in the analysis. We include them into the analysis as negative controls to see if ADS can detect their performance differences. We further introduce a novel methods E, that by definition monitors both false negatives and false positives.

Methods A to F are functions of threshold th and gene i. To convert these into scalar values for the predicted annotation P we calculate the selected method for each threshold th and select the maximum value across all possible thresholds. Let sem be any core semantic similarity function and S any summation method for the similarity matrix. Semantic similarity metric EM_{sem} is then:

$$EM_{sem}(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} S(P_i^{th}, T_i, sem)$$

We combined three sem functions with the six S functions to defined altogether 18 variations of semantic similarity metrics. In the following section we give equations for EM_{sem} based on $Resnik$. Equations for EM_{sem} based on Lin and $AJacc$ are similar.

$$Resnik\ A(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} A(P_i^{th}, T_i, Resnik) \tag{14}$$

$$Resnik\ B(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} B(P_i^{th}, T_i, Resnik) \tag{15}$$

$$Resnik\ C(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} C(P_i^{th}, T_i, Resnik) \tag{16}$$

$$Resnik\ D(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} D(P_i^{th}, T_i, Resnik) \tag{17}$$

$$Resnik\ E(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} E(P_i^{th}, T_i, Resnik) \tag{18}$$

$$Resnik\ F(P,T) = \max_{th} \frac{1}{n(th)} \sum_{i}^{n} F(P_i^{th}, T_i, Resnik) \tag{19}$$

3.8 Comparing semantic similarity summation methods

We demonstrate the principle summation methods for semantic similarities with a toy dataset in table 1. We explain strengths and weaknesses of each method especially when the final score is taken to be the maximum score over threshold th positions.

First, it is easy to see that the average of column maxima is expected to grow as the threshold goes lower even with a randomly sampled matrix, as each maximum is selected from larger set of numbers. This means that when the maximum score over th for method B is selected, it will be always at lowest positions. So method B clearly *cannot monitor false positives*. Therefore, average of row maxima would do better job at separating random matrix from one with signal at the lower th positions.

Situation is totally opposite at the highest threshold positions. When only a single row is selected from randomly sampled matrix, one strong value can alter the result for row maximums. This means that the maximum score over th with method C will most likely at higher th positions. However, now we are not paying attention on how many correct GO classes get annotated. Therefore method C clearly *cannot monitor false negatives*.

Method A has a different problem. If there is only one GO class in the correct set, then it is possible to obtain good score for correct prediction at high th position. However, assume that a gene has ten correct GO classes and these are very dissimilar from each other in GO tree. Now if classifier predicts the very same GO classes in top-10 positions, 90% of the values in the matrix will represent dissimilarities between GO-classes. Therefore method A is clearly *affected by the number and the heterogeneity of the correct GO classes*.

Taking an average of B and C, like in method D, mixes weaker and better signal. This might not be good choice when maximum score over th is selected. Indeed, our results show that D is quite weak method.

Concatenated row and column maxima, as in method F, will always pay more attention to longer vector of maxima. It will pay more attention to column maxima at the higher threshold positions and more attention row maxima at the lower threshold positions. Selecting the minimum of B and C, as in method E, requires that both mean of row and column maxima show strong signal. This is more challenging to row maxima at lower positions and to column maxima at higher positions. Therefore we propose methods E and F as new summation methods for semantic similarities.

				Correct GO classes					
				GO-1	GO-2	GO-3	GO-4	GO-5	
		classifier scores	Row Max	0.6	0.2	0	0.7	1	Column Max
Predicted GO classes	GO-6	0.9	0.8	0	0.2	0	0.6	0.8	
	GO-7	0.86	0.6	0.6	0	0	0	0	
	GO-5	0.84	1	0	0	0	0.7	1	
	GO-8	0.8	0.2	0.1	0.2	0	0.2	0	$thr = 0.8$
	GO-9	0.76	0.8	0	0.2	0.8	0	0	
	GO-1	0.74	1	1	0	0.1	0	0	
	GO-10	0.74	0.2	0	0.2	0	0.2	0	
	GO-11	0.7	0.3	0.1	0	0.2	0.1	0.3	

(a)

Summation methods for Semantic Similarities			Score for threshold	
Method abbr.	Description	Equation	$thr = 0.8$	$thr = 0.7$
A	matrix mean	$\sum_j \sum_i x_{ij}/(NM)$	0.22	0.19
B	mean of col.maxima	$\sum_{j=1}^{M} \max_{1 \le i \le N} x_{ij}/M$	0.5	0.74
C	mean of row maxima	$\sum_{i=1}^{N} \max_{1 \le j \le M} x_{ij}/N$	0.575	0.538
D	mean of B and C	$(B + C)/2$	0.538	0.639
E	min of B and C	$min(B, C)$	0.5	0.538
F	mean of concatenated row and col. maxima	$(\sum_{j=1}^{M} \max_{1 \le i \le N} x_{ij} + \sum_{i=1}^{N} \max_{1 \le j \le M} x_{ij})/(N + M)$	0.533	0.615

(b)

Table 2. Two tables demonstrate the different summation methods for GO semantic similarities, used here. First part shows small toy data, where rows represent predicted GO classes and columns represent correct GO classes. Predicted GO classes are ordered using the classifiers prediction score. Next, a semantic similarity measure is calculated for every in the upper matrix. Finally, a threshold value *thr* is defined to select accepted predictions. Lower table shows different summation methods that we tested. We show results for two threshold settings from toy data.

References

1. Clark WT, Radivojac P. Information-theoretic evaluation of predicted ontological annotations. Bioinformatics. 2013;29(13):i53–i61.

2. Pesquita C, Faria D, Falcao AO, Lord P, Couto FM. Semantic similarity in biomedical ontologies. PLoS comput biol. 2009;5(7):e1000443.

3. Pesquita C, Faria D, Bastos H, Ferreira AE, Falcão AO, Couto FM. Metrics for GO based protein semantic similarity: a systematic evaluation. BMC Bioinformatics. 2008;9(Suppl 5):S4. doi:10.1186/1471-2105-9-s5-s4.

4. Ferri C, Hernández-Orallo J, Modroiu R. An experimental comparison of performance measures for classification. Pattern Recognition Letters. 2009;30(1):27–38.

5. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009;45(4):427–437.

6. Seliya N, Khoshgoftaar TM, Van Hulse J. A study on the relationships of classifier performance metrics. IEEE; 2009. p. 59–66.

7. Martin DM, Berriman M, Barton GJ. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. BMC bioinformatics. 2004;5(1):178.

8. Engelhardt BE, Jordan MI, Muratore KE, Brenner SE. Protein molecular function prediction by Bayesian phylogenomics. PLoS computational biology. 2005;1(5):e45.

9. Götz S, García-Gómez JM, Terol J, Williams TD, Nagaraj SH, Nueda MJ, et al. High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic acids research. 2008;36(10):3420–3435.

10. Friedberg I. Automated protein function prediction—the genomic challenge. Briefings in bioinformatics. 2006;7(3):225–242.

11. Hawkins T, Chitale M, Luban S, Kihara D. PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. Proteins: Structure, Function, and Bioinformatics. 2009;74(3):566–582.

12. Wass MN, Sternberg MJ. ConFunc—functional annotation in the twilight zone. Bioinformatics. 2008;24(6):798–806.

13. Chitale M, Hawkins T, Park C, Kihara D. ESG: extended similarity group method for automated protein function prediction. Bioinformatics. 2009;25(14):1739–1745.

14. Engelhardt BE, Jordan MI, Srouji JR, Brenner SE. Genome-scale phylogenetic function annotation of large and diverse protein families. Genome research. 2011;21(11):1969–1980.

15. Fontana P, Cestaro A, Velasco R, Formentin E, Toppo S. Rapid annotation of anonymous sequences from genome projects using semantic similarities and a weighting scheme in gene ontology. PLoS One. 2009;4(2):e4619.

16. Minneci F, Piovesan D, Cozzetto D, Jones DT. FFPred 2.0: improved homology-independent prediction of gene ontology terms for eukaryotic protein sequences. PLoS One. 2013;8(5):e63754.

17. Radivojac P, Clark WT, Oron TR, Schnoes AM, Wittkop T, Sokolov A, et al. A large-scale evaluation of computational protein function prediction. Nature methods. 2013;10(3):221–227.

18. Gillis J, Pavlidis P. Characterizing the state of the art in the computational assignment of gene function: lessons from the first critical assessment of functional annotation (CAFA). In: BMC bioinformatics. vol. 14. BioMed Central; 2013. p. S15.

19. Koskinen P, Törönen P, Nokso-Koivisto J, Holm L. PANNZER: high-throughput functional annotation of uncharacterized proteins in an error-prone environment. Bioinformatics. 2015;31(10):1544–1552.

20. Kahanda I, Funk CS, Ullah F, Verspoor KM, Ben-Hur A. A close look at protein function prediction evaluation protocols. GigaScience. 2015;4(1):41.

21. Jiang Y, Oron TR, Clark WT, Bankapur AR, D'Andrea D, Lepore R, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome biology. 2016;17(1):184.

22. Clark WT, Radivojac P. Information-theoretic evaluation of predicted ontological annotations. Bioinformatics. 2013;29(13):i53–i61.

23. Gentleman R. Visualizing and distances using GO. URL http://www bioconductor org/docs/vignettes html. 2005;38.

24. Pesquita C, Faria D, Bastos H, Falcao A, Couto F. Evaluating GO-based semantic similarity measures. In: Proc. 10th Annual Bio-Ontologies Meeting. vol. 37; 2007. p. 38.

25. Resnik P, et al. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. J Artif Intell Res(JAIR). 1999;11:95–130.

26. Lin D, et al. An information-theoretic definition of similarity. In: Icml. vol. 98. Citeseer; 1998. p. 296–304.

27. Hand DJ, Till RJ. A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine learning. 2001;45(2):171–186.

28. Friedberg I, Radivojac P. Community-Wide Evaluation of Computational Function Prediction. ArXiv e-prints. 2016;.

29. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning. ACM; 2006. p. 233–240.

30. Lord PW, Stevens RD, Brass A, Goble CA. Semantic similarity measures as tools for exploring the gene ontology. In: Biocomputing 2003. World Scientific; 2002. p. 601–612.

31. Couto FM, Silva MJ, Coutinho PM. Measuring semantic similarity between Gene Ontology terms. Data & knowledge engineering. 2007;61(1):137–152.

32. Schlicker A, Domingues FS, Rahnenführer J, Lengauer T. A new measure for functional similarity of gene products based on Gene Ontology. BMC bioinformatics. 2006;7(1):302.

33. Azuaje F, Wang H, Bodenreider O. Ontology-driven similarity approaches to supporting gene functional assessment. In: Proceedings of the ISMB'2005 SIG meeting on Bio-ontologies; 2005. p. 9–10.