Notes on Pathway Enrichment Analysis by Different Methods

Alina Selega

November 2023

1 Introduction

I was trying to understand the differences between how pathway enrichment analysis is performed by the GSEA [6] and gProfiler [3] methods. I used to refer to this task in general as "gene set enrichment analysis", but I guess this name, at least in the abbreviation, is reserved for the GSEA method and it is more correct to refer to the overall task as "pathway enrichment analysis".

There are different approaches for pathway enrichment analysis [2]. All of them aim to find pathways that are enriched in a list of genes (or other entities) more than by chance, by using statistical methods. There are two classes of these approaches:

1. gene set enrichment analysis (GSEA)
2. overrepresentation analysis (ORA)

Other distinctions between these classes explained in [2] were unfortunately not understandable to me so this where I leave this classification. As a note, while the description of competitive and self-contained tests in [2] was not clear, it is much better explained in [4].

2 GSEA: the analysis and the method

GSEA is both the name of the tool [6] and the type of analysis this tool implements. The problem definition for GSEA is:

- for a given gene list L, ranked by the correlation between each gene's expression and the phenotype of interest
- and a pre-defined gene set S
- determine whether genes from S tend to occur close to the top or bottom of L.

If they do, then the genes (anti-)correlated with a certain phenotype can be characterized by the enrichment for the function or location that grouped together genes in S. GSEA is implemented by `fgsea`, `clusterProfiler`, *GSEApy* and others.

2.1 Enrichment score

The method moves down the list L (Figure 1), computing a running-sum statistic x_t for each gene g_t, for $t \in 1, ..., G$ as:

$$x_0 = 0 \tag{1}$$

$$x_{t+1} = \begin{cases} x_t + c_t & \text{if } g_t \in S \\ x_t - c_t & \text{if } g_t \notin S \end{cases} \tag{2}$$

where G is the total number of genes and c_t is the phenotype correlation of each gene.

Once done for all genes in L, the enrichment score for gene set S is the maximum deviation from zero (Figure 1). Thus, if we had genes present in S throughout the list L, we'd get more or less equal amounts of up and down movements of the walk, with not many genes in a row being a part of S, and consequently, a low enrichment score.

2.1.1 Original implementation

It is worth noting that the original implementation of GSEA did a simple update of 1 unit for encountering a gene belonging or not belonging to S. However, they found that that formulation would generate a high enrichment score if genes in S were located in the middle of L (understandably). They didn't like this situation because genes in the middle of a ranked list probably don't have a strong association with the phenotype, defeating the whole purpose. To address this, they introduced updates weighted by the correlation so that contributions of the middle genes would be small.

2.2 Estimating significance

They then need to estimate the significance of a computed enrichment score. They suggest doing it by permuting *phenotype labels* for each sample and recomputing the scores.

2.2.1 Permuting phenotype labels of samples

It is important to note here that the input to GSEA is a gene-by-sample matrix of expression values and a phenotype label for each sample. So how do we end up with a single ranked list of genes if we had multiple samples? GSEA can

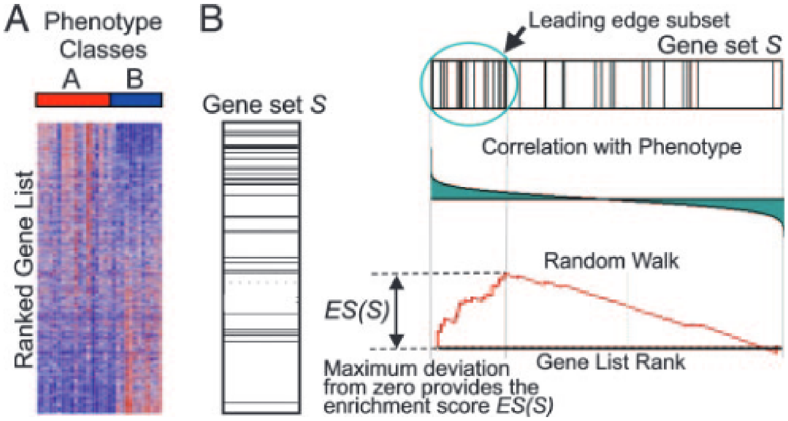

Figure 1: Overview of GSEA. Source: [6].

take a ranking method to perform this ranking. By default they use signal-to-noise-ratio but in practice, it can also be the numbers coming out of differential expression analysis between a pair of phenotype classes.

So the preferred method is to re-assign phenotype labels, re-compute the ranked list, and compute the enrichment score on it for a given gene set S. Do that multiple times and now you have a null distribution of enrichment scores. Compare your real score to it, get an empirical p-value out, and you're done.

The argument for permuting the labels says that this "preserves gene-gene correlations and, thus, provides a more biologically reasonable assessment of significance than would be obtained by permuting genes."

However, they note in their paper that you can also take a second approach. e.g. if you have too small of a dataset to allow label permutations or if you have a single phenotype class.

2.2.2 Creating random gene sets of size $|S|$

This second approach creates a random gene set S' that has the same size as the original S. The original paper doesn't talk much about it but I am assuming that S' is created from the full set of genes in your input data (as there is really no other information given to the method). For this reason, it seems important for your data to include *all* genes so that they represent the whole universe of known genes that you then can generate S' from to see how often your list would be enriched in a random set of genes. (Indeed, this is the advised usage [4].)

In practice, this second case is the most common, when we have a measure of importance that we rank the genes by rather than two explicit conditions or phenotypes we're comparing. In fact, `clusterProfiler` only takes the ranked gene list as input so therefore must be using random gene sets for significance estimation (as there are no samples provided to re-assign labels to). The authors caution that “This approach is not strictly accurate: because it ignores gene-gene correlations, it will overestimate the significance levels and may lead to false positives”. Perhaps using a stricter significance level would be useful in this case.

Another note is that the original paper [6] describes this case as “P value can be estimated by permuting the genes, with the result that genes are randomly assigned to the sets while maintaining their size.” I found this explanation quite misleading as I originally took it as permuting gene list L and testing that permutation against gene set S. In reality, they create random gene sets S' and test L against those. I am not entirely clear on the difference between these two approaches.

2.3 Multiple testing correction

When doing this analysis for many sets S and the same ranked list L, we need to adjust for many comparisons by adjusting the significance level.

GSEA normalizes the enrichment score (ES) for each S by the mean of ES across all permutations and uses the false discovery rate approach to adjust significance.

I will note here that the definition of the normalized enrichment score (NES) is quite misleading in the literature: I have seen two papers (including the original GSEA paper) [6, 4] saying that “The ES score is [...] normalized relative to pathway size” implying that $NES = \frac{ES}{|S|}$. However, according to the guidelines from the Broad Institute [1], this is incorrect and NES is normalized as I stated above.

2.4 Leading edge subset

They introduce something called the leading edge subset which is the set of genes in S that correspond to the part of the running-sum curve from the start until its highest point (Figure 1). This represents the set of genes that contribute the most to the enrichment of the gene set S.

3 Over-representation analysis: the analysis

Over-representation analysis (ORA) tests for enrichment in a bag-of-words list based on the hypergeometric distribution and the p-value is calculated as:

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}} \tag{3}$$

where N is the total number of genes in the background, M is the number of genes in the background that belong to gene set S, n is the number of genes in the bag-of-words list L, and k is the number of genes in L that belong to gene set S.

3.1 gProfiler: the method

gProfiler [3] uses the above to analyze unranked gene lists using Fisher's exact test and ranked gene lists using a modified Fisher's exact test, with multiple test correction. I am unable to find the exact desciption of what a "modified Fisher's exact test" entails as, perplexingly, neither the original gProfiler paper [5] nor the later one [3] mention the word "Fisher" anywhere in the text.

3.1.1 Ranked lists

For ranked lists, gProfiler calls it an "ordered query" and iteratively performs the hypergeometric test against the gene set S for groups of genes taken from the top of the list L: e.g. first testing the group with n top genes, then $n+1$ top genes, $n+2$ and so on (presumably they don't start with $n = 1$ but I haven't checked the implementation to know). The analysis then reports the size of the group of genes that received the lowest p-value for that pathway S. I am assuming the analysis continues checking groups until the last group's size is equal to the size of the query list L but I haven't checked that.

Both ordered and unordered analyses require the definition of background, which is typically specified as all genes in the library.

In author's words [4], "ordered enrichment test, [which] is suitable for lists of up to a few thousand genes that are ordered by a score, whereas the rest of the genes in the genome lack meaningful signal for ranking." This is in contrast to GSEA, which is designed to run on *all* genes without prior filtering. However, in the gProfiler paper [3], they state that "This option is very similar to the idea of the GSEA analysis method".

4 Difference between GSEA and gProfiler ranked list enrichment analyses

One difference I can see with using gProfiler vs. GSEA on a ranked list of all genes is that gProfiler can return pathways that are enriched in groups spanning any length of the ranked list, for example from the top to the middle. On the other hand, GSEA returns pathways enriched at "the top" (or "the bottom") of the list, specifically taking care not to include the genes in the middle of the list (as they would be uncorrelated with the phenotype of interest). This seems to be the main reason why gProfiler is advised to be applied to ranked gene lists of interest, i.e. some curated list one can score by importance rather than all genes like GSEA.

Two ways to account for that when using gProfiler would be to (i) analyze a truncated version of the list L containing top X genes or (ii) analyze the whole list but filter the results by setting a threshold for the maximum allowed query size, giving the maximum number of top genes that can be enriched for any pathway. While these two cases are similar, I think they might return different results as e.g. the full list L might not be able to score a significant p-value against any pathway.

References

[1] Gene set enrichment analysis guide. `https://docs.gsea-msigdb.org/#GSEA/GSEA_User_Guide/`. Accessed: 2023-11-23.

[2] Davide Chicco and Giuseppe Agapito. Nine quick tips for pathway enrichment analysis. *PLoS computational biology*, 18(8):e1010348, 2022.

[3] Liis Kolberg, Uku Raudvere, Ivan Kuzmin, Jaak Vilo, and Hedi Peterson. gprofiler2–an r package for gene list functional enrichment analysis and namespace conversion toolset g: Profiler. *F1000Research*, 9, 2020.

[4] Jüri Reimand, Ruth Isserlin, Veronique Voisin, Mike Kucera, Christian Tannus-Lopes, Asha Rostamianfar, Lina Wadi, Mona Meyer, Jeff Wong, Changjiang Xu, et al. Pathway enrichment analysis and visualization of omics data using g: Profiler, gsea, cytoscape and enrichmentmap. *Nature protocols*, 14(2):482–517, 2019.

[5] Jüri Reimand, Meelis Kull, Hedi Peterson, Jaanus Hansen, and Jaak Vilo. g: Profiler—a web-based toolset for functional profiling of gene lists from large-scale experiments. *Nucleic acids research*, 35(suppl_2):W193–W200, 2007.

[6] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.