Supplementary Online Materials

blitzGSEA: Efficient computation of Gene Set Enrichment Analysis through Gamma distribution approximation

Alexander Lachmann[1,*], Zhuorui Xie[1], Avi Ma'ayan[1]

[1]Department of Pharmacological Sciences, Icahn School of Medicine at Mount Sinai, One Gustave L. Levy Place, Box 1603, New York, NY 10029 USA

Reproducibility of benchmark results

All code and benchmarks are deposited as Jupyter Notebooks in GitHub at: https://github.com/MaayanLab/blitzgsea/tree/main/testing. The Jupyter Notebooks include detailed markup descriptions and figures.

Gamma distribution estimation

The gamma distribution has two parameters: alpha for shape, and beta for scale. These parameters are estimated for each gene set size separately. For a single gene set size, blitzGSEA generates multiple enrichment scores (ES) by permuting the gene labels. By default, blitzGSEA calculates 2,000 enrichment scores (ES). The distribution is then fit with the stat.gamma module from the scipy Python package. While fitting the *loc* parameter of the gamma distribution, which is the offset to 0, is fixed to 0. Since calculating large number of perturbations is costly, blitzGSEA avoids calculating permutations for each gene set size in the gene set library. Parameters are only estimated for a subset of gene set sizes, these are used as calibration anchors. By default, 20 calibration anchors are used. blitzGSEA interpolates the gamma distribution parameters that best fit the calibration anchor sizes by first applying a LOESS fit on the estimated anchor parameters, and then applying linear interpolation to estimate the remaining gene set size parameters. The gamma parameters relative to gene set size for an exemplary gene expression signature are visualized (**Fig. S2a**). The parameters follow a monotonic function relative to the gene set size. The LOESS fit in red shows the smoothed parameters used by blitzGSEA.

Identifying calibration anchors

The number of desired calibration anchors can be modified by the user. The default value is 20. To span the entire range of gene set sizes in the gene set library, blitzGSEA always adds a gene set of size of 1 and the maximum gene set size from the maximum from the gene set library. To evenly cover gene set sizes, blitzGSEA calculates equidistant percentiles of the gene set size distribution. This results in a more granular sampling of more common gene set sizes. Additionally, default anchors for gene sets of sizes 4 and 6 are added. The beta/scale parameter has high variability across small gene set sizes (**Fig. S2b**).

Symmetric Gamma distributions

As mentioned above, the default permutations made by the blitzGSEA algorithm is 2,000. For differential gene expression signatures with weights centered around 0, there are an expected 1,000 negative ES and 1,000 positive ES. In practice, this is sufficient for accurate estimation of the gamma distributions. blitzGSEA supports the option to choose symmetric gamma distributions for positive and negative ES. If this option is chosen, only one gamma distribution is estimated by combining the positive ES and the absolute values of the negative ES. In the case of low number of permutations, this can lead to better results if the assumption of symmetry is accurate. If either the number of positive or negative ES is less than 250, blitzGSEA will automatically default to the symmetric estimation of the gamma distributions.

p-value calculation from ES

For a given gene set of size N, blitzGSEA calculates a null distribution comprised of two gamma distributions. The gamma distributions are either defined for positive or negative values and are weighted relative to the ratio of positive to negative ES. The probability function of the bimodal distribution p(x) is defined by:

$gamma_{pos}(x) = \gamma\left(\alpha pos, \frac{x}{\beta pos}\right)\left(\frac{1}{\Gamma(\alpha pos)}\right)$, if x > 0, else $gamma_{pos}(x) = 1$

$gamma_{neg}(x) = \gamma\left(\alpha pos, \frac{-x}{\beta neg}\right)\left(\frac{1}{\Gamma(\alpha neg)}\right)$, if x < 0, else $gamma_{neg}(x) = 1$

$weight_{pos} + weight_{neg} = 1$

$p(x) = 1 - (weight_{pos} * gamma_{pos(x)} + weight_{neg} * gamma_{neg(x)})$

Note that $gamma_{neg}(x)$ is converting negative ES scores to positive values, as the gamma distribution is only defined for positive values.

Kolmogorov-Smirnov testing to show gamma distribution appropriateness

To prove that the gamma distribution accurately simulates the underlying null distribution of enrichment scores, we calculated the Kolmogorov-Smirnov test (KS) for each gene set size. The fit for the positive and negative ES distributions with the theoretical quantiles are visualized on the x-axis, and the quantiles generated from permutations is visualized on the y-axis (**Fig. S1**). When calculating gamma distribution approximations for calibration anchors, blitzGSEA automatically performs KS tests to validate that the generated ES distribution is well described by the gamma distribution for the positive and negative ES values. In case the KS test p-values are below 0.05, blitzGSEA will output a warning. For the KS test, blitzGSEA is using the stats.kstest module from the scipy Python package.

Differential gene expression signature computations

For all benchmarks we used differential gene expression signatures computed by comparing muscle RNA-seq samples from donors aged 20-29 and 70-79 years of age derived from the Genotype Tissue Expression (GTEx) portal [1]. Differential expression analysis was performed with limma voom [2]. For signature weights, we use the limma voom t-statistic.

P-value saturation benchmark

To calculated enrichment, we used the Enrichr [3] gene set library created from KEGG [4] (KEGG_2019) using GSEApy [5] and blitzGSEA with 250 to 2,500 permutations applied to the muscle aging gene expression signature. Each permutation number is repeated 5 times. The gene sets are ranked from largest to smallest where p-values calculated with blitzGSEA are displayed on the x-axis (**Fig. S3**). The y-axis displays the boxplots where each box plot represents 5 p-values as -log(p-values) for a given permutation number. In black, the p-values of blitzGSEA are shown, and in magenta are the GSEApy results. The minimum p-value line indicates the smallest p-value that can be computed by GSEApy which is not equal to 0.

Variable decimal precision

While computing probabilities, blitzGSEA uses either the scipy gamma.cdf function or if additional precision is required the mpsci.distributions.gamma package. By default, the algorithm uses scipy which is the faster implementation and switch when numeric instability is detected. mpsci can be configured to run at arbitrary precision with the underlying mpmath Python library. This implementation will vary between 100 and 1000 digits.

Signature hashing and parameter reuse

The accurate parameters to estimate ES distributions is dependent on the signature values and the size of the gene set. As such the parameter estimation for anchor gene set sizes can be transferred between gene set libraries. In case enrichment against multiple gene set libraries is run sequentially the algorithm will not recalculate anchor parameters once they are calculated. The signature is hashed into a unique key in which parameters are stored in a global python variable.

P-value consistency benchmark

To compare the reliability of the enrichment analysis performed by blitzGSEA to empirical permutation testing, we tested the reproducibility of p-values given multiple reruns for the same number of permutations. A reliable p-value estimator is unbiased to the number of permutations used. Minor differences are observed across replicates. From the p-value saturation benchmark, we calculate the Pearson correlation coefficient (PC) between all p-value results derived from the same number of permutations (n=5). To place a higher emphasis on smaller p-values, we log transform the p-value vectors first. We then calculate the errors as 1-PC. The errors by the number of permutations are recorded for GSEApy and blitzGSEA (**Fig. S3**).

Choice of default permutation number

blitzGSEA calculates 2,000 permutations by default. The choice was made following the benchmark shown in Fig. S3. Self-consistency saturates 1750 permutations and is higher compared to GSEA-P and fGSEA. Users can change the number of permutations if desired.

Supplementary Figures

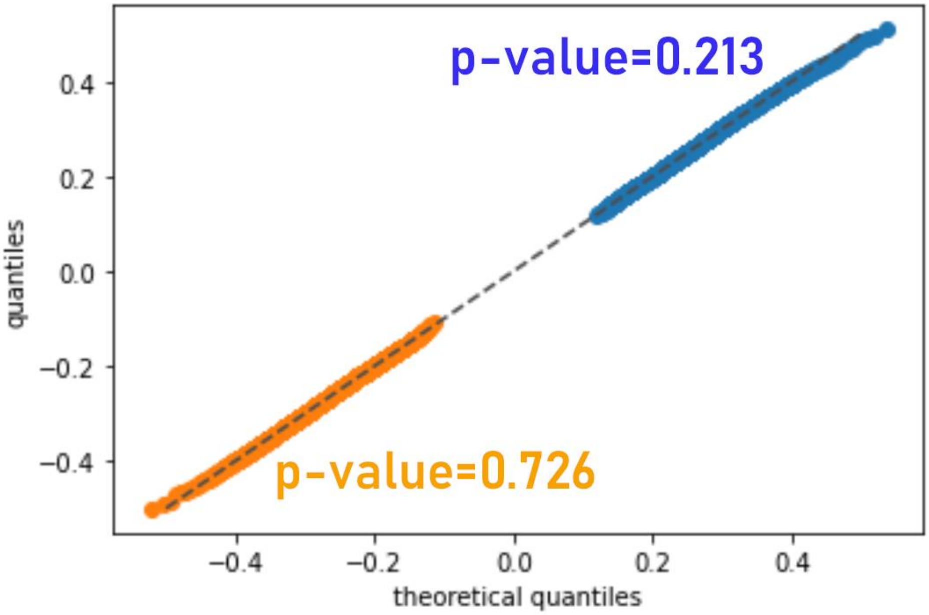


Fig. S1 Kolmogorov-Smirnov test fitting of the gamma distribution for 10,000 sampled enrichment scores for gene sets of size 50.

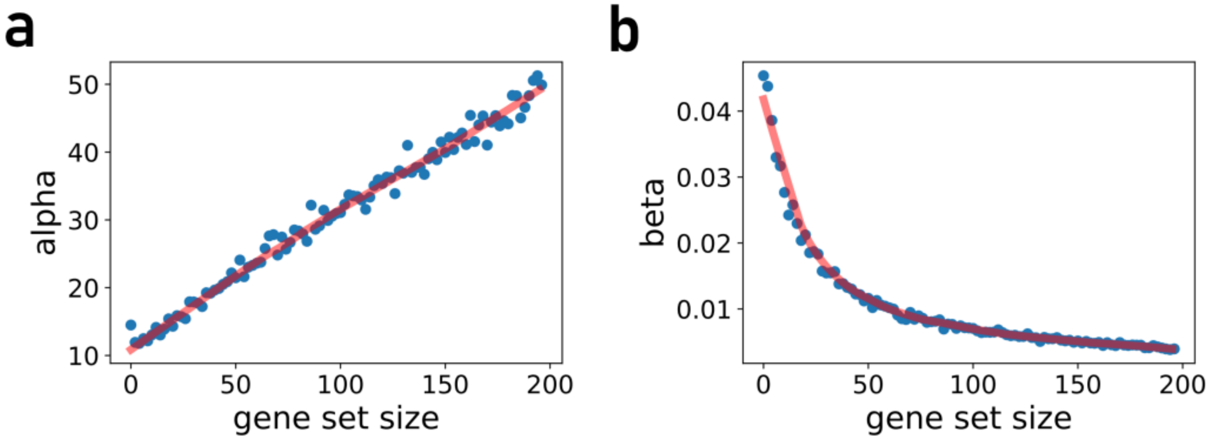


Fig. S2 Gamma distribution parameters relative to gene set size for positive enrichment scores. The red line indicates the LOESS fit. To calculate the parameters, gene set labels were shuffled 2,000 times.

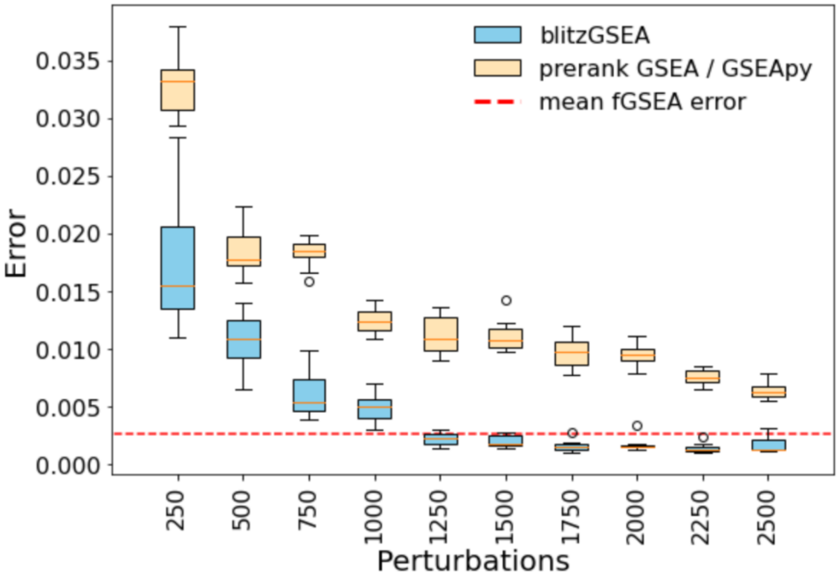
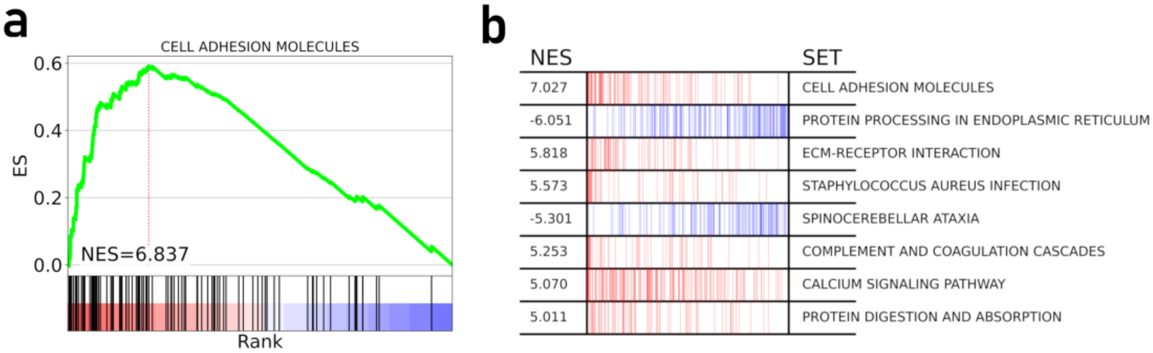


Fig. S3 p-value consistency for different numbers of permutations for GSEA pre-rank, and blitzGSEA with each box representing 10 replicates. Average p-value consistency of fGSEA shown as a constant, since fGSEA does not provide a permutation parameter.



NES	SET
7.027	CELL ADHESION MOLECULES
-6.051	PROTEIN PROCESSING IN ENDOPLASMIC RETICULUM
5.818	ECM-RECEPTOR INTERACTION
5.573	STAPHYLOCOCCUS AUREUS INFECTION
-5.301	SPINOCEREBELLAR ATAXIA
5.253	COMPLEMENT AND COAGULATION CASCADES
5.070	CALCIUM SIGNALING PATHWAY
5.011	PROTEIN DIGESTION AND ABSORPTION

Fig. S4 a) Compact running sum plot made by the blitzGSEA package; b) Top table plot made by blitzGSEA.

References

1. Consortium G: The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. Science 2015, 348(6235):648-660.
2. Law CW, Alhamdoosh M, Su S, Dong X, Tian L, Smyth GK, Ritchie ME: RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR. F1000Research 2016, 5.
3. Kuleshov MV, Jones MR, Rouillard AD, Fernandez NF, Duan Q, Wang Z, Koplev S, Jenkins SL, Jagodnik KM, Lachmann A: Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic acids research 2016, 44(W1):W90-W97.
4. Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M: The KEGG resource for deciphering the genome. Nucleic acids research 2004, 32(suppl_1):D277-D280.
5. Fang Z: GSEApy: Gene Set Enrichment Analysis in Python. Zenodo 2020, 10.5281/ZENODO.3748085.