Orsum: a Python package for filtering and comparing enrichment analyses using a simple principle

Ozan Ozisik, Morgane Térézol, Anaïs Baudot

BMC Bioinformatics

SOFTWARE

Open Access

orsum: a Python package for filtering and comparing enrichment analyses using a simple principle

Ozan Ozisik[1*], Morgane Térézol[1] and Anaïs Baudot[1,2,3*]

*Correspondence: ozan.ozisik@univ-amu.fr; anais.baudot@univ-amu.fr

[1] Aix Marseille University, Inserm, MMG, Marseille, France
[2] Barcelona Supercomputing Center (BSC), Barcelona, Spain
[3] CNRS, Marseille, France

Abstract

Background: Enrichment analyses are widely applied to investigate lists of genes of interest. However, such analyses often result in long lists of annotation terms with high redundancy, making the interpretation and reporting difficult. Long annotation lists and redundancy also complicate the comparison of results obtained from different enrichment analyses. An approach to overcome these issues is using down-sized annotation collections composed of non-redundant terms. However, down-sized collections are generic and the level of detail may not fit the user's study. Other available approaches include clustering and filtering tools, which are based on similarity measures and thresholds that can be complicated to comprehend and set.

Result: We propose orsum, a Python package to filter enrichment results. orsum can filter multiple enrichment results collectively and highlight common and specific annotation terms. Filtering in orsum is based on a simple principle: a term is discarded if there is a more significant term that annotates at least the same genes; the remaining more significant term becomes the representative term for the discarded term. This principle ensures that the main biological information is preserved in the filtered results while reducing redundancy. In addition, as the representative terms are selected from the original enrichment results, orsum outputs filtered terms tailored to the study. As a use case, we applied orsum to the enrichment analyses of four lists of genes, each associated with a neurodegenerative disease.

Conclusion: orsum provides a comprehensible and effective way of filtering and comparing enrichment results. It is available at https://anaconda.org/bioconda/orsum.

Keywords: Over-representation analysis, Enrichment analysis, Filtering, Neurodegenerative diseases

Background

Enrichment analyses are widely used to investigate lists of genes of interest, such as genes differentially expressed or genes associated with disease variants. However, the outputs of enrichment analyses are often long lists of redundant annotation terms. These long lists make the interpretation, reporting and comparison of enrichment analysis results complicated.

Redundancy in enrichment results is often due to the hierarchical structure of certain annotation databases, such as Gene Ontology (GO) [1, 2] or Reactome [3]. In these databases, the annotation terms are linked by parent-child relationship where ancestor terms correspond to more general terms and descendant terms correspond to more specific terms. Importantly, ancestor terms annotate all the genes annotated by their descendants. The overlap of annotated genes causes the risk of both parent and child term to be significant in the enrichment analyses.

A first solution to overcome redundancy consists in using down-sized, non-redundant collections of annotation terms. Such collections include for instance "GO subsets", also known as "GO slims". GO subsets contain only a portion of the GO terms, providing a coarse-grained ontology. Generic and organism-specific GO subsets are maintained by the GO consortium and different communities [4, 5]. The hallmark collection of The Molecular Signatures Database (MSigDB) [6] is another down-sized collection of annotation terms. It contains 50 non-redundant terms obtained by clustering and filtering over 4000 terms from different databases. The enrichment analysis tool FunMappOne [7] has its own down-sized versions of the Kyoto Encyclopedia of Genes and Genomes (KEGG) [8], GO and Reactome databases. The annotation terms are stored in three hierarchical levels, ranging from the full list of terms to the summary list.

Down-sized annotation collections are built a priori; they hence allow directly obtaining non-redundant enrichment results. In addition, using a lower number of annotation terms implies less statistical tests. The multiple testing correction will hence be more gentle. The main drawback of down-sized annotation collections is that the terms may be too general or too specific for the user's study. Additionally, these collections are not available in all the enrichment analysis tools. Finally, the user is dependent on the curators of these collections for updates; for instance, some GO subsets may become outdated while GO is updated.

The second solution to overcome redundancy in enrichment results is the clustering of the enriched annotation terms. The clustering methods calculate the similarity between two annotation terms either with statistical measures on the corresponding sets of annotated genes or with semantic similarity measures. Various semantic similarity measures exist. The most common ones depend on the frequencies of the assessed terms and of their closest common ancestor term in the annotation database [9, 10]. The reader is referred to [10] for a review on semantic similarity. RedundancyMiner [11] is a tool that clusters enrichment results. The similarity between two terms is measured by Fisher's exact test on a contingency table storing the number of common and different genes between the two terms. G-SESAME [12] is an online set of tools associating the annotation terms based on semantic similarities. DAVID [13], ClueGO [14], clusterProfiler [15, 16], pathfindR [17] and ViSEAGO [18] are among the enrichment tools that provide additional functions to cluster enriched terms. While clusterProfiler and ViSEAGO use semantic similarity measures, the others use kappa statistics on annotated gene sets for clustering.

The third solution to overcome redundancy in enrichment analysis results is filtering. One widely used and highly cited tool in this category is REVIGO [19]. REVIGO is available as a web tool. It selects pairs of terms that are more similar than a threshold based on semantic similarity measures and compares them. In the comparison, REVIGO checks

step by step whether one of the terms is very general, or less significant than the other, or a child term of the other. It then discards the term that satisfies the condition. The Cytoscape [20] app of STRING database [21], stringApp [22], provides functionality for enrichment analysis and removal of redundant terms. From two terms with high number of overlapping genes, the less significant one is discarded. A popular enrichment tool, g:Profiler [23, 24], also provided filtering options that considered term significance and parent-child relations on its web interface (before version e94_eg41_p11_5fca2e9) and in its previous R package, gProfileR. GOsummaries [25], which is an R package for visualizing enrichment results, uses gProfileR package for enrichment analysis and filtering.

Clustering and filtering approaches process the user's enrichment results. They hence provide summarized results tailored to the input enrichment analyses. The tools developed for clustering and filtering can work with any enrichment analysis result; the user is thus free to choose any enrichment analysis tool. However, available clustering and filtering approaches are based on similarity measures that can be complicated to handle. It can indeed be complex for the users to set thresholds for similarity and anticipate the consequences on the obtained summarized results. Finally, many popular tools work with built-in annotation data. In this case, the user is dependent on the developers for updates.

Redundancy complicates the interpretation and reporting of enrichment results. Another challenge is comparing multiple enrichment analyses. Such comparisons are carried out, for example, when multiple lists of genes associated with different conditions are analyzed, or when the same list of genes is analyzed using different enrichment analysis tools. This challenge is particularly tedious because similar annotation terms can be as relevant as the exact term matches. BACA [26] is a tool for enrichment analyses of multiple gene lists. For each input gene list, BACA obtains enrichments through the DAVID web service. It then creates a bubble chart that presents the gene lists enriched in each term. BubbleGUM [27] is a tool that runs Gene Set Enrichment Analysis (GSEA) [28] on expression data by processing multiple conditions in pairs. Similar to BACA, the results are presented on a bubble chart allowing comparison of different analyses. FunMappOne and clusterProfiler, mentioned previously, can also work with multiple input gene lists; they produce plots that allow comparison of the enrichment results. ClueGO and pathfindR merge multiple results showing whether the term is common or specific to one of the results. However, to the best of our knowledge, no tool performs collective summarization considering multiple results together.

In this study, we present *orsum* (which stands for "over-representation summary"), a tool to filter enrichment analysis results. orsum has three main features: (i) The filtering process is straightforward, (ii) The ranking of the terms are considered in order to preserve the main biological information, (iii) Multiple enrichment results are filtered collectively.

Implementation

orsum is a Python-based enrichment results filtering tool. The filtering is based on a simple principle: a term is discarded if there is a more significant term that annotates at least the same genes. In this case, the remaining more significant term becomes the representative term for the discarded term.

The inputs for orsum are enrichment analysis results containing term IDs ordered by statistical significance (from the most to the least significant; significance values are

not given) and Gene Matrix Transposed (GMT) file. GMT is a common format to store annotation data, where each row stores an annotation term and the annotated genes.

orsum has two parameters related to the sizes of the considered terms. The first parameter, "minimum term size", is used to discard terms annotating only a small number of genes. The default value is 10. The second parameter, "maximum representative term size", is a threshold that limits the size of the terms that can represent other terms; it allows retaining terms annotating smaller number of genes in the filtered results. By default, this second parameter is not applied.

orsum starts by reading the input enrichment results and the GMT file. Then, an initial representative term list is created. At this point, this list contains the full list of terms present in the input enrichment analysis results. The terms are accompanied by their ranking values corresponding to the order they appear in the enrichment results. In the case of multiple input enrichment results, the input term lists are merged, duplicate terms are removed, and these terms get the best rank they have in any of the input enrichment results.

Next, the filtering process begins. Starting with the top-ranked terms, pairs of terms are checked iteratively. If a better ranked term covers all the genes annotated by a lower ranked term, then the lower ranked term is discarded and represented by the better ranked term. In other words, more significant ancestor terms represent their less significant descendant terms.

orsum outputs multiple files providing both simplified and detailed views of the filtered results:

- An HTML file with the filtered list of representative terms. The user can click on each representative term to see the discarded terms.
- Two TSV files ("-Summary.tsv", "-Detailed.tsv") with the information contained in the HTML file in different formats. "-Summary.tsv" file contains only representative terms while "-Detailed.tsv" additionally contains the terms represented by them.
- A heatmap presenting the top representative terms, colored according to the quartile of their ranks in each input enrichment result.
- A barplot presenting the top representative terms and how many terms they represent.

In the output figures, the number of terms to be presented is adjustable (default and maximum 50).

orsum is implemented in Python 3. The source code is available on https://github.com/ozanozisik/orsum and the package can be installed via bioconda (https://anaconda.org/bioconda/orsum).

Results

Use case: Four neurodegenerative diseases

orsum is designed to filter enrichment analyses and compare results from different studies. In order to illustrate orsum, we applied it to the enrichment analysis results of four gene lists, each associated with a neurodegenerative disease. These diseases are Alzheimer's disease (AD), amyotrophic lateral sclerosis (ALS), Huntington's disease

(HD) and Parkinson's disease (PD). As these diseases are all neurodegenerative diseases, in addition to the filtering of enrichment results, the common and different enriched annotation terms between diseases are also of interest.

We obtained gene lists associated with the four diseases from DisGeNET [29, 30] (on 20.12.2021). The queried disease terms are:

- C0002395 Alzheimer's Disease
- C0002736 Amyotrophic Lateral Sclerosis
- C0020179 Huntington Disease
- C0030567 Parkinson Disease

We selected the genes with gene-disease association score greater than or equal to 0.3. The number of selected genes are 123 for AD, 59 for ALS, 21 for HD and 92 for PD. These disease-associated genes display some overlap (Fig. 1a). We performed enrichment analyses using GO Biological Process annotation terms with g:Profiler (version e104_eg51_p15_3922dba). We ran g:Profiler with the default options and obtained statistically significant terms. AD genes are enriched in 991 GO terms, ALS genes in 59 GO terms, HD genes in 22 GO terms, and PD genes in 714 GO terms. These long lists of terms that are obtained for all diseases but HD illustrate the difficulty in interpreting enrichment results. The enriched terms have high overlaps but there are also many terms unique to each neurodegenerative disease (Fig. 1b).

We applied orsum to the four enrichment results collectively. We used the default settings except for the number of terms to be plotted; minimum term size was 10, maximum representative size threshold was not used, and top 20 terms were plotted. The command is given below:

```
orsum.py
--gmt hsapiens.GO:BP.name.gmt
--files
 gProfiler-GOBP-AD-IDs.txt
 gProfiler-GOBP-ALS-IDs.txt
 gProfiler-GOBP-HD-IDs.txt
 gProfiler-GOBP-PD-IDs.txt
--fileAliases AD ALS HD PD
--outputFolder orsumResult
--numberOfTermsToPlot 20
```

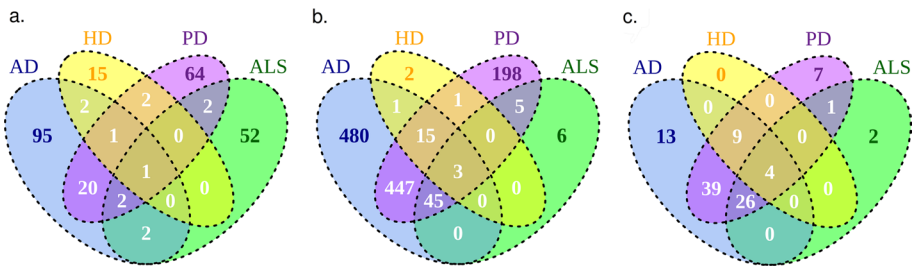

Fig. 1 Venn diagrams presenting **a** the overlap of genes associated with each disease, **b** the overlap of GO Biological Process terms enriched in genes associated with each disease, **c** the overlap of GO Biological Process terms enriched in genes associated with each disease after collective filtering by orsum

The four enrichment results collectively lead to 1203 enriched terms. From these, 40 terms were discarded because they annotated less than 10 genes. And finally, with the application of orsum filtering, the number of terms decreased from 1163 to 101.

In Fig. 1b we saw that enriched terms have high overlap but also there are many terms unique to each disease. After summarization, this distribution is conserved (Fig. 1c). However, common and specific terms can be analyzed more easily.

In Fig. 2, the top representative terms and the quartiles of their ranks according to each input enrichment result are presented. In this figure we can easily see the most significant representative terms and analyze whether these terms are shared and similarly important among the different diseases. For example, we see that "regulation of localization", "neuron death", "response to chemical" and "response to stress" are significant for all the analyzed neurodegenerative diseases. However, "regulation of cell death" and associated terms are significant for AD, HD and PD but not for ALS. In the figure, we can also identify the terms that are specific to a given disease. For example, "Amyloid-beta metabolic process" and "amyloid precursor protein catabolic process" are the two terms specific to AD among top 20 representative terms. Extracellular plaque deposits of the amyloid-β is one of the hallmark pathologies of AD [31]. Another example is "Lipoxygenase pathway", a term specific to ALS, which is significant due to paraoxonase genes associated with the disease [32].

Among the top representative terms, "regulation of multicellular organismal process", "regulation of localization" and "response to organic substance" represent more than 50 discarded terms (Fig. 3). These terms are general terms each annotating more than 2500 genes. We observed that there is a moderate correlation between the number of genes a term annotates and the number of terms it represents (Spearman's rank correlation, r=0.56, p=1.22E-09).

The obtained results show that orsum can filter long lists of enrichment results and highlight common and specific terms in the different enrichment results.

Comparison with REVIGO

As stated in the introduction, REVIGO [19] is a widely used approach to filter enrichment analysis results. We devised two strategies to compare orsum and REVIGO based on the number of representative terms obtained after filtering.

In the first strategy, we applied orsum and REVIGO to the enrichment analysis result of each of the four neurodegenerative diseases that we explored as a use case. REVIGO was run with default parameters except that we selected Homo sapiens as the species; the inputs were the GO terms with their respective p-values. For the sake of comparison with REVIGO, please note that here, orsum needs to be applied separately for each disease and not using its collective filtering feature. In this comparison we observed that REVIGO performed limited filtering (Table 1). Among these results we can comment on the results for HD. In this case, both orsum and REVIGO performed filtering on terms related to cell death, in different ways; as it is the way orsum works, ancestor terms that are more significant than their descendants were never eliminated in favor of the descendants in orsum, while REVIGO favored descendant, more specific terms even if they are less significant. orsum performed additional filterings by making "behavior" representative for "learning or memory" and "memory", and "response to chemical" representative for "response to oxygen-containing compound".

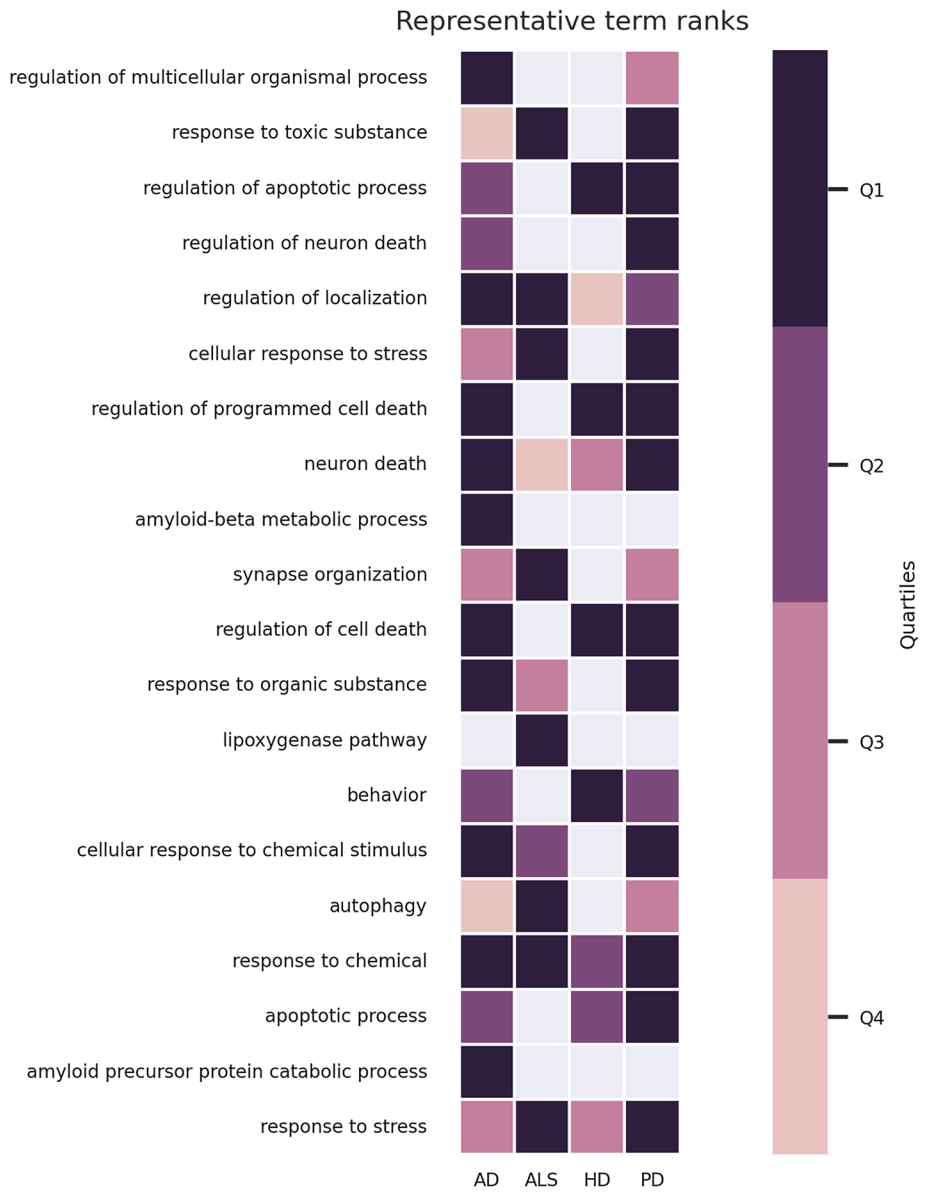

Fig. 2 Top 20 representative terms and the quartiles their ranks belong to according to each input enrichment result

In the second strategy, we devised a systematic approach using enrichment analysis results of artificially generated gene lists. We first generated 100 artificial query gene lists to perform enrichment analysis. However a random gene list will rarely be significantly enriched in any annotation terms. Therefore, we built each query gene list using both random genes and genes annotated by a given annotation term. From GO Biological Process terms, we randomly selected a term that annotates more than 100 genes. Let's say the term annotates n genes. We randomly selected $n/2$ genes annotated by the term, and we added $n/2$ other random genes, getting an artificial query gene list of size n. Next, we performed enrichment analysis on GO Biological Process terms using the hypergeometric test. We selected the terms with p-value< 0.05 after Bonferroni correction.

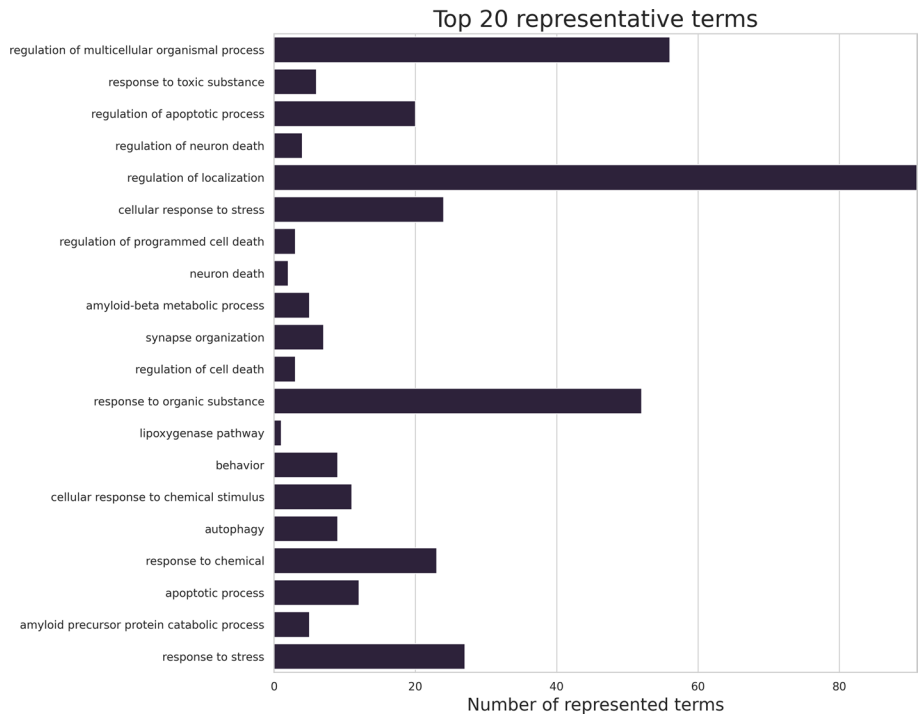

Fig. 3 Top 20 representative terms and the number of terms they represent

Table 1 Numbers of annotation terms in the enrichment analysis results of four neurodegenerative disease gene lists, in total and after filtering by orsum and REVIGO

Disease	Total number of terms	Number of terms after orsum filtering	Number of terms after REVIGO filtering
AD	991	79	540
ALS	59	34	51
HD	22	12	17
PD	714	104	415

Finally, we submitted the enrichment results to orsum and REVIGO and counted the representative term numbers obtained in the filtered results. For REVIGO, we benefited from its REST API. REVIGO allows queries up to 2000 terms but we encountered errors, so we limited our enriched terms list to the top 1000. This limitation concerned only 7 out of the 100 iterations. We observed that orsum resulted in a smaller filtered terms list in 97 out of 100 iterations (Fig. 4).

Discussion

We present orsum, a tool to filter enrichment results. We applied orsum on the enrichment results of four neurodegenerative diseases as a use case. We also compared orsum with the widely used REVIGO approach.

The core of orsum is its filtering principle: more significant ancestor terms represent their less significant descendant terms. In the use case, we demonstrated that this

effectively reduces the number of the enriched terms to a more interpretable level. The main advantage of this principle is its simplicity. The users can easily conceive how orsum filters the terms. The output of orsum is foreseeable and, looking at the representative terms, the user knows that the most significant terms are still belonging to the output list and representing their less significant descendant terms. This simple filtering principle has three consequences: (i) There is a moderate correlation between the number of genes a term annotates and the number of terms it represents. As expected, larger terms represent more terms. (ii) When a term is more significant than its ancestor, both terms are kept as representative terms. Although the redundancy is retained in such a case, this ensures that more significant and specific biological information is presented to the user. (iii) orsum works with only the annotation databases that are hierarchically organized such that terms are in a parent-child relationship, e.g. GO and Reactome. For the summarization of enrichment analysis results obtained from databases that are not hierarchically structured, we recommend the clustering approaches.

orsum can process multiple lists of enriched terms resulting from the analysis of different conditions or from the analysis of the same condition with different tools. This allows comparing and integrating different enrichment results. It should be noted that when applying different tools on the same input data, a proper multiple testing correction might be required.

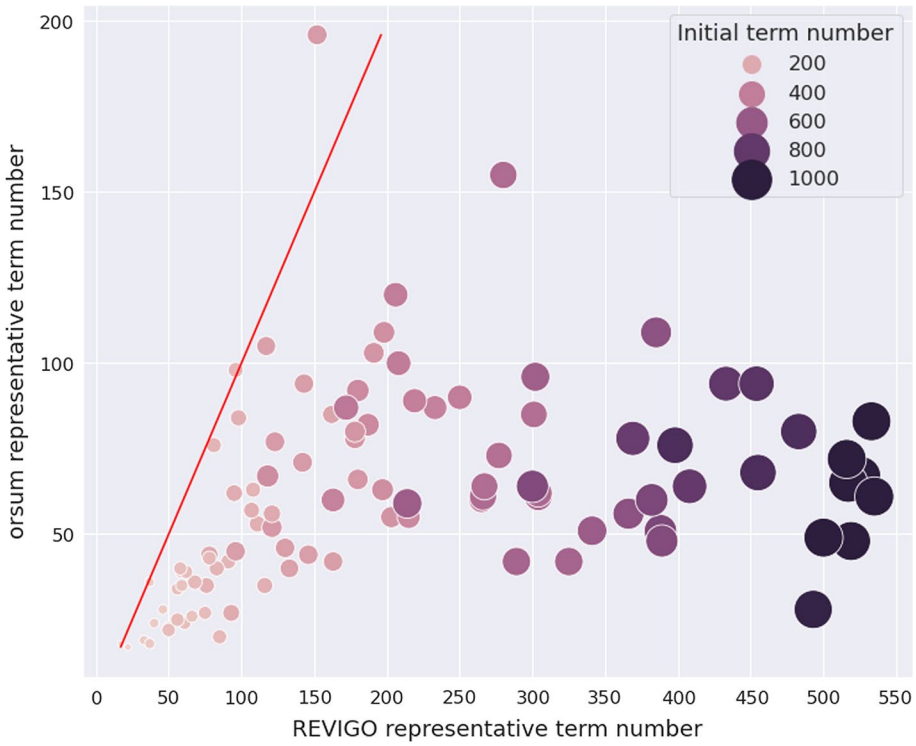

Fig. 4 Numbers of representative terms resulting from orsum and REVIGO applied to the enrichment analyses of artificially generated gene lists. Each point corresponds to an enrichment analysis result obtained for one of the 100 artificially generated gene lists. The size and color of a point indicates the number of terms in the original enrichment analysis result. The red line shows the coordinates where the representative term numbers in orsum and REVIGO are equal

As stated in [19], it is difficult to make a quantitative comparison between enrichment result summarization tools as their success is based on subjective measures like interpretability of the final terms list. We should stress that these tools are aimed to ease the exploration of enrichment results and the user will choose the one that fits to their need and perspective. However, for the sake of completeness, we compared the number of representative terms obtained after applying orsum and REVIGO. We observed that orsum outperformed REVIGO.

Methods that use semantic similarity are critically dependent on the similarity threshold parameters adjusted by the user. For example, in REVIGO, pairs of terms that are more similar than the user adjusted similarity threshold are compared. Additionally there are parameters that cannot be tuned by the user in the REVIGO algorithm. orsum algorithm is not dependent on any parameters; it checks the ranks of the terms and it checks whether all the genes annotated by one term are annotated by the other. The two parameters of orsum are not mandatory, they can improve the final results based on the user's requirements. The minimum term size parameter is used to remove the terms that annotate very small numbers of genes; many other terms might annotate the same set of genes. The maximum representative term size parameter is useful when a term that is more general than the user's requirement is very significant and therefore represents many other terms. Our perspective here is that, if a general term is statistically significant, then this should be highlighted and presented to the user as representative for its less significant descendant terms. The user can further examine the detailed results to check for discarded terms. The user might also choose to see the smaller, descendant terms in the orsum output. Setting the maximum term size parameter to a low value prevents a large term to be representative of other terms. More specific terms remain in the filtered results. Both parameters of orsum are straightforward and their results are foreseeable.

orsum shares the advantages of filtering tools. It runs on enrichment analysis results provided by the user, which means that the user is free to choose any enrichment analysis tool. In addition, as the resulting terms are selected from the input, they are study-specific. In contrary to existing tools, the annotation data (GMT file) is provided by the user. This makes it possible to use the same annotations as the ones used in the enrichment analysis. The user also does not need the developers to update the files in a need for using up-to-date information.

Conclusions

In this study, we developed orsum, a Python-based tool for the filtering of enrichment analysis results. orsum is easy to use, the applied principle for filtering is easy to understand, the sizes of the resulting representative terms can be adjusted, and the output files are informative even at a quick glance. orsum's ability to work with multiple enrichment results will allow it to be used in comparative or integrative studies, for instance, in the investigation of multiple diseases or multiple -omics datasets.

Availability and requirements

Project name: orsum (v1.4)
Project home page: https://anaconda.org/bioconda/orsum
Operating system(s): Platform independent
Programming language: Python
Other requirements: Python 3.6 or higher
License: MIT License
Any restrictions to use by non-academics: None

Abbreviations

AD	Alzheimer's disease
ALS	Amyotrophic lateral sclerosis
GMT	Gene matrix transposed
GO	Gene ontology
GSEA	Gene set enrichment analysis
HD	Huntington's disease
KEGG	Kyoto encyclopedia of genes and genomes
MSigDB	The molecular signatures database
PD	Parkinson's disease

Acknowledgements
We thank the members of European Joint Programme on Rare Diseases, in particular Annika Jacobsen, Eleni Mina, Friederike Ehrhart, Nazli Sila Kara, Nuria Queralt Rosinach and Tooba Abbassi-Daloii for their support and interesting discussions. We thank Alberto Valdeolivas, David P. Hirst, Lionel Spinelli and Thien-Phong Vu Manh for their valuable comments on the manuscript.

Author Contributions
OO conceived the idea for orsum. OO and MT developed the package. AB supervised the study. OO and AB wrote the manuscript. All authors read and approved the final manuscript.

Funding
OO has received funding from the Excellence Initiative of Aix-Marseille University - A*Midex, a French "Investissements d'Avenir" programme - Institute MarMaRa AMX-19-IET-007. MT has received funding from the Excellence Initiative of Aix-Marseille University - A*Midex, a French "Investissements d'Avenir" programme, and the European Union's Horizon 2020 research and innovation programme under the EJP RD COFUND-EJP No 825575.

Availability of data and materials
The source code of orsum is available on https://github.com/ozanozisik/orsum. orsum can be installed via bioconda (https://anaconda.org/bioconda/orsum). All data and materials related to the use case and the comparison with REVIGO are available on https://doi.org/10.5281/zenodo.6036031.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests

Received: 10 February 2022 Accepted: 8 July 2022
Published online: 23 July 2022

References

1. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. Gene Ontol Consortium Nat Genet. 2000;25(1):25–9.
2. The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. Nucleic Acids Res. 2019;47(D1):330–8.
3. Jassal B, Matthews L, Viteri G, Gong C, Lorente P, Fabregat A, Sidiropoulos K, Cook J, Gillespie M, Haw R, Loney F, May B, Milacic M, Rothfels K, Sevilla C, Shamovsky V, Shorser S, Varusai T, Weiser J, Wu G, Stein L, Hermjakob H, D'Eustachio P. The reactome pathway knowledgebase. Nucleic Acids Res. 2020;48(D1):498–503.

4. Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, Foulger R, Eilbeck K, Lewis S, Marshall B, Mungall C, Richter J, Rubin GM, Blake JA, Bult C, Dolan M, Drabkin H, Eppig JT, Hill DP, Ni L, Ringwald M, Balakrishnan R, Cherry JM, Christie KR, Costanzo MC, Dwight SS, Engel S, Fisk DG, Hirschman JE, Hong EL, Nash RS, Sethuraman A, Theesfeld CL, Botstein D, Dolinski K, Feierbach B, Berardini T, Mundodi S, Rhee SY, Apweiler R, Barrell D, Camon E, Dimmer E, Lee V, Chisholm R, Gaudet P, Kibbe W, Kishore R, Schwarz EM, Sternberg P, Gwinn M, Hannick L, Wortman J, Berriman M, Wood V, de la Cruz N, Tonellato P, Jaiswal P, Seigfried T, White R. The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res. 2004. https://doi.org/10.1093/nar/gkh036.
5. Guide to GO subsets. http://geneontology.org/docs/go-subset-guide/. [Online; accessed 27.10.2021]
6. Liberzon A, Birger C, Thorvaldsdóttir H, Ghandi M, Mesirov JP, Tamayo P. The molecular signatures database (MSigDB) hallmark gene set collection. Cell Syst. 2015;1(6):417–25.
7. Scala G, Serra A, Marwah VS, Saarimäki LA, Greco D. FunMappOne: a tool to hierarchically organize and visually navigate functional gene annotations in multiple experiments. BMC Bioinform. 2019;20(1):79.
8. Kanehisa M, Goto S. KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. 2000;28(1):27–30.
9. Yu G, Li F, Qin Y, Bo X, Wu Y, Wang S. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics. 2010;26(7):976–8.
10. Pesquita C, Faria D, Falcão AO, Lord P, Couto FM. Semantic similarity in biomedical ontologies. PLoS Comput Biol. 2009;5(7):1000443.
11. Zeeberg BR, Liu H, Kahn AB, Ehler M, Rajapakse VN, Bonner RF, Brown JD, Brooks BP, Larionov VL, Reinhold W, Weinstein JN, Pommier YG. RedundancyMiner: de-replication of redundant GO categories in microarray and proteomics analysis. BMC Bioinform. 2011;12:52.
12. Wang JZ, Du Z, Payattakool R, Yu PS, Chen CF. A new method to measure the semantic similarity of GO terms. Bioinformatics. 2007;23(10):1274–81.
13. Huang DW, Sherman BT, Tan Q, Collins JR, Alvord WG, Roayaei J, Stephens R, Baseler MW, Lane HC, Lempicki RA. The DAVID gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biol. 2007;8(9):183.
14. Bindea G, Mlecnik B, Hackl H, Charoentong P, Tosolini M, Kirilovsky A, Fridman WH, Pagès F, Trajanoski Z, Galon J. ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. Bioinformatics. 2009;25(8):1091–3.
15. Yu G, Wang LG, Han Y, He QY. clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS. 2012;16(5):284–7.
16. Wu T, Hu E, Xu S, Chen M, Guo P, Dai Z, Feng T, Zhou L, Tang W, Zhan L, Fu X, Liu S, Bo X, Yu G. clusterProfiler 4.0: a universal enrichment tool for interpreting omics data. Innovation. 2021. https://doi.org/10.1016/j.xinn.2021.100141.
17. Ulgen E, Ozisik O, Sezerman OU. pathfindR: an R package for comprehensive identification of enriched pathways in omics data through active subnetworks. Front Genet. 2019;10:858.
18. Brionne A, Juanchich A, Hennequet-Antier C. ViSEAGO: a Bioconductor package for clustering biological functions using gene ontology and semantic similarity. BioData Min. 2019;12:16.
19. Supek F, Bošnjak M, Škunca N, Šmuc T. REVIGO summarizes and visualizes long lists of gene ontology terms. PLoS One. 2011;6(7):21800.
20. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 2003;13(11):2498–504.
21. Szklarczyk D, Gable AL, Lyon D, Junge A, Wyder S, Huerta-Cepas J, Simonovic M, Doncheva NT, Morris JH, Bork P, Jensen LJ, Mering CV. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res. 2019;47(D1):607–13.
22. Doncheva NT, Morris JH, Gorodkin J, Jensen LJ. Cytoscape StringApp: network analysis and visualization of proteomics data. J Proteome Res. 2019;18(2):623–32.
23. Reimand J, Kull M, Peterson H, Hansen J, Vilo J. g:Profiler-a web-based toolset for functional profiling of gene lists from large-scale experiments. Nucleic Acids Res. 2007. https://doi.org/10.1093/nar/gkm226.
24. Raudvere U, Kolberg L, Kuzmin I, Arak T, Adler P, Peterson H, Vilo J. g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). Nucleic Acids Res. 2019;47(W1):191–8.
25. Kolde R, Vilo J. GOsummaries: an R Package for Visual Functional Annotation of Experimental Data. F1000Res 4, 574 (2015)
26. Fortino V, Alenius H, Greco D. BACA: bubble chArt to compare annotations. BMC Bioinformatics. 2015;16:37.
27. Spinelli L, Carpentier S, Montañana Sanchis F, Dalod M, Vu Manh TP. BubbleGUM: automatic extraction of phenotype molecular signatures and comprehensive visualization of multiple Gene Set Enrichment Analyses. BMC Genomics. 2015;16:814.
28. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci USA. 2005;102(43):15545–50.
29. Piñero J, Ramírez-Anguita JM, Saüch-Pitarch J, Ronzano F, Centeno E, Sanz F, Furlong LI. The DisGeNET knowledge platform for disease genomics: 2019 update. Nucleic Acids Res. 2020;48(D1):845–55.
30. DisGeNET - a database of gene-disease associations. https://www.disgenet.org/. [Online; accessed 20.12.2021]
31. Murphy MP, LeVine H. Alzheimer's disease and the amyloid-beta peptide. J Alzheimers Dis. 2010;19(1):311–23.
32. Ticozzi N, LeClerc AL, Keagle PJ, Glass JD, Wills AM, van Blitterswijk M, Bosco DA, Rodriguez-Leyva I, Gellera C, Ratti A, Taroni F, McKenna-Yasek D, Sapp PC, Silani V, Furlong CE, Brown RH, Landers JE. Paraoxonase gene mutations in amyotrophic lateral sclerosis. Ann Neurol. 2010;68(1):102–7.

Publisher's Note