GeneSetCluster 2.0: a comprehensive toolset for summarizing and integrating gene-sets analysis

Asier Ortega-Legarreta[1]*, Alberto Maillo[2]*, Daniel Mouzo[1], Ana Rosa López-Pérez[1], Lara Kular[3], Majid Pahlevan Kakhki[3], Jesper Tegner[2,4,5,6], Maja Jagodic[3], Vincenzo Lagani[2,7], Ewoud Ewing[3&], David Gomez-Cabrero[1,2&].

[1] Translational Bioinformatics Unit, Navarrabiomed, Hospital Universitario de Navarra (HUN), Universidad Pública de Navarra (UPNA), IdiSNA, 31008 Pamplona, Spain.

[2] Biological and Environmental Sciences and Engineering Division, King Abdullah University of Science and Technology, Saudi Arabia.

[3] Department of Clinical Neuroscience, Karolinska Institute, Stockholm, Sweden.

[4] Unit of Computational Medicine, Department of Medicine, Center for Molecular Medicine, Karolinska Institutet, Karolinska University Hospital, L8:05, SE-171 76, Stockholm, Sweden

[5] Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia.

[6] Science for Life Laboratory, Tomtebodavagen 23A, SE-17165, Solna, Sweden

[7] SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence, Thuwal 23952, Saudi Arabia

* Contributed equally [&] Corresponding author

Background

GeneSetCluster is an R package designed to summarize and integrate results from multiple gene-set analyses (GSA), regardless of the specific method used, as long as gene-sets are the primary focus. GeneSetCluster clusters gene-sets based on shared genes, providing a high-level overview of the involved biological pathways, thereby facilitating interpretability. However, the original version faced limitations in managing redundancies across multiple gene sets. Additionally, programmatic tools like GeneSetCluster can be challenging for users with limited coding expertise.

Results

We introduce GeneSetCluster 2.0 which substantially improves upon its predecessor. This update presents a new methodology for addressing duplicated gene-sets and incorporates a seriation-based clustering algorithm that reorders data, enabling the identification of patterns. The updated version is optimized for faster computational performance through parallelization, reducing execution times. Furthermore, the 2.0 version also enhances cluster annotation by identifying relevant tissues and biological processes associated with each cluster. Finally, to improve accessibility and usability, we have developed a user-friendly web application, making GeneSetCluster 2.0 suitable for a broader audience, including those without programming expertise. We also ensured the interplay between the R package, targeting users with programming knowledge, and the web-application, which targets the rest of the users.

Conclusion

GeneSetCluster 2.0 significantly improves the original version by providing a more comprehensive, intuitive, and efficient exploration of GSA results. On the other hand, it bridges the gap between bioinformaticians and clinicians in multidisciplinary teams. Both GeneSetCluster 2.0 web version and R package, along with detailed installation and usage documentation, are available on GitHub at https://github.com/TranslationalBioinformaticsUnit/GeneSetCluster2.0. The web application can be accessed at https://translationalbio.shinyapps.io/genesetcluster/.

1. Background

In molecular biology experiments, researchers often identify molecular elements, such as genes or proteins, associated with specific traits or conditions. Gene-set analysis (GSA) is a key approach for interpreting these findings by transforming basic gene lists into more interpretable sets of the biological processes (a.k.a. pathways or gene-sets) which are most represented in the data. GSA methods help researchers understand complex biological systems, discover potential therapeutic targets, and uncover disease mechanisms (1). These gene-sets can be originated from gene expression studies or genomic region analyses, such as DNA methylation or chromatin accessibility studies, using tools like GREAT (2). Other GSA methods, such as over-representation analysis (ORA)(3) and gene-set enrichment analysis (GSEA)(4), identify significant gene-sets from genes. ORA evaluates whether the proportion of relevant genes within a gene-set is higher than would be expected by chance. GSEA, on the other hand, ranks all genes based on their association with the trait under study and then tests if genes in a particular gene-set tend to appear toward the top of this ranking, indicating their overall importance about the trait.

Regardless of the methodology used, interpreting GSA results remains challenging. Researchers frequently need to analyze multiple contrasts within a single study (e.g., contrasting several drugs against controls), which generates several (possibly large) lists of relevant gene-sets. Integrating results across different GSA analyses is further complicated by variations in gene-set definitions and naming conventions. These discrepancies can introduce inconsistencies that hinder the identification of meaningful and/or accurate conclusions. Moreover, many gene-sets contain overlapping genes, making it difficult to distinguish between functional roles that involve similar genes. This redundancy often leads to the identification of an excessive number of gene-sets, some of which may lack biological relevance, thereby complicating the determination of the most critical pathways.

Addressing these challenges, GeneSetCluster 1.0 was developed to improve the interpretability of GSA results (5). GeneSetCluster 1.0 integrates and synthesizes outputs from multiple gene-set tools and comparisons and identifies relevant candidate pathways and conditions by clustering gene-sets. This clustering methodology is based on distance scores between gene-sets,

calculated using relative risk (RR) measurements similar to those utilized in comorbidity statistics. The RR metric assesses the observed number of shared genes between two gene-sets against the expected number. By identifying clusters of gene-sets, the tool provides a summarized representation of the key biological functions within the system of interest, enhancing result interpretability.

However, we realized – through the feedback from users - that GeneSetCluster 1.0 has certain biases and limitations, particularly in the clustering process. One issue is the identification of identical gene-sets across multiple analyses, which can unintentionally reinforce certain clusters of gene-sets. Conversely, including “outlier” gene-sets, which show minimal or no similarities to others, presents another challenge. These outliers often form small clusters while simultaneously forcing all other gene-sets into a single larger cluster. Moreover, the tool’s accessibility is limited, as it is available only as an R package, making it suitable primarily for users with bioinformatics and programming experiences.

To overcome these limitations, we have developed GeneSetCluster 2.0. It incorporates a new method for handling duplicated gene-sets, and an alternative gene-set clustering approach using seriation-based algorithms, which better handle outlier gene-sets. Additionally, GeneSetCluster 2.0 is now accessible online through a user-friendly web interface, making it usable to a wider range of users, including clinicians and biologists. GeneSetCluster 2.0 enables seamless interaction between the R package and the web interface, allowing data and analysis workflows to be shared and refined across platforms, thereby enhancing collaboration within multidisciplinary teams. Finally, we have implemented new features, such as an annotation toolkit and tissue enrichment analysis for each gene-set cluster, providing deeper biological insights.

2. Implementation

2.1 GeneSetCluster 1.0

GeneSetCluster 1.0 is an R package originally developed to address a fundamental challenge in gene-set analysis: interpreting results that often contain dozens or hundreds of gene-sets with overlapping genes. Many of these gene-sets represent similar biological processes under different names, complicating the identification of overarching themes.

The first version introduced a methodology to cluster all input gene-sets, regardless of duplication or overlap, based on their shared gene content. It employed a RR distance metric, inspired by comorbidity statistics, to measure the similarity between gene-sets (6). Clustering was performed using either k-means or hierarchical methods, ensuring that all input gene-sets were assigned to a cluster. This approach helped researchers cluster related gene-sets and identify major biological themes within their data.

Additionally, GeneSetCluster 1.0 enabled the integration of results from different GSA tools and experimental conditions, offering a unified framework for analyzing multiple analyses simultaneously.

2.2 Approaching duplicated gene-sets

Multiple GSA results often identify the same gene-sets (e.g., identical Gene Ontology IDs), even though the sub-set of genes associated with each GSA result may vary slightly. In GeneSetCluster 1.0, these duplicated gene-sets were treated independently. We refer to this approach as the "*Raw Genesets*" methodology. However, this methodology occasionally introduced bias into the clustering process, resulting – for instance - in the largest cluster composed of these gene-sets and distinct from the others in the analysis. This could impact the overall biological interpretation of the clusters.

To address this limitation, GeneSetCluster 2.0 introduces an alternative approach called the "*Unique Genesets*" methodology. This method detects repeated gene-sets with identical ID labels and merges them into a single, unified GOID containing the union of all genes associated with

both sets. For example, GO:0007612 (biological process related to “learning”) is identified in the first GSA analysis, due to the genes *Pak6*, *Reln*, and *Adcy3* and in the second analysis due to *Reln*, *Adcy3*, and *Eif2ak4* genes. The "*Unique Genesets*" methodology merges these results, counting GO:0007612 only once and consolidates the genes in a single list: *Pak6*, *Reln*, *Adcy3*, and *Eif2ak4*. Consequently, each gene-set is then treated as a unique entity within the clustering process, eliminating the bias caused by handling duplicated entries separately (Fig. 1a).

This refined approach simplifies the analysis, resulting in clearer and more interpretable clusters. It also facilitates more consistent comparisons across studies and enhances the understanding of the biological significance across different research contexts.

2.3 Clustering approaches

GeneSetCluster 2.0 enhances its predecessor by offering two clustering approaches: 1) the "*Classic*" approach, originally implemented in GeneSetCluster 1.0, and 2) a new “*Seriation-based*” clustering approach, which utilizes seriation algorithms to provide automated clustering solutions. For continuity, the “*Classic*” method remains available in the updated version.

The “*Classic*” approach implements established clustering methods such as hierarchical clustering and k-means (7)(8). These methods require users to specify the number of clusters, and thus GeneSetCluster 1.0 incorporates silhouette analysis (9) and the elbow method (10) to suggest the optimal number.

The “*Seriation-based*” clustering leverages seriation techniques to identify coherent groups by arranging gene-sets in a linear sequence based on their pairwise similarities or dissimilarities. Unlike “*Classic*” clustering, which strictly divides gene-sets into separate groups, seriation emphasizes uncovering patterns within the data (11).

The “*Seriation-based*” approach tests 32 seriation algorithms (Table S1) from the *seriation* R package (12) to automatically find the optimal clustering solution (Fig. 1b). Briefly, each seriation algorithm undergoes a four-step evaluation process:

1. **Initial Seriation**: The distance matrix, which represents the pairwise similarities between gene-sets, is reordered by the algorithm, optimizing the placement of similar gene-sets next to each other.
2. **Threshold-Based Segmentation**: Potential clusters are identified within the reordered matrix by applying a predefined similarity threshold (default set at 0.6). This process groups adjacent gene-sets with similarity scores above this threshold. The similarity threshold can be manually adjusted by the users.
3. **Cluster Size Optimization**: Clusters are refined to meet a specified size requirement (4 to 10 gene-sets per cluster) ensuring each cluster contains a meaningful number of gene-sets. This requirement prevents excessive fragmentation and over-large clusters that may overlook biologically relevant relationships. Users can also modify this size threshold manually.
4. **Score computation:** The output of each algorithm is scored based on three weighted criteria: Hamiltonian path length (minimizing distances between consecutive gene-sets) (13), anti-Robinson form criterion ensuring dissimilarities increase when moving away from the diagonal (14), and a total number of clustered gene-sets (maximizing the number of gene-sets successfully grouped), weighted at 40%, 40%, and 20%, respectively (see Supplementary material for details).

Finally, the seriation algorithm with the highest score from step 4 is automatically selected as the optimal solution. However, users can override this automatic selection by specifying their preferred algorithm.

The "*Seriation-based*" method provides several key advantages: 1) it effectively isolates outlier gene-sets that share low similarity with others, preventing them from introducing noise into the clustering results; 2) the optimal seriation algorithm is automatically selected based on objective scoring criteria, minimizing user intervention; and 3) It can detect gradual transitions or hierarchical structures that might be overlooked by traditional clustering methods (see *Results section*). This new method provides a more detailed and comprehensive understanding of the gene-sets relationships.

GeneSetCluster 2.0 incorporates "*Classic*" and "*Seriation-based*" clustering techniques, providing a robust toolkit for exploring and understanding complex gene-set relationships. The "*Classic*" method offers well-defined patterns, while the "*Seriation-based*" approach reveals subtle transitions or hierarchical structures.

2.4 Sub-clustering

GeneSetCluster 2.0 introduces *BreakUpCluster*, a refinement feature for the "*Classic*" clustering approach that enables the subdivision of large gene-sets clusters into smaller sub-clusters using k-means or hierarchical clustering (Fig. 2). This targeted refinement addresses the issue of overly large clusters caused by noise, which can obscure distinct biological patterns.

By allowing researchers to focus on specific clusters of interest, *BreakUpCluster* provides a detailed exploration of finer gene-set relationships while preserving the overall clustering framework. This flexibility enhances the interpretability of complex datasets, making it easier to uncover biologically relevant insights.

2.5 Functional annotations

To further enhance the interpretability of gene-set clusters, GeneSetCluster 2.0 offers multiple annotation methods. A key concept used during annotation is *cluster-associated genes*, which correspond to the union of the genes contained in all gene-sets of a specific cluster.

Annotations

We apply ORA, using the *enrichGO* function (7), to the *cluster-associated genes,* allowing the identification of enriched biological processes within each cluster. This helps researchers quickly determine whether particular pathways or processes are enriched, providing a deeper understanding of the biological relevance and the potential functional roles of the genes involved.

In addition to ORA, GeneSetCluster 2.0 incorporates a semantic enrichment analysis when utilizing Gene Ontology (GO) terms to organize terms around shared biological themes. We apply this enrichment to each cluster's gene-sets using the *simplifyGO* function from the

SimplifyEnrichment package (8). Internally, *simplifyGO* extracts biological themes by computing how closely related GO terms are, leveraging their shared ancestry in the GO hierarchy. This process clarifies the biological context of each cluster by emphasizing functional themes within the GO terms.

Tissue enrichment

GeneSetCluster 2.0 incorporates a tissue enrichment analysis (limited to human data) that identifies tissues where *cluster-associated genes* show significant activity. The analysis is based on a custom tissue expression database derived from GTEx (15), which contains median expression levels across 54 human tissues. For each tissue, genes are ranked according to their expression levels, establishing tissue-specific expression profiles.

The tissue enrichment analysis employs GSEA to evaluate whether *cluster-associated genes* are significantly enriched in these tissue-specific rankings, generating Normalized Enrichment Scores (NES) that quantify the degree of enrichment. This approach provides insights into the tissue-specific context of each cluster, enhancing the biological interpretation of the results.

For computational efficiency, users can access the tissue expression database either through the API integrated in GeneSetCluster 2.0 or direct download from the repository for local use https://figshare.com/articles/dataset/TissueDatabase/25965664?file=46802263.

2.6 Parallelization

GeneSetCluster 1.0 was slow in the case of a large number of gene sets. Therefore, GeneSetCluster 2.0 includes optimized parallelization to improve computational efficiency, especially in the *CombineGeneSets* step (Fig. 2), where the distance between gene-sets is calculated. This crucial step, which often demands significant computational resources, leverages now parallel processing techniques facilitated by the R packages *doParallel* and *foreach*. This optimization ensures that the computational workload is distributed efficiently across multiple processors or threads, resulting in faster execution times and improved overall performance.

To assess the effectiveness of this optimization, the *CombineGeneSets* function was tested using datasets of various sizes: small (239 gene-sets), medium (1000 gene-sets), and large (2287 gene-sets), with different numbers of threads: 1, 2, 4, 6, 8, and 10. The execution time was measured for each condition repeated ten times. The tests were conducted on a workstation equipped with Apple M1 Pro processor with a 10-core CPU and 16 GB of RAM.

The mean execution times (in seconds) for each configuration are shown in Table S2. Results showed that performance was slowest with a single thread, but execution times decreased significantly as threads were added, especially for medium and large datasets, up to four threads. For instance, the medium dataset's processing time dropped from 125 seconds with one thread to 42 seconds with four. Similarly, the large dataset achieved substantial gains, reducing processing time to 440 seconds with six threads, down from 1945 seconds with a single thread. This pattern suggests that using four to six threads effectively balances computational load and efficiency in GeneSetCluster 2.0.

By implementing parallelization, GeneSetCluster 2.0 enhances the scalability and speed of the *CombineGeneSets* step, allowing users to analyze large datasets more effectively.

3. Web Application

3.1 Overview

To enable experienced non-programming users, we developed a web-based Shiny application. The Shiny application, developed with R 4.3, is hosted on a Shiny server at the URL https://translationalbio.shinyapps.io/genesetcluster/. It is fully compatible with all operating systems and web browsers. Upon accessing the application, users are prompted to select the mandatory input parameters and upload their GSA results. Once the "*Run Analysis*" button is clicked, the results, along with corresponding plots, are generated and displayed. Users can then examine the results, add additional annotations, and perform further analyses to gain biological insights. All results and plots are available for download. Additionally, user can save their analysis by downloading an *RData* file for future use, allowing them to continue by re-uploading it in the Shiny application or moving to the R package version. This transition can also be made from the R package to the Shiny application. For a more user-friendly experience, the application can also be deployed locally from the R package using the command *run_app()* (Fig. 3a).

3.2 Graphical user interface

The user interface of the application is divided into two main sections, as illustrated in Fig. 3b. The first section, the input section, allows users to specify parameter values such as *Source*, *Gene ID*, *Organism*, and upload GSA results or template in supported formats (.txt, .csv, .xlsx). Users can also start from a previously saved analysis or import an analysis performed using the R package.

The second section, the results section, is further divided into two parts. At the top, users can view plots generated, including heatmaps of the RR matrix for the "*Classic*" clustering (under the *Heatmap* tab) and the "*Seriation-based*" algorithm (under the *Heatmap_S* tab), along with their tissue enrichment plots (available in the *Tissue* and *Tissue_S* tabs respectively). Below the plots, the results are displayed in tables, showing which gene-sets belong to each cluster (under the *Data* tab), ORA details (under the *ORA* tab), gene information (under the *Genes* tab), and tissue

data (under the *Tissue enrichment* tab). These outputs are provided for both, the "*Classic*" and "*Seriation-based*" methods, the latter being accessible under the *Seriation* tab.

The source code of the application can be accessed at https://github.com/TranslationalBioinformaticsUnit/GeneSetCluster2.0.

3.3 Results interaction features

After the user clicks the "*Run analysis*" button, the "*Classic*" method is executed by default based on the selected input parameters. Once the analysis is completed, the results and corresponding plots are displayed. Users can then choose to perform tissue enrichment analysis or apply the "*Seriation-based*" method for alternative clustering (described in Sections 2.3 and 2.5). Additionally, they can interact with the results for deeper insights. Some key functionalities for exploring the results include:

Targeted analysis: Users may want to investigate specific conditions or phenotypes in their data, focusing on a particular group of genes. This functionality allows them to assess whether their *cluster-associated genes* are enriched in these specific gene groups. Users can either import a custom gene list or select a phenotype from well-known databases for enrichment analysis. Two key databases are accessible through the application: 1) the Human Phenotype Ontologies database (HPO), which provides standardized terms for human disease-related phenotypes (16), and 2) the Mammalian Phenotype (MP) database, which offers terms for annotating mouse phenotype data (17). Overall, this feature helps uncover relevant genetic insights by emphasizing specific groups of genes within the *cluster-associated genes*.

Gene-set cluster functional characterization: Users may need to explore data at the gene level rather than just focusing on gene-sets. In this case, the application displays a table listing all *cluster-associated genes* and the frequency of each gene in the cluster. This allows them to filter genes based on frequency, helping them identify the most prevalent genes within each cluster, discover genes exclusive to a specific cluster, or search for a particular gene. Moreover, an ORA can be conducted on the filtered gene list to streamline the biological interpretation. Additionally, each gene is linked to external databases such as GeneCards for human data (18) and Mouse

Genome Informatics for mice (19), providing further context about gene functions and implications. This feature enhances flexibility and depth in gene exploration, allowing users to focus on the most relevant genes and obtain more detailed biological insights.

Export functionality: Results can be downloaded directly from the application, and the plots can be in several formats, including .jpg, .png, and .pdf. Furthermore, users can save their analysis as an *RData* file, allowing them to easily resume their work by re-uploading it to the application or importing it into the R package for further analysis. This functionality fosters effective interaction and collaboration among multidisciplinary teams, improves workflow efficiency, and supports ongoing research efforts.

In summary, GeneSetCluster is accessible online and allows users to perform all its functions in a user-friendly manner. A comprehensive and informative user guide is available at the *About* tab from https://translationalbio.shinyapps.io/genesetcluster/.

4. Results

To evaluate the performance of GeneSetCluster 2.0 and compare it with GeneSetCluster 1.0, we applied it to data from a published study (20), which explored the molecular basis of myelodysplastic syndromes with del(5q) by analyzing the transcriptional and regulatory landscape of CD34+ progenitor cells using single-cell RNA-seq. The study also assessed the impact of lenalidomide treatment on transcriptional alterations.

The authors conducted differential expression analyses across five comparisons: 1) Non-del(5q) cells of complete responders versus at diagnosis 2) Non-del(5q) cells of partial responders versus at diagnosis, 3) Del(5q) cells of partial responders versus non-responders 4) Non-del(5q) cells of complete responders versus healthy cells 5) Non-del(5q) cells of partial responders versus healthy cells. Complete responders refer to patients who achieved complete cytogenetic response (CCR), while partial responders refer to those with partial cytogenetic response (PCR) to lenalidomide treatment. From each comparison, GSEA was conducted, identifying 20 distinct gene-sets: 12 gene-sets were found in the first three comparisons, while the remaining 8 were identified in the last two comparisons. The authors manually grouped them into 7 clusters based on the biological implications: 1) Cluster 1: Related to ubiquitin processes (7 gene-sets) 2) Cluster 2: Focused on proteasome-mediated processes (2 gene-sets). 3) Cluster 3: Linked to autophagy (2 gene-sets) 4) Cluster 4: Erythropoietin signaling (2 gene-sets). 5) Cluster 5: PD-L1/PD-1 checkpoint pathway (1 gene-set) 6) Cluster 6: Phosphatidylinositol signaling system (1 gene-set). 7) Cluster 7: Mitochondrial and ribosomal translation (8 gene-sets). These original results are visualized in figure 7a of the publication.

GeneSetCluster 1.0 and GeneSetCluster 2.0 were applied to these gene-sets. In the previous version, using "*Raw Genesets*" and "*Classic*" methods, only two large clusters emerged: one on *mitochondrial translation* (8 gene-sets) and another on *proteasome-mediated processes* (12 gene-sets). In contrast, GeneSetCluster 2.0, applying "*Unique Genesets*" and "*Seriation-based*" methods, generated four distinct clusters: 1) Cluster 1: Mitochondrial translation (8 gene-sets). 2) Cluster 2: protein polyubiquitination (4 gene-sets) 3) Cluster 3: Proteasome-mediated processes (2 gene-sets) 4) Cluster 4: Autophagosome assembly (2 gene-sets). Three gene-sets

were left not clustered, as the new method does not force gene-sets into clusters and requires a minimum of two pathways per cluster. The biological significance of each cluster was determined through an ORA of the *cluster-associated genes* against the biological process database. Fig. 4 illustrates these results. This analysis could be performed using both the R package and the web version of the tool.

5. Discussion

The development of GeneSetCluster 2.0 aimed to provide better ways to interpret gene-set analysis results. We focused on three aims: methodological development, improved computational times, and a user-friendly version of the tool.

The improvements in clustering precision of GeneSetCluster 2.0 can be directly observed in its ability to avoid the over-aggregation of unrelated gene-sets that occurred in the previous version. The tool more effectively separates gene-sets by utilizing the "*Unique Genesets*" approach and the "*Seriation-based*" clustering method, allowing biologically related processes to be grouped with greater exactness. For example, when GeneSetCluster 1.0 combined 12 gene-sets into a broad *proteasome-mediated processes*, GeneSetCluster 2.0 refined this into biologically distinct clusters, including *protein polyubiquitination*, proteasome-mediated processes, and *autophagosome assembly*. This more granular clustering enables more precise biological interpretations and reduces noise from unrelated pathways being grouped. The fact that GeneSetCluster 2.0 left three gene-sets unclustered ensures that unrelated gene-sets are not forced into clusters, maintaining a higher degree of biological relevance. Thus, the new version demonstrates superior clustering capability, yielding clearer and more meaningful biological insights from complex gene-set data.

Another key improvement is the significant reduction in computational times achieved through optimized parallelization in GeneSetCluster 2.0. The *CombineGeneSets* step, which calculates distances between gene-sets, now leverages parallel processing techniques. By distributing the workload efficiently across multiple processors, the tool achieves faster execution times, especially for larger datasets. This optimization enhances scalability and ensures that users can analyses extensive data more effectively.

Additionally, incorporating the web-based version makes GeneSetCluster 2.0 more accessible to users with limited bioinformatics expertise. This accessibility fosters greater participation in the analysis process, promoting collaboration and more efficient communication within multidisciplinary teams. A standout feature of GeneSetCluster 2.0 is the seamless integration

between the R package and the web application—a capability that significantly enhances the user experience and analytical flexibility. This integration allows users to transition their analyses and results between platforms effortlessly. For instance, analyses initiated in the R package can be saved and uploaded into the web application to continue exploration with its intuitive interface. Similarly, work started in the web application can be exported back to R for more detailed or customized workflows. This two-way compatibility ensures that users can tailor their workflow to suit their preferences and expertise, facilitating real-time collaboration and efficient sharing of insights.

6. Conclusion

In conclusion, GeneSetCluster 2.0 represents a significant improvement over its predecessor, improving both the interpretability and flexibility of gene-set analysis (Fig. 2 and S1). The updated version introduces an approach for handling duplicated gene-sets and alternative clustering method, enabling clearer and more biologically relevant insights. Computational optimizations further ensure faster analysis of large datasets. Additionally, the new web application enhances accessibility for researchers with diverse coding expertise. Its integration with the R package facilitates collaboration between bioinformaticians and biologists, supporting multidisciplinary research efforts. The web application is available at https://translationalbio.shinyapps.io/genesetcluster/. The updated R package and comprehensive documentation can be downloaded on GitHub at https://github.com/TranslationalBioinformaticsUnit/GeneSetCluster2.0.

7. References

1. Mooney MA, Wilmot B. Gene set analysis: A step-by-step guide. American Journal of Medical Genetics Part B: Neuropsychiatric Genetics. 2015 Oct 8;168(7):517–27.
2. McLean CY, Bristor D, Hiller M, Clarke SL, Schaar BT, Lowe CB, et al. GREAT improves functional interpretation of cis-regulatory regions. Nat Biotechnol. 2010 May 2;28(5):495–501.
3. Khatri P, Sirota M, Butte AJ. Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges. PLoS Comput Biol. 2012 Feb 23;8(2):e1002375.
4. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences. 2005 Oct 25;102(43):15545–50.
5. Ewing E, Planell-Picola N, Jagodic M, Gomez-Cabrero D. GeneSetCluster: a tool for summarizing and integrating gene-set analysis results. BMC Bioinformatics. 2020 Dec 7;21(1):443.
6. Moni M, Liò P. comoR: a software for disease comorbidity risk assessment. J Clin Bioinforma. 2014;4(1):8.
7. Yu G, Wang LG, Han Y, He QY. clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters. OMICS. 2012 May;16(5):284–7.
8. Gu Z, Hübschmann D. simplifyEnrichment: A Bioconductor Package for Clustering and Visualizing Functional Enrichment Results. Genomics Proteomics Bioinformatics. 2023 Feb;21(1):190–202.
9. Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987 Nov;20:53–65.
10. Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognit Lett. 2010 Jun;31(8):651–66.
11. Liiv I. Seriation and matrix reordering methods: An historical overview. Statistical Analysis and Data Mining: The ASA Data Science Journal. 2010 Apr 11;3(2):70–91.
12. Hahsler M, Hornik K, Buchta C. Getting Things in Order: An Introduction to the R Package seriation. J Stat Softw. 2008;25(3).
13. Caraux G, Pinloche S. PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order. Bioinformatics. 2005 Apr 1;21(7):1280–1.
14. Earle D, Hurley CB. Advances in Dendrogram Seriation for Application to Visualization. Journal of Computational and Graphical Statistics. 2015 Jan 2;24(1):1–25.
15. Aguet F, Anand S, Ardlie KG, Gabriel S, Getz GA, Graubert A, et al. The GTEx Consortium atlas of genetic regulatory effects across human tissues. Science (1979). 2020 Sep 11;369(6509):1318–30.
16. Robinson PN, Köhler S, Bauer S, Seelow D, Horn D, Mundlos S. The Human Phenotype Ontology: A Tool for Annotating and Analyzing Human Hereditary Disease. The American Journal of Human Genetics. 2008 Nov;83(5):610–5.
17. Smith CL, Goldsmith CAW, Eppig JT. The Mammalian Phenotype Ontology as a tool for annotating, analyzing and comparing phenotypic information. Genome Biol [Internet]. 2004;6(1):R7. Available from: https://doi.org/10.1186/gb-2004-6-1-r7

18. Rebhan M, Chalifa-Caspi V, Prilusky J, Lancet D. GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support. Bioinformatics. 1998 Jan 1;14(8):656–64.
19. Eppig JT. Mouse Genome Informatics (MGI) Resource: Genetic, Genomic, and Biological Knowledgebase for the Laboratory Mouse. ILAR J. 2017 Jul 1;58(1):17–41.
20. Serrano G, Berastegui N, Díaz-Mazkiaran A, García-Olloqui P, Rodriguez-Res C, Huerga-Dominguez S, et al. Single-cell transcriptional profile of CD34+ hematopoietic progenitor cells from del(5q) myelodysplastic syndromes and impact of lenalidomide. Nat Commun. 2024 Jun 20;15(1):5272.

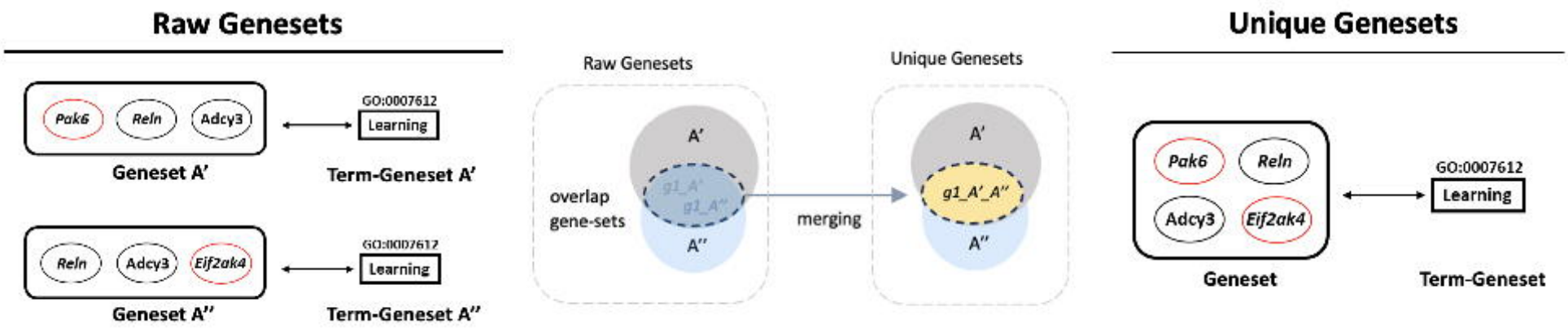
Figures Captions

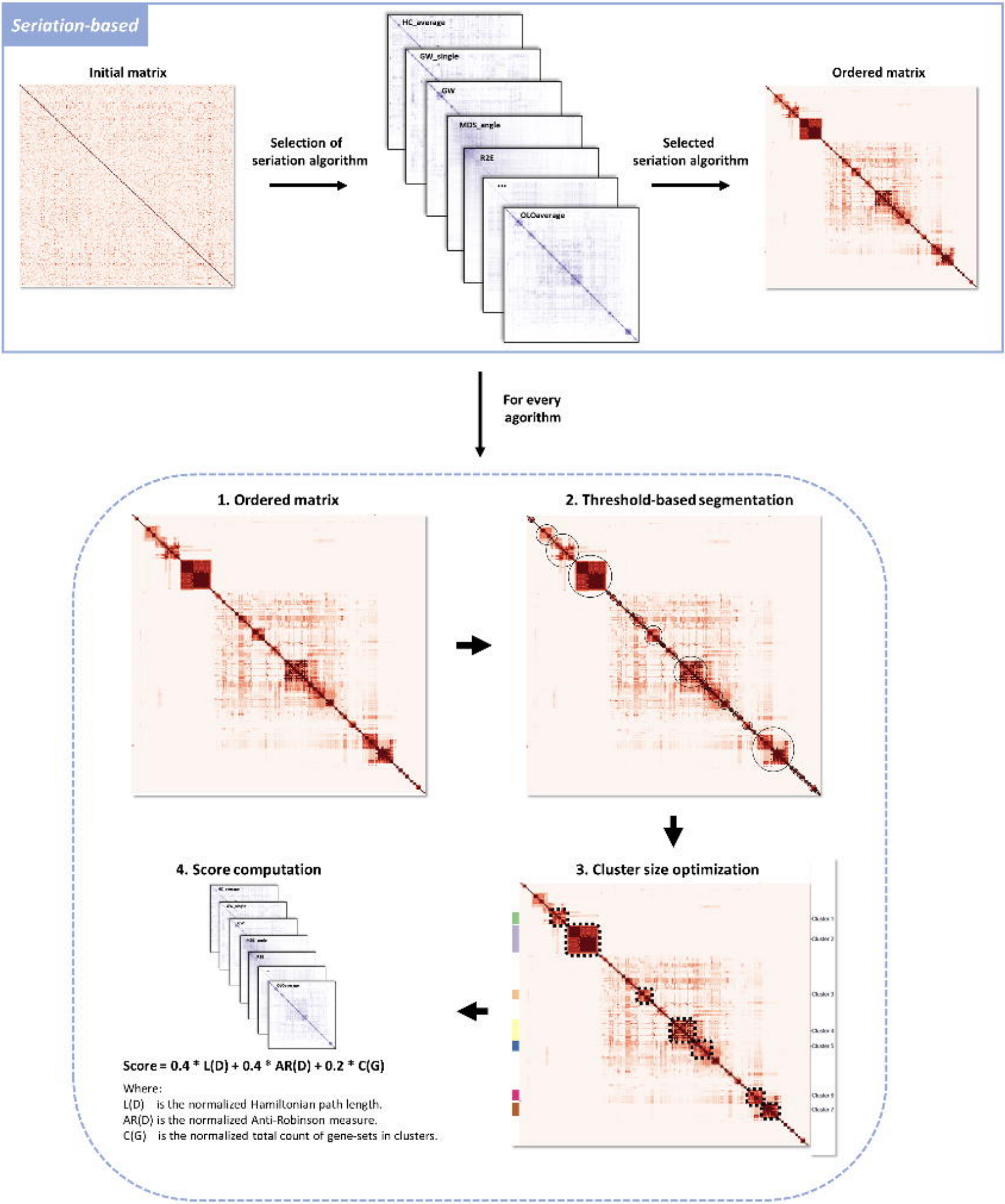
Fig 1. New implementations of GeneSetCluster 2.0. (A) Methods for handling duplicate gene-sets. Left side illustrates "*Raw Genesets*" method, considering all input gene-sets as separate entities. Right side depicts the new "*Unique Genesets*" approach, which merges duplicate gene-sets by combining their genes. (B) Visualization of the "*Seriation-based*" method.

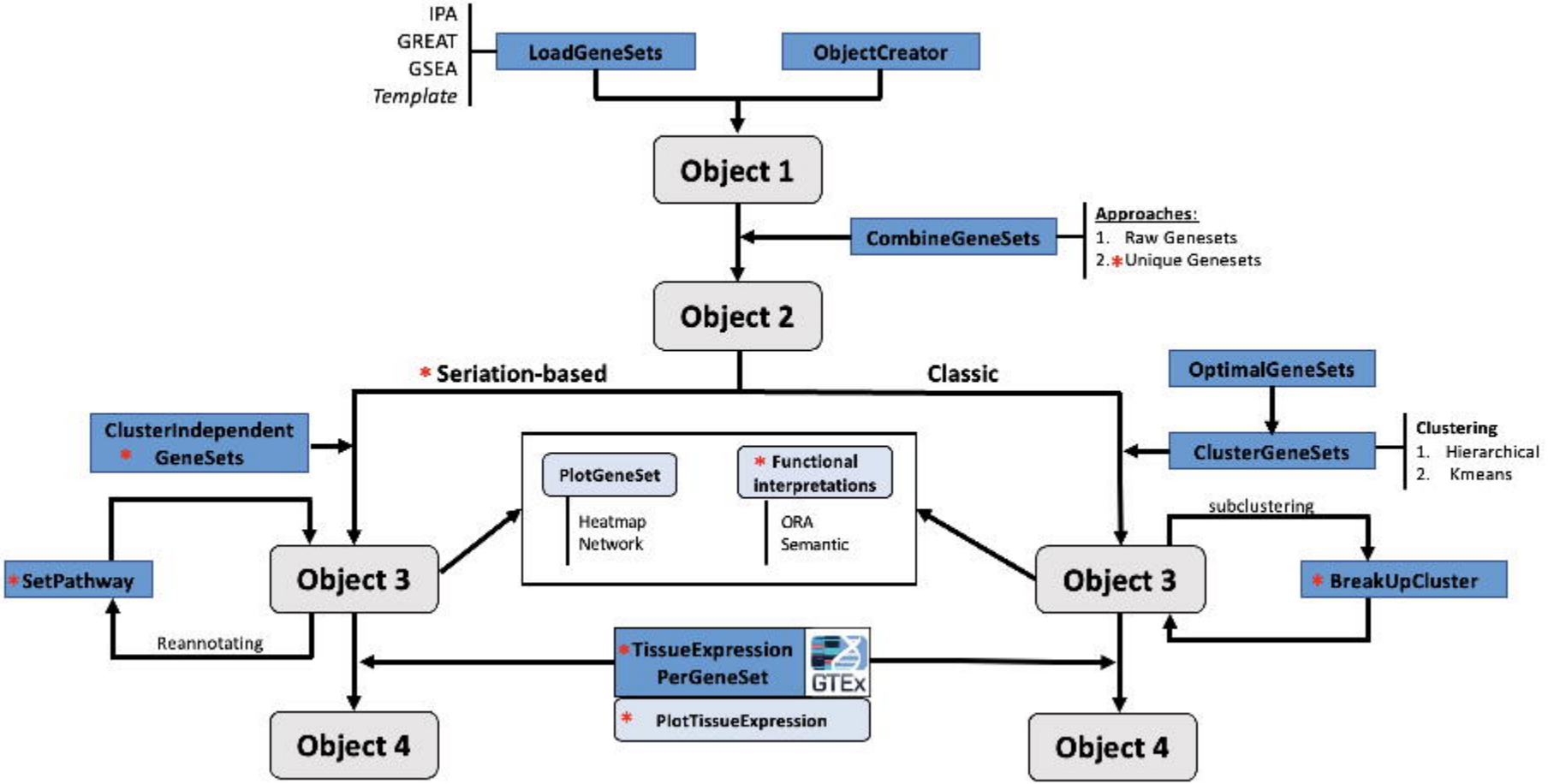
Fig 2. Workflow of GeneSetCluster 2.0. The illustration outlines all the features, with new implementations indicated by red asterisks. After uploading the GSA results, users can choose between two methods for handling duplicated gene sets: "*Raw Genesets*" and "*Unique Genesets*" during the *CombineGeneSet* step. Subsequently, the new "*Seriation-based*" clustering approach can be applied. Within this clustering method, the function *ClusterIndependentGeneSets* calculates the groups based on an optimal number, which can be manually adjusted using *SetPathway*. On the other hand, with the "*Classic*" method, clusters can now be subdivided into smaller clusters using *BreakUpCluster*. Finally, the outputs of both clustering methods can be annotated with additional functional interpretations (ORA and *Wordcloud*) or through tissue enrichment analysis using the *TissueExpressionPerGeneSet* function and its corresponding plot, *PlotTissueExpression*.

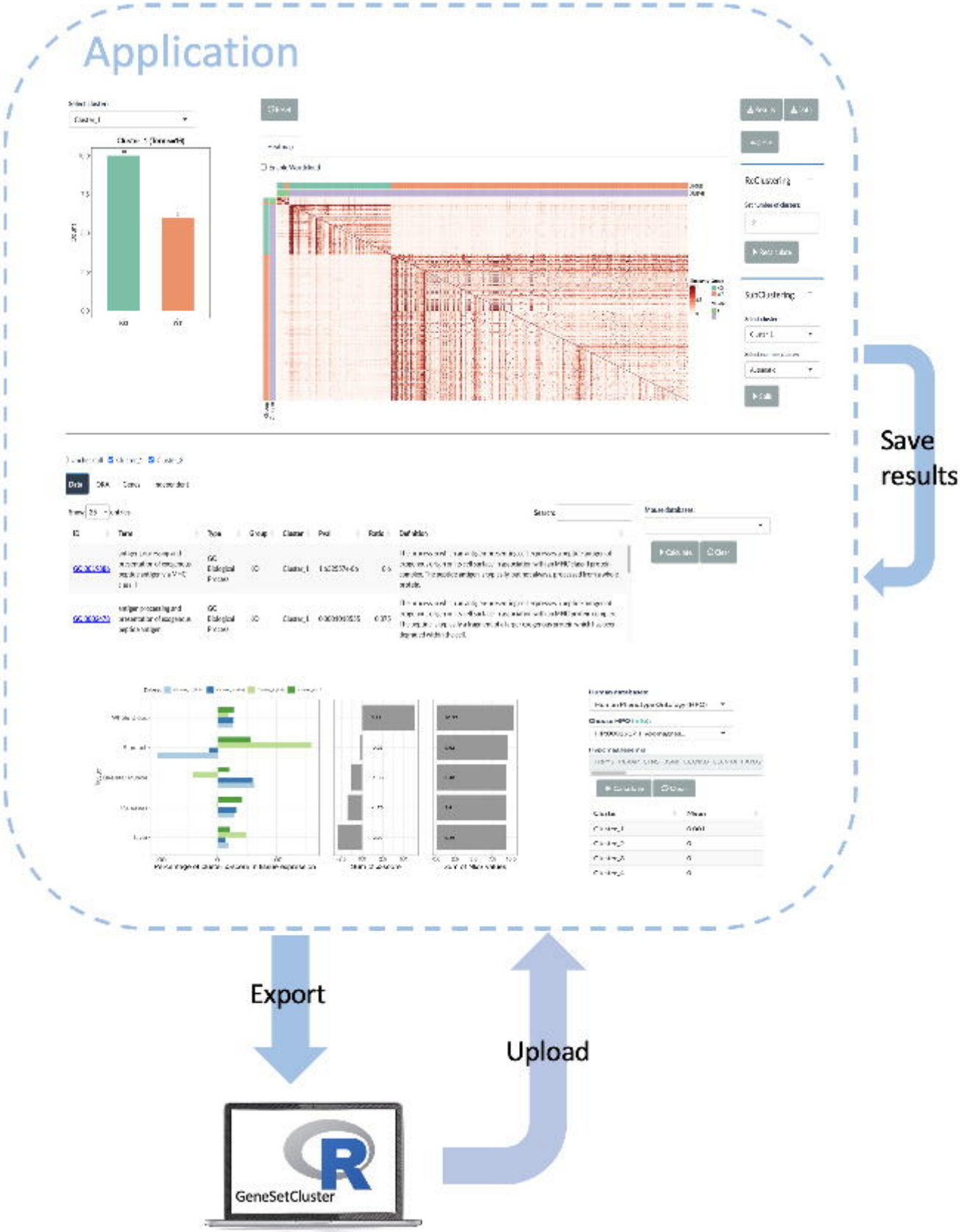
Fig 3. GeneSetCluster 2.0 web-application (A) Overview of the application and R package connection. (B) Application screenshot. The screenshot presents two main sections. The input section, highlighted in blue, allows the user to specify parameter values. After the user clicks on "*Run Analysis*" the result section will display, highlighted in green. This section displays the results in tables at the bottom and plots at the top.

Fig 4. Performance comparison from GeneSetCluster 1.0 and GeneSetCluster 2.0. The first column shows the 20 gene-sets being analyzed. The "Manual" column presents the clusters manually curated by experts. The "GeneSetCluster 1.0" column displays results generated using the "*Raw Genesets*" and "*Classic*" methods. The "GeneSetCluster 2.0" column shows the improved results using the "*Unique Genesets*" and "*Seriation-based*" methods in GeneSetCluster 2.0.


Raw Genesets
Pak6
Reln
Adcy3
GO:0007612
Learning
Geneset A'
Term-Geneset A'
Reln
Adcy3
Eif2ak4
GO:0007612
Learning
Geneset A''
Term-Geneset A''
Raw Genesets
Unique Genesets
A'
overlap
gene-sets
g1_A'
g1_A''
A''
merging
A'
g1_A'_A''
A''
Unique Genesets
Pak6
Reln
Adcy3
Eif2ak4
GO:0007612
Learning
Geneset
Term-Geneset


Seriation-based
Initial matrix
Selection of seriation algorithm
HC_average
GW_single
GW
MDS_angle
R2E
OLOaverage
Selected seriation algorithm
Ordered matrix
For every agorithm
1. Ordered matrix
2. Threshold-based segmentation
3. Cluster size optimization
4. Score computation
Score = 0.4 * L(D) + 0.4 * AR(D) + 0.2 * C(G)
Where:
L(D) is the normalized Hamiltonian path length.
AR(D) is the normalized Anti-Robinson measure.
C(G) is the normalized total count of gene-sets in clusters.


IPA
GREAT
GSEA
Template
LoadGeneSets
ObjectCreator
Object 1
CombineGeneSets
Approaches:
1. Raw Genesets
2. * Unique Genesets
Object 2
* Seriation-based
Classic
OptimalGeneSets
ClusterIndependent * GeneSets
ClusterGeneSets
Clustering
1. Hierarchical
2. Kmeans
PlotGeneSet
Heatmap
Network
* Functional interpretations
ORA
Semantic
subclustering
* SetPathway
Object 3
Object 3
* BreakUpCluster
Reannotating
* TissueExpression PerGeneSet
GTEx
* PlotTissueExpression
Object 4
Object 4


Application
Save results
Export
Upload
GeneSetCluster

GeneSet Cluster | Main | About

New | Upload | Results

Download template | Examples

Source

○ GREAT ○ IPA ◉ GSEA ○ Template

Gene ID

○ Ensembl ID ○ Symbol ◉ Entrez ID

Organism

◉ Homo sapiens ○ Mus musculus

Approaches

○ Unique GeneSet ◉ Unique Pathways

Upload files

Browse... No file selected

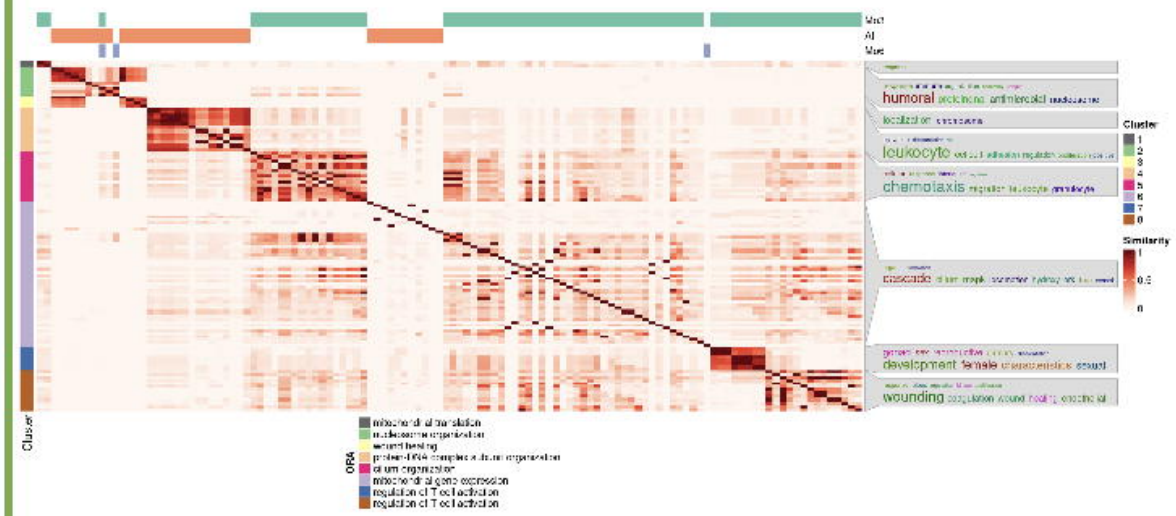
File	Group
Covid19AI_Healthy.csv	AI
Covid193Mo_Healthy.csv	Mo3
Covid196Mo_Healthy.csv	Mo6

Run analysis

Reset

Heatmap | Tissue | Heatmap_S | Tissue_S

Enable Wordcloud



Results | Data

Images

ReClustering +

SubClustering +

☐ Uncheck all ☑ Cluster_1 ☑ Cluster_2 ☑ Cluster_3 ☑ Cluster_4 ☑ Cluster_5 ☑ Cluster_6 ☑ Cluster_7 ☑ Cluster_8

Data | ORA | Genes | Tissue enrichment | Seriation

Show 25 entries

Search:

Human databases:

Calculate | Clear

ID	Term	Type	Group	Cluster	Pval	Ratio	Definition
GO:0046683	response to organophosphorus	GO Biological Process	Mo3	Cluster_1	0.0082759831848814	92.478	Any process that results in a change in state or activity of a cell or an organism (in terms of movement, secretion, enzyme production, gene expression, etc.) as a result of an organophosphorus stimulus. Organophosphorus is a compound containing phosphorus bound to an organic molecule; several organophosphorus compounds are used as insecticides, and they are highly toxic cholinesterase inhibitors.
GO:0014074	response to purine-containing compound	GO Biological Process	Mo3	Cluster_1	0.0178345205269334	85.08	Any process that results in a change in state or activity of a cell or an organism (in terms of movement, secretion, enzyme production, gene expression, etc.) as a result of a purine-containing compound stimulus.
GO:0006334	nucleosome assembly	GO Biological Process	AI	Cluster_2	1.34375e-07	58.423	The aggregation, arrangement and bonding together of a nucleosome, the beadlike structural units of eukaryotic chromatin composed of histones and DNA.
GO:0034728	nucleosome organization	GO Biological Process	AI	Cluster_2	1.34375e-07	54.737	A process that is carried out at the cellular level which results in the assembly, arrangement of constituent parts, or disassembly of one or more nucleosomes.

Showing 1 to 25 of 122 entries

Previous 1 2 3 4 5 Next

Gene-sets	Manual	GeneSetCluster 1.0	GeneSetCluster 2.0
Ubiquitin-protein transferase activity	Related to Ubiquitin processes	Proteasome-mediated ubiquitin-dependent protein catabolic process	Protein polyubiquitination
Ubiquitin-like protein conjugating enzyme binding			
Ubiquitin conjugating enzyme binding			
Ubiquitin mediated proteolysis			
Proteasome-mediated ubiquitin-dependent protein catabolic process	Proteasome mediated processess		Proteasome-mediated ubiquitin-dependent protein catabolic process
Proteasomal protein catabolic process			
Autophagosome assembly	Related to autophagy		Autophasome assembly
Autophagosome organization			
Autophagy			-
Signaling By Erythropoietin	Erythropoietin signaling		-
PD-L1 expression and PD-1 checkpoint pathway in cancer	PD-L1/PD-1 pathway in cancer		-
Phosphatidylinositol signaling system	Phosphatidylinositol signaling system		-
Mitochondrial translational elongation	Related to mitochondrial and ribosomal translation	Mitochondrial translation	Mitochondrial translation
Translational termination			
Mitochondrial translational termination			
Mitochondrial translation			
Translational elongation			
Translation			
Peptide biosynthetic process			
rRNA processing			