NAR Genomics and Bioinformatics, 2026, **8**, lqag043
https://doi.org/10.1093/nargab/lqag043
Computational Methods for Transcriptome Analysis

Analysis and visualization of expression patterns with fuzzy sets as FlowSets

Felix Offensperger *,† , **Markus Joppich** *,† , **Evi Sinn** , **Ralf Zimmer** *

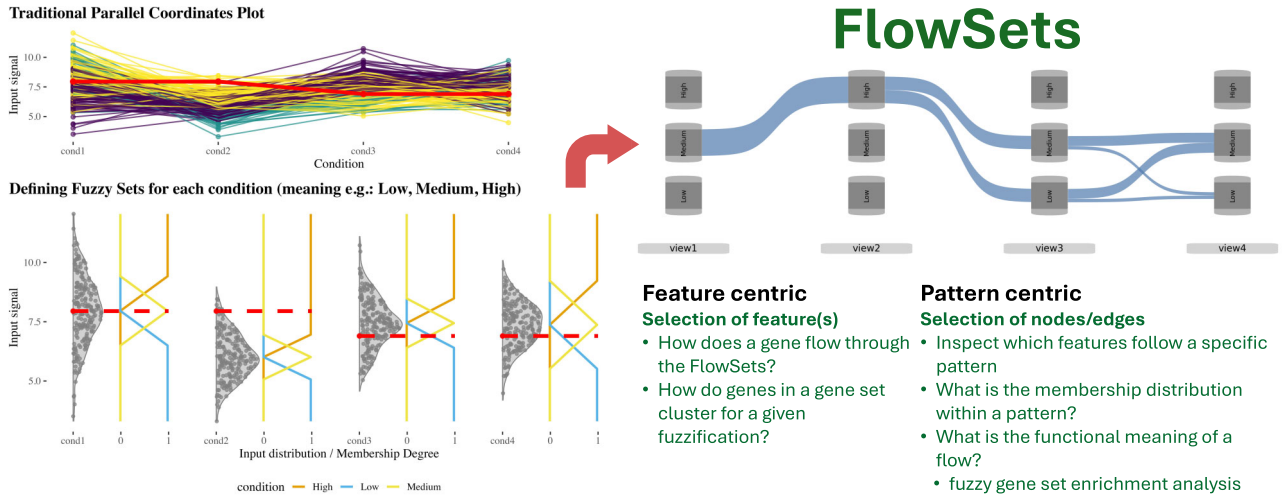
Institute of Bioinformatics, Department of Informatics, Ludwig-Maximilians-Universität München, Amalienstr. 17, 80333 München, Germany

*To whom correspondence should be addressed. Email: felix.offensperger@bio.ifi.lmu.de
Correspondence may also be addressed to Markus Joppich. Email:joppich@bio.ifi.lmu.de
Correspondence may also be addressed to Ralf Zimmer. Email:zimmer@ifi.lmu.de
†The first two authors should be regarded as Joint First Authors

Abstract

The growing availability of high-dimensional, complex datasets demands analysis methods that are both interpretable and flexible. We introduce FlowSets, a novel framework for identifying and analysing flows—fuzzy, interpretable patterns—across multiple, diverse, and heterogeneous data sets. Views are constructed as an interpretation of biological features of the data sets by grouping absolute or relative values, summary statistics (such as fold changes or *P*-values), or higher-order comparisons into fuzzy, linguistically defined categories based on their underlying distributions. FlowSets builds on these fuzzy categorizations and then tracks how features transition between categories across different conditions or data types, uncovering structural patterns that conventional methods often overlook. The FlowSets framework enables users to define, analyse, and manipulate complex patterns across heterogeneous datasets in a flexible and interpretable manner. With FlowSets, users can visualize feature flows, quantify pattern memberships, and perform enrichment analysis explicitly designed for sets with gradual memberships. This approach offers a robust and customizable alternative to rigid clustering or hard thresholding, allowing for a more transparent and insightful interpretation of multidimensional biological data.

Graphical abstract



Introduction

Genome-wide high-throughput technologies, such as proteomics, bulk and single-cell RNA sequencing, are increasingly accessible and widely adopted. The rapid growth in single-cell RNA-seq (scRNA-seq) studies, for instance, reflects both the scalability of current protocols and the growing interest in resolving expression dynamics across cell types, conditions, and time points. Simultaneously, proteomics and bulk RNA-seq remain standard tools, often across large numbers of replicates or experimental conditions.

This expansion of available techniques has enabled complex, multi-dimensional studies, with increasingly complex

Received: December 12, 2025. **Revised:** March 18, 2026. **Accepted:** April 2, 2026

setups spanning genes, transcripts, proteins, metabolites, and cell states, but it has also introduced new challenges for data interpretation.

Discovering features that behave similarly within omics data has become a central task in computational biology. Parallel coordinates plots and heatmaps provide a simple visual summary of behaviour that can be used alongside more advanced computational approaches [1]. Classical clustering algorithms, including k-means [2], hierarchical clustering [3], and fuzzy c-means [4, 5], have historically provided the foundation for pattern detection in single-omics datasets. While effective for broad structure discovery, these methods assume comparable signal distributions across features and are sensitive to normalization, distance metrics, and arbitrary parameter choices. As a result, features may share overarching cluster assignments despite exhibiting substantially different magnitudes or shapes. Methods tailored to omics-specific characteristics have attempted to alleviate these issues. For example, WGCNA [6, 7] constructs weighted gene co-expression networks to identify modules with coordinated behaviour, emphasizing correlation-based structure rather than absolute expression levels. Likewise, trajectory-aware approaches such as Tempora [8] leverage temporal labels and pathway information to infer dynamic transitions within time-series single-cell RNA-seq data, integrating both temporal and functional context into pattern discovery.

With the rise of multi-omics experiments, integrative pattern detection has become increasingly important [9, 10]. Component- and factor-based approaches such as DIABLO from the mixOmics suite [11] and Multi-Omics Factor Analysis [12] enable joint exploration of multiple molecular layers by identifying latent components that capture shared or complementary structure. These models allow users to extract cross-layer signatures, though their interpretability can be limited by model complexity and reliance on latent variable assumptions.

If patterns are first identified separately within each omics layer, they can subsequently be compared or combined across layers [13, 14]. Tools such as MiBiOmics [15] facilitate this by relating modules between omics types. Likewise, OmicsTIDE [16] enables pairwise trend comparison through clustered profile matching and linked visualizations. While effective for cross-layer comparison, these methods rely on discrete clustering, which may obscure subtle or heterogeneous signal patterns.

The purpose of all of these methods is to exploratively identify patterns. However, when one has a hypothesis about the experiment and wants to explicitly analyse a specific pattern, these methods do not provide the possibility to define and analyse such a pattern. Additionally, a directed analysis is hampered, as the methods are operating on summarized, complex, and non-interpretable metrics.

In our approach, the signals of a feature (gene, protein, etc.) are first categorized independently within each (omics) layer, producing interpretation views such as cell type cluster characterizations. These views are then each represented by fuzzy sets [17], in which features receive degrees of membership rather than binary assignments. This allows a feature to partially belong to multiple categories, providing a more nuanced and information-rich representation of its behaviour.

Features or linkages that appear across all views can subsequently be connected through their fuzzy memberships, forming the FlowSets diagram. This representation is inherently time- and order-agnostic, yet supports targeted exploration of specific genes or expression patterns. The FlowSets framework is designed for scalability and enables users to analyse large series datasets by leveraging the flexibility of fuzzy set theory.

A key advantage of fuzzy sets is their ability to represent gradual signal differences more naturally than discrete classification. They also allow expression values to be interpreted as linguistic classes, such as 'low,' 'medium,' or 'high,' improving interpretability. Additionally, FlowSets can be applied to differential analysis results, where fold changes serve as signals. This makes sequences of differential comparisons both comparable and comprehensible, extending the framework beyond raw expression data.

The FlowSets framework, incorporating the above-described concepts and visualizations, is available on GitHub (https://github.com/zimmerlab/FlowSets).

Materials and methods

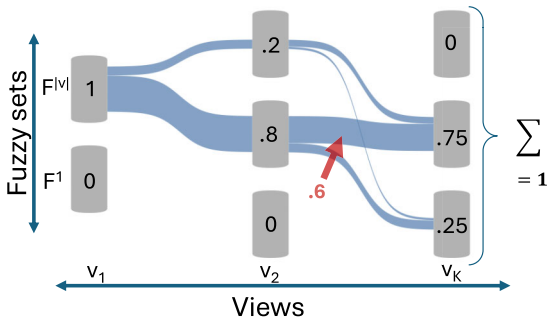
FlowSets is an alluvial-type analysis framework for omics data that is based on features with partial set membership. Signal values are transformed into linguistically defined categories based on their underlying distributions (e.g. from 'low' to 'medium' to 'high'). Features are assigned fuzzy set memberships, enabling them to be assigned to multiple classes rather than to strict, discrete categories. These per-layer fuzzy interpretations are referred to as *views*. Each view, such as cell-type clusters, conditions, or time points, offers a distinct perspective on the observed system by grouping measured or derived feature values into fuzzy sets. The transitions of these features across views are visualized as flows in FlowSets (Fig. 1a).

FlowSets provides two complementary representations. The first is a fuzzy extension of an alluvial diagram [18] (the *FlowSets diagram*). Because alluvial diagrams are defined visually rather than mathematically, the underlying structure is formalized as an ordered, multipartite, weighted graph (the *FlowSets graph*) whose edges encode joint fuzzy membership. A key principle of FlowSets is the focus on specific subsets within the FlowSets graph, referred to as *FlowSets patterns*.

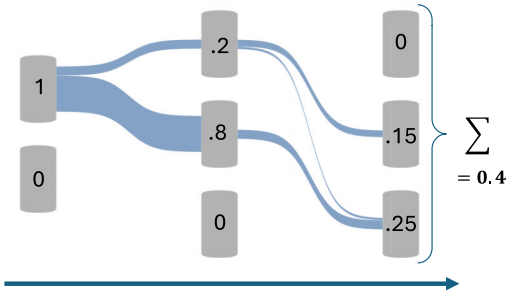
FlowSets input: fuzzy values

Fuzzy set theory [17] is well suited for discretization when the assignment to a category is unclear (e.g. for values close to a cutoff), because it allows partial membership across categories. Gene-expression measurements, for instance, are exact in form but uncertain in reality, so modelling them as a distribution rather than a single value provides a more accurate description of the underlying signal.

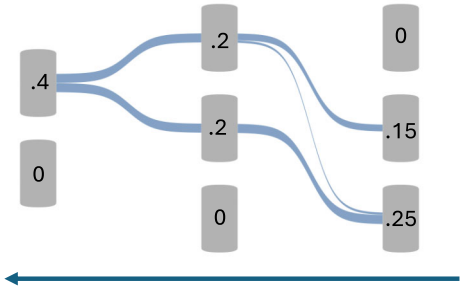
The continuous value of features (e.g. gene expression) is divided into user-defined fuzzy sets representing linguistic classes, according to the signal distribution of the feature and the membership functions. The meaning of a fuzzy set, its size, and membership function are problem-specific. FlowSets is based on fuzzy values as input; to allow continuous input, FlowSets offers a preliminary fuzzification procedure, where Gaussian, triangular, and crisp membership functions are available. The positions of the fuzzy sets can be specified either by providing their centres, by distributing them evenly from the minimum to the maximum values, or by using quantiles of the data distribution. But any other fuzzification approach is also incorporable, so that even more complicated approaches for deriving fuzzy values can serve as input [19].



(a) FlowSets diagram for a single feature



(b) DP step of a restricted FlowSets



(c) Backtracking step of a restricted FlowSets

Figure 1. (**a**) FlowSets for a single feature, including the naming conventions. At each view and fuzzy set, the membership of the observed feature is given in the grey box. There are four paths through the multi-partite graph for the feature. For the unrestricted FlowSets, the width of each edge shows the joint membership of two fuzzy sets. The edge highlighted by the red arrow has a joint membership of 0.6, represented by the width of its alluvium, based on memberships of 0.8 at the source node and 0.75 at the target node. (**b**) When restricting the flow of the observed feature to just a selection of edges (pattern: all decreasing edges in the last transition, selected edges are still shown), the sum of the membership of each feature is reduced to the membership in the remaining three paths. A dynamic programming algorithm is implemented, which provides the desired reduced membership for the pattern. However, in this example, the membership of the feature decreases in the last transition, meaning that the displayed memberships (and the subsequent visualization of the weights on the edges) may be misleading. (**c**) To address this, a backtracking step is introduced that propagates the pattern membership back to the initial view according to the weights of the remaining edges. The weights for a feature are then the same for each view and transition of the restricted FlowSets.

Every feature $f \in \mathcal{F}$ with signal x, and for every view $v_k \in V$ has certain membership values $F^i_{v_k}(x)$ which assigns feature f with a given membership value to the respective fuzzy set i in this view. A fuzzy concept is the set of all desired membership functions within a single view $FC_{v_k} = \{F^1_{v_k}, \ldots, F^{|v_k|}_{v_k}\}$. Note that the fuzzy concept does not have to be the same across all views, and even $|v_k|$ the number of fuzzy sets for view v_k can differ. The fuzzy values of x in FC_{v_k} in a view v_k is thus a vector of length $|v_k|$. The membership reflects the extent to which a feature is contained within a specific view. It typically is ensured that for every feature f the sum $\sum_{i=1}^{|v_k|} F^i_{v_k} = 1$.

FlowSets diagram: visual appearance

A FlowSets diagram generalizes classical alluvial diagrams by replacing crisp set assignments with fuzzy set memberships. Consider a sequence of views $v \in V$, each corresponding to a layer in the diagram. Every view v contains a finite collection of sets, each representing a linguistic variable that characterizes the distributional position of a feature (e.g. 'low,' 'medium,' 'high,' to represent, for example gene expression levels). Let $\mathcal{F}$ denote the set of genome-wide features.

Instead of assigning each feature $f \in \mathcal{F}$ to exactly one set within a view, FlowSets employs graded (fuzzy) membership functions. Thus, a feature may simultaneously belong to multiple linguistic sets within the same view, with degrees of membership in [0,1] that reflect its relative position in the underlying distribution.

A FlowSets diagram is defined as an ordered collection of K views, $v_1, \ldots, v_K$, displayed along the x-axis (Fig. 1a). For any two neighbouring views v_k and v_{k+1}, the alluvia linking sets across these views represent joint membership of features in the corresponding fuzzy sets. For a feature f, this joint membership is computed using a fuzzy 'AND' operator applied to its membership degrees in the two sets. The width of each alluvium is then given by the sum of these joint memberships over all $f \in \mathcal{F}$.

FlowSets graph: underlying data structure

The FlowSets graph representation consists of nodes, representing all fuzzy sets in all views $F^i_{v_k}$, and edges, connecting fuzzy sets of neighbouring views according to the chosen order of views, e.g. $(F^i_{v_k}, F^j_{v_{k+1}})$ for $0 < i \leq |v_k|$ and $0 < j \leq |v_{k+1}|$. These edges are representing the joint membership of features from a specific fuzzy set i (e.g. 'low') in a view v_k to a fuzzy set j (e.g. 'high') in the next view v_{k+1}. Given these properties, a weighted, ordered, and multipartite graph is constructed.

In order to calculate the joint membership for each feature, which is depicted as weight on the edge between two fuzzy sets, the respective memberships are multiplied (*edge weight*): $F^i_{v_k} \cdot F^j_{v_{k+1}}$. For a set of features, the weight of an edge is defined as the sum of all feature edge weights. In the traditional alluvial diagrams, a feature is contained with a membership of 1 in exactly one set of each view. Then, the membership of this feature in the edge from a crisp classification for a feature contributes with an edge weight of 1 to exactly one edge between subsequent views in the FlowSets graph. In the fuzzy case, this edge membership can be distributed over many or all edges from view v_k to view v_{k+1}.

Table 1. FlowFinder language for selecting pattern of interest

Symbol	Meaning
?	Any transition
=	Stays in the same class
x	Changes the class in any direction
∼	Remains approx. at same class (± 1)
>	Decreases class
≫	Decreases class > 1
<	Increases class
≪	Increases class > 1

Between each view, the transition can be specified using the symbols below. At each view the user can also specify minimum or maximum class per view.

Feature-centric analysis

The feature-centric analysis in FlowSets allows to focus on a subset of features $\mathcal{F}' \subset \mathcal{F}$ in the entire FlowSets representation. Subsetting features does not change the membership of the other features, because they are independent of each other, but the summarized weights differ. Individual features or groups of features can be compared to the weights of the entire dataset, and also be highlighted in comparison to the background in the visualization. Complex regulatory patterns across views of subsets of features can become apparent and further analysed.

Pattern-centric analysis

A pattern-centric analysis focuses on analysing how features behave across a selected subset of edges and nodes that define a specific pattern. Such a selection of specific fuzzy sets or edges over all views is also called a *restricted FlowSets graph* (Fig. 1b).

We define a path within the FlowSets graph as an ordered list of nodes, which traverses all views exactly once $F^*_{v_1} \rightarrow \ldots \rightarrow F^*_{v_K}$, where $*$ is a selected node. From the definition of edges it follows that within each view exactly one fuzzy set is covered. Paths are observations of specific fuzzy sets or changes across different views. The membership of a feature within a path expresses the membership with which the feature is contained in all fuzzy sets of the path. A closed formula for the *path weight* can be derived as:

$$pw(f) = \prod_{k=1}^{K} F^*_{v_k}. \tag{1}$$

In the unrestricted FlowSets (the unmodified alluvial diagram applied independently to each feature), the fuzzy memberships of all possible paths for a given feature sum to 1.

A *FlowSets pattern* is defined as a set of paths, leading to a restricted FlowSets. Here, the memberships for a feature do not sum up to one, but rather to a membership of every feature to the pattern.

Patterns are specified using the FlowFinder pattern language (Table 1) to support user-friendly specification, FlowFinder identifies nodes and edges that satisfy the defined pattern criteria. These criteria may describe transitions between views, requiring paths to remain within the same fuzzy set, or, as fuzzy sets often express relative positions, directional changes used to analyse trends. Node subsetting can be expressed by selecting the minimum or maximum linguistic class within individual views.

By constraining both nodes in each view and the edges between them, users can define complex FlowSets patterns.

The membership of a feature for a FlowSets pattern is simply the sum over the membership in all paths of the pattern (*pattern weight*).

$$Pw(f) = \sum_{i \in paths} pw_i(f) \tag{2}$$

and the pattern weight for a set of features $\mathcal{F}'$ is:

$$Pw(\mathcal{F}') = \sum_{f \in \mathcal{F}'} Pw(f). \tag{3}$$

Complexity of pattern membership calculation

In traditional alluvial diagrams, subsetting is a straightforward task based on simple Boolean logic (included or excluded). However, in FlowSets, features are partially reduced based on their fuzzy memberships.

The computational complexity of the subsetting problem depends critically on whether the restriction is applied only to nodes v_k leading to v'_k, or also to edges between the remaining nodes in v'_k. When subsetting the nodes of a FlowSets only (leading to the induced subgraph), the distributive property allows the pattern weight of a feature

$$Pw(f) = \prod_{k=1}^{K} \sum_{i=1}^{|v'_k|} F^i_{v'_k}, \tag{4}$$

can be computed in linear time, which is of the order $\mathcal{O}(K * n)$, where n is the maximum number of fuzzy sets per view. In contrast, edge subsetting, and thereby supporting the full range of FlowSets patterns, induces a considerably higher level of computational complexity. The most direct approach requires computing the memberships of a feature across every possible path within the restricted FlowSets. However, because the number of such paths can be exponential within the number of views K, this approach quickly becomes computationally infeasible $\mathcal{O}(n^K)$, particularly for datasets with many views.

Fortunately, it is not necessary to explicitly enumerate all individual paths. Since only the sum of the memberships of all selected paths is relevant for determining the membership for each feature in a pattern, the individual paths themselves do not need to be explicitly computed. Making use of the prefix property for pattern weights in FlowSets, a dynamic programming algorithm provides the same reduced memberships while reducing the complexity from exponential to quadratic, approximately in $\mathcal{O}\left(\sum_{k=1}^{K-1} n_{v'_k} * n_{v'_{k+1}}\right) \leq \mathcal{O}(K * n^2)$.

Calculating the membership for a pattern

Starting from the first view, the membership of the remaining fuzzy sets v'_k are considered and all remaining edge memberships are computed in a dynamic programming fashion (Fig. 1b). The prefix scores $D^i_{v'_k}$ for the memberships for all prefix paths up to view v'_k ending in $F^i_{v'_k}$ are calculated. For each selected edge, the membership can flow proportionally to the next view from one fuzzy set $F^i_{v'_k}$ to another fuzzy set $F^j_{v'_{k+1}}$ in the next view, by a factor of the membership in the next view. For each node in the next view, the edge memberships for all connected nodes I is summed up to obtain a prefix score that serves as new starting membership for the next step.

$$D^j_{v'_{k+1}} = \sum_{i \in I} D^i_{v'_k} * F^j_{v'_{k+1}}. \tag{5}$$

This constrains the overall membership of the feature in all classes of the view to be at most equal to its value in the previous view, but it is further reduced if edges or fuzzy sets with membership of this feature are not selected.

The steps are consecutively performed for all views. At the end of this iterative process the memberships for all fuzzy sets in the final view are calculated. Here, these memberships are summed up for each feature, which then equals the total (remaining) membership for each feature within the selected pattern.

It is then known how much of the membership can still be passed through the graph. However, the weights of the flow may be misleading, as the incoming weight is not necessarily the same as the outgoing weight for every node (e.g. due to deselected edges and dead ends). This can optionally be adjusted for every feature with a backtracking step (shown in Fig. 1c), so that the cut of each bipartite graph equals the pattern membership.

The memberships in the fuzzy sets of the last view are backtracked to the first view, for which a corresponding backtracking matrix B is constructed. By iterating through all views in reverse order, the memberships are distributed proportionally to the edges used in the dynamic programming phase. The final memberships for each node are computed as

$$B^j_{v'_{k-1}} = \sum_{i \in I} \frac{D^j_{v'_{k-1}} * B^i_{v'_k}}{\sum_{l \in L} D^l_{v'_{k-1}}}, \tag{6}$$

where I are all nodes in view v'_k connected to $F^j_{v'_{k-1}}$ and L are all nodes in view in v'_{k-1} connected to $F^i_{v'_k}$. These memberships can then be summed for each edge and node, providing the true weight of the edges for the selected pattern.

Once the membership of each feature for a pattern in the system is known, various downstream analyses can be performed, such as identifying the features with the highest membership in the given pattern or performing pathway over-representation analysis as described below.

Enrichment of resulting flows

To assess the membership of pathways or gene sets within a pattern, two approaches can be used. First, the memberships of features within the pattern can be thresholded, enabling the application of traditional gene set enrichment analysis. Alternatively, we propose a fuzzy gene set over-representation analysis, which directly incorporates the membership of each feature into the enrichment process. Similar to the traditional approach, gene-set over-representation analysis here is viewed as an over-representation of a particular set of features within the data. Thus, for each gene set $\mathcal{F}'$, we sum up the membership of the genes from the gene set within a given pattern $Pw(\mathcal{F}')$. We then calculate the gene set coverage (the fraction of the summed-up memberships) by dividing the number of measured genes from the gene set. Typically the distribution of gene set coverage is zero-inflated (many gene sets are not covered) and their coverage is highly dependent on the gene set size [20]. Therefore, using a simple standard approach, all gene sets were assigned to bins based on the number of genes they contain. By default, the bin boundaries are 2, 10, 50, and 100. Within these bins, all non-zero gene set coverages are z-scaled, and for the positive z-values (over-representation), P-values are then derived from the survival function of the underlying normal distribution. The resulting P-values for all gene sets are multiple testing corrected (Benjamini–Hochberg). As input for gene sets, any list definition of features can be used (gmt format), such as Reactome pathways [21], Gene Ontology [22], or MSigDB pathways [23], while custom gene sets can also be added.

Use-case data

The data for the use-case study was obtained from a SARS-CoV-2-related study in which pneumonic (symptomatic) and non-pneumonic (non-symptomatic) patients were characterized using scRNA-seq analysis [24]. In this use case, we focus only on pneumonic versus non-pneumonic samples and therefore exclude the control group from the analysis, which considers three time points of infection. All analyses shown here were performed on the monocyte subset of the full dataset.

Gene set over-representation was performed on Gene Ontology [22] biological processes with five or more associated genes per gene set.

Results

With the FlowSets framework, it is possible to analyse complex data sets using fuzzy set theory for discrete measurement data. The framework is agnostic to the used input data and can even combine multiple input data types (e.g. transcriptomic measurements and proteomic ones) by abstracting measured signals into linguistic classes (fuzzy sets).

To highlight the applicability of FlowSets, we demonstrate the analysis of biomedical data. The use-case dataset consists of human peripheral blood mononuclear cells (PBMCs) of patients with pneumonic (symptomatic) and non-pneumonic (non-symptomatic) COVID-19, analysed at three distinct timepoints [24]. This dataset is thus well suited for a FlowSets analysis because it allows for (i) an analysis of gene expression data along the distinct time points for each disease state and (ii) a double-differential comparison along the time point series and between the two disease states. In this study we focus, like in the original publication, on monocytes. Hence, this use case operates only on (differential) gene expression data from the monocyte cluster.

The fuzzy nature of FlowSets applies exceptionally well to zero-inflated distributions like scRNA-seq data because linguistic classes can be used to differentiate between cells expressing and cells not expressing a gene within one cluster. Additional fuzzy sets can be used to describe the distribution of gene expression values and thus not only enable to see changes in the intensity of gene expression, but also in the fraction of expressing cells, offering an alternative method for managing the inherent complexity of gene expression patterns in scRNA-seq datasets.

FlowSets provides instructions for creating gene expression output directly from Seurat [25] or scanpy [26] objects (see GitHub repository), resulting in a file with one mean expression value per gene, and corresponding standard deviation and percent expressing cells. Given this additional information about the distribution of the gene expression values, the fuzzification can be performed by sampling from such defined distribution (here: 1000 samples). These are then fuzzified and averaged for each sample, and weighted by the fraction of cells expressing the particular gene. This way, scRNA-seq gene expression is accurately represented in a fuzzified form. The summarized expression is read in by FlowSets, and for each gene,

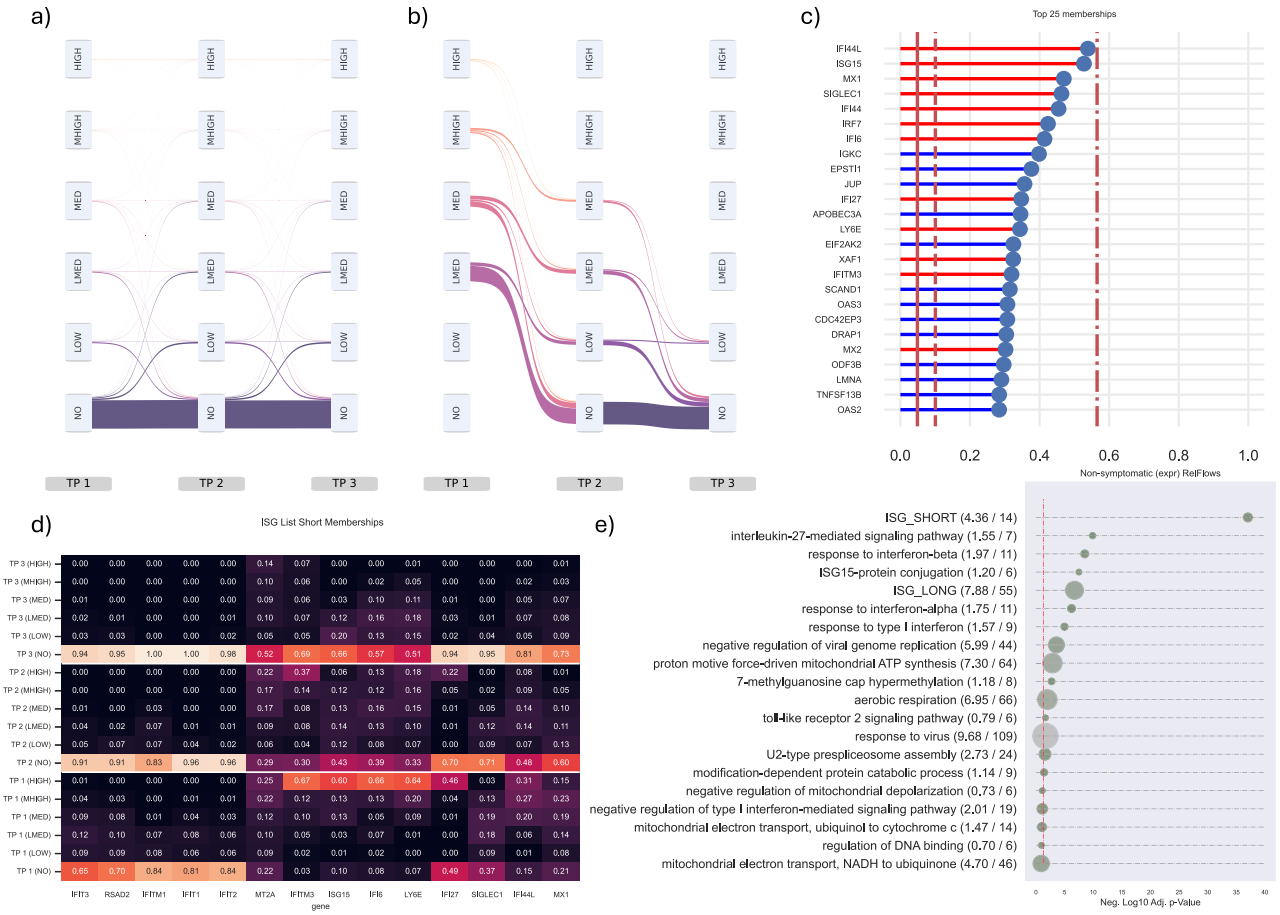


Figure 2. (**a**) FlowSets illustrates the flow of 25 767 genes in the non-symptomatic monocytes of the scRNA-seq dataset. The signal intensity of the starting set determines the colour of the edges. (**b**) A FlowSets pattern was applied to highlight a subset of flows in the FlowSets plot that indicate decreasing gene activity across time points 1 to 3, as identified by the FlowFinder routine. (**c**) Gene memberships for the pattern in the non-symptomatic sample. Highlighted in red are the members of the long ISG signature. Mean and standard deviation are calculated for all non-zero memberships, with red lines indicating the mean, a *z*-score of 1, and a *z*-score of 10. (**d**) Membership of all ISG signature features across all views and fuzzy sets, aggregated along the *y*-axis. (**e**) Results of gene set over-representation analysis on combined target flows for monocytes from non-symptomatic patients, filtered for size (>5 genes) and significance (adjusted *P*-value <.05). In brackets first the fraction of the summed-up memberships and then the gene set size are shown, which also is visualized as size of the dots. The red line indicates a cut-off of 0.05.

at each view, the expression is transformed into a membership for each designated fuzzy set of the fuzzification. In this analysis, we choose six fuzzy sets (`'NO'`, `'LOW'`, `'LOWMED'`, `'MED'`, `'MEDHIGH'` and `'HIGH'`) as triangular membership functions with their peak/centre at log-UMI 0.4, 0.8, 1.2, 1.6, 2, and 2.4.

When analysing the data, we first want to get an overview of gene expression patterns over the disease states of the monocyte cluster (Fig. 2a). In the complete FlowSets diagram, it can be seen that a majority of genes are not expressed and do not change over the series and remain in the `'NO'` set. Only a few genes are in the medium or high expression range, which is depicted by only thin lines. Nonetheless, we are interested in specific expression patterns that are indicative of the studied SARS-CoV-2 infection. These are genes, which are initially highly expressed in order to tackle the infection and then return to a lower expression set (Fig. 2b). Flows following such a pattern can be selected using the FlowFinder routine (sequence=[‘>’, ‘≥’], minLevels=[‘LOWMED’, None, None], maxLevels=[‘HIGH’, ‘MED’, ‘LOW’]).

For the selected pattern, it is possible to identify the genes with the highest membership scores within these flows (Fig. 2c). From the original publication it is known that the interferon-stimulating genes play a major role in early stages of SARS-CoV-2 infection and signatures were derived (ISG-signature short and long). Unsurprisingly, many genes from these ISG signatures are among the top genes representing the pattern. On the other hand, it also becomes apparent that the membership within the pattern drops fast to below 0.25, indicating that these genes have the majority of their membership in other flows. With the FlowSets framework it is possible to have a closer look at the memberships of specific genes over all views and fuzzy sets (Fig. 2d). For instance, the genes IFI27, IFI44L, IFI6, IFITM3, ISG15, and LY6E show a high expression level at TP1 and then a decreasing expression at views TP2 and TP3. With this functionality it is possible to differentiate the behaviour of specific genes.

The FlowSets framework provides means to analyse the memberships for the full FlowSets or a selected pattern also on a pathway or gene set basis. For each pattern, it is checked whether a specific gene set (e.g. from a collection like Reactome [21] or Gene Ontology [22]) is over-represented. Here, we performed such analysis on the pattern using Gene Ontology biological process gene sets and

the two ISG gene signatures from the original publication [24]. Gene set over-representation results were filtered for significance (adjusted *P*-value <.05) and minimum gene set size (>5 genes), as very small pathways were not considered of interest, and visualized in Fig. 2e. It is reassuring that the ISG signatures are found significantly enriched in the pattern. Moreover, compared to the symptomatic FlowSets, we can see that the short ISG gene set is better covered (pathway coverage $Pw_{non\_sympt}(ISG\_SHORT) = 0.31$ versus $Pw_{sympt}(ISG\_SHORT) = 0.26$). Interestingly, several mitochondrial processes appear among the top 20 over-represented gene sets in the non-symptomatic data, suggesting that their upregulation may contribute to coping with the disease, consistent with previous reports [27].

Next, we are interested in identifying key differences between symptomatic and non-symptomatic patients. Again, we concentrate on the monocytes within the dataset because these are known to be key indicators for disease outcome [28, 29]. A differential comparison between the two disease-state groups was therefore performed for each view using Seurat's FindMarkers function. The thereby returned single differential expression results were concatenated into a discrete series, and the $\log_2$ fold changes are used as input for FlowSets. The fuzzy sets are defined such that a gene is clearly up-regulated [abs(fc) of 1] in one group (`SYMPT` or `ASYMPT`), most abundant [abs(fc) of 0.5] in one group (`sympt` or `asympt`) or `nodiff` if either the fold change is too small, or there was no significant fold change. Given the centres of the fuzzy concepts, their widths are automatically derived by FlowSets. Again, we first checked the overall patterns of the 314 Seurat-identified genes (Fig. 3a). A larger set of genes, which is more prevalent in `sympt` at TP1 can be noted, while no such flow is visible in `asympt`. This warrants a closer look at flows that have a disease-associated pattern (e.g. changing from one extreme at TP1 to the other, or no regulation, at TP3; Fig. 3c and d). For both restricted FlowSets, gene set enrichment on the pattern membership was performed. Again, results were filtered for significance, and gene set size and finally sorted by adjusted *P*-value (Fig. 3e and f).

In the symptomatic cases (Fig. 3e), it is interesting to see an up-regulation of apoptotic process as well as a viral entry into host cell (Fig. 3g). Albeit not significant anymore, among the top 20 gene sets in the symptomatic case, many apoptosis- and cell death-related gene sets are identified. Again, this is expected, as programmed cell death is a hallmark of severe COVID-19 infection [30]. Likewise, there are many gene sets associated with T-cell activation found for genes up-regulated in the non-symptomatic case (Fig. 3h). It is also possible to confirm the finding that ISG-related gene signatures are over-represented in non-symptomatic patients.

These results show that FlowSets is able to easily extract relevant processes and activities from gene expression or differential gene expression data. It was possible to recapitulate the results from the original publication and to identify key processes like cell cycle activity only uncovered recently.

An illustrative comparison with unsupervised clustering is shown in Fig. 3b. For the differential expression results generated with Seurat, clustering fold change profiles is more straightforward than clustering expression values directly, making comparison with clustering-based approaches more appropriate. Accordingly, the same fold change values used for FlowSets were imported, non-significant fold changes were set to zero, and the genes were clustered by k-means across a range of cluster numbers. The resulting clusters were then tested for over-representation against the same GO annotation using ORA as implemented in the Python package *gseapy* [31]. Across $k = 3$ to $k = 9$, k-means recovers cluster profiles that are very similar to each other. For direct comparison, $k = 4$ was selected; in this setting, clusters 0 and 1 most closely matched the two FlowSets patterns of interest. For cluster 0, gene set enrichment identified only a strict subset of the significant pathways detected with the FlowSets-based gene set analysis, whereas cluster 1 yielded no significantly enriched terms. This is most likely due to the clustering itself, as some of the genes in the cluster have a quite different trajectory than the cluster profile and thus also do not follow the FlowSets pattern. Furthermore, the ISG-related signature that was important according to the original publication was not recovered by the k-means approach. In this example, FlowSets, therefore, provided a more sensitive enrichment analysis. Overall, these results indicate that targeted rule-based filtering with FlowSets provides a straightforward and effective strategy for addressing focused research questions.

Discussion

Here, we present the FlowSets framework, which allows fuzzy set-based analysis of (discrete) series data on any kind of signal, including expression and differential values. While we are showcasing the use on scRNA-seq data, the framework can be applied to a whole range of data, from RNA-seq, over other omics data and even spatial technologies.

FlowSets is an explicit, rule-based method for defining patterns of expression. Visual methods, like parallel coordinate plots, often only visualize input signals for patterns or clusters and require an analytical method to identify them. FlowSets can be used for assigning features into soft clusters and could be integrated in a parallel coordinate plot with pattern memberships visualized as colour or transparency. Unlike all traditional approaches that use linear modelling on (pseudo-)time or unsupervised clustering (e.g. WGCNA), FlowSets enables direct selection and interpretation of user-defined or biologically guided expression patterns. Because the methodology is qualitatively different and more flexible by defining explicit patterns on custom views, a direct comparison to tools like WGCNA is not fully appropriate.

In our showcase, we had a consistent fuzzy set structure across multiple measurement data. This consistency enables straightforward comparison across dimensions in multidimensional datasets, making it easier to identify systematic patterns and dynamic changes. But it would also be possible to use multiple views on a single measurement. As biological features (e.g. genes or proteins) are often described using multiple metrics, such as statistical significance and effect size. These metrics can sometimes lead to contradictory conclusions. By combining or visualizing them together a more realistic and informative view of the data can be achieved.

An additional layer of information can be obtained by integrating different omics technologies (e.g. transcriptomics, proteomics, and epigenomics), each of which captures a distinct view. When related entities are present in the different omics layers, a clear linkage can be used as a feature (e.g. transcript–protein relation). In more complicated scenarios involving 1:m mappings between entities (e.g. gene–transcript relation), either every combination can be used as a feature, or the features of one of the layers can be summarized to

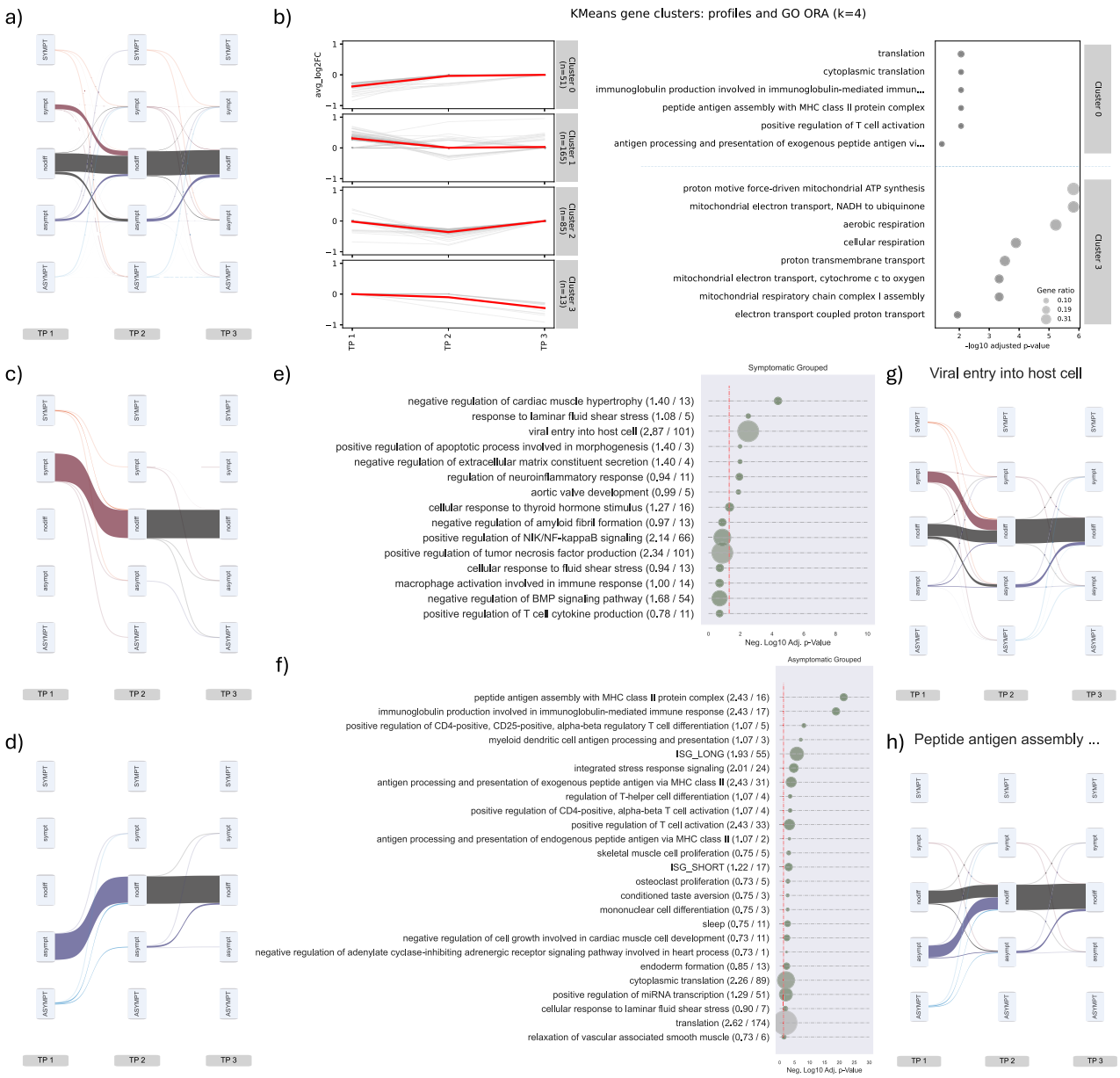


Figure 3. (**a**) FlowSets depicting fold changes in gene expression between symptomatic and non-symptomatic patients' monocytes ($n = 314$ genes). The edges are coloured according to the direction of the fold change of the starting set. (**b**) Shows a conventional clustering-based analysis of fold change profiles using k-means. For $k = 4$, the resulting cluster-specific expression profiles are displayed. On the right, over-representation analysis (ORA) was performed using the same GO Biological Process gene sets, and only significantly enriched terms were retained (adjusted *P*-value <.05). Similar patterns were selected in the restricted FlowSets indicative of (**c**) symptomatic or (**d**) non-symptomatic disease progression, selected based on decreasing expression classes across all views of the series. (**e**+**f**) FlowSets enrichment analysis on pattern membership for (e) symptomatic and (g) non-symptomatic cases using Gene Ontology (biological process) gene sets. (**g**+**h**) Shows the flow for one selected gene set for each pattern.

resolve the linkage. Creating an integrated pattern that brings these layers together enables biological consistency to be assessed and multi-level regulatory effects to be identified. Linking biological entities can also be used to analyse more complex relationships, such as gene regulation or protein–protein interactions, which depend on multiple features and types of evidence. For instance, regulatory edges can be used as a feature, described by the expression levels of the regulator and target, and a computed regulation score. Combining all relevant views into a single composite pattern via FlowSets enables more accurate modelling and provides greater insight into complex biological systems.

With the analyses performed, we are able to efficiently reproduce known facts about COVID-19 and reproduce findings from the selected use-case publication. An essential part of the analysis was querying disease-related processes based on condition-specific expression patterns. Given a new dataset and scientific question, FlowSets provides a framework for the discovery of relevant gene expression patterns, and identification of key gene sets and active biological processes. This facilitates both a broad overview of relevant mechanisms and the targeted investigation of pathways that exhibit consistent patterns across multiple views. It is also easy to perform a targeted analysis by highlighting genes or gene sets within the

flows between views or fuzzy sets. In contrast to explorative methods, FlowSets allows both a feature-centric and a pattern-centric analysis of the data: it can be performed on any target gene or pattern, regardless of whether it has been identified as specifically interesting by the method.

Conclusion

In summary, the FlowSets framework enables easy and comprehensible analysis of complex multi-dimensional data. FlowSets can be applied to any type of expression data, in particular to the results of differential expression analysis. The user can select expression patterns of interest, and FlowSets is able to identify even subtle differences in expression between user-defined views. Views are represented as well-defined fuzzy concepts enabling the definition and interpretation of patterns. At all stages of the analysis, the interpretation of the data is supported by meaningful visualizations. Thus, FlowSets is a versatile tool for analysis of many different modalities, even beyond the gene expression community.

Acknowledgements

Author contributions: Felix Offensperger: Conceptualization, Methodology, Software, Data Curation, Visualization, Investigation, Writing—Original Draft; Markus Joppich: Conceptualization, Methodology, Software, Visualization, Investigation, Writing—Original Draft, Supervision; Evi Sinn: Writing—Review & Editing; Ralf Zimmer: Conceptualization, Methodology, Writing—Review & Editing, Supervision, Project administration, Funding acquisition.

Conflict of interest

None declared.

Funding

Deutsche Forschungsgemeinschaft (grant/award number: 'SFB1123').

Data availability

The FlowSets framework is available on GitHub (https://github.com/zimmerlab/FlowSets), along with notebooks containing the code to reproduce the figures presented in this paper. All shown analyses have been conducted in Python 3.12 with Polars version 1.19.0. Data used are publicly available from Pekayvaz *et al*. [24].

Submitted versions are also accessible via FigShare: https://figshare.com/projects/FlowSets/263524.

References

1. Rutter L, Lauter ANM, Graham MA *et al.* Visualization methods for differential expression analysis. *BMC Bioinformatics* 2019;20:458. https://doi.org/10.1186/s12859-019-2968-1
2. MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. Berkeley, CA, USA: University of California Press, 1967, 281–97. Project Euclid, bsmsp/1200512992
3. Murtagh F, Legendre P. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *J Classif* 2014;31:274–95. https://doi.org/10.1007/s00357-014-9161-z
4. Bezdek JC. *Pattern recognition with fuzzy objective function algorithms*. New York, NY: Springer, 1981. https://doi.org/10.1007/978-1-4757-0450-1
5. Loubaton R, Champagnat N, Vallois P *et al.* MultiRNAflow: integrated analysis of temporal RNA-seq data with multiple biological conditions. *Bioinformatics* 2024;40:btae315. https://doi.org/10.1093/bioinformatics/btae315
6. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* 2008;9:559. https://doi.org/10.1186/1471-2105-9-559
7. Group MS, Analysts L. Temporal dynamics of the multi–omic response to endurance exercise training. *Nature* 2024;629:174–83. https://doi.org/10.1038/s41586-023-06877-w
8. Tran TN, Bader GD. Tempora: cell trajectory inference using time-series single-cell RNA sequencing data. *PLoS Comput Biol* 2020;16:e1008205. https://doi.org/10.1371/JOURNAL.PCBI.1008205
9. Subramanian I, Verma S, Kumar S *et al.* Multi-omics data integration, interpretation, and its application. *Bioinform Biol Insights* 2020;14:1177932219899051. https://doi.org/10.1177/1177932219899051
10. Li Y, Wu FX, Ngom A. A review on machine learning principles for multi-view biological data integration. *Brief Bioinform* 2016;19:325–40. https://doi.org/10.1093/bib/bbw113
11. Rohart F, Gautier B, Singh A *et al.* mixOmics: an R package for 'omics feature selection and multiple data integration. *PLoS Comput Biol* 2017;13:e1005752. https://doi.org/10.1371/journal.pcbi.1005752
12. Argelaguet R, Velten B, Arnol D *et al.* Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Mol Syst Biol* 2018;14:e8124. https://doi.org/10.15252/msb.20178124
13. Rappoport N, Shamir R. Multi–omic and multi–view clustering algorithms: review and cancer benchmark. *Nucleic Acids Res* 2018;46:10546–62. https://doi.org/10.1093/nar/gky1226
14. Hoogendijk A, Pourfarzad F, Aarts CEM *et al.* Dynamic transcriptome–proteome correlation networks reveal human myeloid differentiation and neutrophil–specific programming. *Cell Rep* 2019;29:2505–19. https://doi.org/10.1016/j.celrep.2019.10.082
15. J Z, JF G, M N *et al.* MiBiOmics: an interactive web application for multi-omics data exploration and integration. *BMC Bioinformatics* 2021;22:6. https://doi.org/10.1186/s12859-020-03921-8
16. Harbig TA, Fratte J, Krone M *et al.* OmicsTIDE: Interactive exploration of trends in multi-omics data. *Bioinform Adv* 2023;3:vbac093. https://doi.org/10.1093/bioadv/vbac093
17. Zadeh LA. Fuzzy sets. *InformControl* 1965;8:338–53. https://doi.org/10.1016/S0019-9958(65)90241-X
18. Brunson JC. ggalluvial: layered grammar for alluvial plots. *J Open Source Softw* 2020;5:2017. https://doi.org/10.21105/joss.02017
19. Consiglio A, Mencar C, Grillo G *et al.* Fuzzy method for RNA-seq differential expression analysis in the presence of multireads. *BMC Bioinformatics* 2016;17:345. https://doi.org/10.1186/s12859-016-1195-2
20. Rahmatallah Y, Emmert-Streib F, Glazko G. Gene set analysis approaches for RNA-seq data: performance evaluation and application guideline. *Brief Bioinform* 2016;17:393–407.https://doi.org/10.1093/bib/bbv069
21. Jassal B, Matthews L, Viteri G *et al.* The reactome pathway knowledgebase. *Nucleic Acids Res* 2020;48:D498–503. https://doi.org/10.1093/nar/gkz1031
22. Ashburner M, Ball CA, Blake JA *et al.* Gene Ontology: tool for the unification of biology. *Nat Genet* 2000;25:25–9. https://doi.org/10.1038/75556

23. Subramanian A, Tamayo P, Mootha VK *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA* 2005;102:15545–50. https://doi.org/10.1073/pnas.0506580102
24. Pekayvaz K, Leunig A, Kaiser R *et al.* Protective immune trajectories in early viral containment of non-pneumonic SARS-CoV-2 infection. *Nat Commun* 2022;13:1018. https://doi.org/10.1038/s41467-022-28508-0
25. Butler A, Hoffman P, Smibert P *et al.* Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;36:411–20. https://doi.org/10.1038/nbt.4096
26. Wolf FA, Angerer P, Theis FJ. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol* 2018;19:15. https://doi.org/10.1186/s13059-017-1382-0
27. Srinivasan K, Pandey AK, Livingston A *et al.* Roles of host mitochondria in the development of COVID-19 pathology: Could mitochondria be a potential therapeutic target? *Mol Biomed* 2021;2:38. https://doi.org/10.1186/s43556-021-00060-1
28. Merad M, Martin JC. Pathological inflammation in patients with COVID-19: a key role for monocytes and macrophages. *Nat Rev Immunol* 2020;20:355–62. https://doi.org/10.1038/s41577-020-0331-4
29. Schulte-Schrepping J, Reusch N, Paclik D *et al.* Severe COVID-19 is marked by a dysregulated myeloid cell compartment. *Cell* 2020;182:1419–40. https://doi.org/10.1016/j.cell.2020.08.001
30. Bader SM, Cooney JP, Pellegrini M *et al.* Programmed cell death: the pathways to severe COVID-19? *Biochem J* 2022;479:609–28. https://doi.org/10.1042/BCJ20210602
31. Fang Z, Liu X, Peltz G. GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. *Bioinformatics* 2023;39:btac757. https://doi.org/10.1093/bioinformatics/btac757

Received: December 12, 2025. **Revised:** March 18, 2026. **Accepted:** April 2, 2026