scientific reports

Check for updates

OPEN

CytoAnalyst web platform facilitates comprehensive single cell RNA sequencing analysis

Phi Bya[1], Duy Tran[1], Khoi Nguyen[1], Sorin Draghici[2,3] & Tin Nguyen[1]✉

Single-cell technologies have revolutionized our ability to study cellular heterogeneity and dynamics at unprecedented resolutions. In this fast-growing field, it becomes increasingly challenging to navigate the vast number of tools and steps for analysis. It is particularly difficult to integrate and analyze large datasets that require extensive collaborations and customized pipelines to obtain robust results. We present CytoAnalyst, a web-based platform that offers a number of important advantages over existing tools for single-cell analysis. First, the platform enables custom pipeline configuration using an efficient study management system and a broad range of analysis modules. Second, the platform supports parallel analysis instances, facilitating the comprehensive comparison of different methods or parameter settings available at each analysis step. Third, the advanced sharing system facilitates real-time synchronization among team members and seamless analysis continuation across different devices. Finally, the grid-layout visualization system supports simultaneous displays of different data aspects, allowing for the comparison of multiple labels and plots side-by-side for comprehensive data insights, with the ability to save and reload visualization settings at any analysis step. The platform incorporates multiple blending modes, allowing users to combine plots in various ways for comprehensive data exploration. CytoAnalyst supports a high level of analytical rigor while providing user-friendly and flexible operations through its carefully designed interface and extensive documentation. The platform supports all major web browsers and is freely available at https://cytoanalyst.tinnguyen-lab.com.

Keywords Single-cell analysis, Interactive visualization, Embedding analysis, Clustering analysis, Cell type annotation, AI-guided cell annotation

Background

Single-cell RNA sequencing (scRNA-Seq) has emerged as a transformative technology in biomedical research, enabling unprecedented insights into cellular heterogeneity, rare cell populations, and dynamic biological processes[1]. Single-cell analysis often follows a complex workflow that involves multiple steps such as quality control, normalization, feature selection, dimensionality reduction, clustering, differential expression (DE) analysis, cell annotation, and trajectory inference. While numerous methods and tools are available for each analysis step, integrating them into a cohesive workflow remains a challenge[2].

Current single-cell analysis tools are available as command-line packages and/or web-based platforms. Command-line packages offer extensive analytical capabilities but require coding and bioinformatics skills[3–6]. Web-based platforms make single-cell analysis more accessible to life scientists by providing an intuitive graphical user interface. These tools can be installed on local computers[7–18], or are available as public servers[19–22]. The analysis capabilities of these platforms vary substantially. Some platforms focus solely on data visualization (e.g., iSEE, ShinyCell, SCope, UCSC Cell Browser) or cluster and DE analysis (Loupe Browser, CellxGene, SPRING). Others offer more comprehensive workflows, from data pre-processing to clustering, DE analysis, and other analytical steps (Cellenics, Asc-Seurat, SCTK2, ASAP, Cellar, ezSingleCell, ICARUS, SingleCAnalyzer, Granatum). Despite the extensive functionalities provided by these platforms, important challenges persist. It is particularly challenging to integrate and analyze large datasets that require coordinated efforts/collaborations and customized analysis workflows to obtain robust results and valid conclusions.

[1]Department of Computer Science and Software Engineering, Auburn University, Auburn 36849, AL, USA. [2]Department of Computer Science, Wayne State University, Detroit 48202, MI, USA. [3]AdvaitaBio, Ann Arbor 48105, USA. ✉email: tinn@auburn.edu

Here, we introduce CytoAnalyst, a new web-based platform that advances workflow flexibility, parallel processing, and team collaboration in single-cell data analysis. We systematically compare CytoAnalyst's features with available single-cell analysis tools (Supplementary Section 1 and Supplementary Figure S1). While some platforms offer broader data modality support or specialized analytical modules, CytoAnalyst's strength lies in its integrated approach to workflow flexibility, parallel processing, and team collaboration.

CytoAnalyst offers a number of advantages over existing tools. First, instead of imposing a rigid workflow, CytoAnalyst enables custom pipeline configuration using an efficient study management system and seven analysis modules (embedding, clustering, DE analysis, gene set management, enrichment, annotation, and trajectory inference). Researchers can navigate between analysis modules and start/resume any analysis steps without losing results or visualization configuration. Second, the platform supports parallel analysis instances, facilitating the comparison of methods or parameter settings available at each step. Third, the advanced sharing system facilitates real-time interaction among team members and seamless analysis continuation across different devices. Access permissions for a shared study can be controlled at granular levels, allowing collaborators to view or alter parameter settings and analysis pipelines. Finally, the grid-layout visualization system supports simultaneous displays of different data aspects, allowing for the comparison of multiple labels and plots side-by-side for comprehensive data insights, with the ability to save and reload visualization settings at any analysis step.

To make CytoAnalyst accessible to all researchers, we host it on a high-performance infrastructure with optimized networking and storage capabilities (Dual AMD EPYC 9654 96-Core Processor, 192 cores and 384 threads, 3TB DDR5 RAM, 112TB usable NVMe SSD storage, 4 NVIDIA H100 GPUs). The platform supports all major web browsers without installation or registration requirements.

Implementation

Functional overview

Figure 1 shows the overall structure of CytoAnalyst. The platform consists of three main systems: 1) study management and data sharing, 2) grid-layout visualization system, and 3) core analysis system with seven analytical modules. CytoAnalyst supports a typical single-cell RNA sequencing analysis workflow that guides users from data upload to biological interpretation. Here we summarize each system and the typical user workflow, then describe them in detail in the following sections.

A typical analysis workflow in CytoAnalyst begins with data uploading, where users can import 10X Genomics Cell Ranger[9] output (.tar.gz or .h5 format) or AnnData[23] objects (.h5ad format), along with additional metadata (.csv/.tsv format). Following data upload, users perform quality control and preprocessing steps, including cell and gene filtering based on UMI counts, gene counts, and mitochondrial/ribosomal gene percentages. The next step involves normalization (log-normalization or SCTransform) and data integration across multiple samples using methods such as RPCA[24], Harmony[25], or CCA[26] when analyzing multiple datasets. Users then proceed to dimensionality reduction and visualization using PCA[27], UMAP[28], or t-SNE[29] for transcriptome landscape visualization. The workflow continues with clustering analysis using Leiden[30] or Louvain[31] algorithms to identify distinct cell populations, followed by cell annotation using marker genes, reference databases, or AI-powered inference tools. Differential expression (DE) analysis identifies marker genes and investigates expression differences between cell groups using statistical methods such as the Wilcoxon rank-sum test[32]. For deeper biological insights, users can perform gene set enrichment analysis using curated pathways and marker sets, and conclude with pseudo-time trajectory analysis using Slingshot to investigate cellular development and differentiation processes.

This interconnected workflow is supported by the study management and data-sharing system that allows users to create and manage their projects. The system maintains detailed analysis logs and enables collaboration through secure sharing capabilities, allowing researchers to distribute both data and analysis outcomes with colleagues while all analysis parameters are automatically documented. The platform provides extensive documentation and tutorials at each analytical step.

The grid-layout visualization system enables dynamic exploration of single-cell data through multiple complementary approaches. The system supports flexible and simultaneous display of multiple plot types: scatter plots, violin plots, dot plots, heatmaps, histograms, volcano plots, and trajectory plots. A distinctive feature of CytoAnalyst lies in its emphasis on interactive visualization and real-time analysis. The platform implements a flexible visualization framework that facilitates the creation of customized plots with adjustable parameters, overlay of multiple data types, and interactive cell selection for focused examination.

Finally, the core analysis system consists of independent modules for embedding analysis, clustering, DE analysis, gene set management, cell enrichment, cell annotation, and pseudo-time trajectory inference. The modules are interconnected, enabling users to seamlessly transition between different steps while maintaining complete control over parameter settings. Advanced users can skip any steps and begin with any process based on their research needs and dataset characteristics. For computationally demanding tasks, CytoAnalyst employs an advanced job queuing system that efficiently manages server resources while delivering real-time progress updates. Users can initiate multiple analyses without waiting for previous tasks to complete. This functionality allows researchers to explore different parameter configurations simultaneously. All analytical results are securely stored on the server and remain accessible through the platform's interface.

All analysis results can be exported for downstream publication and sharing. Clustering results, DE analysis outputs, cell type annotations, and pseudo-time trajectory inference results can be exported as CSV files. All visualizations, including scatter plots, violin plots, dot plots, heatmaps, histograms, volcano plots, and trajectory plots, can be exported in multiple high-quality formats (PNG, JPEG, SVG) with customizable dimensions. Additionally, complete analysis workflows including all embeddings (PCA[27], t-SNE[29], UMAP[28]), clustering assignments, cell enrichment statistics, annotation labels, trajectory inference outputs, and associated metadata can be exported as comprehensive AnnData objects[23] (.h5ad format) for seamless integration with other single-

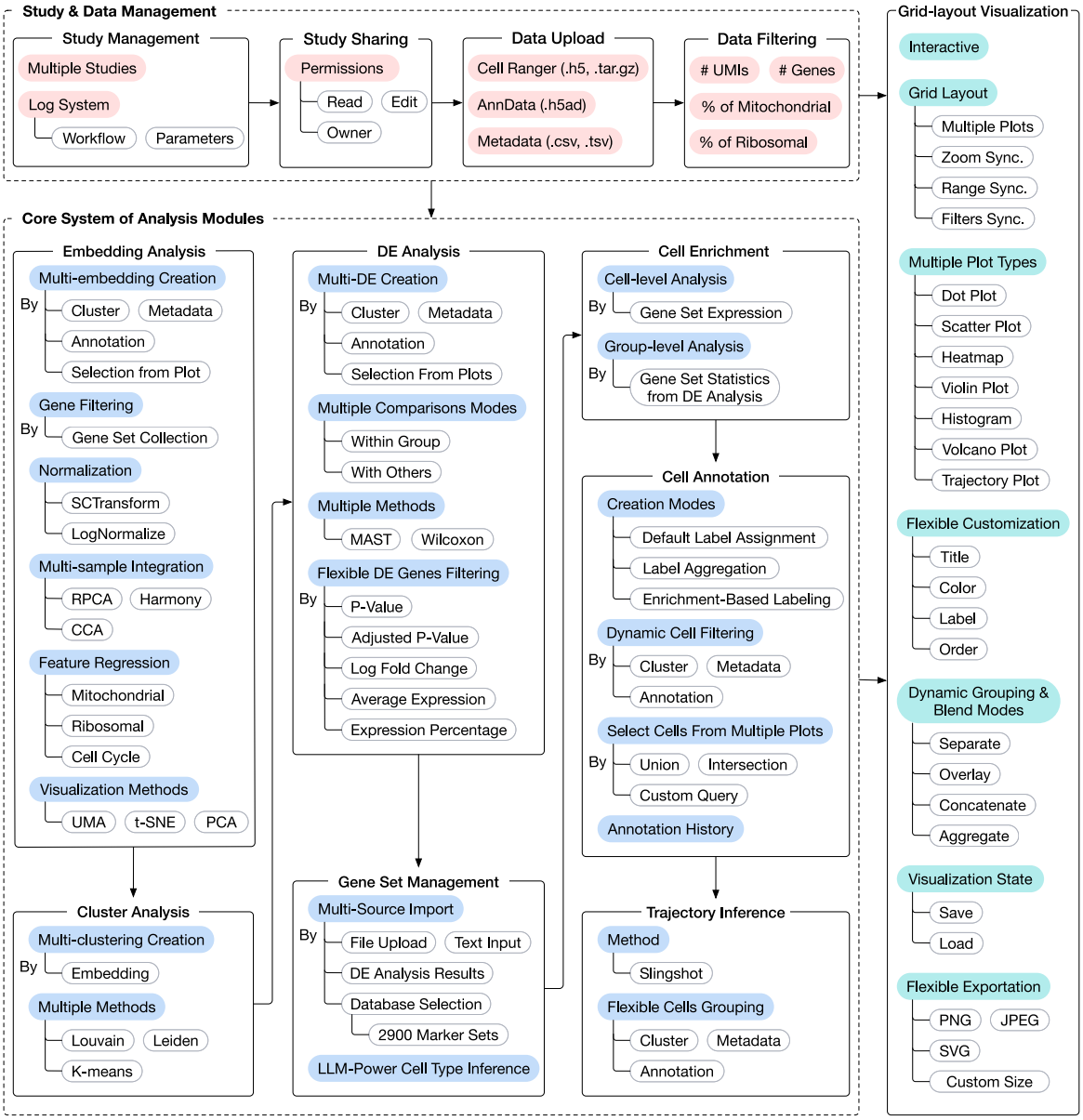

Fig. 1. Overview of the CytoAnalyst's capabilities and core analytical components. The platform consists of three main modules: 1) study management and sharing system, 2) grid-layout visualization system, and 3) core analysis system of seven analytical modules. The management system allows users to create and manage projects and data. The system supports 10X Genomics Cell Ranger in .tar.gz or .h5 format, and AnnData objects in .h5ad format, as well as additional metadata in .csv/.tsv format. Following data upload, users can perform quality control and preprocessing before performing downstream analyses. All analysis parameters and results are automatically logged and can be shared with collaborators through a secure sharing system. The grid-layout visualization system supports flexible and simultaneous display of multiple plot types: scatter plots, violin plots, dot plots, heatmaps, histograms, volcano plots, and trajectory plots. The platform's blending modes allow users to combine plots in various ways for comprehensive data exploration. The core system consists of seven analysis modules: embedding analysis, clustering, differential expression (DE) analysis, gene set management, cell enrichment, cell annotation, and pseudo-time trajectory inference. These analysis modules are interconnected, enabling users to seamlessly transition between different steps while maintaining complete control over parameter settings. Advanced users can skip any steps and begin with any process based on their research needs and dataset characteristics.

cell analysis platforms such as Scanpy[4] and Seurat[33], enabling collaborative research and extended downstream analysis.

Study management and sharing

Study management

The study management system allows users to efficiently manage, share, and monitor their analyses. Researchers can create multiple studies within the platform to organize their data, which can be from different experiments or conditions. Each study maintains an independent analysis workflow, allowing focused investigation of related datasets. Users can transition among studies seamlessly through a user-friendly interface, with all analysis results and parameters preserved within their respective contexts.

The platform also provides a secure sharing system for efficient collaborations. Individual studies can be shared through protected links, enabling real-time interaction among team members. Access permissions for a shared study can be controlled at granular levels, allowing collaborators to view or alter parameter settings and the analysis pipeline. The owner of a study can grant or revoke access to the study at any time, ensuring data security and privacy. The sharing system also facilitates seamless analysis continuity across different devices.

To ensure reproducibility and accountability, CytoAnalyst maintains a comprehensive log of all analytical operations. Users can review the analysis history for each study, encompassing data upload details, preprocessing steps, parameter settings, analysis workflow, and visualization. This logging mechanism guarantees that all analytical decisions are documented and retrievable if necessary.

Data upload and pre-processing

CytoAnalyst accepts two standardized formats for single-cell data: output from 10X Genomics Cell Ranger[34] in .tar.gz or .h5 format, and AnnData[23] objects in .h5ad format. Researchers can supplement each dataset with metadata (sample information, experimental conditions, etc.). Multiple files can be uploaded for the same study, enabling integrated analysis across samples or conditions. Upon data selection, CytoAnalyst performs automatic format detection and validation to ensure data consistency. The platform generates an interactive preview of uploaded data, allowing verification of gene identifiers, cell barcodes, and metadata fields.

Following data upload, users can perform quality control and pre-processing before other downstream analyses. The platform computes and visualizes key quality metrics, including unique gene counts per cell, unique molecular identifier (UMI) counts per cell, percentage of mitochondrial genes, and/or ribosomal genes. These metrics are displayed as interactive violin plots showing value distributions across all cells. Users can dynamically adjust filtering thresholds while observing effects on cell populations in real time through violin plots and dimensional reduction visualizations.

For multi-sample experiments, quality metrics are computed and displayed independently for each sample, enabling sample-specific quality control thresholds. Sample identity information is preserved throughout the analysis to facilitate downstream data integration and comparative analyses. After initial data processing, users retain the flexibility to incorporate additional samples into the study or supplement existing samples with new metadata. This adaptive approach allows seamless progression to subsequent analysis steps, including data integration, dimensionality reduction, clustering, cell type annotation, differential expression analysis, and trajectory inference.

Grid-layout visualization

Figure 2 illustrates the visualization capabilities of CytoAnalyst that enable dynamic exploration of single-cell data through multiple complementary approaches. The visualization architecture centers around a grid-layout system for the simultaneous display of different data aspects (Fig. 2A1–8). The framework supports diverse visualization types, including scatter plots, violin plots, dot plots, heatmaps, trajectory plots, histograms, and volcano plots. Each plot is optimized for specific analysis contexts and data types.

Scatter plots facilitate visualization of both continuous and categorical variables using two-dimensional spaces from t-SNE[29] and UMAP[28]. These variables encompass metadata, gene expression levels, cluster assignment, cell type annotation, enrichment results, and trajectory information (Fig. 2A1–2). Trajectory plots visualize inferred cellular paths and associated gene expression dynamics. These plots can overlay gene expression levels, pseudotime ordering, or cluster assignments to provide comprehensive views of biological processes (Fig. 2A3). Violin plots represent gene expression distributions across cell populations. Users can examine individual genes or gene sets, with options to group cells by categorical variables or their combinations (Fig. 2A4). Dot plots and heatmaps are best for the visualization of expression patterns across multiple genes and cell populations. Heatmaps provide detailed expression patterns, while dot plots offer concise representations highlighting informative genes with their expression levels and percentages (Fig. 2A5–6). Histograms display the distribution of individual variables across all cells. Users can adjust bin sizes and range to focus on specific distribution aspects (Fig. 2A7). Volcano plots illustrate differential expression results with adjustable significance thresholds and effect size cutoffs (Fig. 2A8).

The platform incorporates multiple plot-blending modes, allowing users to combine plots in various ways for comprehensive data exploration. Figure 2B1–3 shows three examples: 1) two categorical variables are aggregated to create a new plot with hierarchical labels, 2) a categorical and a continuous variable are aggregated to create a new plot with a gradient color mapping for each category, allowing for adjusting filtering on each category independently, and 3) an overlay mode for transparent overlaying of two plots for comparison. Overall, supported blending modes include replace, separate, aggregate, overlay, and concatenate. Replace mode enables straightforward feature visualization by substituting plots in the grid layout. Separate mode facilitates side-by-side comparison through the addition of new plots. Aggregate mode combines multiple variables into unified visualizations. For categorical variables, the system generates plots with combined variable coloring. For

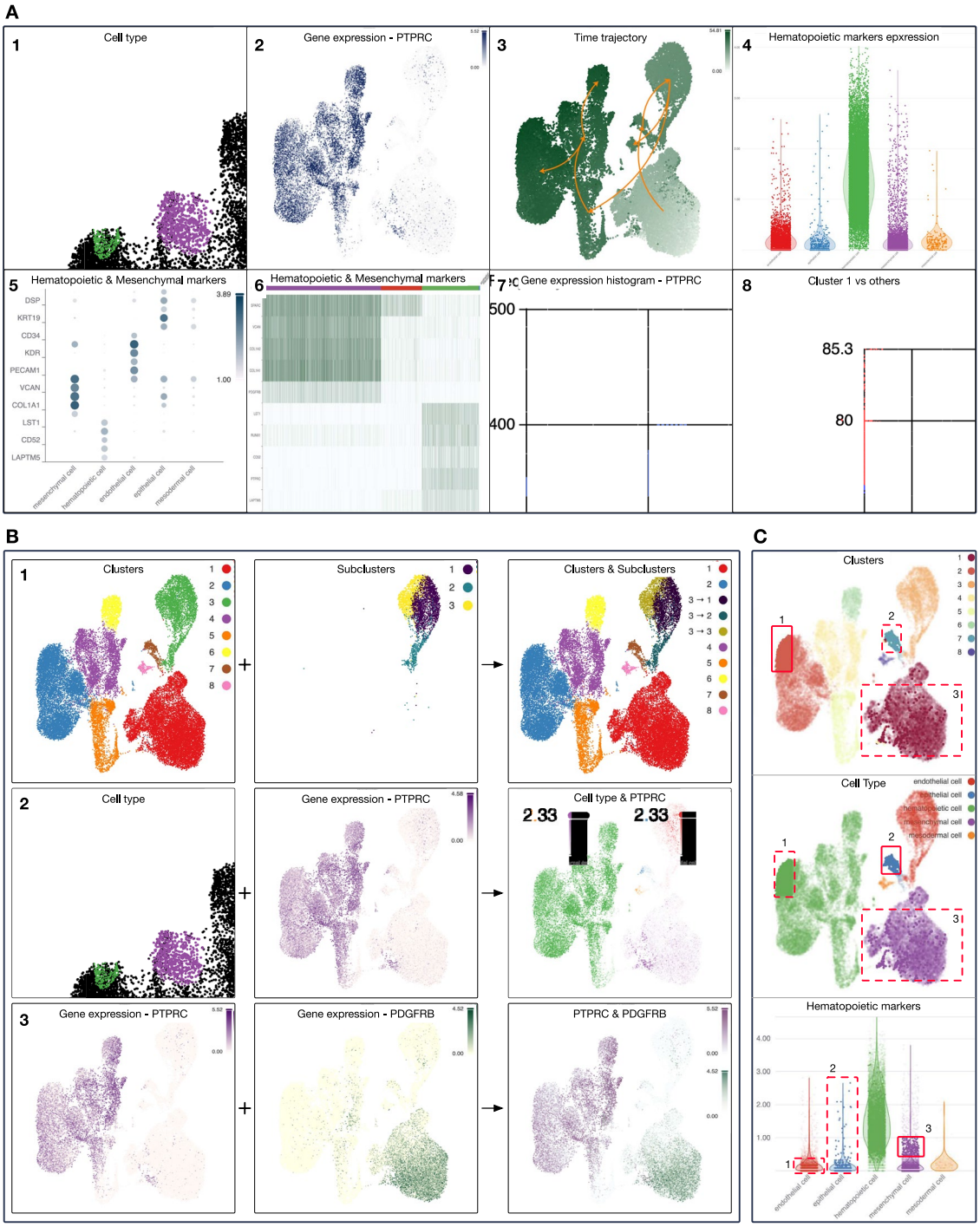

Fig. 2. Interactive visualization in CytoAnalyst. (**A**) Chart types supported by the platform, including 1) scatter plot for categorical data (e.g., metadata, clustering, annotation labels), 2) scatter plot for continuous data (e.g., UMI counts, gene expression, enrichment scores, pseudotime), 3) graph plot overlaying scatter plot (i.e., trajectory inference), 4) violin plot (i.e., distribution of gene expression), 5) dot plot (i.e., expression of genes across cell groups), 6) heatmap (i.e., expression of genes across individual cells), 7) histogram (e.g., distribution of gene expression, UMI counts), and 8) volcano plot (i.e., differential expression analysis results). With violin plots, dot plots, and heatmaps, users can group cells by any categorical variables or their combinations. (**B**) Blending modes in CytoAnalyst, including 1) aggregate mode for combining two categorical variables, 2) aggregate mode for combining a categorical and a continuous variable, resulting in a new plot with a gradient color mapping for each category, and 3) overlay mode for transparent overlaying of two plots for comparison. C) Cell selection across multiple plots using the grid layout, enabling focused examination of specific cell populations. In this example, there are 3 sets of cells selected in 3 different plots. Red solid-lined boxes indicate the selected area/cells in each plot. Red dashed-line boxes indicate the corresponding selected area/cells in other plots.

continuous variables, it calculates and displays average values. When variable types are mixed, i.e., continuous and categorical, the system separates data points by categories and colors them using a gradient scale to allow users to compare values across categories. Overlay mode enables transparent plot overlaying to examine relationships between variables. Concatenate mode stacks multiple features in heatmaps and dot plots for direct comparison.

All visualizations support interactive features, including zooming, panning, cell selection, and data point tooltips. Data filtering can be applied to any categorical or continuous variable, affecting individual plots or the entire grid layout. Figure 2C demonstrates the cell selection across multiple plots using the grid layout in CytoAnalyst. When there are multiple plots in the grid layout, users can select cells in one plot and see the corresponding selection in other plots. In this example, there are three different plots with a selected region indicated by a red solid-line box in each plot. Two red dashed-line boxes in each plot indicate the corresponding selected cells in other plots. This feature is particularly useful for examining feature expression patterns across multiple visualizations or selecting cells for annotation using complex criteria, such as the union or intersection of multiple regions.

Plot arrangements can be modified through drag-and-drop interactions with real-time updates. Users can synchronize zoom levels across plots for direct feature comparison and modify individual plot parameters without altering source data. Color mapping in CytoAnalyst supports both categorical and continuous variables through predefined color presets and custom color palettes, and gradients. For categorical variables like cluster assignments, users can define custom colors for each category. For continuous variables, adjustable color gradients for min and max values are available. Users can export any figure in PNG, JPEG, or SVG formats with user-defined dimensions and resolution. They can save and load visualization settings as profiles for future use or collaboration, with all parameters and configurations preserved. Users can load saved visualization profiles with only a few clicks.

Core analysis modules

At the core of its capabilities, CytoAnalyst consists of seven analysis modules: embedding analysis, clustering, DE analysis, gene set management, cell enrichment, cell annotation, and pseudo-time trajectory inference. These modules are interconnected, allowing for flexible analyses and collaborations. Depending on data and research goals, advanced users can start the analysis using any module and subsequently refine the results based on their preferred analysis pipelines.

Embedding analysis

Figure 3 shows CytoAnalyst's embedding analysis workflow for one or multiple samples. The workflow is compatible with that of Seurat for data integration and dimensionality reduction. Users start the embedding analysis by selecting the cells of interest using sample information, metadata variables, cluster labels, existing cell annotations, and manual selection. Users can filter cells based on any categorical or continuous variable,

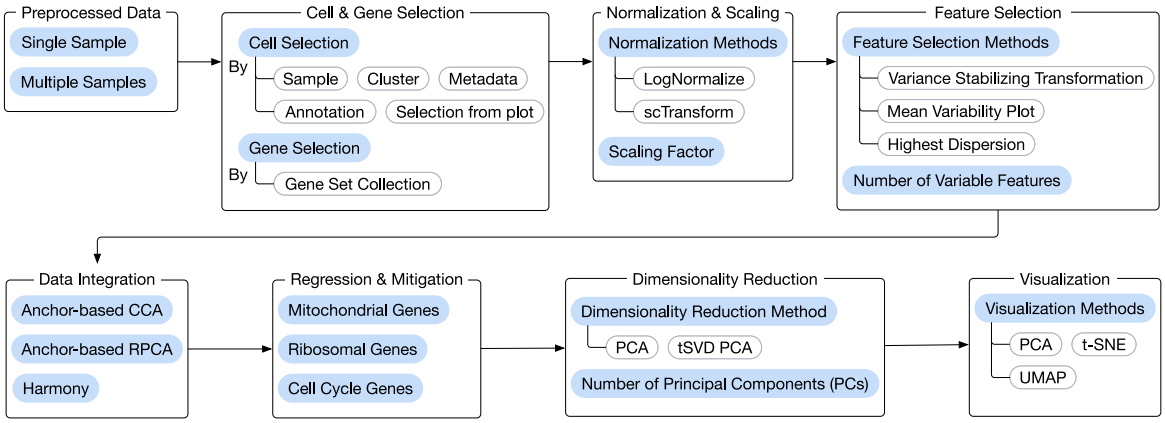

Fig. 3. Embedding analysis for one or multiple samples. Users start by selecting the cells of interest using sample information, metadata, cluster labels, existing annotations, and different visualization plots. They can also choose to focus on a pre-defined set of genes using custom gene sets or cell type markers. After cell and gene selection, the next step is to scale/normalize the data (using LogNormalize or scTransform) and to identify highly variable features/genes (using variance stabilizing transformation, mean variability plot, or highest dispersion). The platform includes three integration methods (anchor-based CCA, anchor-based RPCA, and Harmony) for integration and batch correction of single-cell data and samples obtained from different experiments or sources. Users can also regress out unwanted sources of biological variation related to mitochondrial, ribosomal, or cell cycle genes. Finally, users can perform dimensionality reduction and transcriptome landscape visualization using PCA, t-SNE, or UMAP. The platform also facilitates the rapid creation of multiple embedding analyses for categorical variables, such as generating separate embeddings for each cluster. This capability proves particularly valuable for sub-clustering analysis, as global embeddings may not capture the hierarchical data structure.

or directly select the cells from the visualization interface. In parallel, the gene filtering option offers additional analytical flexibility by allowing users to focus their analyses on pre-defined gene sets. This is particularly useful when focusing on specific cell types using known marker genes.

After selecting cells and genes of interest, users can normalize the data using LogNormalize[26] or SCTransform[35]. They can perform feature selection to identify highly variable genes using Variance Stabilizing Transformation[36], Mean Variability Plot[37], or Highest Dispersion[38]. When analyzing samples and data from multiple sources, users can perform data integration and batch correction using three established approaches: Anchor-based CCA Integration[26], Anchor-based RPCA Integration[24], or Harmony[25]. Users can also regress out unwanted sources of biological variation related to mitochondrial, ribosomal, or cell cycle genes. Finally, users can perform dimensionality reduction and visualize the embedding results using truncated SVD PCA (tSVD PCA)[39,40], vanilla PCA[27], t-SNE[29], or UMAP[28]. The interactive interfaces enable customization of advanced parameters, allowing users to maintain complete control over the analysis.

To extend embedding functionalities, the platform also facilitates the rapid creation of multiple embedding analyses for categorical variables, such as generating separate embeddings for each cluster. This capability proves particularly valuable for sub-clustering analysis, as global embeddings may not capture the hierarchical data structure. With the built-in grid-layout visualization ability, generated embeddings can be visualized with any existing labels. These embeddings serve as foundations for downstream analyses, including clustering, cell type annotation, and trajectory inference.

Cluster analysis

CytoAnalyst implements a flexible cluster analysis framework with multiple algorithms to identify distinct cell populations. The platform includes Louvain[31], Leiden[30], and K-means[41] as clustering methods to accommodate diverse data structures. Louvain and Leiden utilize adjustable resolution parameters to control cluster granularity, while K-means requires the specification of cluster numbers. The graph-based Louvain and Leiden methods excel at global-level clustering analysis for identifying cellular populations across complete datasets, while K-means is more suitable for sub-clustering where the number of clusters is known[42].

To perform clustering, users can choose one or multiple embeddings from the embedding analysis and specify the clustering algorithms with corresponding parameters. The platform enables the simultaneous creation of multiple clustering analyses, facilitating the comparison of different parameter settings and algorithms. This capability proves essential for identifying optimal parameters and cluster numbers. The grid-layout visualization system enables direct comparison of clustering results across different embeddings and parameters.

For investigating finer population structures, CytoAnalyst supports hierarchical sub-clustering analysis. Users can perform embedding analysis on specific clusters from initial clustering results, followed by subsequent clustering analysis on these focused embeddings. This approach reveals heterogeneity within major cell types that may be obscured in global embedding spaces.

Once the cluster analysis is done, the cluster labels can be transferred to other embeddings for visualization. In other words, one can use a specific embedding for clustering and then choose any other embeddings to display the cell labels. This flexibility enables visualization of sub-clustering results within global embedding spaces alongside primary cluster labels. The platform's aggregation blend mode facilitates the visualization of multiple clustering levels in unified plots, providing comprehensive views of cellular hierarchies. This visualization approach extends to combining clustering results with metadata labels, enabling the exploration of relationships between cell populations and experimental conditions.

Differential expression (DE) analysis

Figure 4 shows an example of DE analysis using the user interface implemented in CytoAnalyst. Figure 4A shows an example DE analysis configuration in which users can choose to compare cells from different clusters (by cluster), cell groups separated by conditions or other variables (by metadata), annotated cell types (by annotation), or customized groups (custom). Via comparison mode, users can choose to compare each cell group against all other groups (with others) or compare cells from different conditions (within cluster). The cell filtering setting allows users to refine the comparative analysis by choosing samples, metadata, cluster labels, and annotated cell types in each of the two groups involved. The method configuration setting allows users to choose the method (Wilcoxon, MAST) and other important parameters (max cell, min percent, log fold-change). Figure 4B shows the preview table with details of the cell groups involved (cell count, selected samples, and clusters) and the total number of cells. Figure 4C shows DE analysis results for a comparison, in which the results can be sorted and/or filtered using any of the computed statistics of the genes (p-value, log fold-change, average expression, percentage of cells with positive expression, and percentage difference). Figure 4D displays the volcano plots in which genes with adjusted p-values less than 5% and absolute log fold-change higher than 2 are highlighted in red.

Overall, we implement a comprehensive framework for comparative analysis among different types/clusters, conditions, and time points. Users can perform DE analysis on any cell subset using an advanced selection system that combines interactive visualization and filtering based on both categorical and continuous variables. The DE analysis workflow supports three distinct comparison modes: between groups, within groups, and user-defined group comparisons. Between-group analysis identifies genes differentially expressed between clusters or categorical variables, such as comparing a cluster against all other cells. Within-group analysis examines condition-specific effects in defined populations, enabling comparison of identical clusters across different conditions. Custom analysis refers to a customized grouping defined by users. The platform facilitates the creation of multiple DE analyses simultaneously, providing a preview interface that displays comparison groups with their respective cell counts and gene numbers.

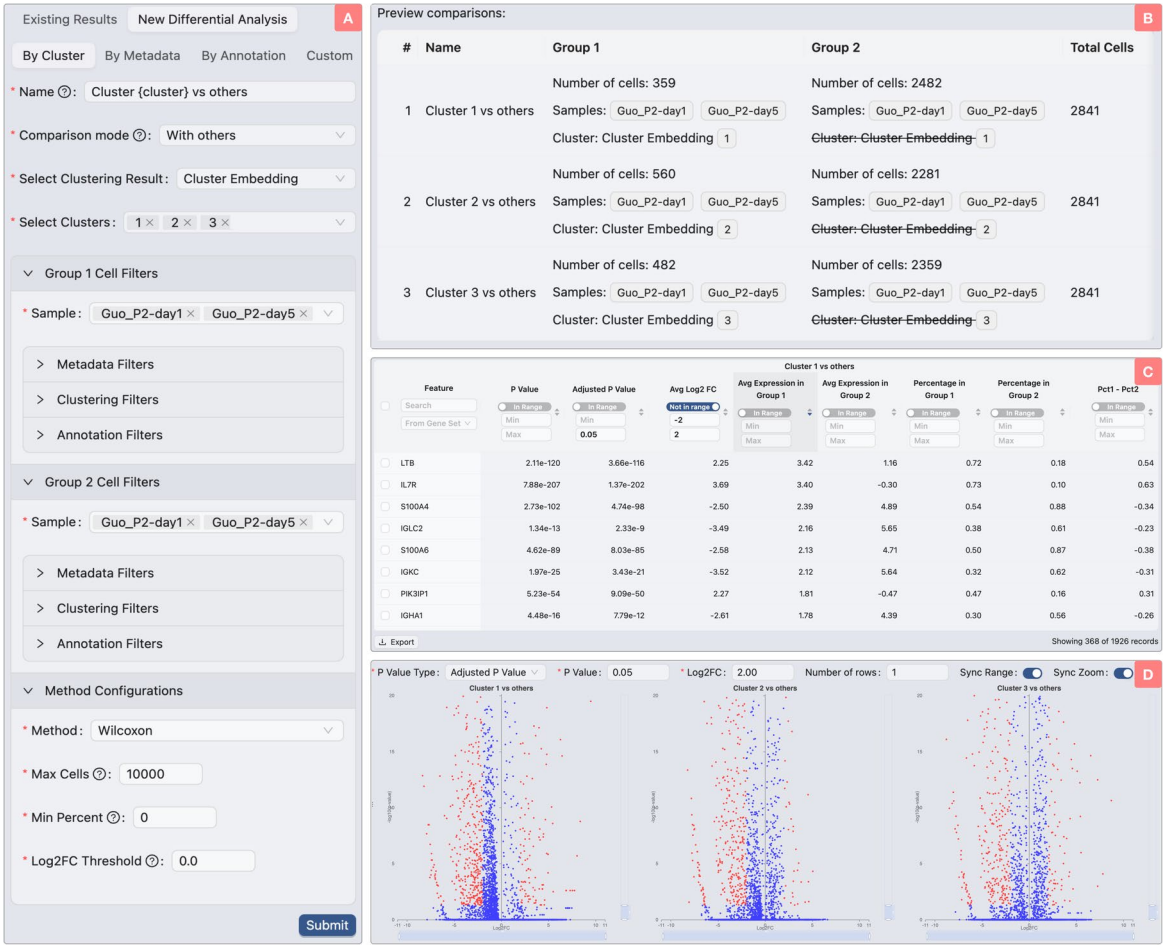

Fig. 4. User interface for DE analysis. A) DE analysis configuration. Users can choose to compare cells from different clusters (by cluster), cell groups separated by conditions or other variables (by metadata), annotated cell types (by annotation), or groups manually selected by users (custom). In this example, we perform DE analysis by comparing cell groups obtained from cluster analysis. Via comparison mode, users can choose to compare each cluster against all other clusters (with others) or compare cells from different conditions (within cluster). The cell filtering setting allows users to choose specific samples, metadata, cluster labels, and annotated cell types. The method configuration setting allows users to choose the hypothesis testing method (Wilcoxon, MAST) and important parameters: 1) total number of cells involved (max cell), 2) minimum expression percentage for genes (min percent), and 3) minimum log fold-change for genes. B) Preview table before performing the DE analysis. In this example, users are comparing each of the three clusters against all other clusters. The table displays the name for each comparison, details of the cell groups involved (cell count, selected samples, and clusters), and the total number of cells. C) DE analysis results for one of the three comparisons. The results can be sorted and/or filtered using any of the computed statistics of the genes: 1) p-value and adjusted p-value, 2) log fold-change, 3) average expression in each group, 4) percentage of cells in which the genes are expressed, and 5) percentage difference. D) Volcano plots of all comparisons that show the adjusted p-values and log fold-change. In this example, genes with adjusted p-values less than 5% and absolute log fold-change higher than 2 are highlighted in red, while the rest of the genes are highlighted in blue.

For hypothesis testing, users can choose between the Wilcoxon rank-sum test[32] and the Model-based Analysis of Single-cell Transcriptomics (MAST)[43] to compute the p-values for the genes, followed by a correction for false discovery rate using Benjamini-Hochberg[44]. After hypothesis testing, users can refine the list of DE genes using p-value, log fold-change, average expression, and minimum expressing cell percentage (in each group or between groups). Result exploration for DE analysis utilizes interactive volcano plots arranged in grid layouts, accompanied by comprehensive statistical tables. The interface enables gene searching, filtering, and highlighting across plots. Selected genes can also be incorporated into existing gene set collections for other downstream analyses.

Gene set management

A gene set is a collection of unique genes grouped together based on a shared characteristic or function. A gene set can represent a cellular process or functional module, a signaling pathway, a set of markers of a specific cell type, a set of markers for a condition or disease, or simply a set of DE genes obtained from a comparative analysis.

There are multiple applications of gene sets in CytoAnalyst, including: 1) gene filtering in embedding analysis, 2) cell enrichment and annotation, 3) artificial intelligence (AI)-based cell type inference using Large Language Models (LLMs), and 4) visualization and identifying patterns of gene expression associated with specific cellular processes or phenotypes.

CytoAnalyst implements a hierarchical system for creating, importing, and managing gene sets. The platform organizes gene sets into collections, enabling logical grouping of closely related gene sets. This hierarchical structure facilitates the management of different biological contexts, such as varying cell type granularity levels or pathway-specific gene groups. Users can create distinct collections for broad cell types, cell subtypes, or biological pathways to explore enriched processes in specific cell populations. The platform supports multiple methods for gene set creation and management. Users can create new collections, import gene sets from external sources in tabular or GMT formats, or choose from curated pre-defined collections. Within each collection, users maintain full control over gene set composition, including renaming, deletion, and modifying the gene set (adding and removing genes).

By default, CytoAnalyst embeds a comprehensive collection of cell type markers available from CellMarker 2.0[45]. The database contains a curated compilation of experimentally validated markers for known cell types in human and mouse tissues. For *Homo sapiens*, users can select from more than 400 tissues, 1,500 cell types, and 15,000 markers. Similarly, the *Mus musculus* reference collection offers users approximately 300 tissues, 1,400 cell types, and 12,000 markers. Users can easily search for these reference gene sets (by tissue or cell type) and incorporate them into their analysis pipelines with minimal effort.

The gene set management interface also provides a cell-type inference tool that leverages AI to infer potential cell types associated with a gene set. Given the tissue and the gene set (marker genes), the tool returns a list of cell types most likely present in the sample, structured in a cell ontology hierarchy. The inference tool is based on Meta's llama 3.3[46], a recent 70B parameter model, using the Ollama framework to create a responsive API system that enables efficient LLM communication. To ensure an accurate inference with a consistent output format, we design a prompt template that directs the LLM with specific guidance and context. This AI-based inference feature is a proof-of-concept tool that demonstrates the potential of integrating large language models into single-cell analysis workflows.

Cell enrichment analysis

CytoAnalyst supports both cell-level and group-level enrichment analyses. Cell-level enrichment basically performs enrichment analysis of pre-defined gene sets for each individual cell. Given a cell, CytoAnalyst first calculates the z-score of each gene and then compares the z-scores of genes in a gene set against genes in all other gene sets using one-sided Welch's t-test[47]. In addition to the p-value for each gene set, the software also returns a score (average z-score of genes in the gene set) and score difference (by subtracting the average z-score of genes in the gene set from the average z-scores of genes in other gene sets). Users can visualize the p-values, scores, and score differences across cells and gene sets for cell annotation or developmental stages.

In addition to cell-level enrichment, CytoAnalyst also supports enrichment analysis for a group of cells (group-level enrichment). Using the DE analysis results, the platform computes the p-values and statistics of pre-defined gene sets for the cell group. The platform first ranks the genes using log2FC and p-values and then applies FGSEA[48] to compute enrichment scores and statistical significance for each gene set. The platform also calculates the enrichment score difference as the difference between the enrichment score of the target gene set and the average enrichment score of all other gene sets. The obtained enrichment score, enrichment score difference, and statistical significance are then assigned to all cells of the underlying group.

Cell enrichment primarily facilitates cell type annotation and cell state identification. For example, users can enrich cells with known cell type markers to assign labels to distinct populations, or enrich DE genes with biological pathways to characterize cellular states. In both cases, the platform calculates enrichment scores for the target gene set, enrichment score differences between the target gene set and other genes, and the statistical significance for individual cells or cell groups.

The sets of DE genes identified through the differential analysis between the chosen groups can be exported to be subsequently analyzed through a comprehensive omics analysis platform such as iPathwayGuide[49], or others. Furthermore, the DE genes identified by the differential analysis can also be used as annotations/markers for the identification of cell types in subsequent experiments (see below).

Cell annotation

Cell annotation is a critical step in scRNA-Seq analysis, enabling researchers to assign meaningful labels to distinct cell populations. CytoAnalyst implements a flexible annotation system combining manual curation, marker-based enrichment, and AI-based inference for cell type assignment.

Users can create a new annotation instance through three available options: 1) default label assignment (Fig. 5A), 2) label assignment using metadata, clustering, and former annotations (Fig. 5B), and 3) assignment using cell enrichment results (Fig. 5C). In the first option (default), users can assign an unknown label to all cells in the dataset and then gradually refine the annotation through manual examination and analysis. In the second option (label aggregation), users can combine variables from metadata, clustering, and former annotations to generate new cell labels. In the third option (enrichment analysis), users can assign initial cell labels based on enrichment statistics, metadata, clustering, and other variables.

The annotation feature is not an isolated module but is supplemented by other modules, including *Gene Set Management* and *Cell Enrichment Analysis*, so that users can create accurate and informative annotations (see sections above). Users can switch between the annotation interface and other interfaces while editing the annotation. Figure 5D shows the interface for annotation editing. The interface allows users to flexibly filter cells using metadata, clustering, or any other labels (Fig. 5D1). The platform displays cells in a tabular format

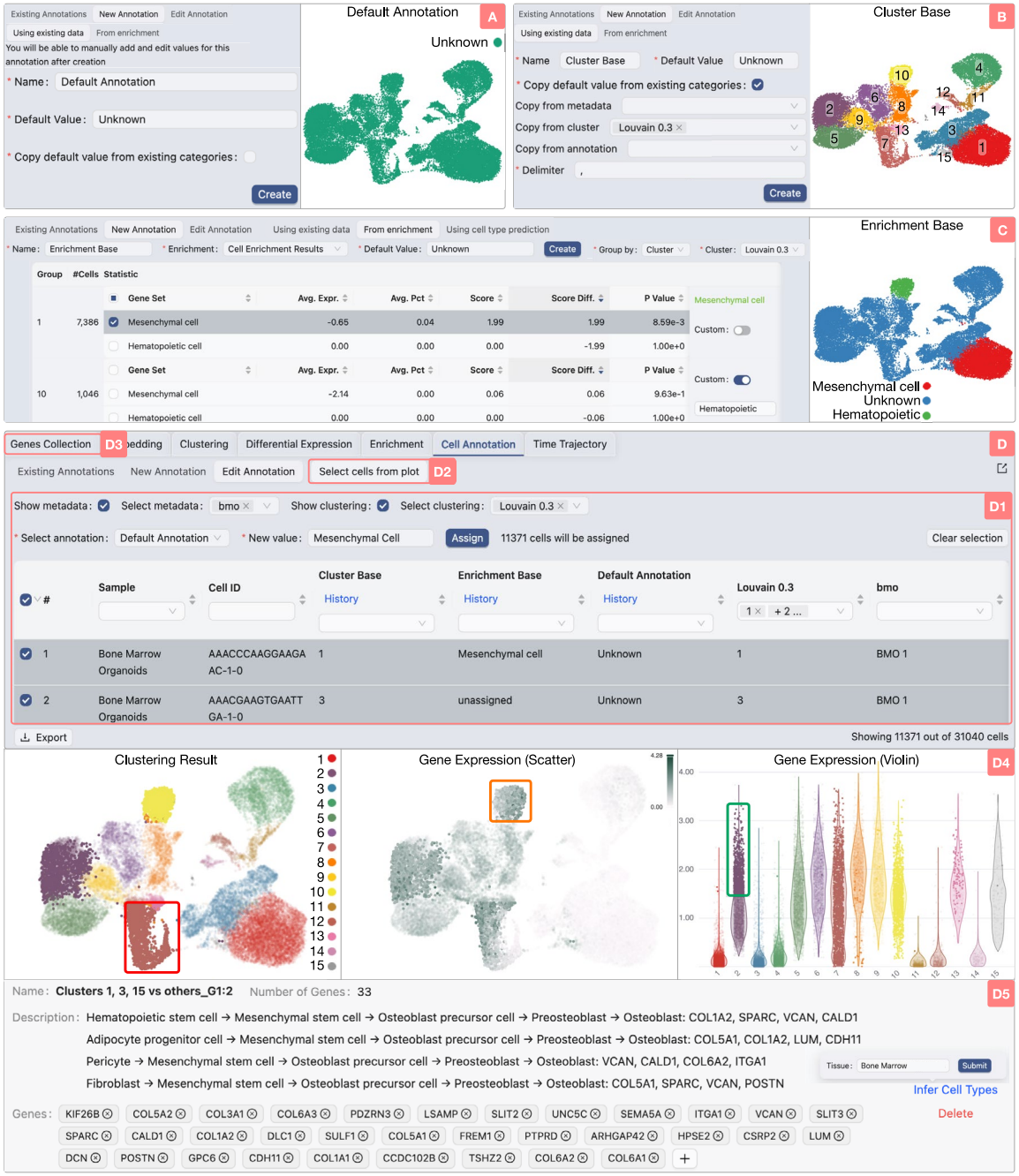

Fig. 5. Cell annotation interface. (**A**) Default annotation initialization. Users can create a default annotation in which all cells are unknown, and then gradually refine the annotation. (**B**) Annotation initialization using clustering results, metadata, and/or former annotations. (**C**) Annotation initialization using enrichment analysis results. Using enrichment results, users can assign labels to a group of cells by either selecting a gene set with satisfying statistics in the group or assigning a custom label to cells in a group. (**D**) Annotation editing interface. CytoAnalyst allows users to flexibly choose and assign new labels to cells (D1). They can view all cell labels in the annotation table. In addition, users can choose to display specific information from metadata (show metadata) or clustering results (show clustering) as columns. In this example, we select 11,371 cells (out of 31,040) from clusters 1, 3, and 15, and then assign them as Mesenchymal (new value). Moreover, users can select the cells from different plots (D2, D4). They can also use the AI-based tool to infer the cell type based on the markers of the cells (D3, D5).

with current annotations, metadata variables, and cluster information. This table supports additional filtering operations and cell export capabilities. The annotation interface also integrates with the grid-layout visualization system, allowing users to select cells from multiple plots (Fig. 5D2 and D4). For complex selections across multiple plots, users can define how to combine these regions using union, intersection, or custom queries that

support AND, OR, and NOT operators. CytoAnalyst also provides users with a way to infer potential cell types for each group of cells based on their marker genes and tissue information (Fig. 5D3 and D5). By switching to the Genes Collection interface, users can utilize an AI model to infer the cell type based on the marker genes of a cell group. Finally, users can assign new labels to the selected cells, and the platform will update the annotation results immediately. For each session, the platform maintains the annotation history, allowing users to review and revert to previous assignments.

Trajectory inference

Trajectory inference in single-cell analysis is a computational approach that aims to infer dynamic biological processes, such as cellular differentiation, developmental pathways, or disease mechanisms, by mapping the gene expression changes cells undergo on a pseudo-timeline[5]. Even though all cells are usually part of the same sample taken at the same time, the main assumption behind the trajectory inference is that various cells are caught in different stages of a process. By grouping the cells with similar expression profiles and analyzing the differences between these groups, one can presumably establish an order for the changes that the cells go through, and thus create a trajectory in this pseudo-time. CytoAnalyst utilizes Slingshot to infer cell lineage trajectories and pseudo-temporal ordering from single-cell RNA-seq data[50]. The method has been shown to perform robustly across diverse datasets, balancing accuracy and flexibility in identifying branching lineages[51]. Slingshot treats cell clusters as nodes in a graph and constructs a minimum spanning tree (MST) connecting these nodes, thereby identifying the global lineage structure (e.g., the number of lineages and branching points). Next, it assigns pseudo-times to individual cells by fitting principal curves[52] to the data along each lineage, starting from a user-specified or algorithm-determined root cluster. These curves model smooth trajectories through the cells' expression space, capturing their presumed progression along divergent differentiation paths.

CytoAnalyst enhances trajectory inference by letting researchers group cells in multiple ways, using clusters, annotated cell types, or metadata (like treatment groups or patient cohorts). This flexibility allows researchers to focus on specific populations, such as tracking differentiation in cells from a particular experiment or comparing trajectories between healthy and diseased samples. Additionally, CytoAnalyst offers customizable settings for Slingshot, including the ability to choose how cluster distances are measured, modify the convergence threshold to control the precision of principal curve fitting, and so on. These customizations help researchers reproduce or apply them to their specific case studies.

After inferring trajectories, CytoAnalyst provides an intuitive visualization for users to explore and interpret the results. Researchers can visualize trajectories and gene expression in overlay mode to observe how a marker gene peaks at a branching point, and in aggregate mode to visualize multiple lineages on a single plot or in separate mode to examine individual lineages in detail.

Technical implementation

Figure 6 shows the overall architecture of CytoAnalyst. The web-based platform utilizes modern web technologies to provide a seamless user experience across different devices and browsers. At the user interface level, we use the React framework to enable the creation of dynamic, responsive components that are updated in real time as users interact with the platform. To enable real-time collaboration features (analysis sharing, result updating, etc.), we built the back end using WebSocket along with Meteor, a full-stack JavaScript platform that simplifies the development of real-time web applications. It provides a scalable, reactive architecture that enables efficient data synchronization between clients and servers. With this back-end architecture, CytoAnalyst provides real-time updates during data processing and synchronizes both individual and shared analyses across all clients, sessions, and devices. This enables users to monitor and alter analysis progress, intermediate results, and visualization. In

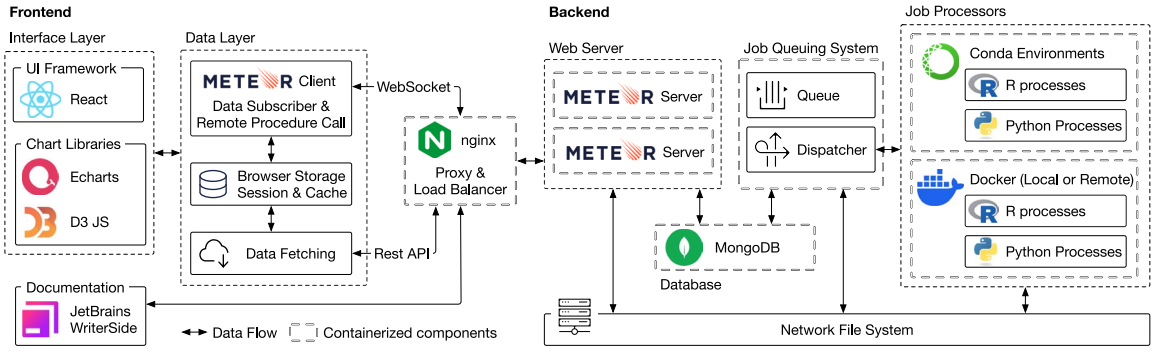

Fig. 6. CytoAnalyst software architecture. The platform is built on a modern web stack with React for the front-end interface, Echarts and D3 for interactive visualization, JetBrains WriterSide for documentation, Meteor for the back-end server, MongoDB for database management, and R and Python for data processing and analysis. CytoAnalyst implements a high-performance job queuing system to manage computational tasks and resource allocation, synchronizing real-time progress updates across sessions and devices. The platform is deployed using Docker containerized technologies to ensure portability and scalability, and is hosted on an enterprise-grade server infrastructure with optimized networking, storage, and computational capabilities to handle large-scale single-cell datasets.

addition, data management is one of the key components of CytoAnalyst's architecture. CytoAnalyst leverages MongoDB, a scalable NoSQL database for efficient storage and retrieval of large datasets. This robust database infrastructure supports CytoAnalyst's data management system, including automatic archiving of unused projects to optimize server resources.

The core sharing system in CytoAnalyst utilizes a flexible permission management framework. Project owners can share analyses as read-only, read-write, or full ownership. This granular permission system enables team leaders to maintain appropriate control while facilitating collaborative analysis. For each shared project, CytoAnalyst maintains complete version control. All analytical steps, including parameter settings and computations, are automatically logged and preserved. We implement a token-based authentication system tied to each study, ensuring only authorized users can view or modify analysis results. The platform maintains detailed access logs and enables project owners to revoke sharing permissions when needed. All analytical steps, including parameter settings and computations, are automatically logged and preserved.

CytoAnalyst uses corresponding modules from Seurat[33] for the majority of analytical functions. For data upload and preprocessing, we use Seurat's CreateSeuratObject and standard quality control functions. For normalization, we implement Seurat's NormalizeData with log-normalization. For data integration across samples, we use Seurat's integration workflow, including FindIntegrationAnchors and IntegrateData. For dimensionality reduction, we use Seurat's RunPCA, RunUMAP, and RunTSNE functions. For clustering analysis, we use Seurat's FindNeighbors and FindClusters functions, implementing the Louvain and Leiden algorithms. For k-means clustering, we use MiniBatchKmeans from the ClusterR[53] package to handle large datasets efficiently. For differential expression analysis, we use Seurat's FindMarkers and FindAllMarkers functions. For gene set enrichment analysis, we use the FGSEA[48] package for fast gene set enrichment analysis. For trajectory inference, we use Slingshot[50] for pseudo-time analysis and lineage reconstruction.

For the computational backend, we use a custom job queuing system to manage and distribute analytical tasks across multiple worker processes. This system enables parallel processing of multiple analyses while maintaining system responsiveness. The platform architecture is designed with extensibility as a core principle to accommodate the rapidly evolving landscape of single-cell analysis methodologies. The modular structure of our seven core analysis modules allows new functionalities to be added as independent modules without disrupting existing workflows. Our containerized job processing system, using Docker and Conda environments, enables seamless integration of new R packages, Python libraries, or different computational frameworks. The flexible parameter management system accommodates new method-specific parameters and configurations. The job queuing system dynamically allocates computational resources based on the specific requirements of each analysis module, ensuring optimal performance across different analytical workflows.

The platform is optimized for datasets ranging from thousands to hundreds of thousands of cells. Performance benchmarks (Supplementary Section 2) demonstrate excellent responsiveness for datasets up to 50K cells (analysis completion in under 10 minutes), good performance for datasets up to 200K cells (10-60 minutes), and full support for datasets up to 1M cells with longer computation times (1-24 hours depending on analysis complexity) (Supplementary Figure S2).

CytoAnalyst's extensible architecture ensures the platform can evolve to support emerging methodologies and data types. The data management infrastructure can accommodate different data types beyond single-cell RNA-seq, enabling support for technologies like spatial transcriptomics and single-cell proteomics, similar to how tools like Cellar[17] have expanded to handle spatial transcriptomics and proteomics data. Other platforms have successfully demonstrated such extensibility, including ezSingleCell[18], which supports multiple modalities, including spatial transcriptomics, and ASAP[16], which has extended beyond RNA-seq to include scATAC-Seq analysis. New analytical methods can be integrated through our modular framework by implementing the computational backend, defining parameter interfaces, and developing visualization components. The flexible parameter management system accommodates method-specific configurations, while our job queuing system handles computational demands with varied resource requirements. The platform's comprehensive logging systems ensure that new methods maintain the same level of reproducibility and traceability as existing modules.

Results

To demonstrate the capabilities of CytoAnalyst, we analyze three single-cell datasets obtained from previous studies[54–58] to showcase the platform's features and functionalities. The first dataset consists of 31,040 cells in bone marrow organoids[54]. The second dataset comprises 15,457 cells collected from the sun-protected inguinoiliac region of whole-skin samples from five male donors[55]. The third dataset consists of 5,828 bone marrow cells that were collected from three experimental studies[56–58].

Case Study 1: Annotation of bone marrow organoids

Here, we analyze the single-cell dataset (31,040 cells) from Frenz-Wiessner et al.[54], in which bone marrow organoids were generated from human induced pluripotent stem cells. Through manual examination of both scRNA-Seq and matched flow cytometry data, the authors identified five major cell types. However, for the purpose of illustrating the capabilities of CytoAnalyst, we assume that we do not know the number or cell types, nor the true annotation of the single-cell data. We aim to identify potential cell types based on the unbiased analysis of the single-cell data alone, without using external data from the matched flow cytometry. We will do so following a logical order of analysis steps: transcriptome landscape visualization, cluster analysis, DE analysis, and cell type annotation.

Transcriptome landscape visualization and louvain clustering: To initiate the analysis, we upload the file Bone_Marrow_Organoids.h5ad obtained from Frenz-Wiessner et al.[54]. Note that the authors have already filtered barcodes with fewer than 400 detected genes, more than 40,000 counts in total, and mitochondrial genes that

exceeded 10% of the total number of gene counts. They also employed Scrublet v.0.2.3[59] with default parameters, filtering transcriptomic profiles with a predicted doublet score exceeding 0.2 for the removal of doublets.

Figure 7A shows the transcriptome landscape of the updated data using UMAP. The visualization clearly shows that the transcriptome landscape consists of three major cell populations (marked as I, II, and III). To determine potential cell groups, we perform Louvain clustering[31] with the default resolution of 0.3 (Fig. 7B).

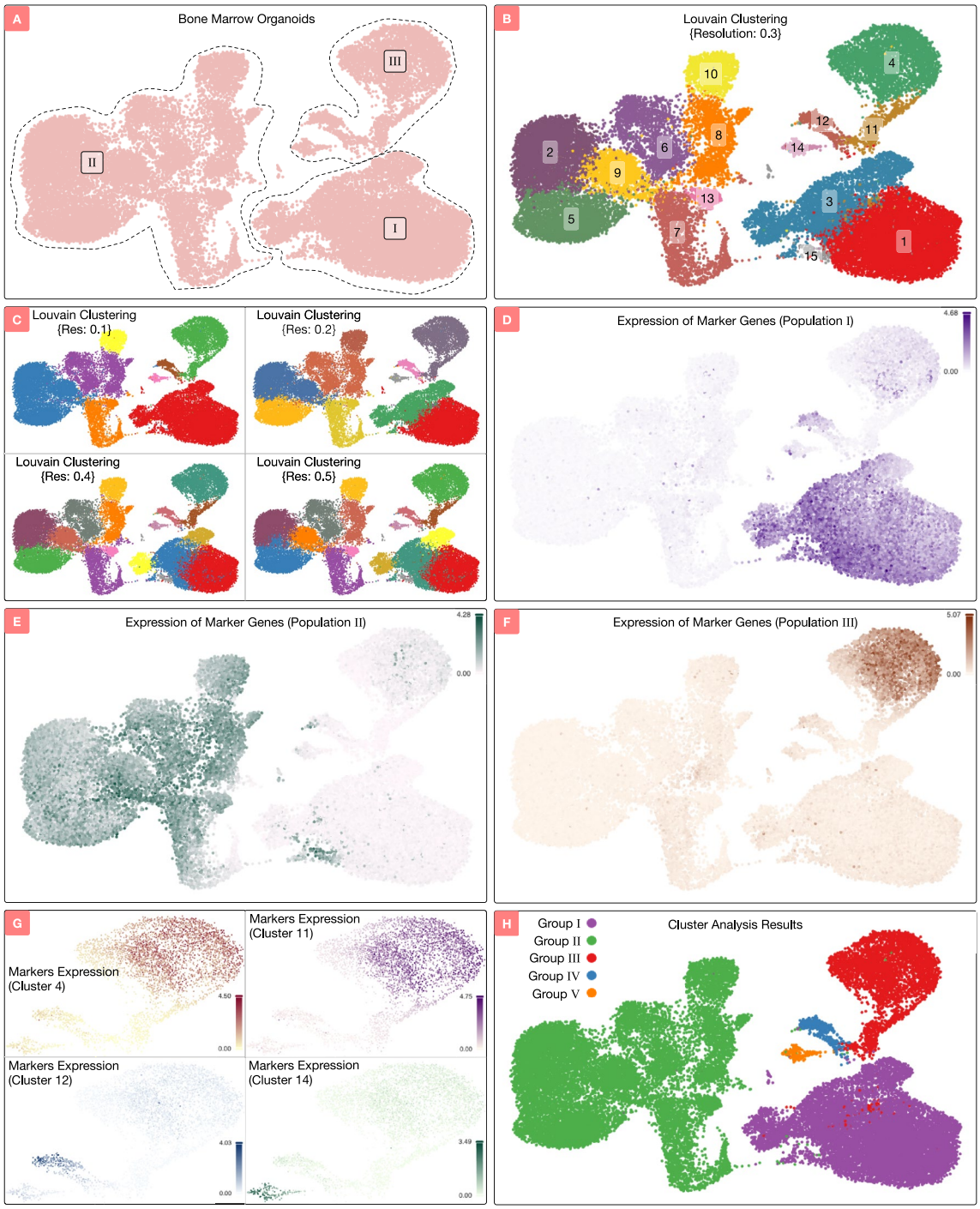

Fig. 7. Visualization, Louvain clustering, and differential expression (DE) analysis. (**A**) Transcriptome landscape visualization using UMAP. The landscape can be separated into three populations (I, II, and III) that are highlighted by the three dashed lines. (**B**) Louvain clustering using the default resolution of 0.3. (**C**) Louvain clustering results using resolutions of 0.1, 0.2, 0.4, and 0.5. The next step is to perform DE analysis and visualize the expression of the marker genes across the transcriptome landscape. (**D**-**F**) Expression of marker genes for populations I, II, and III, respectively. (**G**) Expression of marker genes for clusters 4, 11, 12, and 14 of population III. H) Final cell grouping based on visualization, Louvain clustering, and DE analysis.

Acknowledging that Louvain clustering has the potential to produce a large number of clusters, we execute the algorithm with different resolutions to explore cluster granularity and to identify robust boundaries (Fig. 7C).

Figures 7B and C show that, regardless of resolution settings, Louvain is able to separate the cells into three major populations, which is consistent with what we observe from the transcriptome landscape. Louvain with a resolution of 0.3 identifies 15 clusters, in which clusters 1, 3, and 15 correspond to population I; clusters 2, 5, 6, 7, 8, 9, 10, and 13 correspond to population II; clusters 4, 11, 12, and 14 correspond to population III. Similarly, Louvain with resolution 0.1 (panel C) identifies 8 clusters, in which cluster 1 corresponds to population I; clusters 2, 4, 5, and 6 correspond to population II; clusters 3, 7, and 8 correspond to population III. In summary, Louvain analysis results with different resolutions all separate the transcriptome landscape into three major cell populations, as we observed in Fig. 7A–C. For the next step of the analysis, we proceed with Louvain results using the default resolution of 0.3 (Fig. 7B), but all other resolutions are likely to lead to similar annotation results, as we will show in the following text.

DE analysis and cluster verification: We proceed with the Louvain clustering results using the default resolution of 0.3 (Fig. 7B). There are three major cell populations (I, II, and III). Next, we perform DE analysis to identify the marker genes of each population and visualize their expression. The goal is to verify whether we should further divide each population into smaller cell groups. We conduct three DE analyses: population I (clusters 1, 3, 15) versus others; population II (clusters 2, 5, 6, 7, 8, 9, 10, 13) versus others; population III (clusters 4, 11, 12, 14) versus others. We use the Wilcoxon rank sum test to calculate the p-values of the genes and then adjust the p-values using Benjamini-Hochberg[44]. We then use the following criteria to identify the marker genes of each population: 1) log fold change of 3 or higher, 2) FDR p-value less than 5%, 3) average expression more than 1, and 4) the difference in the percentage of cells expressing the gene between groups is at least 50%. We then visualize the expression of the marker genes.

Figures 7D–F show the average expression of the marker genes for each of the three cell populations. The marker genes of population I have high expression in the population and have negligible expression in any other populations (Fig. 7D). Therefore, we are confident that the first population consists of a single cell type. Similarly, the marker genes of population II have high expression in the population and have negligible expression in any other populations (Fig. 7E). Therefore, it is most likely that population II consists of a single cell type as well. However, the markers of the third population (clusters 4, 11, 12, and 14) have high expression only in cluster 4 and not in clusters 11, 12, and 14. This suggests that cluster 4 represents a distinct cell type from clusters 11, 12, and 14.

Consequently, we perform additional DE analyses for each cluster in population III. These are smaller groups of cells, and we could not find the markers for clusters 11 and 12 using the stringent criteria used above. Therefore, we relax the log2FC threshold to 2 or higher for cluster 11, and reduce the percentage of cells expressing the genes between groups to 40% or higher for cluster 12. Figure 7G shows the expression of marker genes in each of the four clusters. The top two panels show the expression of marker genes in clusters 4 and 11. Interestingly, the expression of marker genes in these two clusters shares very similar patterns. Therefore, we merge these two clusters together. The bottom two panels show the expression of marker genes in clusters 12 and 14, in which marker genes of each cluster are highly expressed in their respective cluster but not in other clusters. At the end, we divide the population III into three cell groups: clusters 4 and 11 together, cluster 12, and cluster 14. In summary, through visualization, Louvain clustering, and DE analysis, we identify five cell groups. Figure 7H and Table 1 show the final clusters and their respective markers.

Cell type annotation using built-in AI-based Inference: Through visualization, clustering, and DE analysis, we identify five groups of cells. Here, we aim to assign the cell groups to known cell type labels. For each of the five groups and their associated markers (Fig. 7H and Table 1), we use the built-in inference tool to search for potential cell types. Given the tissue and associated markers, the inference tool returns a list of cell types most

Group	Cluster	Marker genes
I	1, 3, 15	KIF26B, GULP1, COL5A2, COL3A1, ANTXR1, COL6A3, PDZRN3, LSAMP, SLIT2, TSPAN5, PALLD, TENM3, EDNRA, UNC5C, SEMA5A, ITGA1, VCAN, PRR16, PDGFRB, SLIT3, SPARC, PEG10, CALD1, MEST, PCOLCE, COL1A2, DLC1, SULF1, SDC2, FAM110B, COL5A1, FREM1, PTPRD, ARHGAP42, ANK3, HPSE2, BAMBI, EPS8, CSRP2, LUM, DCN, POSTN, GPC6, LRFN5, FLRT2, NID2, CDH11, COL1A1, CCDC102B, TSHZ2, FBLN1, COL6A2, COL6A1
II	2, 5, 6, 7, 8, 9, 10, 13	LAPTM5, PTPRC, CD52, S100A4, STK17B, ARHGAP15, FYB1, MCTP1, DOCK2, LST1, PIM1, IKZF1, RAB11FIP1, SYK, DOCK8, SPI1, MAP3K8, SRGN, CELF2, CHST11, ARHGDIB, CD69, LCP1, ATP8B4, PRKCB, CORO1A, GMFG, TYROBP, NCF4, RAC2, RUNX1, SAMSN1
III	4, 11	GJA4, TIE1, ADGRL4, ARHGAP29, CD34, CHRM3, HSPG2, CRIM1, CALCRL, DYSF, SCN9A, RASGRP3, MECOM, WWTR1, LDB2, LIMCH1, EMCN, VEGFC, KDR, CCSER1, AFAP1L1, LIFR, PLK2, FILIP1, GJA1, ADGRF5, PTCHD4, MYCT1, PLEKHG1, RAPGEF5, DOCK4, KLHL4, STC1, DEPTOR, PREX2, CSGALNACT1, NFIB, RGS3, GALNT18, TSPAN18, ESAM, JAM3, MMRN2, ZEB1, ABLIM1, BICD1, PTPRB, FLT1, EFNB2, DOCK9, RHOJ, CLEC14A, PPP1R13B, RNASE1, PRKCH, DLL4, CDH5, IGFBP4, PECAM1, ICAM2, PTPRM, CD93, SOX18, PLVAP, CLDN5, SHANK3, CYYR1
IV	12	L1TD1, KANK4, SLC2A1, PATJ, VTCN1, EPCAM, FIGN, ITGB6, LYPD6B, VIT, DOCK3, THRB, NDNF, TENM3, ANXA3, FRAS1, CCSER1, GABRP, EFNA5, ADAMTS19, CTNND2, ADAMTS6, PERP, KHDRBS2, COL12A1, DSP, GCNT2, SEMA3C, MET, SEMA3D, SDK1, NRK, GPC3, GRHL2, PCSK5, KANK1, SLN, SESN3, FAT3, SHANK2, KCNMA1, KIAA1217, TMTC2, KRT18, KRT8, FREM2, GPC5, MTUS2, SAMD4A, PRTG, ADAMTS18, CLDN6, CDH3, EMP2, KRT19, EPB41L3, CCBE1, GREB1L, PCDH11Y
V	14	L1TD1, SPATA6, FMN2, PDPN, ALPL, ITGA6, CACNA2D3, HERC5, COL23A1, INSYN2B, PRDM1, GMPR, RMND1, POU5F1, SLC16A10, GALNT17, SUGCT, NRCAM, EDA, FGF13, GABRA3, DMD, PCSK1N, TRPC5, RTL4, GNA14, CFAP95, CACNA1B, ASRGL1, USP28, SLC25A16, UTF1, CTNNA3, PLCE1, TMTC1, PPM1H, TMEM132D, PLBD1, TCL1A, WDHD1, FRMD6, NPAS3, HS3ST4, NETO2, ZNF66, NANOS3

Table 1. Marker genes obtained from the DE analysis. We use the following criteria to identify the genes that are differentially expressed: 1) Log2FC $\geq$ 3, 2) FDR p-value $\leq$ 0.05, 3) expression $\geq$ 1, and 4) percentage of cells expressing the genes between groups $\geq$ 50%.

likely present in the sample, structured in a cell ontology hierarchy. Table 2 displays the top predictions for each cell group, sorted in descending order of likelihood.

Our strategy for cell type assignment is to search for the label that appears most frequently in the top predictions. If there are multiple labels with the same frequency, we choose the cell type with the lowest order (more fine-grained) in the cell ontology hierarchy. The goal is to reduce the assignment error since a more fine-grained label needs more evidence. Based on this strategy, we label Group I as Mesenchymal cells because Mesenchymal appears in most predictions. Similarly, we classify Group II as Hematopoietic stem cells because the top three predictions all point to this cell type. Group III is classified as Endothelial cells, as the cell type appears four times in the five predictions. Interestingly, the AI suggests that Group IV should be assigned to Mesenchymal cells (similar to Group I). Therefore, we merge Group I and Group IV together and label them as Mesenchymal cells. Finally, group V is assigned to Mesodermal cells.

Figure 8 summarizes the complete analysis for this case study. Data visualization shows that there are three major cell populations in the transcriptome landscape (Fig. 8A). Following the analysis of Louvain clustering and expression patterns of the marker genes of each population, we determine that the dataset consists of five cell groups (Fig. 8B). Using the AI cell-type inference tool, we assign each cell group to a known cell type label with the highest likelihood (Fig. 8C). The final annotation determined by CytoAnalyst is highly similar to the annotation provided by the authors (Fig. 8D). The two annotations share 99% similarity, with the difference being that CytoAnalyst assigns cells in group IV (blue cells in Fig. 8B) to Mesenchymal instead of Epithelial cells. Endothelial cells can undergo a process called endothelial-to-mesenchymal transition (EndMT), where they acquire mesenchymal characteristics. We hypothesize that the authors of the dataset were able to distinguish between the two cell types using external evidence from flow cytometry data (e.g, cell size, morphology, etc.)[54], evidence that may not be present or visible in the gene expression data.

Case study 2: Cell type markers of skin samples

We analyze the single-cell data from Solé-Boldo et al.[55]. The data has a total of 15,457 cells from whole-skin samples of five male donors: two young donors (25 and 27 years old) and three old donors (53, 69, and 70 years old). The authors performed standard data processing, data integration, and dimension reduction using Seurat[3]. They performed cluster analysis and cell type annotation, resulting in 17 clusters (using Louvain) and 9 main cell types using known cell markers: keratinocytes, fibroblasts, macrophages/dendritic cells, T cells, vascular and lymphatic endothelial cells, pericytes, erythrocytes, and melanocytes. Next, they isolated 5,948 fibroblasts and performed functional enrichment to identify four fibroblast subtypes: secretory-reticular fibroblasts, pro-inflammatory fibroblasts, secretory-papillary fibroblasts, and mesenchymal fibroblasts. The authors also performed DE analysis to identify the markers for the clusters, cell types, and subtypes. As before, we analyzed the data in an unbiased way, ignoring the clusters and cell types reported by the original authors.

In this case study, we use the DE analysis module from CytoAnalyst to conduct four distinct DE analyses: (1) identification of cluster markers, (2) identification of fibroblast-specific markers in young samples, (3) identification of fibroblast-specific markers in old samples, and (4) comparison of fibroblast subpopulations in young samples. In the first analysis, cluster markers are identified by comparing each cluster against all other

Group	Prediction
I	1. Pericyte →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: PDGFRB, COL1A1, COL1A2, ANTXR1, PDZRN3
	2. Smooth Muscle Cell →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: CALD1, COL1A1, COL1A2, COL3A1, COL5A1, COL5A2
	3. Fibroblast →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: COL1A1, COL1A2, COL3A1, COL5A1, COL5A2, COL6A1, COL6A2, COL6A3
	4. Mesenchymal Stem Cell →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: COL1A1, COL1A2, PDGFRB, VCAN, CDH11, DCN
	5. Osteoblast →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: COL1A1, COL1A2, SPARC, POSTN, DCN
II	1. Myeloid Cell →**Hematopoietic Cell**→ Immune System Cell → Nucleated Cell → Eukaryotic Cell → Cells: CD52, SYK, SPI1, DOCK2, RAB11FIP1
	2. Dendritic Cell → Myeloid Cell →**Hematopoietic Cell**→ Immune System Cell → Nucleated Cell → Cells: CD52, DOCK2, SYK, SPI1, FYB1
	3. Neutrophil → Myeloid Cell →**Hematopoietic Cell**→ Immune System Cell → Nucleated Cell → Cells: LAPTM5, PTPRC, S100A4, RAC2, NCF4
	4. Macrophage → Myeloid Cell →**Hematopoietic Cell**→ Immune System Cell → Nucleated Cell → Cells: CD52, SYK, SPI1, DOCK2, TYROBP
	5. Monocyte → Myeloid Cell →**Hematopoietic Cell**→ Immune System Cell → Nucleated Cell → Cells: LAPTM5, PTPRC, S100A4, STK17B, ARHGAP15
III	1. **Endothelial Cell**→ Angioblast → Vascular Progenitor Cell → Mesodermal Cell → Ectoderm Or Mesoderm Derived Cell → Cells: KDR, VEGFC, PECAM1
	2. Lymphatic Endothelial Cell →**Endothelial Cell**→ Angioblast → Vascular Progenitor Cell → Mesodermal Cell → Cells: FLT1
	3. Venous Endothelial Cell →**Endothelial Cell**→ Angioblast → Vascular Progenitor Cell → Mesodermal Cell → Cells: KDR, EFNB2
IV	1. Osteoblast →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: MET, SEMA3C
	2. Adipocyte →**Mesenchymal Cell**→ Connective Tissue Cell → Somatic Cell → Eukaryotic Cell → Cells: NRK
	3. Hematopoietic Stem Cell → Hematopoietic Progenitor Cell → Hematopoietic Cell → Immune System Cell → Eukaryotic Cell → Cells: EMP2, KRT18, KRT19
V	1. Mesenchymal Stem Cell →**Mesodermal Cell**→ Ectoderm And Mesoderm Derived Cell → Somatic Cell → Eukaryotic Cell → Cells: ALPL
	2. Osteoblast → Mesenchymal Cell →**Mesodermal Cell**→ Ectoderm And Mesoderm Derived Cell → Somatic Cell → Cells: COL23A1
	3. Mesenchymal Cell →**Mesodermal Cell**→ Ectoderm And Mesoderm Derived Cell → Somatic Cell → Eukaryotic Cell → Cells: ITGA6

Table 2. Predicted cell types using the AI-based inference tool.

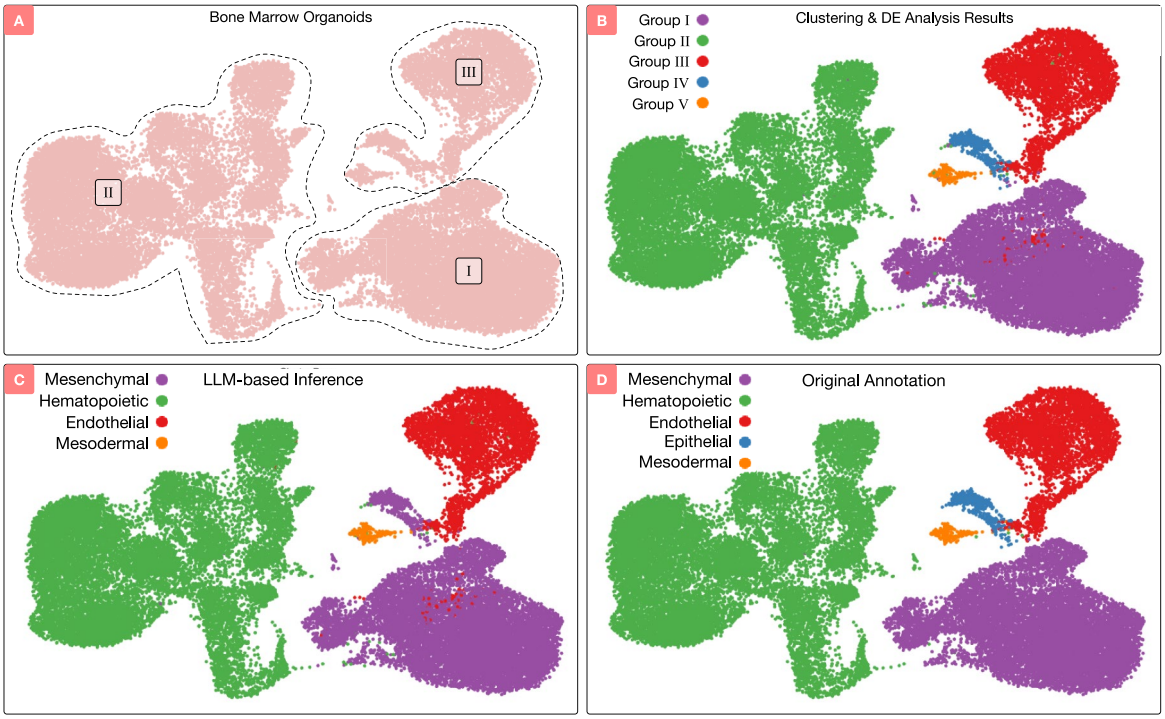

Fig. 8. Final cell type annotation results. A) Transcriptome landscape visualization. B) Final grouping using clustering and DE analysis. C) Cell types annotated by CytoAnalyst using the built-in AI-based inference tool. D) Cell types annotated by Frenz-Wiessner et al. using single-cell and flow cytometry data. The two annotations share high similarity, with the only difference being that the AI assigns group IV to Mesenchymal instead of Epithelial.

clusters. With 17 clusters in total, this results in 17 DE analyses. In the second and third analyses, we compare each fibroblast cluster against all other cells in young and old samples, respectively. With four clusters within the fibroblast population, this results in a total of 8 DE analyses. In the final analysis, we compare each fibroblast subpopulation against other fibroblast subpopulations in young samples, resulting in 4 DE analyses. For all comparisons, we use the Wilcoxon rank-sum test to calculate the p-values.

The parameter settings used for all analyses are shown in Supplementary Figures S3 and S8. The volcano plots are shown in Supplementary Figures S4–S7, and S9. The DE genes for each of the four analyses are reported in Supplementary Tables S1–S4, respectively. Overall, the results from CytoAnalyst are consistent with the published results. In the first analysis (17 DE analyses for 17 clusters), most of the markers identified by CytoAnalyst for each cluster are also confirmed by the authors (88.4%). In the second analysis, fibroblast-specific markers identified by CytoAnalyst in young samples share high similarity (91.9%) with the list of markers provided by the authors. In the third analysis, fibroblast-specific markers identified by CytoAnalyst in old samples have 91.1% similarity with the markers identified by the authors. In the fourth analysis, markers identified by CytoAnalyst have 84.7% similarity with the markers identified by the authors. Detailed analysis workflow and results are reported in Supplementary Section 3.

Case study 3: Trajectory inference of bone marrow cells

We analyze the bone marrow dataset obtained from Björklund et al.[60] in which the authors integrated the data from three different experiments[56–58]. The authors processed the data using standard Seurat protocol, including cell filtering, normalization, feature selection, dimensionality reduction, and clustering. The authors performed trajectory inference using Slingshot to identify developmental trajectories of bone marrow cells. They also used tradeSeq[61] to identify genes that change during the developmental trajectories of each lineage: Cd34 (Hematopoietic stem/progenitor cells, or HSPC), Ms4a1 (B cells), Ltf (Granulocyte cells), and Siglech (Dendritic cells). The authors published the lineages obtained for only HSPC, which can serve as a reference for our analysis. Here, we aim to reproduce the reported HSPC lineages using the same data. In addition, we also infer lineages for B cells, Granulocyte cells, and Dendritic cells.

The dataset contains integrated embeddings, clustering results, and cell type annotations provided by the authors. Here, we use CytoAnalyst to infer developmental trajectories given the provided embeddings and clusters. Specifically, we use the expression of four genes corresponding to four cell type lineages to infer the trajectories: Cd34 (HSPC), Ms4a1 (B cells), Ltf (Granulocyte), Siglech (Dendritic cells). Supplementary Figures S10, S13, S14, and S15 show the expression of these genes in the dataset, side-by-side with the clustering results. For each gene, we examine the expression pattern across clusters and manually select the start and end points for the trajectory inference. Supplementary Figure S11 shows the parameter settings we use for trajectory inference,

including embeddings, start groups, end groups, distance method, convergence threshold, etc. Supplementary Figures S12, S16, S17, and S18 show the inferred trajectories using CytoAnalyst. We also compare our results against the results reported by the authors. Supplementary Figure S19 shows that the results from CytoAnalyst match the published results, demonstrating the platform's ability to reproduce complex trajectory inference analyses. Supplementary Section 4 provides details for the complete analysis workflow and results.

Conclusions

In this article, we present CytoAnalyst, a web-based platform for single-cell data analysis that is both powerful and easily accessible. By combining state-of-the-art analytical methods with advanced visualization techniques and an intuitive user interface, the platform enables researchers of all backgrounds to perform rigorous and comprehensive single-cell analysis. CytoAnalyst's comprehensive analysis workflow includes data filtering, quality control, multi-sample integration, dimensionality reduction, cluster analysis, marker identification through DE analysis, cell type annotation, and pseudo-time trajectory inference. These analytical modules are interconnected with interactive visualization and systematic study management, enabling seamless transitions between steps while maintaining full parameter customization capabilities.

Several aspects distinguish CytoAnalyst from existing tools for single-cell analysis. First, its advanced visualization framework with multiple blending modes and interactive selection capabilities facilitates deep exploration of cellular heterogeneity. Second, the platform's robust annotation system, powered by extensive reference databases and machine learning approaches, enables reliable cell type identification. Third, the implementation of real-time collaboration features and comprehensive project sharing capabilities promotes team-based analysis and reproducible research. CytoAnalyst's scalable architecture and high-performance computing capabilities ensure it can handle the growing scale and complexity of single-cell datasets. Finally, to our knowledge, CytoAnalyst is the first single-cell analysis platform to incorporate an AI module for cell type inference.

While CytoAnalyst provides a comprehensive platform for single-cell analysis, several limitations should be acknowledged to help users make informed decisions about its applicability to their research needs. CytoAnalyst currently focuses exclusively on scRNA-Seq data and does not support scATAC-Seq, spatial transcriptomics, or single-cell proteomics. The platform accepts data in standardized formats (10X Genomics Cell Ranger[9] output and AnnData[23] objects) and assumes that basic quality control steps such as ambient RNA removal[62,63] or doublet detection[64,65] have been performed using external tools.

Regarding annotation approaches, CytoAnalyst employs a marker-based annotation workflow similar to established cell type annotation pipelines used throughout the field. Like standard manual annotation approaches in Seurat[3], SingleR[66], and other widely-used tools[67], CytoAnalyst relies on users to make informed decisions about marker gene selection, cell type assignment, and validation of annotations based on their biological expertise. The platform provides several annotation modes, including manual assignment, marker gene enrichment analysis, and AI-assisted inference, but all require user interpretation and validation. This approach is consistent with established annotation pipelines where the accuracy ultimately depends on the quality of marker genes, the researcher's domain knowledge, and careful validation rather than fully automated classification. For novel cell types or complex populations, we recommend the same validation practices used in standard workflows: literature review, additional marker analysis, and comparison with reference datasets.

CytoAnalyst's current implementation prioritizes ease of use and accessibility over computational flexibility. Advanced users who require custom algorithm implementations or non-standard analysis approaches may prefer command-line tools or programming environments like R or Python. However, our modular architecture and containerized infrastructure position the platform well for future expansion to address these limitations as the field continues to evolve.

As single-cell technologies continue advancing, CytoAnalyst will evolve to incorporate new analytical methods while maintaining its core mission of democratizing single-cell transcriptomics analysis.

Data availability

The datasets analysed during the current study are available at: https://cellxgene.cziscience.com/collections/59cd85c5-3b22-4035-b628-2a20810ad54b. https://cellxgene.cziscience.com/collections/c353707f-09a4-4f12-92a0-cb741e57e5f0. https://zenodo.org/records/15319627.

Received: 22 May 2025; Accepted: 30 July 2025
Published online: 06 August 2025

References

1. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology.* **15**(6), e8746 (2019).
2. Zappia, L., Phipson, B. & Oshlack, A. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Computational Biology.* **14**(6), e1006245 (2018).
3. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology.* **36**(5), 411–420 (2018).
4. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology.* **19**, 1–5 (2018).
5. Trapnell, C. et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology.* **32**(4), 381–386 (2014).
6. Gayoso, A. et al. A Python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology.* **40**(2), 163–166 (2022).
7. Rue-Albrecht, K., Marini, F., Soneson, C. & Lun, A. T. iSEE: interactive summarizedexperiment explorer. F1000Research. **7**:741 (2018).

8. Ouyang, J. F., Kamaraj, U. S., Cao, E. Y. & Rackham, O. J. ShinyCell: simple and sharable visualization of single-cell gene expression data. *Bioinformatics.* **37**(19), 3374–3376 (2021).
9. 10x Genomics.: 10x Genomics Loupe Browser v8.0.0. https://www.10xgenomics.com/software
10. Biomage Ltd.: Cellenics. https://www.biomage.net
11. Pereira, W. J. et al. Asc-Seurat: analytical single-cell Seurat-based web application. *BMC Bioinformatics.* **22**, 1–14 (2021).
12. Wang, Y., Sarfraz, I., Pervaiz, N., Hong, R., Koga, Y., Akavoor, V. et al. Interactive analysis of single-cell data using flexible workflows with SCTK2. Patterns. 4(8) (2023).
13. CZI Cell Science Program and Abdulla S, Aevermann B, Assis P, Badajoz S, Bell SM, Bezzi E, et al. (2025) CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. Nucleic Acids Research. ;53(D1):D886–D900.
14. Weinreb, C., Wolock, S. & Klein, A. M. SPRING: a kinetic interface for visualizing high dimensional single-cell expression data. *Bioinformatics.* **34**(7), 1246–1248 (2018).
15. Davie, K. et al. A Single-Cell Transcriptome Atlas of the Aging Drosophila Brain. *Cell.* **174**(4), 982–998 (2018).
16. Gardeux, V., David, F. P., Shajkofci, A., Schwalie, P. C. & Deplancke, B. ASAP: a web-based platform for the analysis and interactive visualization of single-cell RNA-seq data. *Bioinformatics.* **33**(19), 3123–3125 (2017).
17. Hasanaj, E., Wang, J., Sarathi, A., Ding, J. & Bar-Joseph, Z. Interactive single-cell data analysis using Cellar. *Nature Communications.* **13**(1), 1998 (2022).
18. Sethi, R. et al. ezSingleCell: an integrated one-stop single-cell and spatial omics analysis platform for bench scientists. *Nature Communications.* **15**(1), 5600 (2024).
19. Speir, M. L. et al. UCSC Cell Browser: visualize your single-cell data. *Bioinformatics.* **37**(23), 4578–4580 (2021).
20. Jiang, A., Lehnert, K., You, L. & Snell, R. G. ICARUS, an interactive web server for single cell RNA-seq analysis. *Nucleic Acids Research.* **50**(W1), W427–W433 (2022).
21. Prieto, C., Barrios, D. & Villaverde, A. SingleCAnalyzer: interactive analysis of single cell RNA-Seq data on the cloud. *Frontiers in Bioinformatics.* **2**, 793309 (2022).
22. Zhu, X. et al. Granatum: a graphical single-cell RNA-Seq analysis pipeline for genomics scientists. *Genome Medicine.* **9**, 1–12 (2017).
23. Virshup, I., Rybakov, S., Theis, F. J., Angerer, P. & Wolf, F. A. anndata: Access and store annotated data matrices. *Journal of Open Source Software.* **9**(101), 4371 (2024).
24. Hao, Y. et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology.* **42**(2), 293–304 (2024).
25. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods.* **16**(12), 1289–1296 (2019).
26. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell.* **177**(7), 1888–1902 (2019).
27. Abdi, H. & Williams, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics.* **2**(4), 433–459 (2010).
28. McInnes, L., Healy, J. & Melville, J. UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426. (2018).
29. Maaten, L. V. D. & Hinton, G. Visualizing data using t-SNE. Journal of Machine Learning Research. 9(11) (2008).
30. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports.* **9**(1), 1–12 (2019).
31. Blondel, V. D., Guillaume, J. L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment.* **2008**(10), P10008 (2008).
32. Wilcoxon, F., Katti, S. & Wilcox, R. A. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected Tables in Mathematical Statistics.* **1**, 171–259 (1970).
33. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial Reconstruction of Single-Cell Gene Expression Data. *Nature Biotechnology.* **33**, 495–502 (2015).
34. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications.* **8**(1), 14049 (2017).
35. Hafemeister, C. & Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology.* **20**(1), 296 (2019).
36. Durbin, B., Hardin, J., Hawkins, D. & Rocke, D. A variance-stabilizing transformation for gene-expression microarray data. Bioinformatics. 18 (2002).
37. Patlak, J. B. Measuring kinetics of complex single ion channel data using mean-variance histograms. *Biophysical Journal.* **65**(1), 29–42 (1993).
38. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell.* **184**(13), 3573-3587.e29 (2021).
39. Baglama, James & Reichel, Lothar. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM Journal on Scientific Computing.* **27**(1), 19–42 (2005).
40. Baglama, J., Reichel, L. & Lewis B. W. irlba: Fast Truncated Singular Value Decomposition and Principal Components Analysis for Large Dense and Sparse Matrices. R package version 2.3.5.1. Available from: https://github.com/bwlewis/irlba.
41. Kodinariya, T. & Makwana, P. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies.* **01**(1), 90–95 (2013).
42. Zhang, S., Li, X., Lin, J., Lin, Q. & Wong, K. C. Review of single-cell RNA-seq data clustering for cell-type identification and characterization. *RNA.* **29**(5), 517–530 (2023).
43. Finak, G. et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology.* **16**, 278 (2015).
44. Thissen, D., Steinberg, L. & Kuang, D. Quick and easy implementation of the Benjamini-Hochberg procedure for controlling the false positive rate in multiple comparisons. *Journal of Educational and Behavioral Statistics.* **27**(1), 77–83 (2002).
45. Hu, C., Li, T., Xu, Y., Zhang, X., Li, F., Bai, J. et al. CellMarker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scRNA-seq data. Nucleic Acids Research. 51(D1):D870–D876 (2023).
46. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A. et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783. (2024).
47. Welch, B. L. The generalization of "Student's" problem when several different population varlances are involved. *Biometrika.* **34**(1–2), 28–35 (1947).
48. Korotkevich, G., Sukhov, V., Budin, N., Shpak, B., Artyomov, M. N. & Sergushichev, A. Fast gene set enrichment analysis. bioRxiv. p. 060012 (2016).
49. Ahsan, S. & Drăghici, S. Identifying Significantly Impacted Pathways and Putative Mechanisms with iPathwayGuide. *Current Protocols in Bioinformatics.* **57**, 7–15 (2017).
50. Street, K. et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics.* **19**, 477 (2018).
51. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nature Biotechnology.* **37**(5), 547–554 (2019).
52. Hastie, T. & Stuetzle, W. Principal curves. *Journal of the American Statistical Association.* **84**(406), 502–516 (1989).
53. Mouselimis, L., Sanderson, C., Curtin, R., Agrawal, S., Frey, B. & Dueck, D. ClusterR: Gaussian mixture models, k-means, mini-batch-kmeans, k-medoids and affinity propagation clustering. R package version. 1(0) (2019).

54. Frenz-Wiessner, S. et al. Generation of complex bone marrow organoids from human induced pluripotent stem cells. *Nature Methods.* **21**, 868–881 (2024).
55. Solé-Boldo, L. et al. Single-cell transcriptomes of the human skin reveal age-related loss of fibroblast priming. *Communications Biology.* **3**(1), 188 (2020).
56. Han, X. et al. Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell.* **172**(5), 1091–1107 (2018).
57. Schaum, N. et al. Single-Cell Transcriptomics of 20 Mouse Organs Creates a Tabula Muris. *Nature.* **562**(7727), 367–372 (2018).
58. Dahlin, J. S. et al. A single-cell hematopoietic landscape resolves 8 lineage trajectories and defects in Kit mutant mice. *Blood, The Journal of the American Society of Hematology.* **131**(21), e1–e11 (2018).
59. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell Systems.* **8**(4), 281–291 (2019).
60. Björklund,Å., Czarnewski, P., Reinsbach, S. & Francis, R.: Trajectory inference using Slingshot. NBIS Workshop on Single-cell RNA-seq Analysis. https://nbisweden.github.io/workshop-scRNAseq/labs/seurat/seurat_07_trajectory.html.
61. Van den Berge, K. et al. Trajectory-based differential expression analysis for single-cell sequencing data. *Nature Communications.* **11**(1), 1201 (2020).
62. Young, M. D. & Behjati, S. SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data. Gigascience. 9(12):giaa151 (2020).
63. Fleming, S. J. et al. Unsupervised removal of systematic background noise from droplet-based single-cell experiments using Cell Bender. *Nature Methods.* **20**(9), 1323–1335 (2023).
64. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell Systems.* **8**(4), 329–337 (2019).
65. Germain, P. L., Lun, A., Meixide, C. G., Macnair, W. & Robinson, M. D. Doublet identification in single-cell sequencing data using scDblFinder. F1000Research. 10:979 (2022).
66. Aran, D. et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature Immunology.* **20**(2), 163–172 (2019).
67. Abdelaal, T. et al. A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biology.* **20**, 1–19 (2019).

Author contributions

P.B. designed and implemented the systems. D.T. and K.N. helped with case studies, tutorials, documentation, and bug fixing. S.D. contributed with specifications definition, workflow design, GUI design, and optimizing user experience. T.N. provided project guidance and oversight throughout all phases of the research. All authors participated in manuscript writing and system testing.

Funding

This work was partially supported by National Science Foundation (2343019 and 2203236), National Cancer Institute (U01CA274573), National Institute of General Medical Sciences (R44GM152152), and National Institute of Food and Agriculture (2023-67022-40041). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the funding agencies.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-14398-x.

Correspondence and requests for materials should be addressed to T.N.