OmicVerse: A single pipeline for exploring the entire transcriptome universe

Zehua Zeng [ab1], Yuqing Ma[cd1], Lei Hu[ae1], Peng Liu[a], Bowen Tan[a], Yixuan Wang[a], Cencan Xing [ab✉], Yuanyan Xiong[f✉], Hongwu Du [ab✉]

a School of Chemistry and Biological Engineering, University of Science and Technology Beijing, Beijing 100083, China

[b] Daxing Research Institute, University of Science and Technology Beijing, Beijing 100083, China.

[c] Center of Precision Medicine and Healthcare, Tsinghua-Berkeley Shenzhen Institute, Shenzhen, Guangdong Province, 518055, China.

[d] Institute of Biopharmaceutics and Health Engineering, Tsinghua Shenzhen International Graduate School, Shenzhen, Guangdong Province, 518055, China..

[e] School of Life Sciences, Westlake University, Hangzhou, Zhejiang, 310030, China.

[f] Key Laboratory of Gene Engineering of the Ministry of Education, Institute of Healthy Aging Research, School of Life Sciences ,Sun-Yat-sen University, Guangzhou, Guangdong, 510006, China

[1] These authors contributed equally to this work

✉email: cencanxing@ustb.edu.cn; xyyan@mail.sysu.edu.cn; hongwudu@ustb.edu.cn

Abstract

Single-cell sequencing is frequently marred by "interruptions" due to limitations in sequencing throughput, yet bulk RNA-seq may harbor these ostensibly "interrupted" cells. In response, we introduce the single cell trajectory blending from Bulk RNA-seq (BulkTrajBlend) algorithm, a component of the OmicVerse suite that leverages a Beta-Variational AutoEncoder for data deconvolution and graph neural networks for the discovery of overlapping community. This approach proficiently interpolates and restores the continuity of "interrupted" cells within single-cell RNA sequencing dataset. Furthermore, OmicVerse provides an extensive toolkit for bulk and single cell RNA-seq analysis, offering uniform access to diverse methodologies, streamlining computational processes, fostering exquisite data visualization, and facilitating the extraction of novel biological insights to advance scientific research.

Main

Single-cell RNA sequencing (scRNA-seq) and bulk RNA sequencing (RNA-seq) have emerged as essential techniques for exploring cellular heterogeneity, differentiation, and disease mechanisms[1-6]. These technologies facilitate numerous applications, including the conversion of bulk-seq data into single-seq analyses[7], performing differential expression analysis[8], pathway enrichment[9], gene co-expression network analysis in bulk RNA-seq[10], cell annotation[11], cell interaction analysis[12], cell-trajectory inference[13], evaluation of cell-state in gene sets, and prediction of drug response in scRNA-seq[14]. Many of these approaches harness open-source algorithms contributed by the research community [15,16].

Nevertheless, the burgeoning variety and number of omics algorithms pose challenges in selecting tools that are accurate, user-friendly and appropriate for specific analyses. Learning to use diverse algorithms often leads to computational inefficiencies, as users are required to acclimate to various system. Moreover, for analyses involving low-data quantities, researchers commonly employ web servers and the R language [17], whereas Python is the language of choice for processing large-scale datasets[18].

Integrating single-cell and bulk sequencing results can be intricate, producing complex, multi-layered data sets that challenge the exaction of meaningful biological insights. A recognized impediment in single-cell sequencing is the 'interruption' -- the omission of certain cell types due to technological constraints on the sequencing platform and interruption the trajectory of cell differentiation, such as the enzymatic lysis-related loss of podocytes and intercalated cells[19], the differentiation from HPC to podocytes was interrupted, and the filtering-induced absence of neutrophils, cardiomyocytes, neuronal cells, and megakaryocytes and the differentiation from nIPC to neurons was interrupted[20-22]. The BD Rhapsody™ single-cell platform overcomes granulocyte loss by accommodating their natural sedimentation[23]. Conversely, bulk RNA-seq of whole tissues intrinsically includes these 'interrupted' cells. Current algorithm for isolating 'interrupted' cells from bulk RNA-seq are non-existent, revealing a gap in the tools available for reconciling bulk RNA-seq and scRNA-seq data

To address these challenges, we have developed OmicVerse (https://omicverse.readthedocs.io/), a comprehensive Python library designed for transcriptomic research. OmicVerse streamlines access to a spectrum of model/algorithms for bulk-seq and single-seq analyses, enhancing computational efficiency and visual engagement. Rewritten model/algorithms and integrated different pre-processing options stem from benchmark testing[24] (Supplementary Note 1). Moreover, OmicVerse features ingle cell trajectory blending from Bulk RNA-seq (BulkTrajBlend), an algorithm specifically adept at resolving 'interruptions' in single-cell data. BulkTrajBlend employs a beta-variational autoencoder and graph neural network-based algorithm to deconvolve single-cell data from bulk RNA-seq, facilitating the identification of 'interrupted' cells within the reconstructed single-cell landscape

Results

Design concept of BulkTrajBlend and Benchmarking

The conceptualization of BulkTrajBlend draws upon prior research, proposing that Bulk RNA-seq data is a composite of scRNA-seq data through a nonlinear superposition mechanism[25,26]. Central to this notion is the implementation of the beta-variational autoencoder (β-VAE), a potent tool for approximating Bulk RNA-seq data to scRNA-seq representation[27,28]. The incorporation of β-VAE facilitates the construction of an encoder and decoder from single-cell data, traditionally characterized by unconstrained attributes.

BulkTrajBlend advances the foundational structure of autoencoders (AE) and β-VAE. These enhancements encompass (1) employing an AE to construct a Bulk RNA-seq generator analogous to real Bulk RNA-seq inspired by TAPE[29], subsequently utilizing ground truth bulk RNA-seq as input for calculating the true fraction of cells, (2) deploying unsupervised clustering to denoise and refine the outcomes of the decoder, (3) substituting the percentage of cells in the output with a secondary VAE network employing a linear estimation approach, and (4) employing a graph neural network (GNN) to sample the generated single-cell data, thereby identifying overlapping cell communities.

The methodology based on β-VAE approximates the joint distribution of data x and latent generating factors z by estimating the probability distribution $q_\theta(z|x)$ relative to the true posterior $q_\theta(x|z)$. Here, x denotes gene expression data, and z characterizes the normally distributed parameters of x post-sampling. It is noteworthy that this approximation introduces a level of noise and bias into the generated data. Consequently, unsupervised clustering is employed as a data refinement strategy to mitigate the impact of noise and enhance data robustness.

Another salient constraint of β-VAE pertains to the unconstrained nature of the decoder's output. This contrasts with the real Bulk environment, where the cellular ratios are not rigidly fixed. To address this discrepancy, a simulated Bulk environment is constructed through the sampling of single-cell data, with the procedural details outlined in the "Methods" section. This process is facilitated by a deep neural network (DNN)-based autoencoder model, wherein the simulated Bulk serves as input, the encoder's output reflects the proportions of actual cells, and the simulated Bulk constitutes the decoder's output. Mean absolute error (MAE) is adopted as the evaluation metric for both the encoder and decoder. Subsequent to model convergence, the real Bulk data is utilized as input for the AE model, with the critical requirement being the alignment of the generation, based on the best-pretrained decoder, with the real Bulk data. At this juncture, the cell proportions output by the encoder accurately mirror the cell proportions of the actual Bulk (Fig.1a).

Given that BulkTrajBlend's primary objective is to interpolate data from original scRNA-seq data, the focus shifts to the targeted extraction of cells from the generated single-cell data. Considering the inherent challenges associated with cell annotation,

the input single-cell data encompassing diverse cell types is expected to exhibit overlaps in real-world scenarios. Most of the existing community discovery (cell clustering) algorithms are non-overlapping[30], real communities are overlapping[31], and the GNN-based Neural Overlapping Community Detection (NOCD) algorithm achieves the best level in the existing baseline[32]. Using NOCD enabling the identification of overlapping cell communities. This insight is integral to the subsequent task of recovering and reconstructing cell differentiation trajectories within the single-cell sequencing data (Fig.1b).

To assess the efficacy and accuracy of BulkTrajBlend in the context of cell differentiation trajectory recovery, a rigorous benchmarking exercise is undertaken. The VAE module within BulkTrajBlend is systematically compared against alternative generative models, including conditional generative adversarial networks (CGAN) and auxiliary conditional GANs (ACGAN). The benchmarking process encompasses the evaluation of various performance metrics, encompassing the correlation of cell-type marker gene expression, marker gene similarity (quantified via cosine similarity), probability of trajectory conversion post-interpolation, and the degree of data variability following interpolation. Notably, the findings consistently underscore BulkTrajBlend's superior performance, manifesting as heightened correlations in marker gene expression, marker gene similarity, trajectory conversion probabilities, and minimal post-interpolation data variability in the generated single-cell data (Fig.1c-1f, Supplementary Note 2, Extended Data Fig.2-4).

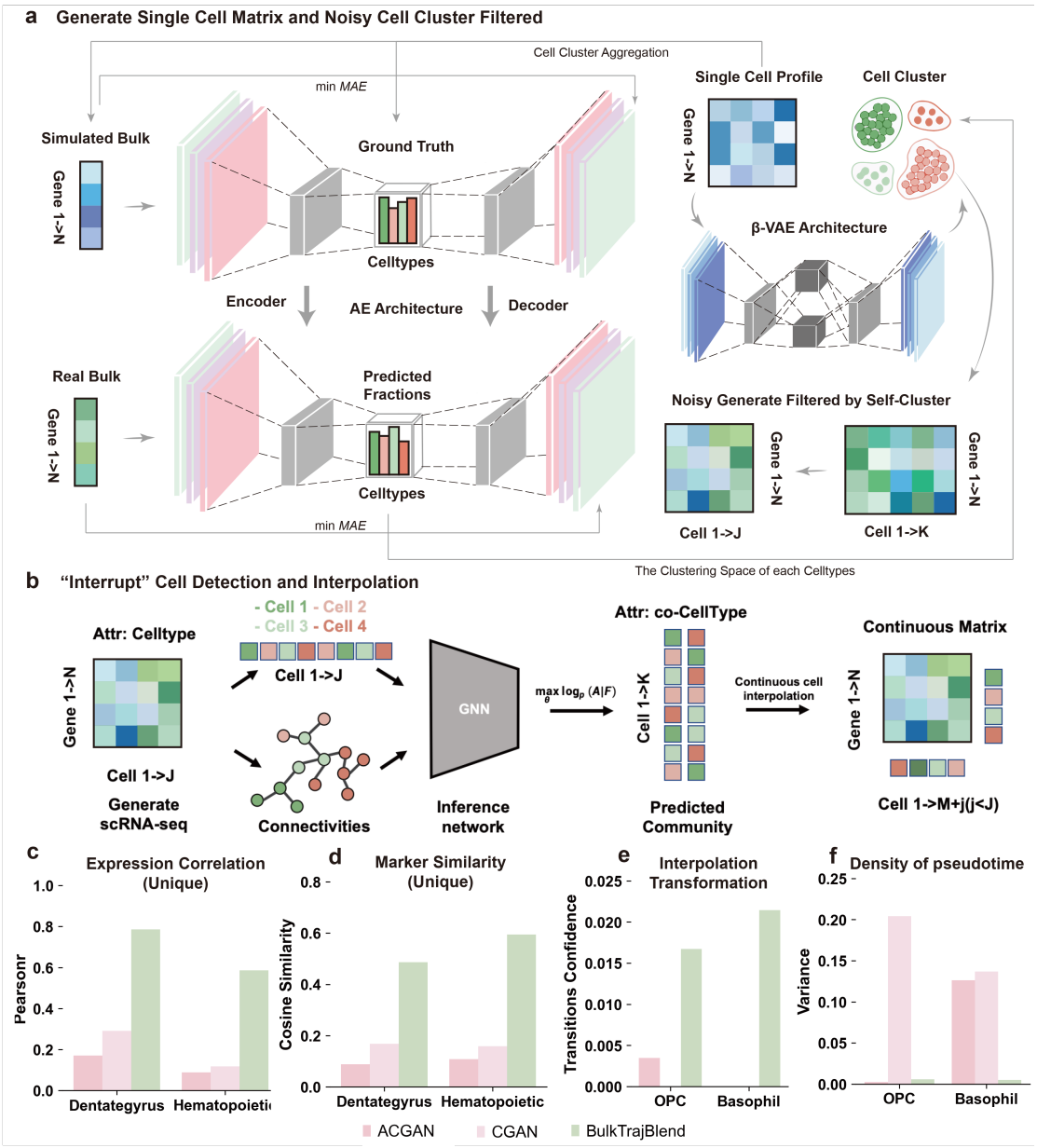

Fig 1 Architecture of the BulkTrajBlend framework.

(a) Single-Cell Profile Generation in BulkTrajBlend: This stage outlines the creation of single-cell profiles. An initial single-cell profile, representing the ground truth for cell fractions, and simulated bulk transcriptome data are fed into an autoencoder (AE). Simultaneously, real bulk transcriptome data provides the optimal input for the AE. The AE's predicted cell fractions define the clustering space of the resulting single-cell profile, which is then processed by a β-VAE to generate a profile akin to that of real bulk data. Any noise in this profile is refined using unsupervised clustering.

(b) 'Interrupt' Cell Detection in BulkTrajBlend: Here, a neighborhood graph constructed via UMAP based on the generated single-cell data identifies nodes corresponding to individual cells and demarcates distinct communities by cell type. The annotated graph is the input for a Graph Neural Network (GNN) that detects overlapping communities and identifies mixed cell types, which are then reintegrated into the original single-cell profile.

(c) Correlation Score of Cell-Type Marker Gene Expression: This component exhibits correlation scores for cell-type marker gene expression across three models within the Dentate Gyrus and Hematopoietic

datasets.

(d) Cell-Type Marker Similarity Assessment Using Cosine Similarity: This part addresses the assessment of similarities between cell-type marker genes using cosine similarity.

(e) Probability of Cell Conversion: The framework evaluates the likelihood of nIPC (neurogenic intermediate progenitor cells) becoming OPC (oligodendrocyte progenitor cells) against the backdrop of interpolated OPC cells in the Dentate Gyrus dataset, and the corresponding likelihood for the conversion of HSC (hematopoietic stem cells) to Basophil cells with interpolated Basophil cells in the Hematopoietic dataset.

(f) Pseudotime Density for OPC Cells: This final component depicts the pseudotime density of OPC cells incorporating interpolated OPC cells in the Dentate Gyrus dataset, coupled with an analogous representation for Basophil cells post-interpolation in the Hematopoietic dataset..

Impact of Varied Hyperparameters on Interpolation Performance in BulkTrajBlend

This study explores the effect of varying hyperparameter settings on the performance of BulkTrajBlend, an tool reconstruction OPC trajectories in the Dentate gyrus dataset and interpolating Basophil within the HPC dataset. We analyzed the impact of hyperparameter variations by examining five key factors: (1) the number of interpolated cells, (2) the correlation of marker gene expression between interpolated and actual cells, (3) marker gene similarity, (4) transition probabilities following interpolation, and (5) the prevalence of noise clusters.

Initially, the effect of changing the size of the input single-cell data, ranging from 1,000 to 20,000 cells, was investigated. An increase in data size resulted in higher correlations of marker gene expression and improved single-cell similarity as performed by BulkTrajBlend (Fig.2a-2b). The transition probabilities, however, were only slightly better (Fig.2c). Notably, an inverse relationship was found between the saturation of cell numbers and the frequency of noise clusters (Fig.2d).

Next, the effect of interpolation size was examined, with sizes ranging from 1 to 10 times the original number of target ‘interrupted’ cells. Marker gene correlation and single-cell similarity improved significantly within the 1-4x interpolation range, outperforming the 6-10x range. Conversely, larger interpolation sizes were correlated with a marked increase in noise clusters (Fig.2e-2h).

Counter to expectations, a detailed analysis of the number of neurons in BulkTrajBlend's hidden layer, with a range from 64 to 1024, revealed that a hidden layer with only 64 neurons exhibited the highest marker gene correlation, similarity, and transition probability for interpolated single cells, while also reducing noise cluster occurrences (Fig.2i-2l).

In conclusion, the ideal hyperparameter setting involve using the entire single-cell dataset, interpolating at a scale of 2x or 4x, and configuring a hidden layer with 64 neurons. Under these optimal hyperparameters, BulkTrajBlend effectively reconstructs the nIPC-OPC developmental flow pattern in dentate gyrus datasets and the HSC-Basophil flow pattern in hematopoietic system development datastes (Fig.2m-2n). It is

important to note that using the full single-cell dataset improves accuracy, it also significantly increases computational demands (Fig.2o).

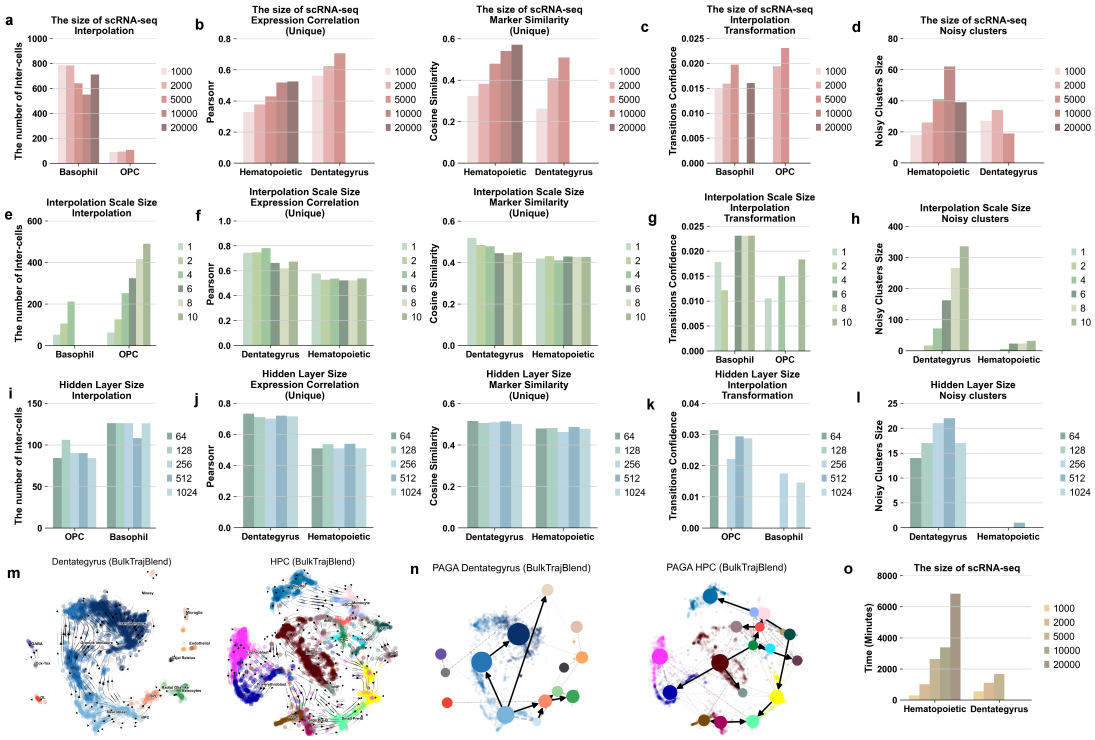

Fig 2 The systematic hyperparameter testing for 'interrupt' performance. The tests consider varying sizes of raw single-cell profiles as input in a-d:

(a) The quantity of 'interrupt' cells generated from Basophil cells in the Hematopoietic dataset and OPC cells in the Dentate Gyrus dataset, respectively.

(b) The analysis juxtaposes two aspects: on the left panel, the expression trends' correlation of marker genes between the reference and generated single-cell profiles; and on the right panel, the similarity between marker genes of the two profiles.

(c) The transition probability of the generated target cells is computed along a cellular developmental trajectory, with Basophil cells in the Hematopoietic dataset and OPC cells in the Dentate Gyrus dataset.

(d) The extent of noise clusters present in single-cell profiles, with the Hematopoietic dataset on the left and Dentate Gyrus on the right.

(e-h) The scale size of the generated target cells utilized as input is scrutinized.

(i-l) The size of neurons in the hidden layer varies as input.

(m) The flow trend of cell developmental trajectories of neurogenic intermediate progenitor cells (nIPC) are visualized on UMAP plots for the Dentate Gyrus on the left and the Hematopoietic dataset on the right.

(n) Cell state transition directed graphs within the trajectory of Partition-based Graph Abstraction (PAGA) graphs are presented for the Dentate Gyrus on the left and Hematopoietic dataset on the right.

(o) The model's runtime in relation to different sizes of raw single-cell profile inputs is illustrated.

Proficient Reconstruction of Cell Developmental Trajectories in Simulated "interruptions" Single-Cell Profiles

Our study extended beyond evaluating BulkTrajBlend's ability to reconstruct developmental trajectories in real datasets, by also examining its performance within simulated datasets. We crafted three simulated datasets with specific "interruptions": the first omitted a subset of $Ngn3^{High}$ endocrine progenitor-precursor ($Ngn3^{High}$ EP) cells in mouse pancreatic development, the second removed immature granule from mouse dentate gyrus neurons development, and a third excluded hematopoietic stem cells (HSC) mesomorphic cells from the human bone marrow development. The reconstructed developmental trajectories within these simulated "interruptions" datasets were successful (Fig.3a-3c, Extended Data Fig.5a-5c, Extended Data Fig.5g-5i).

Within the mouse pancreatic development dataset, PAGA plots illustrated a baseline probability of 0.04 for $Ngn3^{High}$ EP cells differentiating into Pre-endocrine cells. In the corresponding "interruptions" dataset, this probability was 0. BulkTrajBlend interpolation increased the probability to 0.035 (Fig.3d-3g, Extended Data Fig.6a-6c). In the mouse dentate gyrus neurons development, Granule Immature cells had baseline differentiation probability to Granule Mature cells of 0.018, while no probability was observed in simulated "interruptions" dataset. BulkTrajBlend's interpolation resulted in a probability elevation to 0.019 (Fig.3g, Extended Data Fig.5d-5f, Extended Data Fig.6d-6f). In human bone marrow development, hematopoietic stem cells stage 2 (HSC 2) cells showed a differentiation probability into monocytes of 0.082, compared to 0 in the simulated "interruptions" dataset. Following BulkTrajBlend interpolation, the probability rose to 0.079 (Fig.3g, Extended Data Fig.5j-5l, Extended Data Fig.6g-6i). Notably, the original pseudotime variability in the three datasets was preserved after interpolation (Supplement Note 3). These analyses collectively highlight BulkTrajBlend's effectiveness in accurately reconstructing authentic developmental trajectories.

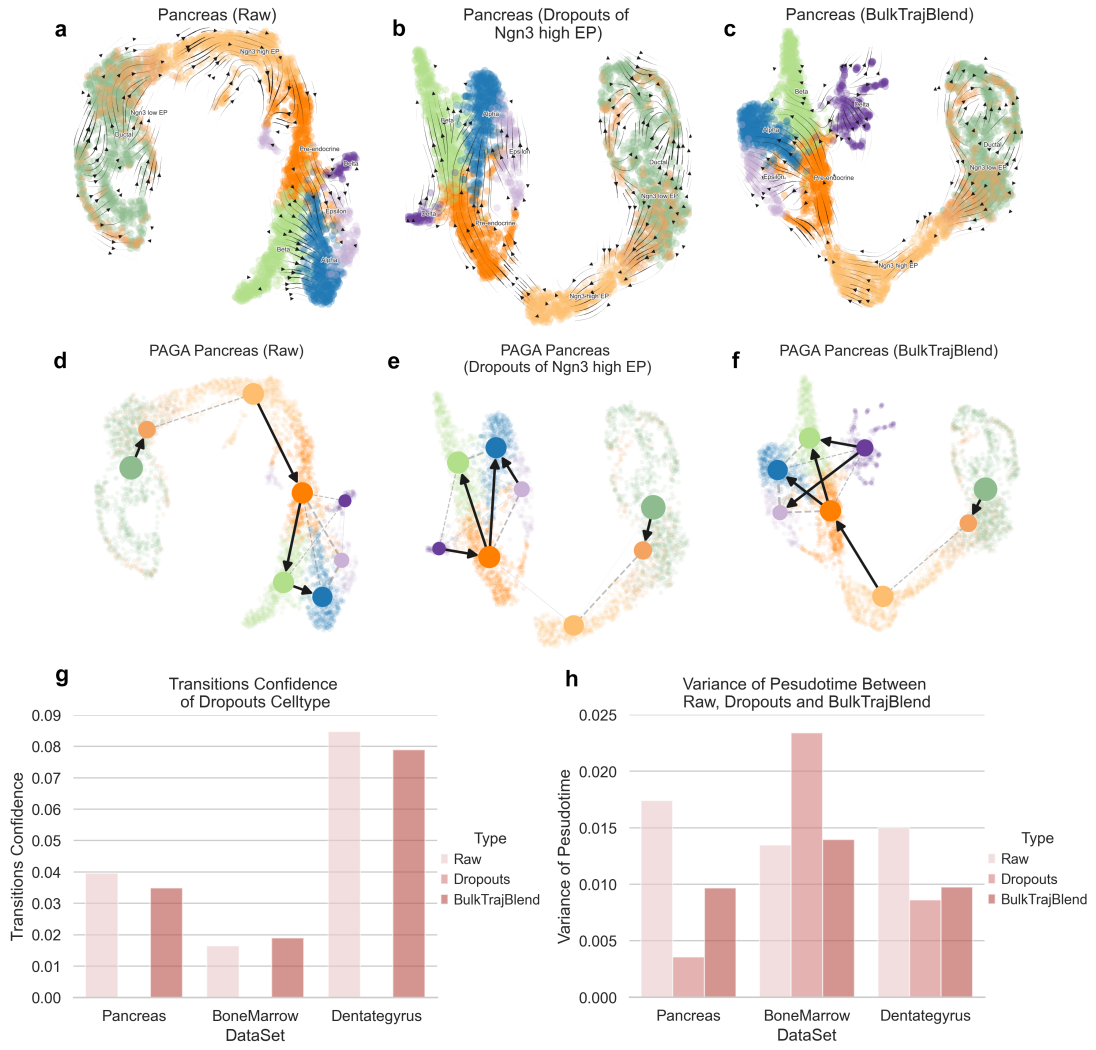

Fig 3 | Reconstruction of cell developmental trajectories in simulated "interruptions" within single-cell Profiles.

(a-c), Sequentially depicted are the raw pancreas dataset's velocity stream, the effect of simulated 'interruptions' via cell dropouts, and the refined dataset post-interpolation with BulkTrajBlend for dropout imputation as determined by pyVIA. The UMAP embedding is color-coded by cell type, consistent with the initial cluster annotations. Explained are the following cell types: $Ngn3^{High}$ EP, $Ngn3^{High}$ endocrine progenitor-precursor; $Ngn3^{Low}$ EP, $Ngn3^{Low}$ endocrine progenitor-precursor, Alpha, glucagon- producing α-cells; Beta, insulin-producing β-cells; Delta, somatostatin-producing δ-cells and Epsilon, ghrelin-producing ε-cells.

(d-f), Displayed in sequence is the directed graph overlaid on the UMAP embeddings for the raw pancreas dataset, the dataset with 'interruptions' in cell dropouts, and the dataset post-BulkTrajBlend interpolation based on pyVIA's dropout assessments.

(g), The confidence in cell state transitions as determined by pyVIA is presented for various datasets and experimental conditions. The corresponding color bars signify the methodology employed. Specifically, for the pancreas dataset with $Ngn3^{High}$ EP dropouts, the displayed confidence relates to the transition from $Ngn3^{High}$ EP to pre-endocrine cells. In the bone marrow dataset with HSC dropouts, the values indicate the transition confidence from HSC to Monocytes. Likewise, the Dentate Gyrus dataset with dropouts of Granule Immature cells shows the transition confidence from Granule immature to Granule

mature cells.

(h), The variance in pseudotime, as estimated by pyVIA, is documented across different datasets and experimental manipulations.

OmicVerse provides a comprehensive analysis platform for Bulk RNA-seq data.

The OmicVerse platform provides a sophisticated environment for the analysis of Bulk RNA-seq data. Bulk RNA-seq is an established method for investigating the transcriptome of combined cellular samples, tissue or biopsies[6]. It probes gene expression, isoform variations, alternative splicing, and single-nucleotide polymorphisms, unveiling critical biological information such as copy number variations, microbial contamination, transposable elements, cell-types deconvolution, and neoantigens. Advances in bioinformatics have enhanced the ability to reveal these hidden dimensions in Bulk RNA-seq data, expanding its analytical applications.

OmicVerse integrates an extensive collection of Bulk RNA-seq analysis algorithms, previously developed mostly in R but now increasingly in Python, to promote their use and interconnectivity[33], to promote their use and interconnectivity. Our integration enhances the existing repertoire of analysis algorithms catering to single-cell, spatial transcriptomics, as well as machine learning and deep learning models[34].

The platform hosts a comprehensive assortment of Bulk RNA-seq algorithms, including pyComBat[35] for batch correction, pyDEG for differential expression analysis using Deseq2[36], t-test, and Wilcoxon tests, pyPPI for protein-protein interaction network using STRING web API[37], pyWGCNA for gene co-expression network[38] , pyGSEA for gene set enrichment analysis[39], and pyTCGA for The Cancer Genome Atlas (TCGA) data analysis, complete with survival analysis (Fig.4a).

To evaluate the OmicVerse's analytical pipeline, we analyzed Alzheimer's disease (AD) data, beginning with pyDEG to identify differential expressed genes between AD patients and controls, highlighting the top 10 foldchange genes. Then, we executed Gene Set Enrichment Analysis at the gene level using pyGSEA, ordering genes according to p-values derived from pyDEG's differential expression analysis. We further built a co-expression network from the top 5000 genes exhibiting the highest absolute median difference (MAE), selecting the most differential expression module for visualization (See Supplementary Note 4 for Methods).

OmicVerse's workflow simplifies Bulk RNA-seq analyses with minimal coding required (Fig.4b). Parameter adjustments may enhance visual outputs. Our analysis revealed 62 genes differentially expressed in AD--52 upregulated and 10 downregulated. Box plots showcased the most altered genes (Fig.4c-4e) .And Gene Set Enrichment Analysis exposed over-represented pathways pertinent to Alzheimer's, consistent with established literature (Fig.4f-4g). Moreover, we refined our focus to the most variable genes from the top 5000, discerning 12 modules through pyWGCNA at 5 soft threshold. Notably, modules 4 and 5 showed the highest rates of differential gene

expression, with module 5 containing APP proteins. Further probing of these modules gives insight into their network connectivity (Fig.4h-4j).

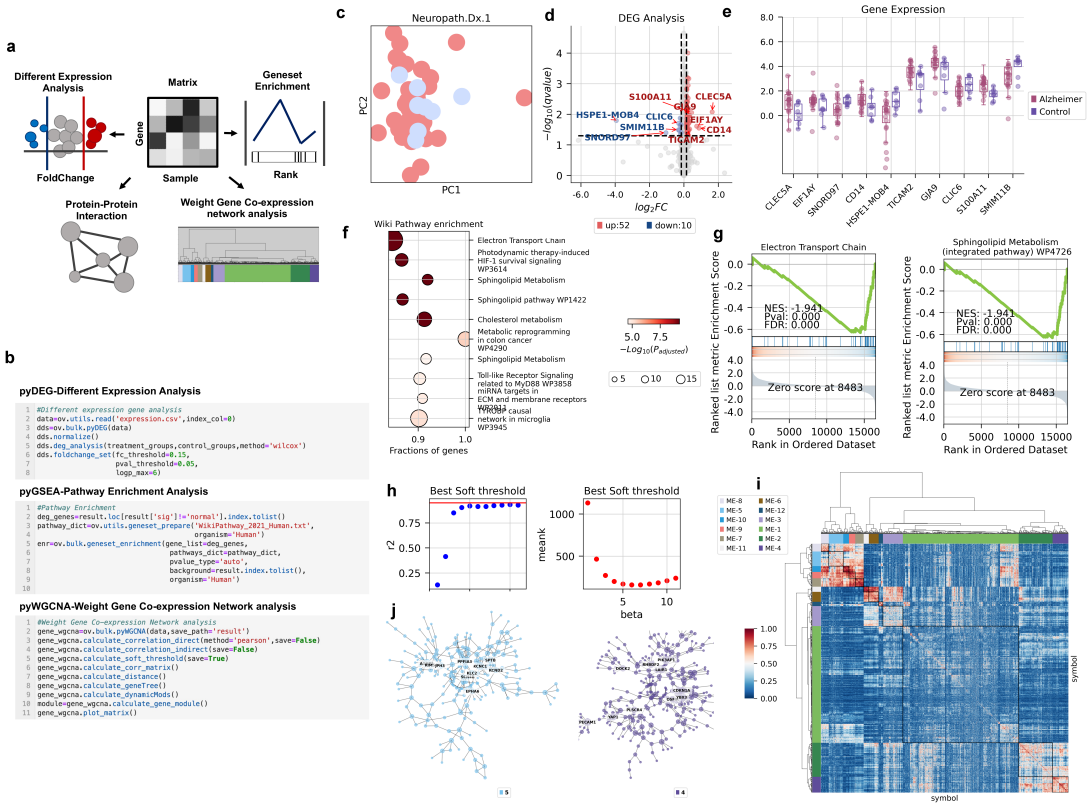

Fig 4 A comprehensive overview of Bulk RNA-seq data analysis utilizing OmicVerse.

(a) A graphical depiction showcases various analyses: differential expression analysis (pyDEG), gene set enrichment analysis (pyGSEA), protein-protein interaction analysis (pyPPI), and weighted gene co-expression network analysis (pyWGCNA).

(b) A code snippet demonstrates how to import data and execute pyDEG, pyGSEA, and pyWGCNA, incorporating continuous covariates.

(c) Principal Component Analysis (PCA) embeddings characterize samples within Alzheimer's and control groups.

(d) A volcano plot highlights differentially expressed genes; those upregulated are marked in red, while downregulated genes are indicated in blue.

(e) A box plot reveals the top 10 genes with the most significant fold change between Alzheimer's and control groups.

(f) WikiPathways enrichment results are visualized, with dot size correlating to the gene count for each function and color intensity reflecting p-value significance – darker hues indicate higher pathway enrichment.

(g) Gene set enrichment analysis (GSEA) is executed using WikiPathways gene sets, with enrichment scores and p-values derived from a weighted two-sided Kolmogorov–Smirnov-like statistic and normalized for gene set size, producing the Normalized Enrichment Score (NES).

(h) The optimal soft threshold is determined, where the horizontal axis represents the soft threshold gradient, the left vertical axis corresponds to the scale-free fit index (with higher values preferred), and the right vertical axis reflects the average node connectivity (with lower values preferred).

(i) A gene clustering dendrogram illustrates dissimilarity based on topological overlap, combined with module color assignments. Consequently, twelve co-expression modules are identified, each displayed in a distinct color. An accompanying heatmap depicts the correlation among the 5,000 genes within each module.

(j) Modules 4 and 5, which are scale-free networks, are shown where each node represents a gene. The node size corresponds to gene connectivity, and color denotes the module affiliation, with the five most central genes in each module labeled.

OmicVerse provides a versatile multifaceted framework for Single-Cell RNA-Seq Analysis

Single-cell RNA-seq is a powerful high-throughput technique that enables the measurement of gene expression patterns and cell types at the single-cell level. It has become an crucial technique for delineating cellular heterogeneity, differentiation, and disease mechanisms, particularly within cancer research. scRNA-seq unravels tumor cell diversity and tracks tumor progression to preempt cellular deterioration[40]. The breadth of scRNA-seq data analysis facilitated by OmicVerse includes cell annotation, examination of cell interactions, trajectories inference, states evaluation within gene sets, and drug responses prediction[41]. The framework supports Anndata-standardized data processing for integrated downstream analysis and benefits from benchmarked data transformations[24]. Preprocessing methods in OmicVerse feature optimal logarithmic transformation with pseudo-count addition, principal-component analysis (PCA), and Pearson residual normalization. For visualizing reduced dimensions, it employs GPU-accelerated Uniform Manifold Approximation and Projection (UMAP) through pymde[42].

Incorporating a suite of state-of-the-art scRNA-seq algorithms, OmicVerse's integrated toolset includes pyHarmony[43], pyCombat[35], scanorama[44] for batch correction, pySCSA[45], updated with CellMarker[46] 2.0 and CancerSEA[47] for enhanced cell-type annotation; CellPhoneDB[12] for cell-cell interactions analysis pyVIA[13] for trajectory inference; AUCell for geneset score evaluates based on Area Under the Curve[48]; and scDrug for drug prediction[14] (Fig.5a). The OmicVerse framework also introduces SEACells for metacell analysis, effectively minimizing data noise[49]. Importantly, the data format input for all the aforementioned methods is consistent, enabling users to conduct analyses using Anndata, with significantly improved visualization for more elegant results. OmicVerse's user-friendly nature and straightforward application are exemplified in Fig.5b.

Illustrating Omicverse's practical application in scRNA-seq, we analyzed a colorectal cancer (CRC) dataset, emphasizing the tumor microenvironment (TME) cell atlas integration[50,51]. Beginning with cell automatical annotation via pySCSA, the results showed high concordance with manual annotations (Fig. 5c) , and the f1_score reached 0.856, attesting to OmicVerse's annotation prowess(Fig. 5d). Employing AUCell, we confirmed the expected signaling pathway enrichment in cell-specific receptor pathways: B-cell receptor signaling pathway exhibited prominence in B cells, while the T-cell receptor signaling pathway was most pronounced in T cells and NK cells (Fig.5e). In addressing the sparsity inherent in previous CRC single-cell data analysis and enhancing resolution and depth, we harnessed SEACells to extract metacells from the scRNA-seq data. After 39 epochs, the metacell aggregation iteration converged, attaining high cell purity of 0.98, with compactness and separation values closely approximating 0 (Fig.5f, Extended Data Fig.7a-7c). The SEACells algorithm enhanced cell type differentiation, with the signal intensity for receptor pathways

significantly accentuated (Extended Data Fig.7d).

Further, we traced epithelial-to-cancer cell differentiation trajectories using pyVIA and annotated cancer cell types within the epithelial population with pySCSA, delineating distinct pathways including EMT and Metastasis. This provided deep insights into cancer progression (Fig. 5g). By commencing the trajectory with stemness as the starting point, we delineated the pseudotime trajectory of cancer cell differentiation, revealing three distinctive directions: EMT-Differentiation, Metastasis, representing two stages in the transition from epithelial cells to cancer cells. This analysis furnished invaluable insights into the dynamics of cancer evolution. In a parallel approach, metacells within the epithelial cell subpopulation were subjected to further aggregation analysis. Due to the inherent similarities among epithelial cells, the average cell purity of the metacells obtained reduced to 0.9, while compactness and separation values remained in close proximity to 0 (Extended Data Fig.7e-7f). Consequently, we extrapolated the metacells of epithelial cells into trajectories, revealing that EMT-differentiation and Metastasis served as the two primary differentiation pathways, aligning with the analysis conducted on all cells (Extended Data Fig.7g-7i).

Finally, to investigate the interaction network between epithelial cells and other TME cells, we established a CRC cell communication network using CellPhoneDB (Fig. 5h). The analysis encompassed immune cells, including B-cells, T-cells, NK-cells, and plasma cells, exploring their interactions with eight subtypes of epithelial cells. The analysis unveiled PPIA-BSG and LTB-LTBR as recurrent ligand-receptor pairs mediating the recognition of cancer epithelial cells by immune cells (Fig. 5i). Notably, PPIA-BSG and LTB-LTBR have been linked to a positive correlation in various cancers and associated with poor prognosis[52,53]. OmicVerse's data harmonization significantly streamlines this comprehensive analysis, enabling researchers to delve into personalized explorations as outlined in our detailed tutorial (Refer to Supplementary Note 5 for the Methods).

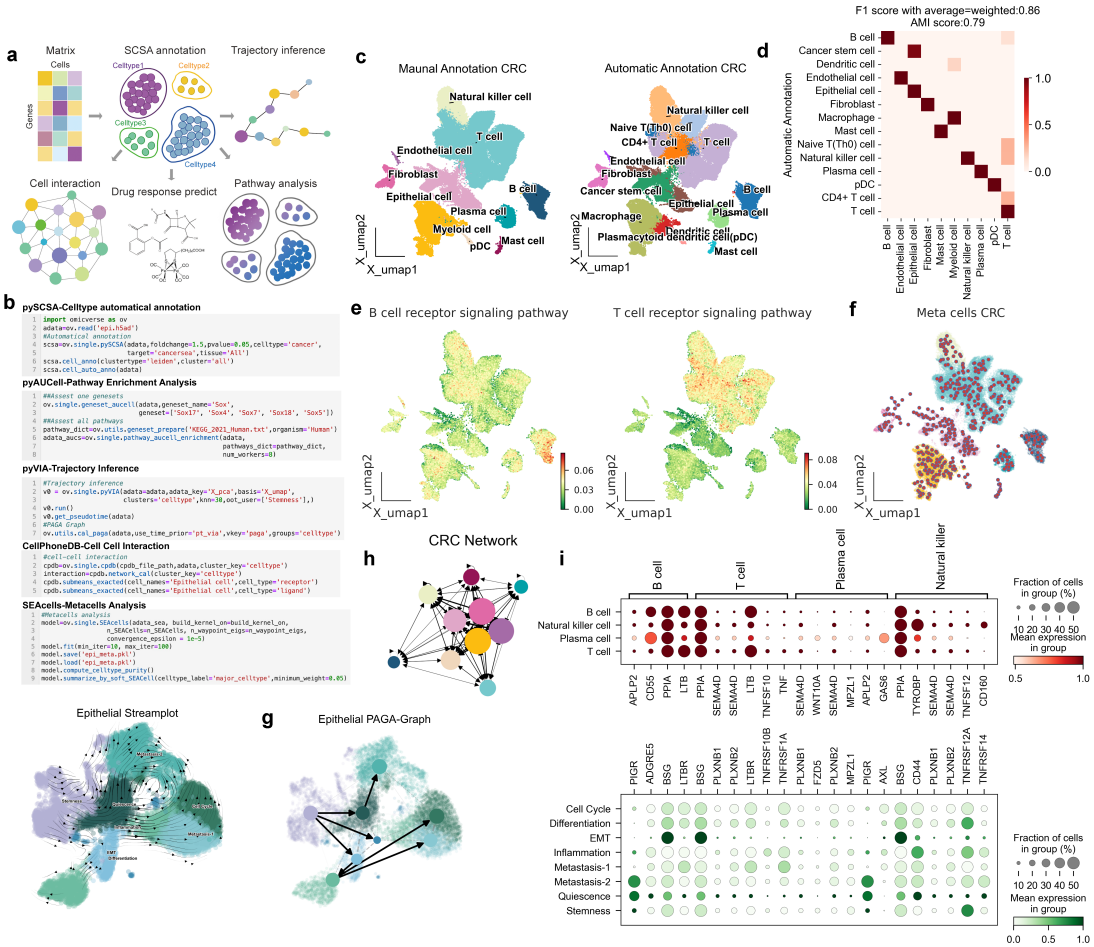

Fig 5 OmicVerse a comprehensive analytical platform for single-cell RNA-seq analysis.

(a) A graphical overview highlights crucial analysis modules: cell type annotation (pySCSA), cellular interactions (CellPhoneDB), trajectory inference (pyVIA), pathway analysis (AUCell), and drug response prediction (scDrug).

(b) An example code snippet illustrates the process for loading data and conducting analyses using pySCSA, CellPhoneDB, pyVIA, AUCell, and SEACells, with the inclusion of continuous covariates.

(c) UMAP plot visualizes single-cell RNA sequencing (scRNA-seq) data from colorectal cancer (CRC) patients. The plot contrasts manual cell type annotations, shown in the left panel, with automatic annotations depicted in the right panel.

(d) The concordance between manual and pySCSA-generated annotations is presented in a row-normalized confusion matrix.

(e) Pathway enrichment within CRC cells is elucidated in a UMAP visualization, with the left side indicating B cell receptor signaling and the right side detailing T cell receptor signaling, as analyzed by AUCell.

(f) Metacell composition within the CRC dataset is revealed in a UMAP plot.

(g) Epithelial cell subpopulations in CRC are displayed in a UMAP plot; automated annotations by pySCSA are demonstrated on the left, complemented by a cell state transition directed graph derived from a Partition-based Graph Abstraction (PAGA) trajectory on the right.

(h) CellPhoneDB computes an interaction network between CRC cell types, offering insights into intercellular communication.

(i) Scaled mean expression levels of genes that code for interacting ligand-receptor proteins, identified

by CellPhoneDB, are shown in dot plots to underscore the supporting interactions between immune and epithelial cells.

OmicVerse performed multi-omics analysis with MOFA and GLUE

Single-cell sequencing advancements enable the investigation of biological systems across different tissue levels. A key element in scRNA-seq is understanding the impact of chromatin accessibility variation, which is quantified by Single-cell sequencing assay for transposase-accessible chromatin (scATAC-seq). The conjoined analysis of scATAC-seq and scRNA-seq data is critical for unravelling transcriptional regulatory complexities. While scNMT-seq can capture both modalities simulaneously, obtaining unpaired data from identical tissues is more common[54]. Addressing this disparity, Graphical Linkage Unified Embedding (GLUE) offers a Graphical Linkage Unified Embedding solution for integrating unpaired data[55], and Multi-Omics Factor Analysis (MOFA) elucidates the variations within omics data[56]. OmicVerse utilizes both GLUE and MOFA to reveal transcriptional regulatory dynamics.

Within OmicVerse, the novel GLUE_pair alhorithm leverages the pearson correlation coefficient to compute cell similarity between scRNA-seq and scATAC-seq base on embedding from GLUE (Fig. 6a). The accuracy of GLUE_pair is verified using the Adjusted Rand Score (ARI) to confirm cell type congruence post-normalization. For analysis of paired cell modalities, OmicVerse applies MOFA's core algorithm, simplifying ensuing data analysis and visualization tasks (Fig. 6a). , all achievable with minimal coding (Fig. 6b).

Demonstrating the integration of GLUE and MOFA, we analyzed simultaneous single-nucleus RNA-seq (snRNA-seq) and single-nucleus ATAC-seq (snATAC-seq) data from cortical regions of Alzheimer's disease patients. Our analysis of aligned cell types uncovered consistent patterns indicative of common cellular states (Fig. 6c-6d). From a random subset of 5,000 paired cells, MOFA unveiled 13 factors (Fig. 6e-6f). The initial six factors accounted for RNA-related variance, while the second for ATAC-related variance. The interplay among these factors and cell types revealed significant associations: EX-signature with Factor 1, PER.END-signature with Factor 5, ASC-signature with Factor 2, MG-signature with Factor 3, and INH-signature jointly detailed by Factors 6 and 4. Additionally, gene weights for each factor uncovered genes with the most considerable influence on their respective signatures (Refer to Supplementary Note 6 for the Methods, Extended Data Fig 8a-8c).

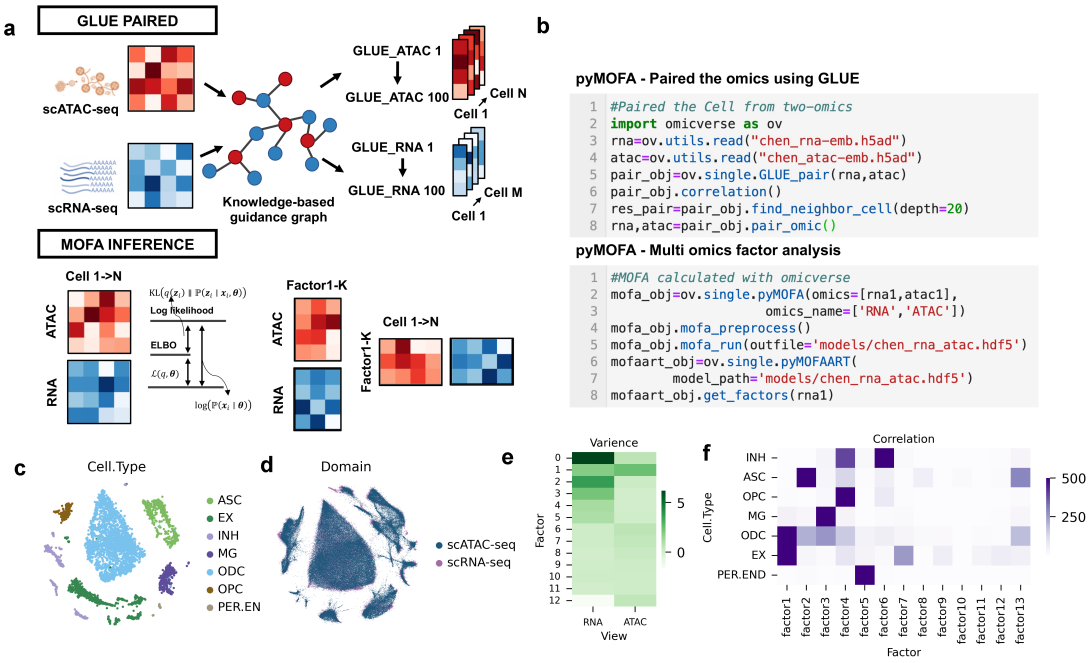

Fig 6 | The integration of multi-omics data analysis by OmicVerse, utilizing both MOFA and GLUE:

(a) The representation includes a graphical model of cell type correlations using GLUE, alongside an illustration of cell variance captured by MOFA, as indicated by the Evidence Lower Bound (ELBO).

(b) A sample code snippet is provided for the import and processing of data via pyMOFA-pair and pyMOFA-object tools.

(c) A UMAP plot showcases the distribution of cell types identified in scRNA-seq data from patients with Alzheimer's Disease.

(d) Integrated cell embeddings from various omics layers are displayed in UMAP visualizations, with color-coding reflecting the respective omic strata.

(e) A heatmap illustrates the percentage of variance accounted for by each factor (displayed as rows) across different omics datasets.

(f) Another heatmap exhibits the results of correlation analyses between cell types and the identified factors.

Discussion

The innovative amalgamation of the variational autoencoder and graph neural networks culminated in the creation of the BulkTrajBlend framework, designed for the deconvolution of scRNA-seq data within Bulk RNA-seq and the elucidation of precise cell-specific developmental trajectories in scRNA-seq. This framework demonstrates significant accuracy and robustness, due in large part to the novel integration of the topological overlap community in graph neural networks, which skillfully addresses the potential bias introduced by unsupervised clustering in the single cell data outcomes.

A conceptual parallel exists between back-calculating cell proportions in Bulk RNA-seq from scRNA-seq and using Bulk RNA-seq as a scaffold for interpolating scRNA-seq. However, the latter is inherently more challenging due to the need to accurately interpolat inadequate target cell type. While numerous single-cell generators perform well in generate scRNA-seq data, the incorporation of unknown information remains an intrinsic challege. For example, scDesign3 is a proficient statistical simulator that creates realistic single-cell data by learning interpretable parameters from actual scRNA-seq data. Nevertheless, reconstructing cell developmental trajectories often requires elusive parameters, which necessitates leveraging known data from Bulk RNA-seq[57]. Hence, BulkTrajBlend is meticulously crafted based on the principles of scDesign3[57] and scGen[28], with the state space and parameters informed by Bulk RNA-seq. Notably, cell categorization in the resulting single-cell data often relies on unsupervised annotation. By introducing GNN, BulkTrajBlend effectively reduces resolution-dependent issues linked to unsupervised clustering.

While BulkTrajBlend can efficiently extract the state space of cells from Bulk RNA-seq and interpolate the original scRNA-seq data, this interpolation relies on the selection of the reference scRNA-seq versus the reference Bulk RNA-seq data. We suggest that users can adopt an additional comprehensive single-cell profile to train BulkTrajBlend and then perform interpolation of their data, greatly avoiding generating BulkTrajBlend without information about the target cells.

Upon devising the interpolation algorithm for Bulk RNA-seq in scRNA-seq, it became apparent that a unified Python-based framework was comprehensive dual analysis of these platforms was missing. To fill this void, we developed OmicVerse, seamlessly integrating single-seq and bulk-seq. OmicVerse introduces a specialized analysis object for each omics layer, facilitating streamlined analysis and ensuring an intuitive user experience. OmicVerse not only has a well-established scRNA-seq ecosystem like Seurat, which complements Scanpy, but also has a unique Bulk RNA-seq ecosystem, thus offering a consistent and user-friendly interface (Supplement Note 7).

As an integrated framework for both Bulk and single-cell RNA-seq analysis, OmicVerse offers a suite of analytical tools that include, but not limited to:

1) Bulk RNA-seq: OmicVerse provides comprehensive functionalities, including multi-sample integration, batch effect correction, differential gene expression analysis, gene set enrichment analysis, protein interaction network construction, the identification of gene co-expression modules, and TCGA database preprocessing.

2) Single-cell RNA-seq: OmicVerse offers robust features, including multi-sample quality control, batch effect removal and integration, automated cell type annotation (with multiple databases support) and migration annotation, cell type and gene set enrichment analysis, developmental trajectory reconstruction, metacell identification, cellular communication network analysis, and drug response prediction. It also covers scATAC-seq integration and multi-omics analysis, inherently linked to RNA-seq.

3) Bulk RNA-seq to scRNA-seq: OmicVerse augments the deconvolution of Bulk RNA-seq, cell proportions estimation, interpolation the scRNA-seq data and the recovery of developmental trajectories within scRNA-seq. Acting as a critical bridge in the transition from Bulk to single-cell RNA-seq.

The OmicVerse documentation provides a detailed Application Programming Interface (API) reference for each algorithm, coupled with tutorials that clarify their functions, limitations, and synergies with other bulk and single-seq analysis tools. These resources are accessible via Google Colab, offering a free computational workspace for pipeline examinations. OmicVerse also has comprehensive developer documentation that makes it easy for users to add tools to the ecosystem following a consistent development logic.

Our primary goal was to foster an ecosystem replete with visually engaging and insightful visualizations, fully integrated within the Python programming environment. OmicVerse allows users to perform extensive transcriptome analysis using a singlular programming language, tapping into the collective machine-learning knowledge and models within the Python community. We anticipate that OmicVerse will continue to grow, with updates introducing new algorithms, features, and models. Ultimately, OmicVerse aims to act as a driving force for the bulk and single-seq community, encouraging the prototyping of new models, establishing the standards for RNA-omics analysis, and expanding the potential for scientific exploration.

Methods

Methods for BulkTrajBlend

BulkTrajBlend is primarily designed to address the issue of "interrupt" cells in single-cell data, making the inference of developmental or differentiation trajectories continuous. To achieve this goal, we designed BulkTrajBlend to generate potential "missing" cells from bulk RNA-seq data for inferring pseudo-temporal cell trajectories. This process consists of the following four steps (where communities represent cell types): (1) Cell fraction calculation: Using AE to construct a similuar bulk RNA-seq generator and ground truth bulk RNA-seq as input to calculate the true fraction of cells. (2) Generation of single-cell data: Using beta-VAE, bulk RNA-seq data is transformed into single-cell data, where each cell represents a node. (3) Computation of single-cell neighborhood graph: The UMAP method is employed to compute the neighborhood graph of the single-cell data, resulting in an adjacency matrix A. (4) Community detection and generation of overlapping cell communities: A Graph Neural Network (GNN) is utilized to identify overlapping cell communities within the single-cell data, generating an affinity matrix F. (5) Community trajectory inference: By incorporating the overlapping cell communities of target cells, the inference of community trajectories in the original single-cell data is improved.

(1) Cell fraction calculation:

To estimate the proportion of cells in Bulk RNA-seq, we first annotated the single-cell data with cell types and summed the gene counts of single cells of different cell types by cell to obtain an $N * M$ matrix, where M represents the number of cell types and N represents the number of genes. We define this $N \times M$ matrix as the simulated Bulk RNA-seq cell type matrix, and then we sum M columns of each row to get the simulated Bulk RNA-seq $B_{simulated}$, and we input the simulated Bulk RNA-seq into the self-encoder. In the self-encoder, we define the output of the encoder as T, and we make T close to $\frac{Number\ of\ the\ cell}{Number\ of\ all\ cells}$, i.e., Cell Fraction, by training AE. we then define the output of the generator as G and we make G and $B_{simulated}$ close to each other by MAE as an evaluation. After training the optimal AE, we change the input to real Bulk RNA-seq $B_{groundtruth}$, at which time the output of the encoder, T, is the Cell Fraction corresponding to real Bulk, which we use as the range of the generator space for the subsequent beta-VAE.

(2) Generation of single-cell data：

Given a set $D = X, V, W$, where $x \in X$ represents gene expression vectors, $v \in V$ represents cell type proportions, satisfying $logp(v|x) = \sum_k l\,ogp(v_k|x)$,where $v \in R^K$; and $w \in W$ represents conditionally correlated generative factors . We assume that gene expression vectors x are generated by a real-world simulator S, with the corresponding generative factors as input, i.e., $p_\theta(x|v, w) = S(v, w)$, where θ represents the generative model parameters.

We aim to develop an unsupervised deep generative model that, using only the samples

of X, learns the joint distribution of the data x and a set of latent variables z ($z \in \mathbb{R}^M$, where $M \geq K$) for generating observed data x, i.e., $p_\theta(x|z) \approx p(x|v,w) = S(v,w)$. However, the marginal likelihood $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$ required for evaluation and differentiation is intractable, making it difficult to compute or derive the true posterior density $p_\theta(z|x)$.

To address this problem, we approximate the true posterior distribution $p_\theta(z|x)$ with an approximate posterior distribution $q_\phi(z|x)$ that is easier to compute. Our goal is to ensure that the inferred latent variables $q_\phi(z|x)$ capture the generative factors v in a disentangled manner. A disentangled representation implies that individual latent units are sensitive to variations in a single generative factor while being relatively invariant to variations in other factors. In a disentangled representation, knowledge of one factor can be generalized to new configurations of other factors. The conditionally correlated generative factors w can remain entangled in a separate subset of z and are not used to represent v.

To achieve this, we minimize the KL divergence between the approximate posterior and the true posterior:

$$\mathcal{KL}(q_\phi(z|x)||p_\theta(z|x)) = -\sum_z q_\phi(z|x)\log\left(\frac{p_\theta(x,z)}{p_\theta(x)q_\phi(z|x)}\right)$$

$$= -\sum_z q_\phi(z|x)\log\left(\frac{p_\theta(x|z)q_\phi(z|x)}{p_\theta(x)}\right)$$

$$= -\sum_z q_\phi(z|x)\log\left(p_\theta(x|z)q_\phi(z|x)\right) + \log(p_\theta(x))$$

Here, $\mathcal{KL}(q_\phi(z|x)||p_\theta(z|x))$ is the variational lower bound and can be written as:

$$\mathcal{L}(\theta,\phi;x) = \sum_z q_\phi(z|x)\log(p_\theta(x|z)) - \mathcal{KL}(q_\phi(z|x)||p_\theta(z))$$

We introduce a constraint to shape the inferred posterior $q_\phi(z|x)$ and match it with a prior $p_\theta(z)$ that controls the capacity of the latent information bottleneck. We set the prior as an isotropic unit Gaussian, $p(z) \sim \mathcal{N}(0,I)$. The constrained optimization problem can be written as:

$$\max_{\phi,\theta}\mathbb{E}_{q_\phi(z|x)}[\log(p_\theta(x|z))] \quad \text{s.t.} \quad \mathcal{KL}(q_\phi(z|x)||p_\theta(z)) < \epsilon$$

Here, ϵ is the strength of the applied constraint. With this optimization based on MLE, the latent variable z can reflect the character of the ground truth data with lower error. By KKT conditions, we can rewrite the problem in Lagrangian form:

$$\mathcal{F}(\theta,\phi,\beta;x,z) = \mathbb{E}_{q_\phi(z|x)}[\log(p_\theta(x|z))] - \beta(\mathcal{KL}(q_\phi(z|x)||p_\theta(z)) - \epsilon)$$

where β is the regularization coefficient of the constraint, which limits the capacity of z and imposes an implicit pressure for independence in learning the posterior distribution due to the isotropic nature of the Gaussian prior $p_\theta(z)$. With both β and ϵ being non-negative, the equation can be rewritten using complementary relaxation of KKT conditions as:

$$\mathcal{F}(\theta, \phi, \beta; x, z) \geq \mathcal{L}(\theta, \phi; x, z, \beta)$$
$$= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right] - \beta\mathcal{KL}(q_\phi(z|x)||p_\theta(z))$$

This is the expression of the β-VAE model with an additional β coefficient.
In this model, different values of β can alter the level of learning pressure imposed during training, encouraging the learning of different representations. We assume a disentangled representation of the conditional independent data generative factors v and therefore set $\beta > 1$ to apply a stronger constraint on the latent variable information bottleneck, exceeding the constraint of the original VAE. These constraints restrict the capacity of z and, combined with the pressure to maximize the log-likelihood of the training data x, encourage the model to learn the most efficient representation of the data.

(3) Computation of single-cell neighborhood graph

Here, we used the scanpy.pp.neighbors function from Scanpy to compute the cell neighborhood graph. For detailed mathematical descriptions, please refer to the relevant papers and documentation of UMAP in Scanpy.

(4) Community detection and generation of overlapping cell communities:

We performed community detection on the cell neighborhood graph using a Graph Neural Network (GNN) model to find overlapping cell communities. GNN can learn relationships between nodes and divide them into different communities based on their similarities. Specifically, we used GNN to generate an affinity matrix F, which represents the degree of association between cells. The computation is as follows:

$$F := \mathrm{GNN}_\theta(A, X)$$

Here, A is the adjacency matrix of the cell neighborhood graph, and X represents cell type as the node feature. To ensure non-negativity of F, we applied element-wise ReLU non-linear activation function to the output layer. For detailed information about the GNN architecture,

$$F := \mathrm{GCN}_\theta(A, X) = \mathrm{ReLU}\left(A\hat{A}\mathrm{ReLU}\left(A\hat{A}XW^{(1)}\right)W^{(2)}\right)$$

Here, $\hat{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix, $\hat{A} = A + I_N$ is the adjacency matrix with self-loops, and $\hat{D} * ii = \sum_j \hat{A} * ij$ is the diagonal degree matrix of the adjacency matrix with self-loops. We considered other GNN architectures and deeper models but did not observe significant improvements. Two main differences between our model and the standard GCN are: (1) batch normalization applied after the first graph convolutional layer, and (2) L2 regularization applied to all weight matrices. We found that both modifications significantly improved the performance.

We measured the fit between the generated affinity matrix F and the neighborhood

graph using the negative log-likelihood function of the Bernoulli-Poisson model:

$$-\log p(A|F) = -\sum_{(u,v)\in E} \log\left(1-\exp(-F_u F_v^T)\right) + \sum_{(u,v)\notin E} F_u F_v^T$$

Here, E represents the set of edges in the graph. Since neighborhood graphs of single-cell data are typically sparse, the second term in the third sum contributes more to the loss. To balance these two terms, we adopted a standard technique known as balanced classification [18], and defined the loss function as follows:

$$L(F) = -E\big((u,v)\sim P_E\big)\big[\log\big(1-\exp(-F_u F_v^T)\big)\big] + E\big((u,v)\sim P_N\big)[F_u F_v^T]$$

Here, P_E and P_N represent uniform distributions over edges and non-edges, respectively.

Instead of directly optimizing the affinity matrix F as in traditional methods, we search for the optimal neural network parameters θ^* to minimize the (balanced) negative log-likelihood function:

$$\theta^* = \mathrm{argmin}_\theta L\big(\mathrm{GNN}_\theta(A,X)\big)$$

Through these steps, the BulkTrajBlend model computes overlapping communities in single-cell data, which can be used to infer "interruption" cells in the original single-cell data. It can help reveal cell type transitions and dynamics, and model and analyze cell developmental trajectories.

(5) Community trajectory inference:

Here, we inserted the overlapping communities of target cells into the original single-cell data and used PyVIA to infer pseudo-temporal trajectories of cell differentiation. For detailed inference methods, please refer to the mathematical description of PyVIA. Additionally, researchers can also use CellRank for community trajectory re-inference.

CGAN and ACGAN model description

CGAN (Conditional Generative Adversarial Nets) is a GAN (Generative Adversarial Nets) based model that generates data by training the generator and discriminator with the data and corresponding labels. The training process can be split into 2 parts. In the first part, latent variables z (dims=100) generated by standardized normal distribution $N(0,1)$ and its generated class label l_g are input into the generator to get the generated data. Here the generator can be summarized as a function g_θ, where θ is the parameter of the MLP and there are 6 layers in that each layer is normalized the hidden dimensions are 128*256*512*1024 and the activation function is LeakyRelu. After getting the generated data $g = g_\theta(z,\ l_g)$, there will be a discriminator d_ϕ, where ϕ is the parameter of the MLP and there are 4 layers in each layer the hidden dimension

is 512, dropout rate is 0.4 and the activation function is LeakyRelu, judging whether g accords with its label l. Therefore, in the second part, d_ϕ will be trained by the real data r and its label l_r with Adam optimizer to improve the judgement level of d_ϕ. Then the loss of g_θ judged by d_ϕ will be employed to enhance the generation ability of g_θ with the same optimizer. The loss functions for g_θ and d_ϕ are both MSEloss and the weights of the loss of the generative data and the real data are both 0.5.

In addition, ACGAN (Auxiliary Classifier GAN), which makes the generative data more authentic, keeps the same structure of the generator as the one in the CGAN, but it adds the classifier that offers the label of the input data on the output of the discriminator. In the training process, the loss function for the added classifier is CrossEntropy.

Data pre-processing

All single-cell data used for BulkTrajBlend training underwent the same quality control steps: Cells with low sequencing counts (<1,000) and a high mitochondrial fraction (>0.2) were excluded in further analysis. The filtered count matrix was normalized by dividing the counts of each cell by total molecule counts detected in that particular cell and logarithmised with Python library scanpy[58]. All Bulk RNA-seq were normalised using DEseq2 and `numpy.log1p` logarithmised using Python's Numpy[59] package. It is worth noting that both Bulk and single-cell data use raw counts during AE estimation of the cell fraction state space, whereas both Bulk and single-cell data use normalised and logarithmised data during training of β-VAE.

Performance evaluation

To evaluated the generated and `interruption` performance of our model, a comprehensive analysis was conducted, encompassing the examination of five critical dimensions:

(1) The count of interpolated cells, we counted the number of cells that were eventually used to interpolate into the raw single-cell profile.

(2) The correlation in marker gene expression between interpolated and authentic cells, we first use scanpy's `scanpy.tl.rank_genes_groups` function to calculate the marker genes for each type of cell subpopulation in the raw single-cell profile (taking the top 200 marker genes). Then, we use the Pearson coefficient to calculate the percentage of these 200 marker genes in the expression correlation between the generated single-cell profile and the raw single-cell profile.

(3) Marker gene similarity, we first used scanpy's `scanpy.tl.rank_genes_groups`

function to calculate the marker genes for each type of cell subpopulation (taking the first 200 marker genes) in the raw single-cell profile versus the generated single-cell profile, respectively. Then, we treated marker genes as words and all the marker genes of each cell class as sentences, and used cosine similarity to calculate the similarity of marker genes of each cell subpopulation.

(4) Transition probabilities post-interpolation We first wrapped `omicverse.pp.scale` and `omicverse.pp.pca` in omicverse, `omicverse.utils.cal_paga`, and computed the principal component PCA of the single-cell profile. We took the first 50 principal components and used the scanpy's `scanpy.pp.neighbour` to compute the neighbourhood map of the single-cell profile. Immediately after that, we calculated the developmental trajectory of single-cell profile with pseudotime using pyVIA, and we calculated the state transfer confidence for each type of cell subpopulation by taking pseudotime as the priority time with the neighbourhood graph as the input of `omicverse.utils.cal_paga`.

(5) The number of noise clusters, we used `scanpy.tl.leiden` in scanpy to perform unsupervised clustering on the generated single-cell profiles, with the resolution set to 1.0, and we identified the categories with less than 25 cells after clustering as noisy clusters and counted the number of noisy clusters as an assessment of the generation quality.

(6) Density assessment of pseudotime, after we obtained the pseudotime of single-cell profiles using pyVIA as described previously, we assessed the variance of the pseudotime of target interpolated cells as one of the metrics for the assessment of developmental trajectory reconstruction

Datasets

Dentate Gyrus:

Single-cell RNA-seq: Data from Hochgerner et al. (2018)[60]. Dentate gyrus (DG) is part of the hippocampus involved in learning, episodic memory formation and spatial coding. The experiment from the developing DG comprises two time points (P12 and P35) measured using droplet-based scRNA-seq (10x Genomics Chromium). The dominating structure is the granule cell lineage, in which neuroblasts develop into granule cells. Simultaneously, the remaining population forms distinct cell types that are fully differentiated (e.g. Cajal-Retzius cells) or cell types that form a sub-lineage (e.g. GABA cells) (Accession ID GSE95753).

Bulk RNA-seq: Data from Cembrowski et al. (2016)[61]. Dentate gyrus (DG) is measured by RNA sequencing (RNA-seq) to produce a quantitative, whole genome atlas of gene expression for every excitatory neuronal class in the hippocampus; namely, granule cells and mossy cells of the dentate gyrus, and pyramidal cells of areas CA3, CA2, and CA1 (Accession ID GSE74985).

Pancreatic endocrinogenesis:

Single-cell RNA-seq: Data from Bastidas-Ponce et al. (2019)[62]. Pancreatic epithelial and Ngn3-Venus fusion (NVF) cells during secondary transition with transcriptome profiles sampled from embryonic day 15.5. Endocrine cells are derived from endocrine progenitors located in the pancreatic epithelium. Endocrine commitment terminates in four major fates: glucagon- producing α-cells, insulin-producing β-cells, somatostatin-producing δ-cells and ghrelin-producing ε-cells (Accession ID GSE132188).
Bulk RNA-seq: Data from Bosch et al. (2023)[63]. RNA-sequencing was performed of pancreatic islets (islets of Langerhans) from mice on PLX5622 or control diet for 5.5 or 8.5 months (Accession ID GSE189434).

Human bone marrow:

Single-cell RNA-seq: Data from Setty et al. (2019)[64]. The bone marrow is the primary site of new blood cell production or haematopoiesis. It is composed of hematopoietic cells, marrow adipose tissue, and supportive stromal cells. This dataset served to detect important landmarks of hematopoietic differentiation, to identify key transcription factors that drive lineage fate choice and to closely track when cells lose plasticity (https://data.humancellatlas.org/explore/projects/091cf39b-01bc-42e5-9437-f419a66c8a45).
Bulk RNA-seq: Data from Myers et al (2018). RNA-Seq of CD34+ Bone Marrow Progenitors from Healthy Donors (Accession ID GSE118944).

Maturation of murine liver:

Single-cell RNA-seq: Data from Liang et al (2022)[65]. A total of 52,834 single cell transcriptomes, collected from the newborn to adult livers, were analyzed. We observed dramatic changes in cellular compositions during liver postnatal development. We characterized the process of hepatocytes and sinusoidal endothelial cell zonation establishment at single cell resolution. We selected 'Pro-B', 'Large Pre-B', 'SmallPre-B','B','HPC','GMP','iNP','imNP','mNP',
'Basophil','Monocyte','cDC1','cDC2','pDC','aDC',Kupffer','Proerythroblast','Erythroblast', erythrocyte' (Annotation could be found in metadata of Data from Liang et al) to performed HPC differentiation analysis (Accession ID GSE171993).
Bulk RNA-seq: Data from Renaud et al (2014)[66]. We analyze gene expression patterns in the developing mouse liver over 12 distinct time points from late embryonic stage (2 days before birth) to maturity (60 days after birth). Three replicates per time point (Accession ID GSE58827).

Construction of Simulated "interruptions" single-cell profile

To simulate the cell "interruptions" in single-cell sequencing, we conducted experiments involving cell dropout across diverse datasets. In the Pancreas dataset, we

employed Leiden clustering and manually excluded specific clusters of Ngn3 high EP, resulting in a reduction of confidence in the transition from Ngn3 high EP to Pre-endocrine to 0. In the Dentategyrus dataset, we applied Leiden clustering and manually removed specific clusters of Granule Immature, leading to a confidence reduction in the transition from Granule Immature to Granule Mature to 0. Furthermore, in the BoneMarrow dataset, we randomly eliminated 80% of the cells from HSC-2, causing a confidence drop in the transition from HSC-2 to Monocyte-2 to 0.

To employ BulkTrajBlend for generating "interrupted" cells across various datasets, we generated single-cell data from the bulk RNA-seq data using BulkTrajBlend and filtered out noisy cells using the size of the Leiden as a constraint. In configuring the model for different datasets, we set the hyperparameter "cell_target_num" to be 1.5 times, 1 time, and 6 times the number of dropped-out cell types, aligning with Pancreas, Dentategyrus, and BoneMarrow, respectively. Subsequently, BulkTrajBlend calculated the overlap of cell types in the generated single-cell data and we annotated the overlapping cell communities. Specifically, we selected the single-cell data in which dropped-out cell types were associated with adjacent cell types.

Methods of OmicVerse integration

We unified the downstream analyses of Bulk RNA-seq, single cell RNA-seq in OmicVerse. Since the downstream analyses are independent of the parameter evaluation of BulkTrajBlend and the analysis modules of each part are independent of each other, we have placed the datasets and methods used in each part in Supplementary, an index of which is provided here.

(1) Bulk RNA-seq: All datasets selected, parameter setting, and methods could be found in Supplementary Note 4.
(2) scRNA-seq: All datasets selected, parameter setting, and methods could be found in Supplementary Note 5.
(3) Multi-omics: All datasets selected, parameter setting, and methods could be found in Supplementary Note 6.

Data availability

All datasets analyzed in this manuscript are public and have been published in other papers. We have referenced them in the manuscript and, when necessary, made them available at https://github.com/Starlitnightly/omicverse-reproducibility.

Code availability

The code to reproduce the experiments of this manuscript is available at https://github.com/Starlitnightly/omicverse-reproducibility . The OmicVerse package can be found on GitHub at https://github.com/Starlitnightly/omicverse Documentation and tutorials can be found at https://omicverse.readthedocs.io.

Acknowledgments

This work was supported by the grants from the National Natural Science Foundation of China (32300682 to C.X.), the National Key Research & Developmental Program of China (92249303 to Y.X.), the Fundamental Research Funds for the Central Universities (FRF-TP-22-007A1 to C.X.), the Student Research Training Program of University of Science and Technology Beijing.
We thank all the Github users who contributed code to OmicVerse over the years. We would like to thank the following WeChat Official Accounts for promoting OmicVerse: pythonic biologists, biotrainee. Pythonic biologist and Biotrainee's article inspired some of the charting in the OmicVerse.

Author contributions

Z.Z., Y.M., and L.H. contributed equally to this work. Z.Z. was responsible for designing the OmicVerse application programming interface and designing the whole BulkTrajBlend framework. Y.M. played a key role in designing and implementing the overlap cell community of BulkTrajBlend, while Z.Z. was responsible for implementing and conducting testing of the single-cell RNA-seq pipeline. L.H. conducted simulated single-cell profile tests for BulkTrajBlend. The multi-omics module of pyMOFA was implemented and tested by L.H., Y.W., and Z.Z. P.L. handled the implementation and testing of the metacells analysis in SEACells. B.T. was responsible for writing the methods for CGAN and ACGAN, as well as reviewing the methods of BulkTrajBlend. H.D., Y.X., and Z.Z. jointly conceived, implemented, and tested the bulk RNA-seq pipeline. C.X. and Y.X. provided the conceptualization of false overlap rate for evaluation of BulkTrajBlend. Y.X., H.D., C.X., and Z.Z. provided supervision and contributed to the conceptualization of the OmicVerse platform. The manuscript was collaboratively written by Z.Z., Y.M., L.H., H.D., and Y.X.

Competing intersets

The authors declare no competing interests.

Reference

1 Kharchenko, P. V. The triumphs and limitations of computational methods for scRNA-seq. *Nature Methods* **18**, 723-732 (2021).

2 Peng, L. *et al.* Single-cell RNA-seq clustering: datasets, models, and algorithms. *RNA biology* **17**, 765-783 (2020).

3 Xu, X., Hua, X., Mo, H., Hu, S. & Song, J. Single-cell RNA sequencing to identify cellular heterogeneity and targ ets in cardiovascular diseases: from bench to bedside. *Basic research in cardiology* **118**, 7, doi:10.1007/s00395-022-00972-1.

4 Derakhshan, T., Boyce, J. A. & Dwyer, D. F. Defining mast cell differentiation and heterogeneity through single-ce ll transcriptomics analysis. *Journal of Allergy and Clinical Immunology* **150**, 739-747, doi:10.1016/j.jaci.2022.08.011.

5 Zeng, L. *et al.* Research progress of single-cell transcriptome sequencing in autoimmun e diseases and autoinflammatory disease: A review. *J Autoimmun* **133**, 102919, doi:10.1016/j.jaut.2022.102919.

6 Thind, A. S. *et al.* Demystifying emerging bulk RNA-Seq applications: the application and utility of bioinformatic methodology. *Briefings in bioinformatics* **22**, bbab259 (2021).

7 Liao, J. *et al.* De novo analysis of bulk RNA-seq data at spatially resolved single-cell resolution. *Nature Communications* **13**, 6498, doi:10.1038/s41467-022-34271-z (2022).

8 Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15**, 550, doi:10.1186/s13059-014-0550-8 (2014).

9 Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpret ing genome-wide expression profiles. *Proc Natl Acad Sci U S A* **102**, 15545-15550, doi:10.1073/pnas.0506580102.

10 Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* **9**, 559, doi:10.1186/1471-2105-9-559 (2008).

11 Hu, C. *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**, D870-D876, doi:10.1093/nar/gkac947 (2023).

12 Efremova, M., Vento-Tormo, M., Teichmann, S. A. & Vento-Tormo, R. CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nature protocols* **15**, 1484-1506 (2020).

13 Stassen, S. V., Yip, G. G. K., Wong, K. K. Y., Ho, J. W. K. & Tsia, K. K. Generalized and scalable trajectory inference in single-cell omics data with VIA. *Nature Communications* **12**, 5528, doi:10.1038/s41467-021-25773-3 (2021).

14 Hsieh, C.-Y. *et al.* scDrug: From single-cell RNA-seq to drug response prediction. *Computational and Structural Biotechnology Journal* **21**, 150-157, doi:10.1016/j.csbj.2022.11.055.

15 Amezquita, R. A. *et al.* Orchestrating single-cell analysis with Bioconductor. *Nature Methods* **17**, 137-145, doi:10.1038/s41592-019-0654-x (2020).

16 Virshup, I. *et al.* The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nature Biotechnology* **41**, 604-606, doi:10.1038/s41587-023-01733-8 (2023).

17 Giorgi, F. M., Ceraolo, C. & Mercatelli, D. The R Language: An Engine for Bioinformatics and Data Science. *Life (Basel)* **12**, 648, doi:10.3390/life12050648.

18 Brittain, J., Cendon, M., Nizzi, J. & Pleis, J. Data scientist's analysis toolbox: Comparison of Python, R, and SAS Performance. *SMU Data Science Review* **1**, 7 (2018).

19 Wu, H., Kirita, Y., Donnelly, E. L. & Humphreys, B. D. Advantages of single-nucleus over single-cell RNA sequencing of adult kidney: rare cell types and novel cell states revealed in fibrosis. *Journal of the American Society of Nephrology: JASN* **30**, 23 (2019).

20 Mereu, E. *et al.* Benchmarking single-cell RNA-sequencing protocols for cell atlas projects. *Nature Biotechnology* **38**, 747-755, doi:10.1038/s41587-020-0469-4 (2020).

21 Denyer, T. & Timmermans, M. C. P. Crafting a blueprint for single-cell RNA sequencing. *Trends Plant Sci* **27**, 92-103, doi:10.1016/j.tplants.2021.08.016 (2022).

22 Thibivilliers, S., Anderson, D. & Libault, M. Isolation of Plant Root Nuclei for Single Cell RNA Sequencing. *Curr Protoc Plant Biol* **5**, e20120, doi:10.1002/cppb.20120 (2020).

23 Gao, C., Zhang, M. & Chen, L. The comparison of two single-cell sequencing platforms: BD rhapsody and 10x genomics chromium. *Current genomics* **21**, 602-609 (2020).

24 Ahlmann-Eltze, C. & Huber, W. Comparison of transformations for single-cell RNA-seq data. *Nature Methods* **20**, 665-672, doi:10.1038/s41592-023-01814-1 (2023).

25 Frishberg, A. *et al.* Cell composition analysis of bulk genomics using single-cell data. *Nature methods* **16**, 327-332, doi:10.1038/s41592-019-0355-5.

26 Wang, X., Park, J., Susztak, K., Zhang, N. R. & Li, M. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature communications* **10**, 380 (2019).

27 Higgins, I. *et al.* beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. (2016).

28 Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nature Methods* **16**, 715-721, doi:10.1038/s41592-019-0494-8 (2019).

29 Chen, Y. *et al.* Deep autoencoder for interpretable tissue-adaptive deconvolution and cell-type-specific gene analysis. *Nature Communications* **13**, 6735, doi:10.1038/s41467-022-34550-9 (2022).

30 Traag, V. A., Waltman, L. & Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* **9**, 1-12 (2019).

31 Yang, J. & Leskovec, J. Structure and overlaps of ground-truth communities in networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* **5**, 1-35 (2014).

32 Shchur, O. & Günnemann, S. Overlapping community detection with graph neural networks. *arXiv preprint arXiv:1909.12201* (2019).

33 Dimitrov, D. & Gu, Q. BingleSeq: a user-friendly R package for bulk and single-cell RNA-Seq data analysis. *PeerJ* **8**, e10469, doi:10.7717/peerj.10469.

34 Flores, M. *et al.* Deep learning tackles single-cell analysis—a survey of deep learning f or scRNA-seq analysis. *Briefings in bioinformatics* **23**, bbab531.

35 Behdenna, A. *et al.* pyComBat, a Python tool for batch effects correction in high-throughput molecular data using empirical Bayes methods. *bioRxiv*, 2020.2003.2017.995431, doi:10.1101/2020.03.17.995431 (2023).

36 Muzellec, B., Telenczuk, M., Cabeli, V. & Andreux, M. PyDESeq2: a python package for bulk

RNA-seq differential expression an alysis. *bioRxiv*, 2022-2012.

37 Szklarczyk, D. *et al.* The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research* **49**, D605-D612 (2021).

38 Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics* **9**, 1-13 (2008).

39 Fang, Z., Liu, X. & Peltz, G. GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. *Bioinformatics* (2022).

40 Zhang, Y. *et al.* Single-cell RNA sequencing in cancer research. *Journal of Experimental &Clinical Cancer Research* **40**, 1-17.

41 Mo, Z. *et al.* Single-cell transcriptomics reveals the role of Macrophage-Naïve CD4+ T cell interaction in the immunosuppressive microenvironment of prima ry liver carcinoma. *Journal of Translational Medicine* **20**, 1-17.

42 Agrawal, A., Ali, A., Boyd, S. & others. Minimum-distortion embedding. *Foundations and Trends® in Machine Learning* **14**, 211-378.

43 Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods* **16**, 1289-+, doi:10.1038/s41592-019-0619-0 (2019).

44 Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nature Biotechnology* **37**, 685-691, doi:10.1038/s41587-019-0113-3 (2019).

45 Cao, Y., Wang, X. & Peng, G. SCSA: a cell type annotation tool for single-cell RNA-seq data. *Frontiers in genetics* **11**, 490 (2020).

46 Zhang, X. *et al.* CellMarker: a manually curated resource of cell markers in human and m ouse. *Nucleic acids research* **47**, D721-D728.

47 Yuan, H. *et al.* CancerSEA: a cancer single-cell state atlas. *Nucleic acids research* **47**, D900-D908.

48 Van de Sande, B. *et al.* A scalable SCENIC workflow for single-cell gene regulatory network analysis. *Nature Protocols* **15**, 2247-2276 (2020).

49 Persad, S. *et al.* SEACells infers transcriptional and epigenomic cellular states from si ngle-cell genomics data. *Nature Biotechnology*, 1-12.

50 Che, L.-H. *et al.* A single-cell atlas of liver metastases of colorectal cancer reveals reprogramming of the tumor microenvironment in response to preoperative chemotherapy. *Cell discovery* **7**, 80 (2021).

51 AlMusawi, S., Ahmed, M. & Nateri, A. S. Understanding cell-cell communication and signaling in the colorectal cancer microenvironment. *Clinical and Translational Medicine* **11**, e308.

52 Han, J. M. & Jung, H. J. Cyclophilin A/CD147 Interaction: A Promising Target for Anticancer The rapy. *International journal of molecular sciences* **23**, 9341, doi:10.3390/ijms23169341.

53 Scarzello, A. J. *et al.* LTβR signalling preferentially accelerates oncogenic AKT-initiated liv er tumours. *Gut* **65**, 1765-1775, doi:10.1136/gutjnl-2014-308810.

54 Clark, S. J. *et al.* scNMT-seq enables joint profiling of chromatin accessibility DNA methy lation and transcription in single cells. *Nature communications* **9**, 781.

55 Cao, Z.-J. & Gao, G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, 1-9 (2022).

56 Argelaguet, R. *et al.* MOFA+: a statistical framework for comprehensive integration of multi-

modal single-cell data. *Genome biology* **21**, 1-17 (2020).

57 Song, D. *et al.* scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nature Biotechnology*, doi:10.1038/s41587-023-01772-1 (2023).

58 Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 1-5 (2018).

59 Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357-362 (2020).

60 Hochgerner, H., Zeisel, A., Lönnerberg, P. & Linnarsson, S. Conserved properties of dentate gyrus neurogenesis across postnatal de velopment revealed by single-cell RNA sequencing. *Nature neuroscience* **21**, 290-299, doi:10.1038/s41593-017-0056-2 (2018).

61 Cembrowski, M. S., Wang, L., Sugino, K., Shields, B. C. & Spruston, N. Hipposeq: a comprehensive RNA-seq database of gene expression in hippo campal principal neurons. *eLife* **5**, e14997, doi:10.7554/eLife.14997 (2016).

62 Bastidas-Ponce, A. *et al.* Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development* **146**, dev173849 (2019).

63 Bosch, A. J. T. *et al.* CSF1R inhibition with PLX5622 affects multiple immune cell compartments and induces tissue-specific metabolic effects in lean mice. *Diabetologia* **66**, 2292-2306, doi:10.1007/s00125-023-06007-1 (2023).

64 Setty, M. *et al.* Characterization of cell fate probabilities in single-cell data with P alantir. *Nature biotechnology* **37**, 451-460, doi:10.1038/s41587-019-0068-4 (2019).

65 Liang, Y. *et al.* Temporal analyses of postnatal liver development and maturation by sin gle-cell transcriptomics. *Developmental cell* **57**, 398-414.e395, doi:10.1016/j.devcel.2022.01.004 (2022).

66 Renaud, H. J., Cui, Y. J., Lu, H., Zhong, X.-b. & Klaassen, C. D. Ontogeny of hepatic energy metabolism genes in mice as revealed by RNA -sequencing. *PloS one* **9**, e104560, doi:10.1371/journal.pone.0104560 (2014).