BMC Bioinformatics

SOFTWARE

Open Access

CIA: unveiling cellular identities with cluster-independent annotation in single-cell RNA sequencing data for comprehensive cell type characterization and exploration

Ivan Ferrari[1,2†], Mattia Battistella[1,2†], Francesca Vincenti[1], Andrea Gobbini[1], Federico Marini[3], Samuele Notarbartolo[1,4], Jole Costanza[1], Stefano Biffo[1,2], Renata Grifantini[1], Sergio Abrignani[1,5] and Eugenia Galeota[1*]

†Ivan Ferrari and Mattia Battistella have contributed equally to this work.

*Correspondence:
Eugenia Galeota
galeota@ingm.org
[1]Fondazione Istituto Nazionale Di Genetica Molecolare 'Romeo ed Enrica Invernizzi' (INGM), Milan, Italy
[2]Department of Biosciences, Università Degli Studi di Milano, Milan, Italy
[3]Institute of Medical Biostatistics, Epidemiology and Informatics, Mainz, Germany
[4]Present address: Infectious Diseases Unit, Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Milan, Italy
[5]Department of Clinical Sciences and Community Health, Università Degli Studi di Milano, Milan, Italy

Abstract

Background Single-cell RNA sequencing (scRNA-seq) has revolutionized our understanding of the transcriptional landscape of complex tissues, enabling the discovery of novel cell types and biological functions. However, the identification and classification of cells from scRNA-seq datasets remain significant challenges.

Results To address this, we developed a new computational tool called CIA (Cluster Independent Annotation), which accurately identifies cell types across different datasets without requiring a fully annotated reference dataset or complex machine learning processes. Based on predefined cell type signatures, CIA provides a highly user-friendly and practical solution to cell-type and functional annotation of single cells. The CIA framework is implemented in both the Python and R programming languages, making it applicable to all main single-cell analysis frameworks, and it is available under the MIT license with its documentation at the following links: Python package: https://pypi.org/project/cia-python/. Python tutorial: https://cia-python.readthedocs.io/en/latest/tutorial/Cluster_Independent_Annotation.html. R package and tutorial: https://github.com/ingmbioinfo/CIA_R.

Conclusions Our results demonstrate that CIA classification performances are comparable to the other state-of-the-art approaches, while requiring a significantly lower computational running time. Overall, CIA simplifies the process of obtaining reproducible signature-based cell assignments that can be easily interpreted through graphical summaries providing researchers with a powerful tool to explore the complex transcriptional landscape of single cells.

Keywords scRNA-seq, Cell-type annotation, Classifier, Gene signature, Scoring, Functional analysis, Clustering-free

Background

The emergence of scRNA-seq technology has opened up unprecedented opportunities for exploring gene expression profiles at the individual cell level [1]. scRNA-seq has quickly become the preferred approach for analyzing cell transcriptional readouts, enabling researchers to identify cell transitions, discover new cell types and investigate cellular heterogeneity at various levels of granularity. Understanding cellular diversity is a significant challenge in many areas of biology and biomedical research, including immunology, developmental and stem cell biology, neurobiology, and cancer research [2, 3].

Once library preparation and sequencing are completed, count matrices are generated by processing and aligning raw data. These matrices serve as the starting point for a single-cell RNA sequencing analysis. The typical computational workflow involves several key steps: quality control and filtering of cells and genes, normalization of gene expression counts and identification of groups of cells with similar transcriptional patterns [4].

Different clustering methods can be employed to identify these groups, but setting the correct clustering resolution is often challenging [5]. General guidelines may not apply equally well to all datasets [6]. Accurate labeling of cells and clusters is crucial for identifying cell populations, their functions, and the biological states represented in the experimental context. Traditionally, cell type annotation relies on manual curation by domain experts—a process that is time-consuming and may lack reproducibility [7]. This approach depends heavily on prior biological knowledge and is not easily automated. However, efforts are underway to provide annotation guidelines using automated cell labeling tools [8].

To address this challenge, various computational approaches have been developed for automated cell type annotation in scRNA-seq datasets. These methods can be broadly categorized into three main strategies: marker-genes database-based annotation, correlation-based annotation, and supervised classification-based annotation [7]. Marker-genes database-based methods rely on known marker genes and their expression patterns, associated with specific cell types, to assign cell identities. Correlation-based methods compare the gene expression profiles of unlabeled cells or clusters with those of reference datasets to infer cell types based on similarity metrics. Supervised classification-based methods leverage machine learning algorithms trained on labeled reference datasets to classify unlabeled cells [7].

Each strategy offers unique advantages in handling complex datasets and capturing subtle variations in gene expression profiles.

However, the identification of reliable reference datasets or gene signatures is complicated by the high degree of heterogeneity characterizing single-cell datasets, making it challenging to label cell types accurately. This heterogeneity arises from various factors, including technical noise, batch effects, and the intrinsic diversity of cellular states, which can obscure the definition of distinct cell population [9]. As highlighted by Abdelaal et al. [10], automated annotation methods must account for this variability to improve accuracy and robustness in single-cell analyses. Their study systematically evaluated multiple cell-type annotation tools, demonstrating that performance can vary significantly across datasets and highlighting the need for methods that are adaptable to different levels of cellular heterogeneity. This underscores the importance of developing approaches that can accurately classify cells despite dataset-specific variations.

Recently, valuable gene signature collections have been published, covering various organisms, tissues and cell types at different levels of granularity such as ACT [11], CellMarker 2.0 [12], PanglaoDB [13], singleCellBase [14], and CellKb[15]. These collections can be utilized to assign cell types in newly produced single-cell RNA-sequencing datasets. Moreover, consortia are producing large-scale single-cell datasets in the form of atlases to serve as a reliable reference for grasping the underlying differences among cell types, tissues and individuals [16]. The availability of reference atlases, coupled with the ability of machine learning approaches to exploit prior domain knowledge, represents a new way of analyzing single-cell datasets [17].

In this context, we present CIA, a software that automatically assigns accurate cell type labels in scRNA-seq datasets requiring only gene signatures. CIA is a versatile framework that supports comprehensive software packages such as SingleCellExperiment, Seurat and Scanpy, allowing researchers to efficiently annotate cell populations without the need for extensive reference training datasets, thus accelerating the process of biological discovery and interpretation across a wide range of single-cell datasets.

Implementation

CIA is a software package developed for the signature-based automatic annotation of cellular populations within scRNA-seq datasets. Designed to operate without relying on clustering, CIA offers a simple and streamlined approach to cell classification. Among the signature-based methods cited in [7], CIA is the only one focused on annotating single cells, as other methods are cluster-based. Although CellAssign operates at the single-cell level, it still depends on a Bayesian probabilistic model to assign labels to clusters, making it fundamentally cluster-based. In contrast, CIA eliminates the need for clustering, model training, or probabilistic frameworks, providing a direct and efficient tool for high-throughput single-cell RNA-seq data analysis.

We opted for a signature-based approach because it enables quick and efficient exploration of datasets, identifying enrichment for specific signatures of interest without the complexity of training models or needing reference datasets. This makes CIA an excellent tool for rapid analysis, allowing researchers to gain valuable insights without the computational overhead associated with traditional machine learning methods.

CIA takes two main inputs: (i) a dataset with normalized gene expression values for individual cells and (ii) a list of gene signatures with the associated cell types. The software is implemented in both Python and R to accommodate diverse user preferences and frameworks commonly used in the scientific community. Signatures can be supplied in Gene Matrix Transposed (GMT) format from different sources, including URLs, local files, Python dictionaries, or R lists of named vectors. CIA is compatible with major single-cell data formats, including Python AnnData [18], R SingleCellExperiment [19], and SeuratObject [20], and is specifically designed to work with datasets that have been normalized using total count scaling, followed by log-transformation. Tutorials included in the CIA packages, demonstrate the required preprocessing steps. This ensures seamless integration into existing analytical workflows.

CIA is composed of several functions, with the core functions being score_*signature*, *score_all_signatures* and *CIA_classify*, and other additional functionalities such as performance reporting and classification refinement: (i) *score_signature* calculates scores (*detailed in Methods*) for a given signature for each cell in the dataset, generating a

score vector. (ii) *score_all_signatures* calculates scores for multiple signatures returning a matrix of scores. This score matrix, where each entry corresponds to the signature scores for every gene signature and cell in the expression matrix, is crucial for the subsequent classification (Fig. 1). Notably, the returned output can be effortlessly integrated into the original input object (as continuous sample metadata), reducing the effort to track all operations performed. (*iii*) Exploiting the score matrix, *CIA_classify* labels each single cell of the dataset according to the best matching signature. To prevent misassignment when signatures yield comparable scores,a threshold parameter allows labeling the cell as 'Unassigned' if not met (see *Methods*). This approach is critical for maintaining classification specificity and avoiding potential misclassifications, which can lead to erroneous biological conclusions.

Moreover, CIA offers additional functionalities to assist users in evaluating both the quality of signatures and the classification performance of CIA itself or other tools:

(i) grouped_distributions calculates and plots median signature scores in groups of cells (e.g. user-defined annotation) for each signature. The function performs comparative statistical analysis on cell populations using Wilcoxon signed-rank tests and Mann–Whitney U tests to evaluate the discriminative power of gene signatures within and across cell groups. The Wilcoxon signed-rank test checks if each pair of gene signatures has significantly different distributions within the same cell group. The Mann–Whitney U test checks if each gene signature has significantly different distributions across different cell groups. In this way the function identifies potential issues such as inadequate gene signatures or flawed clustering. (*ii*) *group_composition* returns and plots a contingency table of the classification results. It is beneficial when a reference classification is available and it can be used to test how assigned labels align with the reference groups. (*iii*) *classification_metrics* and *grouped_classification_metrics* compute sensitivity, specificity, precision, accuracy and F1-score. *classification_metrics* returns the metrics per cell group, providing information about the classification quality for individual cell populations. *grouped_classification_metrics* offers the same metrics for the entire dataset, allowing for rapid comparison of different cell annotations simultaneously.

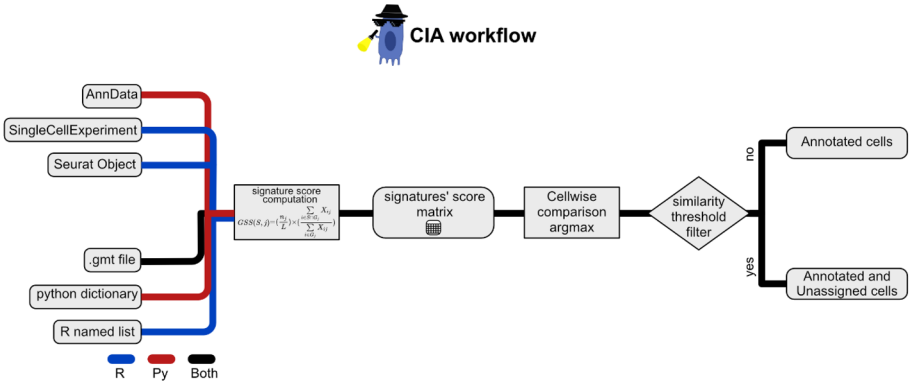

Fig. 1 Schematic representation of CIA workflow. CIA requires as inputs a list of named gene signatures and a single-cell object (Anndata, SingleCellExperiment, SeuratObject) containing the normalized gene expression matrix. The signature_score function calculates for each cell and each signature a gene signature score. Then, raw scores from the signature score matrix are scaled in the interval [0, 1]. Finally, CIA_classify assigns cell labels based on the top scored signature name. At user discretion, it is possible to use a similarity threshold to mark as unassigned cells having top scoring values too close

Additionally, CIA includes several auxiliary functions: *signatures_similarity*, *filter_degs*, and *celltypist_majority_vote*. *signatures_similarity* calculates the similarity between couples of gene signatures using either the Jaccard index or by computing the percentage of overlap of the first signature with the second. *filter_degs* (available only in Python version) subsets differentially expressed genes (DEGs) stored in the 'uns' layer of the AnnData object based on different thresholds of log fold changes, z-scores, mean expression and percentage of cells expressing genes. *celltypist_majority_vote* integrates Celltypist [21] code and is optionally used only after CIA workflow. It extends the most represented cell-level annotations to independently defined cell groups. If reference clusters are not provided by the user, the function employs an over-clustering approach using the Leiden algorithm to define groups (see the Celltypist documentation for further details).

Results

CIA correctly classifies the PBMC3k dataset

We tested CIA on a human PBMC3k single cell dataset from 10X Genomics [22], comprising 2700 cells from a healthy donor. To automatically annotate cells, we utilized differentially expressed genes for specific cell populations, obtained from a publicly available reference atlas of human peripheral mononuclear blood cells (PBMC Atlas) [23], We chose the PBMC Atlas as the reference for our study due to its comprehensive and accurate annotation of human peripheral blood cell types, which includes a variety of well-characterized immune cell populations at both transcriptomic and proteomic levels (Figure S1A). Using such a well-defined atlas provides a solid reference point for cell type classification and allows for a comparative evaluation of CIA's performance against existing classification methods.

The differential expression analysis (see Materials and Methods) yielded seven gene signatures, also available in the CIA tutorials and online repository, that represent immune populations: B lymphocytes, CD4 + and CD8 + T lymphocytes, dendritic cells (DC), monocytes (Mono), natural killer cells (NK) and platelets (Fig. 2A).

We benchmarked our results against the manual classification of the PBMC3k dataset (Fig. 2A, B). First, we applied the *score_all_signature*s function to calculate scaled scores associated with each cell of the PBMC Atlas for each of the seven subpopulation signatures. By utilizing the *grouped_distributions* we obtained the median signature score for each signature within each population. The function also assessed their statistical significance (Figure S1B). We observed that the highest scores for each signature were consistent with the corresponding manually annotated subpopulation.

We then used the signatures obtained from the PBMC Atlas to classify the PBMC3k dataset with the CIA_classify function (Fig. 2B). The heatmaps with median signature scores clearly showed that higher median signature scores corresponded to the correct population (Figure S1C). As expected, the *grouped_distributions* function returned a warning indicating that the signature score distribution of DC and Mono were not significantly different within the manually annotated dendritic cells cluster. This diagnostic information is valuable because it highlights when signatures lack specificity for the target population. For example, the signature for monocytes had lower median scores due to the lack of specific identity markers, while the DC signature exhibited high variability within the DC cluster.

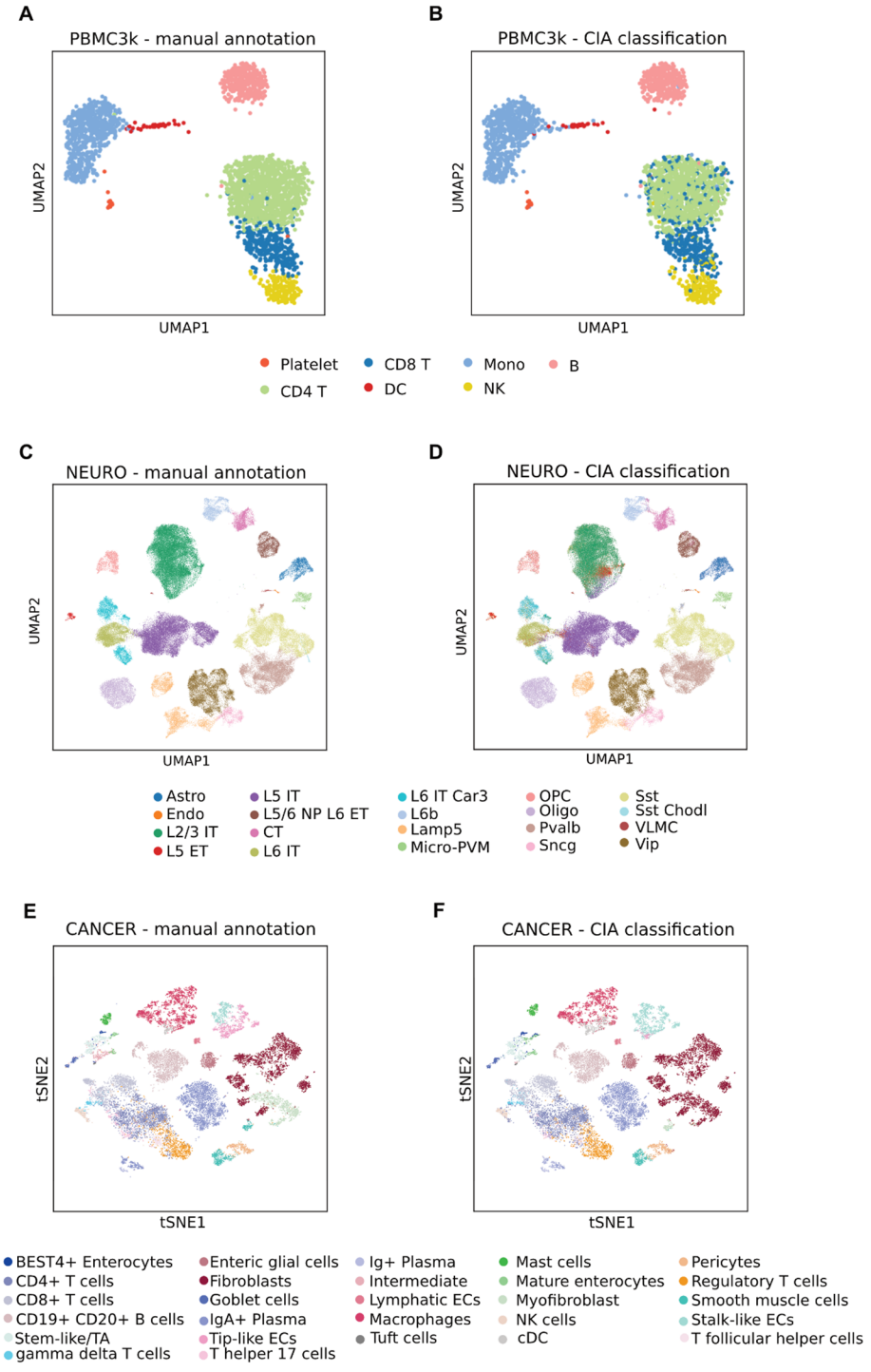

Fig. 2 **A**, **C**, **E** UMAP plot showing the PBMC3k (**A**), NEURO_test (**C**) and KUL3 (**E**) datasets ground truth annotation. **B**, **D**, **F** UMAP plot of PBMC3k (**B**), NEURO_test (**D**) and KUL3 (**F**) datasets showing cell labels predicted by CIA

The statistical tests were crucial for identifying these issues. When a signature is not specific enough, as with the Mono signature, or when scores are highly variable within a population, as with the DC signature, it indicates the need for refinement. To address these problems, it is beneficial to identify more specific population markers or to specify groups of markers that better capture the variability of the given population.

However, since for each distribution, there is only a cell group in which values are significantly higher than all the other groups, the simple inspection of PBMC3k UMAP showing the different signature scores was sufficient to annotate the expected cell populations (Figure S1D).

CIA_classify inferred cell identities with high accuracy (0.98), precision (0.92) and F1 (0.92) values (Table 1) in less than 0.30 s with different computational setups.

Comparison of CIA with state-of-art classifiers

We compared the performance of CIA with representative single-cell classification tools, including the machine-learning-based, Celltypist [21] and Garnett [24], the reference-based SingleR [25] and two deep-learning methods, scANVI [26] and scBalance [27]. The evaluation was conducted on three benchmark datasets of increasing complexity (Table S1): PBMC3k; NEURO_test [28], comprising 119,152 nuclei from the human middle temporal gyrus, used to assess CIA's scalability and robustness on large, granular transcriptomic data; KUL3 [29], a colorectal cancer dataset from six patients (Katholieke Universiteit Leuven) containing 27 distinct cell types, representative of high tissue heterogeneity.

Additionally, we evaluated AUCell [30], a popular signature scoring method, by assigning each cell the label of the signature with the highest AUCell score. However, AUCell scores are not directly comparable across signatures due to their dependence on gene-set size and expression levels. This limitation requires manual threshold selection and may result in multiple high-scoring signatures for the same cell, preventing fully automated

Table 1 Table reporting cell-level classification performances

	Classifier	Classification performance				
		SE	SP	PR	ACC	F1
PBMC	CIA_Python	0,92	0,99	0,92	0,98	0,92
	CIA_R	0,92	0,99	0,92	0,98	0,92
	SingleR	0,76	0,96	0,77	0,93	0,76
	Garnett	0,24	0,94	0,39	0.84	0,29
	Celltypist	0,75	0,96	0,75	0,93	0,75
	AUCell	0,65	0,94	0,65	0,9	0,65
	scANVI	0,85	0,97	0,85	0,96	0,85
	scBalance	0,89	0,98	0,89	0,97	0,89
NEURO	CIA_Python	0,93	1	0,93	0,99	0,93
	CIA_R	0,93	1	0,93	0,99	0,93
	SingleR	0,97	1	0,99	1	0,98
	Garnett	0,03	0,98	0,09	0,93	0,05
	Celltypist	0,85	0,99	0,85	0,99	0,85
	AUCell	0,88	0,99	0,88	0,99	0,88
	scANVI	0,98	1	0,98	1	0,98
	scBalance	0,99	1	0,99	1	0,99
CANCER	CIA_Python	0,78	0,99	0,78	0,98	0,78
	CIA_R	0,78	0,99	0,78	0,98	0,78
	SingleR	0,73	0,99	0,74	0,98	0,74
	Garnett	0,07	1	0,46	0,96	0,13
	Celltypist	0,79	0,99	0,79	0,98	0,79
	AUCell	0,54	0,98	0,54	0,97	0,54
	scANVI	0,78	0,99	0,78	0,98	0,78
	scBalance	0,78	0,99	0,78	0,98	0,78

SE, sensibility; SP, specificity; PR, precision; ACC, accuracy; F1, F1-score. PBMC: PBMC3k, NEURO: NEURO_test, CANCER: KUL3

annotation. In contrast, CIA directly assigns a unique cell type label by identifying the most compatible signature, ensuring comparability across signatures and offering a principled way to handle unassigned cells when signatures are ambiguous.

To ensure a fair comparison, all classifiers relied on the same biological sources. Specifically, CIA and AUCell used identical signature files derived from the PBMC Atlas and CRC_training datasets, while for the NEURO_test dataset, we directly employed subclass-level signatures downloaded from the Azimuth web portal. These Azimuth signatures are publicly available and represent a validated, expert-curated resource.

For model-based tools (Celltypist, Garnett, and SingleR, as well as scANVI and scBalance), we used corresponding annotated training datasets to ensure consistency. PBMC Atlas was used to classify PBMC3k, the Neuro_training dataset (76,533 human primary motor cortex nuclei from the Allen Institute for Brain Science [31]) to classify NEURO_test (Fig. 2D) and the CRC_training dataset (352,187 cells) [32] to classify KUL3.

This setup ensured that all classifiers, whether signature-based or model-based, relied on the same biological sources (see Methods and Supplementary Table S1 and repository https://github.com/ingmbioinfo/CIA_benchmark/ for full reproducibility).

When comparing classifier performance on the PBMC3k dataset, CIA reached the highest accuracy and F1 score while also achieving the shortest runtime (Figs. 2B, 3A, Figure S2, one-sided t-test, p-value < 0.01 for running time).

On the NEURO_test dataset, CIA inferred cell identities with high accuracy (0.99), precision (0.93), and F1 (0.93), demonstrating strong reliability even when using publicly available curated signatures. Its F1 value was only 0.06 points below scBalance, which achieved a nearly perfect classification (F1 = 0.99). Importantly, CIA outperformed all other methods in terms of speed (Figs. 2D, 3B, Figure S4, one-sided t-test, p-value < 0.01), classifying approximately 120,000 cells in one minute or less than one second using 8 CPUs (CIA Python).

For the KUL3 dataset, where high tissue complexity generally reduced classification accuracy, CIA still ranked among the best-performing tools, with an F1 score of 0.78 compared to Celltypist, the top performer (F1 = 0.79) (Fig. 2F, Figure S6). Again, CIA was the fastest method for performing the classification (Fig. 3C, one-sided t-test, p-value < 0.01).

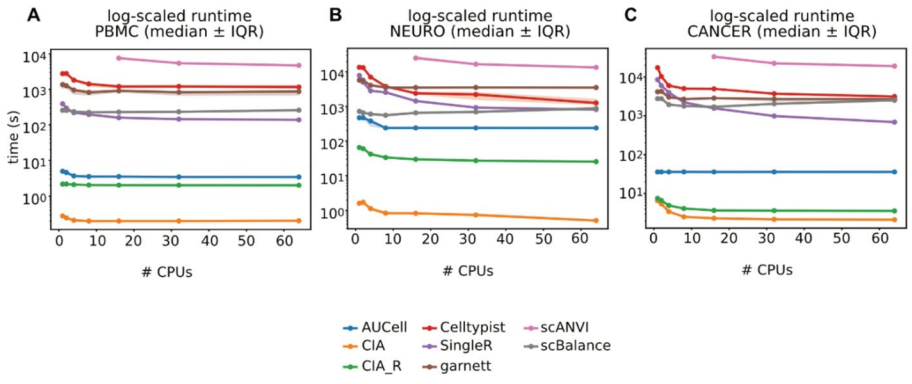

Fig. 3 Log-scaled classification runtime (seconds) versus CPU count for eight classifiers on three datasets: **A** PBMC3k, **B** NEURO_test, **C** KUL3. Curves show the median runtime; vertical bars denote the interquartile range (IQR) at each CPU setting. Repeats: n = 10 for all tool–dataset combinations except scANVI (n = 3 across all datasets) and Celltypist, Garnett, and SingleR on KUL3 (n = 3). N.B. IQR bars are often smaller than the marker size due to highly reproducible timings on our HPC workload manager and the large runtime gaps between signature-based methods (e.g., CIA/CIA_R/AUCell) and the others

Garnett showed the lowest classification performance in our benchmark. We hypothesize that its underlying algorithm may be less effective when handling large or heterogeneous marker sets, and that parameter adjustments are unlikely to yield substantial improvements under these conditions.

Taken together, these results demonstrate the consistent and highly accurate performance of CIA across datasets of increasing complexity (F1 > 0.92 for PBMC3k and NEURO_test, and performance comparable to the best classifier for KUL3), combined with a substantially faster runtime than all other tested methods.

CIA is resilient to gene list trimming

To systematically assess the robustness and stability of CIA classification under conditions of incomplete gene signature information (such as partial lack of marker knowledge for certain cell types) we simulated increasing levels of gene list trimming across the three benchmark datasets (PBMC3k, Neuro_test, and KUL3). For each signature, we progressively removed 10–90% of its genes, repeating the process 10 times for statistical robustness (see Materials and Methods). In all cases, the overall classification performance remained stable up to approximately 30% gene removal, indicating strong resilience to moderate gene loss (Fig. 4A–C). Beyond this threshold, a gradual decline in F1 score was observed, with the steepest drop occurring overall above ~ 60% removal. As expected, initial F1 values reflected dataset complexity, being highest in PBMCK and lowest in KUL3; yet the overall pattern of decay was consistent across datasets. These results demonstrate that CIA maintains reliable classification performance even under partial loss of signature information, with substantial degradation appearing only at high levels of list trimming. Notably, even with 50% of genes removed, CIA still achieved a median F1 score of 0.8 in the Neuro_test dataset, despite most signatures being reduced to only 4–5 genes, underscoring the robustness of CIA robustness in scenarios with limited marker availability.

CIA efficiently highlights unknown cell types

To further test the ability of our classifier to detect the presence of unanticipated cell types, we performed a classification on the Seurat human cord blood mononuclear cells (CBMC) from the 'Using Seurat with Multimodal data' vignette (https://satijalab.org/seurat/articles/multimodal_vignette) (Fig. 5A, B).

The CBMC dataset contains human blood cell types and additional mouse cells (~ 5%) used as negative controls. Signatures previously extracted from the PBMC Atlas were

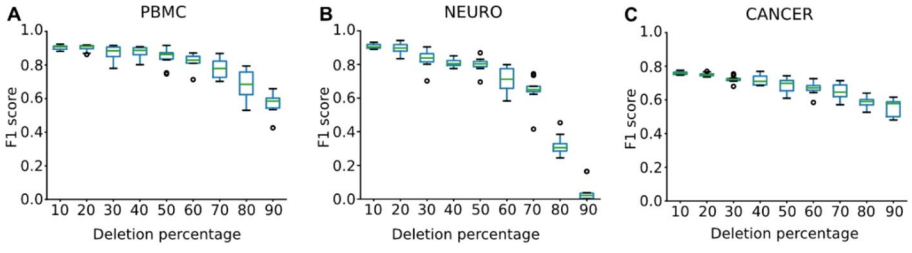

Fig. 4 **A**–**C** Boxplots show the distribution of F1 scores across 10 iterations for increasing levels of random gene deletion (10–90%) in the input signatures. Each reduced signature set was used for classification with *CIA_classify*, and performance was evaluated using *classification_metrics*. Results are shown for **A** PBMC3k, **B** NEURO_test, and **C** KUL3 datasets

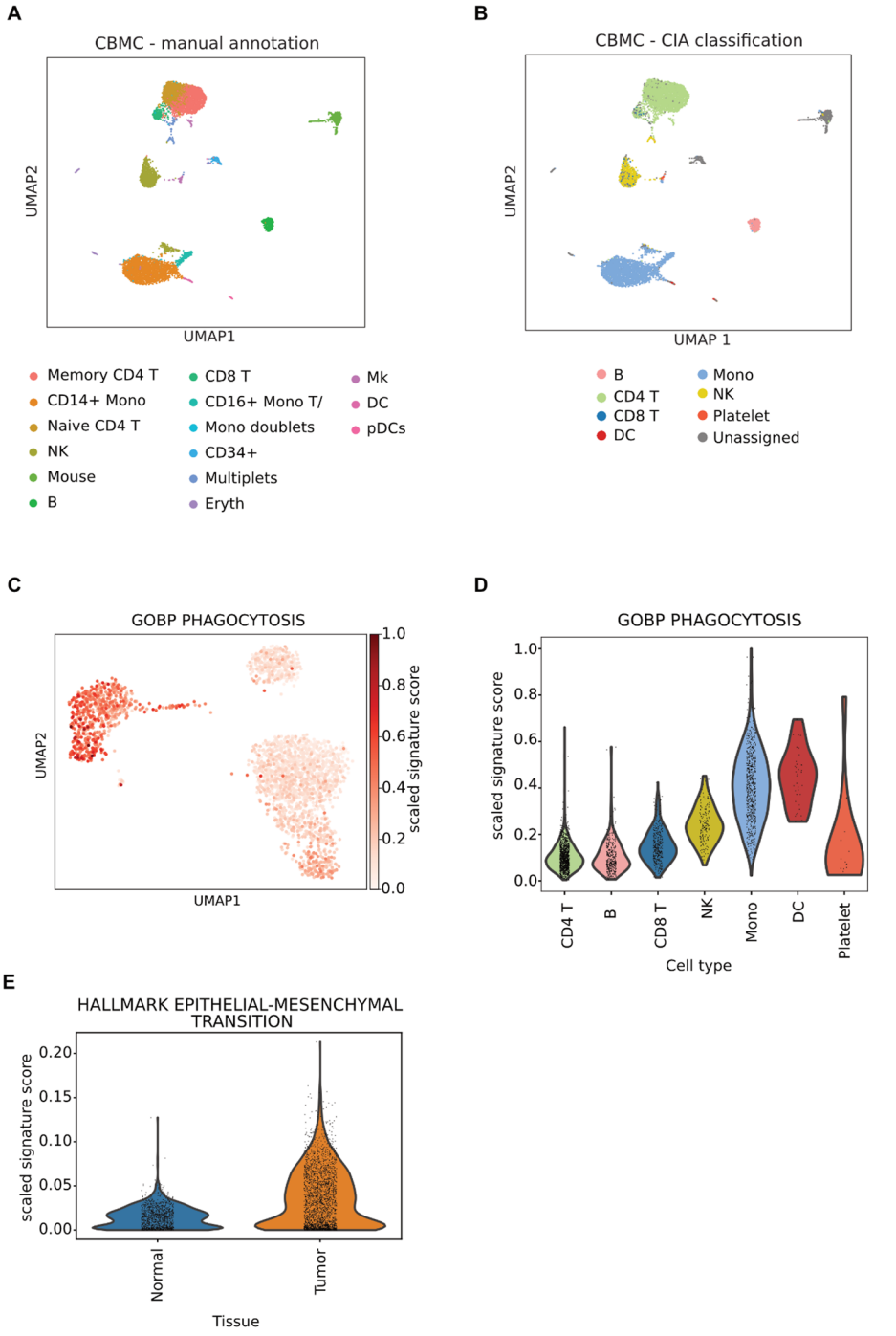

Fig. 5 **A** UMAP of the CBMC dataset with ground-truth annotation. **B** UMAP of CBMC showing CIA labels using PBMC Atlas signatures. The similarity threshold labels as 'Unassigned' (gray) cells whose profiles are not represented by the provided signatures (e.g., mouse cells and doublets). **C** UMAP of PBMC3k showing scaled signature scores of 'GOBP_PHAGOCYTOSIS'. **D** Violin plots of 'GOBP_PHAGOCYTOSIS' scores in PBMC3k grouped by ground-truth labels. **E** Violin plots of 'HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION' across normal and tumor epithelial cells in KUL3 dataset

used for the classification. Although the PBMC Atlas signatures represent cell populations at lower granularity compared to the annotations provided with the CBMC dataset, CIA was able to correctly distinguish such populations. Strikingly, we correctly

labeled as 'Unassigned' those cells whose transcriptional profiles were not represented by the provided signatures (78.45% of mouse cells and 90.38% of human erythrocytes).

CIA enables the fast investigation of functional signatures

Beyond cell type recognition across datasets, we explored the capability of CIA to identify cells exhibiting specific biological functions. To this end, we calculated the scaled signature score of the 'GOBP_PHAGOCYTOSIS' (GO:0006909) signature from the Molecular Signature Database [33] (Fig. 5C). By providing the URL pointing to the GMT file to the score_signature function, we obtained a quantitative layer representing phagocytic activity, which was then superimposed on the UMAP visualization of the entire dataset. The highest score values were observed in the clusters corresponding to Monocytes (Mono) and Dendritic Cells (DC), which are known to exhibit phagocytic activity (Mann–Whitney U test $p < 0.01$), while all the other cell groups had low scores (Fig. 4D).

To further corroborate CIA's ability to highlight specific functional enrichments, we examined additional signatures across distinct datasets.

First, the 'GOMF_STRUCTURAL_CONSTITUENT_OF_MYELIN_SHEATH' (GO:0019911) signature was significantly enriched in oligodendrocytes within the neuro training dataset, consistent with their established role in myelination (Figure S7A, B, one-sided t-test $p < 0.01$).

Secondly, in epithelial cells from the KUL3 colorectal cancer dataset, the 'HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION' signature was significantly upregulated in tumor versus normal samples (Mann–Whitney U test $p < 0.01$), reflecting the activation of the epithelial–mesenchymal transition, a biological process known to be implicated in colorectal cancer progression (Fig. 5E).

These results demonstrate how CIA can be effectively used for rapid functional characterization at single cell resolution.

Methods

Signature score

The *score_all_signatures* function calculates signature scores as previously described in [34]. In brief, this approach takes as input an object with normalized gene counts matrix as AnnData, SeuratObject or SingleCellExperiment object, a dictionary of gene signatures (either directly or in GMT format) and an option to set the score modality. The function calls *score_signature*, in which each cell j is evaluated to determine how representative it is of the predefined gene signature S. Our scoring approach involves the following steps:

First we calculate the proportion of genes within the signature S that are expressed in cell j.

Next, we determine the relative contribution of the expressed signature genes to the total gene expression in cell j. Specifically we sum expression counts of the genes in S that are expressed in cell j and divide this sum by the total expression count of all genes expressed in cell j.

The final score thus reflects both the proportion of signature genes that are expressed and their relative contribution to the total gene expression within the cell.

The 'raw' gene signature score (GSS) for signature S and cell j is defined as:

$$\mathrm{GSS(S,\ j)} = \left(\frac{n_j}{L}\right) \times \left(\frac{\sum_{i \in S \cap G_j} X_{ij}}{\sum_{i \in G_j} X_{ij}}\right)$$

Where:

S is the set of genes in the signature;
n_j is the number of genes in the signature S that have non-zero expression in cell *j;*
L is the total number of genes in the signature S;
G_j is the set of genes expressed (with counts > 0) in cell *j;*
X is the gene expression matrix where X_{ij} represents the expression of gene i in cell j.

The score mode input can be used to specify the type of scores to be computed, with options such as 'raw', 'scaled', 'log', 'log2' or 'log10'.

When the option score_mode is set to 'scaled', raw scores of each cell are divided by the maximum raw score in the dataset, in order to obtain values in the range [0–1].

Classification

The *CIA_classify* function uses *score_all_signatures* in *scaled* mode to compute the scores for input gene signatures and subsequently classify each cell based on these scores. Each cell is assigned the label of the signature with the highest score. To enhance classification accuracy, the function includes a *similarity_threshold* parameter. This threshold specifies the minimum difference required between the highest and the second-highest scores for a confident assignment. If the difference is below this threshold, the cell is labeled as 'Unassigned'. The function also supports parallel processing through the *n_cpus* parameter, enabling efficient computation on large datasets.

Score distribution analysis

The function *grouped_distributions* can be used to determine if the distributions of scores associated with a particular signature are significantly different from others. For each annotated group of cells (e.g. clusters), considered as ground truth annotations, this function applies a Wilcoxon test to verify if the distribution of the scores associated to the signature with the highest median is significantly different from the distributions of scores associated with other signatures. The function then tests, by Mann–Whitney U test, if the distribution of scores of the best signature for each cluster is significantly different from the distribution of scores for the same signature in other clusters. The function returns a heatmap plot of the median scores for each signature in each group, and a warning message if one of the tests has a *p*-value > 0.01.

Data processing

Extraction of signatures from PBMC Atlas

Cell population signatures for the classification of PBMC3k dataset were extracted from the PBMC Atlas [23], a single-cell atlas including 161,764 cells. We omitted 6789 cells labeled as 'other T' since they are not present in the PBMC3k dataset. The 'other' cluster, predominantly platelets, was relabeled as 'Platelet' for clarity. Based on the recent weighted nearest neighbor framework, this atlas integrates gene expression measurements with those of cell surface proteins quantified by the feature barcoding (10 × technology), thus providing a very specific and reliable annotation. Cell populations of the

atlas include three different levels of granularity. For our analyses, the coarser level was chosen (level I). To extract signatures from the atlas we performed a classical differential expression analysis (DEA) exploiting the *scanpy.tl.rank.genes_groups* (with default parameters) of the remaining annotated populations (B lymphocytes, CD4 + and CD8 + T lymphocytes, dendritic cells, monocytes, natural killer cells and 'other' cells, mainly composed by platelets). DEG lists were subsequently filtered through the *filter_degs* to set thresholds on the fold-change (1.5), expression mean (0.25), z-score (5) and percentage of cells within each cluster which express the gene (40%).

NEURO_test dataset preparation

NEURO_test [28] dataset was built starting from the AnnData Object 'Human MTG 10 × SEA-AD' dataset available on the Allen Brain Map website (https://portal.brain-map.org/). Since the Azimuth signatures are not meant to distinguish 'Chandelier' or 'Pax6 + ' progenitor cells from the others, those cells have been removed to avoid the misinterpretation of results.

Colorectal cancer data

CRC_training [32] and KUL3 [29] datasets were reconstructed using the preprocessed raw data deposited at GEO under accession GSE178341 and GSE144735, respectively. Cell annotation between the two datasets was harmonized to facilitate the process of performance evaluation. 'Monocyte', 'Granulocyte', 'cTNI22' cell types from GSE178341 and all the CMS subtypes, 'Unspecified Plasma', 'Epithelial cells', 'Proliferating' and 'Unknown' cell clusters from KUL3 were removed since not cross-annotated between the two datasets. Then, the datasets are processed according to the requirements of each classifier. For the functional analysis, all the cells of KUL3 and the coarse grained annotation provided by the authors were used.

Reference signatures and harmonization across methods

To harmonize biological priors across methods, we explicitly documented the knowledge sources used by each tool (Supplementary Table S1). CIA and AUCell were both provided with the same signature files in GMT format, which were independently derived from annotated reference datasets: the PBMC Atlas for the PBMC3k dataset and Cancer_training dataset (GSE178341) for KUL3. For Neuro_Test we used the curated signatures from Azimuth. These signature files are available at https://github.com/ingmbioinfo/CIA_benchmark/rebuttal_datasets/CIA. Model-based tools such as CellTypist and SingleR, which do not accept marker lists directly, were applied or retrained on exactly the same annotated references used to derive the CIA and AUCell signatures, thus ensuring consistency across approaches. Garnett classifiers were likewise trained on the PBMC Atlas, the NEURO_training dataset obtained from the Allen Institute motor-cortex dataset, and the CRC atlas, following the same rationale. In addition, to broaden the benchmarking we included deep learning–based methods such as scANVI and scBalance, which were trained or applied on the same reference datasets used for the other tools (Table S1). This strategy guarantees that all methods relied on the same prior biological information, thereby preventing biases due to mismatched knowledge sources or potential information leakage from the test sets.

Evaluation of CIA performances and comparison with other classifiers

For classification performance evaluation, we employed our *compute_classification_metrics* function to calculate widely used metrics:

Sensitivity (Recall) TP/(TP + FN), measuring the proportion of true positives correctly identified.

Specificity TN/(TN + FP), measuring the proportion of true negatives correctly identified.

Precision (Positive Predictive Value) TP/(TP + FP), quantifying the fraction of predicted positives that are true positives.

Accuracy (TP + TN)/(TP + TN + FP + FN), representing the proportion of correctly classified cells among all cells.

F1-score (2 × Precision × Sensitivity)/(Precision + Sensitivity), the harmonic mean of precision and sensitivity.

These metrics were derived by comparing the classifications from CIA, Celltypist (v1.6.2), Garnett (v0.2.20), SingleR(v2.4.1), scANVI (v1.4.0), scBalance (Github 2023 release) and AUCell (implemented in the Python package OmicVerse [35] (v1.5.6) with the ground truth annotations provided by the datasets. To assess the computational efficiency, we measured the running time of the classification process while incrementing the number of CPUs allocated (1, 2, 4, 8, 16, 32, and 64 CPUs). Runtime experiments were executed on a single HPC node with ~ 500 GB RAM (RealMemory 506,000 MiB ≈ 494 GiB) and 128 logical CPUs. Each configuration was repeated 10 times per classifier and dataset to quantify run-to-run variability. Exceptions: (i) scANVI was evaluated only at 16, 32, and 64 CPUs with n = 3 repeats across all datasets due to long training times (e.g. ~ 2 days estimated at 1 CPU for the cancer dataset); (ii) SingleR, Garnett, and Celltypist used n = 3 repeats on the cancer dataset (and n = 10 elsewhere). Celltypist classifies cells by applying a combined approach based on logistic regression and stochastic gradient descent. The Celltypist model was trained using the *celltypist.train* function with feature selection enabled. Subsequently, the model was applied to the test dataset using the *celltypist.annotate* function. Garnett trains a classifier using elastic net multinomial regression. Models were trained on the training datasets using the *train_cell_classifier* function with *min_observations* set to 50 and *max_training_samples* set to 800. Cell assignment on the test datasets was performed using the *classify_cells* function. Concerning the SingleR package, the *SingleR* function, which is a wrapper of *trainSingleR* and *classifySingleR* functions, was applied with default parameters. We ran scANVI in a semi-supervised setting: an SCVI model was first pretrained on all cells (training and test dataset are merged) and a 30-dimensional latent space was computed. scANVI was then initialized with the training labels while test cells were set to 'Unknown'. Both stages were trained on CPU (20 epochs; batch size 1,024), with parameters chosen to strike a practical balance between accuracy and running time. Predictions for the held-out test split were obtained via the model's predict API. We executed scBalance on CPU using the intersection of genes shared by train and test; training labels were taken from the 'Cell type' column, and *scBalance* function was applied with default parameters to generate test-set predictions, mapping numeric codes back to the original label names. AUCell was evaluated employing the function *ov.single.pathway_aucell* to obtain AUC scores. Once AUCell scores were obtained, cells were classified according to the

top-scoring signature leveraging the same set of signature genes employed by CIA. All the code used for the benchmark, along with a step by step explanation is publicly available at https://github.com/ingmbioinfo/CIA_benchmark/.

Gene signature list trimming analysis

To assess the stability of CIA classification under gene signature trimming, we generated multiple reduced versions of each signature by randomly deleting 10–90% of genes in 10% increments. For each deletion percentage, we performed 10 independent iterations. Each reduced gene signature was then used to classify the input dataset with *CIA_classify*, and the resulting performances were calculated using *classification_metrics*.

Discussion

In this study we developed a novel computational tool called CIA, which accurately identifies cell types across diverse datasets without relying on a fully annotated reference dataset or complex machine learning processes. Our approach lays its foundation on the unique transcriptional signatures associated with each cell type, enabling CIA to assign cell type labels with high accuracy by comparing the expression profiles of query cells against a set of pre-computed signatures.

CIA is designed for ease of use, featuring comprehensive documentation that facilitates its integration into existing single-cell analysis workflows. Its open-source code is freely available on GitHub, with implementations in both Python (https://github.com/ingmbioinfo/cia) and R (https://github.com/ingmbioinfo/CIA_R). This dual compatibil ity allows for seamless use alongside popular single-cell analysis tools such as Scanpy and Seurat. One of the distinguishing features of CIA is its independence from the clustering step, empowering users to explore biologically relevant cell groups prior to defining clustering resolutions. The rapid classification capabilities of CIA enable it to handle large datasets in seconds, providing an additional layer of prior knowledge that clustering algorithms often overlook. This capability allows for more confident refinement of cluster boundaries, ultimately enhancing the understanding of the dataset under investigation.

While pretrained models like scBERT [36] and scGPT [37] are powerful tools in single-cell research, CIA focuses on simplicity, accessibility, and transparency. It avoids the 'black-box' nature of deep learning by relying on interpretable gene signatures and an intuitive algorithm. CIA is lightweight, requiring only standard computational resources, and is particularly suited for researchers prioritizing rapid and resource-efficient classification without extensive training datasets. By complementing rather than competing with deep learning models, CIA offers a streamlined, user-friendly solution for cell-type annotation.

Unlike many existing tools, CIA does not require a training dataset derived from similar conditions, which can introduce bias associated with specific datasets. By mitigating these issues, CIA facilitates accurate and efficient cell labeling. A notable aspect of CIA is the inclusion of a similarity threshold that can prevent the classification of cells with scores that are closely aligned among signatures. This feature is particularly advantageous as it highlights cells for which the signatures may be inappropriate or those exhibiting ambiguous transcriptional profiles. Consequently, it helps to avert misclassification

and uncovers potential new cell types or biological conditions that might otherwise be overlooked.

As a proof of concept, we demonstrated the efficacy of our classifier using the CBMC dataset annotation, showing that CIA can accurately detect cell populations corresponding to the input signatures, while designating those that do not fit as 'Unassigned' negative controls. Overall, our method is comparable in classification performance to the other state-of-the-art approaches while significantly reducing computational runtime and enhancing interpretability. Although the classification of 'Unassigned' cells may appear as a reduction in performance, it offers an opportunity to generate new hypotheses regarding the nature of these cells within single-cell datasets. Given the proliferation of signature databases and large community cell atlases, computational efficiency may become a critical consideration. The straightforward workflow, coupled with rapid execution and the interpretability afforded by gene signatures, positions CIA as a valuable asset in the era of big-omics.

In addition to its core modules, CIA provides several utilities that enable users to visualize signature enrichment scores on UMAPs, generate contingency matrices, compare classifications against reference datasets, evaluate classification performance, and plot classification results on UMAPs. By integrating this information into existing AnnData, Seurat, or SingleCellExperiment objects, CIA simplifies the graphical representation of signature enrichment scores and classification outcomes, thereby streamlining incorporation into any of the established single-cell workflows. Furthermore, in addition to being a robust alternative to cluster-based methods, CIA can assist in annotating independently obtained clusters. By natively integrating the majority vote feature from Celltypist [21], it also harmonizes labels within clusters, ensuring consistent and accurate annotations. Using standardized types to store data and the computed results, the output of CIA can be seamlessly further explored in interactive applications such as iSEE [38] or CELLxGENE [39], as demonstrated in the package vignette.

Beside these functionalities, it is important to note that the performance of CIA is influenced and limited by the characteristics of the input data. When the starting gene signatures are highly overlapping, their discriminative power is reduced, potentially leading to ambiguous classifications. Likewise, performance may be suboptimal in datasets that are very sparse, poorly aligned with the input signatures, or contain novel or poorly characterized cell types for which robust signatures have not yet been generated. CIA was designed as a fast, easy-to-use classifier purpose-built for iterative, operator-guided workflows: it allows users to perform multiple rapid classification attempts with independently refined signatures, leveraging domain knowledge to converge on robust annotation for new datasets. Recognizing these contexts helps to set realistic expectations for CIA's application and highlights the importance of careful signature selection and data quality assessment when interpreting results. To mitigate these challenges, CIA provides dedicated functionalities and practical guidance. By computing the Jaccard similarity between signatures, users can assess the degree of their overlap and, through the similarity threshold parameter, prevent forced classification—cells receiving comparable scores for overlapping signatures are instead labeled as 'Unassigned,' thereby flagging potential issues in signature design. The combined use of Jaccard similarity and the similarity threshold can also reveal misalignment between input signatures and the dataset under classification: when otherwise distinct signatures yield consistent

'Unassigned' results across multiple cells, this may indicate the presence of a missing or uncharacterized cell identity. Given the ease with which signature-shaped information can be generated or retrieved within the current and future single-cell landscape [19], we anticipate that CIA will become increasingly valuable for obtaining quantitative overviews, ultimately enhancing the extraction of insights and comprehensive interpretation of data. Our findings suggest that CIA is an effective tool for scRNA-seq analysis, enabling researchers to identify cell types and biological functions in their datasets without the need for extensive training or optimization, all within a relatively short time frame. Additionally, we expect CIA's performance to scale linearly with increasing cell numbers, making it amenable for use on standard laptops equipped with typical computational resources.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06320-z.

Supplementary Material 1

Acknowledgements
Not applicable.

Author contributions
I.F. and M.B. contributed equally to this work and share first authorship. The conceptual design of the method was collaboratively developed by M.B., I.F., E.G., and J.C. I.F., M.B., A.G., E.G., and F.V. developed the Python code for the classifier, while I.F. and F.M. were responsible for the development of the R package. Extensive testing of the Python code was conducted by I.F., M.B., A.G., E.G., J.C., and F.V., with F.M. overseeing the testing of the R package. I.F., A.G., and F.M. were responsible for the publication and maintenance of the code on GitHub, ensuring ongoing accessibility and updates. The project was supervised by S.B., R.G., S.A., and E.G., who provided critical oversight and guidance throughout the study. S.N. contributed significantly to the biological validation of the results. The manuscript was primarily written by E.G., I.F., and M.B., with all authors contributing to its revision and approval of the final version.

Funding
This work was funded by: the Italian Association for Cancer Research (AIRC) under the 5x1000 Program ID 21091; the European Union under Grant Agreement No. 101191725 (INFLAME project); the National Recovery and Resilience Plan (PNRR) of the Italian Ministry of University and Research under project code PE00000007 (INF-ACT project); the contribution of resources from the European Union, the Italian State and the Lombardy Region under the Collabora&Innova programme, project code 6013188 (TERA-IMMUN project).

Data availability
The datasets supporting the conclusions of this article are available at the following links: PBMC3k: https://www.10xgenomics.com/datasets/3-k-pbm-cs-from-a-healthy-donor-1-standard-1-1-0 and https://github.com/ingmbioinfo/cia/blob/master/docs/tutorial/data/pbmc3k.h5ad. PBMC Atlas: https://cellxgene.cziscience.com/collections/b0cf0afa-ec40-4d65-b570-ed4ceacc6813. Neuro_Training: https://seurat.nygenome.org/azimuth/demo_datasets/allen_m1c_2019_ssv4.rds. Neuro_Test: https://portal.brain-map.org/atlases-and-data/rnaseq/human-mtg-10x_sea-ad. Cancer Training: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE178341. KUL3: https://ega-archive.org/datasets/EGAD00001008584. CBMC dataset: ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE100nnn/GSE100866/suppl/GSE100866_CBMC_8K_13AB_10X-RNA_umi.csv.gz. Pre-processed datasets and GMT files: https://doi.org/10.5281/zenodo.17241856. Availability and Requirements: Project name: CIA (Cluster Independent Annotation) Project home page: https://pypi.org/project/cia-python/1.0a4/. Project source code: Python: https://github.com/ingmbioinfo/cia. R: https://github.com/ingmbioinfo/CIA_R. Archived version: 1.0a4. Operating system(s): Platform independent. Programming language: Python, R. Other requirements: Python: seaborn, numpy, pandas, AnnData, scanpy. R: Imports section of the DESCRIPTION file available at https://github.com/ingmbioinfo/CIA_R/blob/main/DESCRIPTION. License: Any restrictions to use by non-academics: None. The software is released under the MIT License. Benchmark repository: https://github.com/ingmbioinfo/CIA_benchmark.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 9 June 2025 / Accepted: 6 November 2025

Published online: 17 December 2025

References

1. Chen G, Ning B, Shi T. Single-cell RNA-seq technologies and related computational data analysis. Front Genet. 2019;10:317. https://doi.org/10.3389/fgene.2019.00317.
2. Nguyen QH, Pervolarakis N, Nee K, Kessenbrock K. Experimental considerations for single-cell RNA sequencing approaches. Front Cell Dev Biol. 2018;6:108. https://doi.org/10.3389/fcell.2018.00108.
3. Wagner A, Regev A, Yosef N. Revealing the vectors of cellular identity with single-cell genomics. Nat Biotechnol. 2016;34:1145–60. https://doi.org/10.1038/nbt.3711.
4. Amezquita RA, Lun ATL, Becht E, Carey VJ, Carpp LN, Geistlinger L, et al. Orchestrating single-cell analysis with Bioconductor. Nat Methods. 2020;17:137–45. https://doi.org/10.1038/s41592-019-0654-x.
5. Grabski IN, Street K, Irizarry RA. Significance analysis for clustering with single-cell RNA-sequencing data. Nat Methods. 2023;20:1196–202. https://doi.org/10.1038/s41592-023-01933-9.
6. Duò A, Robinson MD, Soneson C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. F1000Res. 2020;7:1141. https://doi.org/10.12688/f1000research.15666.3.
7. Pasquini G, Rojo Arias JE, Schäfer P, Busskamp V. Automated methods for cell type annotation on scRNA-seq data. Comput Struct Biotechnol J. 2021;19:961–9. https://doi.org/10.1016/j.csbj.2021.01.015.
8. Clarke ZA, Andrews TS, Atif J, Pouyabahar D, Innes BT, MacParland SA, et al. Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. Nat Protoc. 2021;16:2749–64. https://doi.org/10.1038/s41596-021-00534-0.
9. Choi YH, Kim JK. Dissecting cellular heterogeneity using single-cell RNA sequencing. Mol Cells. 2019;42:189–99. https://doi.org/10.14348/molcells.2019.2446.
10. Abdelaal T, Michielsen L, Cats D, Hoogduin D, Mei H, Reinders MJT, et al. A comparison of automatic cell identification methods for single-cell RNA sequencing data. Genome Biol. 2019;20:194. https://doi.org/10.1186/s13059-019-1795-z.
11. Quan F, Liang X, Cheng M, Yang H, Liu K, He S, et al. Annotation of cell types (ACT): a convenient web server for cell type annotation. Genome Med. 2023;15:91. https://doi.org/10.1186/s13073-023-01249-5.
12. Hu C, Li T, Xu Y, Zhang X, Li F, Bai J, et al. Cell marker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scRNA-seq data. Nucleic Acids Res. 2023;51:D870–6. https://doi.org/10.1093/nar/gkac947.
13 Franzén O, Gan L-M, Björkegren JLM. PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. Database. 2019;2019:baz046. https://doi.org/10.1093/database/baz046.
14. Meng F-L, Huang X-L, Qin W-Y, Liu K-B, Wang Y, Li M, et al. singleCellBase: a high-quality manually curated database of cell markers for single cell annotation across multiple species. Biomark Res. 2023;11:83. https://doi.org/10.1186/s40364-023-00523-3.
15. Patil A, Patil A. CellKb Immune: a manually curated database of mammalian hematopoietic marker gene sets for rapid cell type identification. 2020. https://doi.org/10.1101/2020.12.01.389890
16. Lotfollahi M, Naghipourfar M, Luecken MD, Khajavi M, Büttner M, Wagenstetter M, et al. Mapping single-cell data to reference atlases by transfer learning. Nat Biotechnol. 2022;40:121–30. https://doi.org/10.1038/s41587-021-01001-7.
17. Hemberg M, Marini F, Ghazanfar S, Ajami AA, Abassi N, Anchang B, et al. Insights, opportunities and challenges provided by large cell atlases. arXiv; 2024. https://doi.org/10.48550/ARXIV.2408.06563
18. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19:15. https://doi.org/10.1186/s13059-017-1382-0.
19 Lun A, Aut C. SingleCellExperiment. Bioconductor. 2017. https://doi.org/10.18129/B9.BIOC.SINGLECELLEXPERIMENT.
20. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol. 2018;36:411–20. https://doi.org/10.1038/nbt.4096.
21. Domínguez Conde C, Xu C, Jarvis LB, Rainbow DB, Wells SB, Gomes T, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. Science. 2022;376:eabl5197. https://doi.org/10.1126/science.abl5197.
22. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun. 2017;8:14049. https://doi.org/10.1038/ncomms14049.
23. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021;184:3573-3587.e29. https://doi.org/10.1016/j.cell.2021.04.048.
24. Pliner HA, Shendure J, Trapnell C. Supervised classification enables rapid annotation of cell atlases. Nat Methods. 2019;16:983–6. https://doi.org/10.1038/s41592-019-0535-3.
25. Aran D, Looney AP, Liu L, Wu E, Fong V, Hsu A, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. Nat Immunol. 2019;20:163–72. https://doi.org/10.1038/s41590-018-0276-y.
26. Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Mol Syst Biol. 2021;17:e9620. https://doi.org/10.15252/msb.20209620.
27. Cheng Y, Fan X, Zhang J, Li Y. A scalable sparse neural network framework for rare cell type annotation of single-cell transcriptome data. Commun Biol. 2023;6:545. https://doi.org/10.1038/s42003-023-04928-6.
28. Gabitto M, Travaglini K, Ariza J, Kaplan E, Long B, Rachleff V, et al. Integrated multimodal cell atlas of Alzheimer's disease. Nat Neurosci. 2023;27(12):2366–83. https://doi.org/10.21203/rs.3.rs-2921860/v1.
29. Lee H-O, Hong Y, Etlioglu HE, Cho YB, Pomella V, Van Den Bosch B, et al. Lineage-dependent gene expression programs influence the immune landscape of colorectal cancer. Nat Genet. 2020;52:594–603. https://doi.org/10.1038/s41588-020-0636-z.
30. Aibar S, González-Blas CB, Moerman T, Huynh-Thu VA, Imrichova H, Hulselmans G, et al. SCENIC: single-cell regulatory network inference and clustering. Nat Methods. 2017;14:1083–6. https://doi.org/10.1038/nmeth.4463.
31. Bakken TE, Jorstad NL, Hu Q, Lake BB, Tian W, Kalmbach BE, et al. Comparative cellular analysis of motor cortex in human, marmoset and mouse. Nature. 2021;598:111–9. https://doi.org/10.1038/s41586-021-03465-8.
32. Pelka K, Hofree M, Chen JH, Sarkizova S, Pirl JD, Jorgji V, et al. Spatially organized multicellular immune hubs in human colorectal cancer. Cell. 2021;184:4734-4752.e20. https://doi.org/10.1016/j.cell.2021.08.003.

33. Liberzon A, Subramanian A, Pinchback R, Thorvaldsdóttir H, Tamayo P, Mesirov JP. Molecular signatures database (MSigDB) 3.0. Bioinformatics. 2011;27:1739–40. https://doi.org/10.1093/bioinformatics/btr260.
34. Della Chiara G, Gervasoni F, Fakiola M, Godano C, D'Oria C, Azzolin L, et al. Epigenomic landscape of human colorectal cancer unveils an aberrant core of pan-cancer enhancers orchestrated by YAP/TAZ. Nat Commun. 2021;12:2340. https://doi.org/10.1038/s41467-021-22544-y.
35. Zeng Z, Ma Y, Hu L, Tan B, Liu P, Wang Y, et al. Omicverse: a framework for bridging and deepening insights across bulk and single-cell sequencing. Nat Commun. 2024;15:5983. https://doi.org/10.1038/s41467-024-50194-3.
36. Yang F, Wang W, Wang F, Fang Y, Tang D, Huang J, et al. ScBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. Nat Mach Intell. 2022;4:852–66. https://doi.org/10.1038/s42256-022-00534-z.
37. Cui H, Wang C, Maan H, Pang K, Luo F, Duan N, et al. ScGPT: toward building a foundation model for single-cell multi-omics using generative AI. Nat Methods. 2024;21:1470–80. https://doi.org/10.1038/s41592-024-02201-0.
38. Rue-Albrecht K, Marini F, Soneson C, Lun ATL. iSEE: interactive summarized experiment explorer. F1000Res. 2018;7:741. https://doi.org/10.12688/f1000research.14966.1.
39. Megill C, Martin B, Weaver C, Bell S, Prins L, Badajoz S, et al. cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices. 2021. https://doi.org/10.1101/2021.04.05.438318

Publisher's Note