Benchmarking second-generation methods for cell-type deconvolution of transcriptomic data

Alexander Dietrich[1,*], Lorenzo Merotto[2,*], Konstantin Pelz[1], Bernhard Eder[2], Constantin Zackl[2], Katharina Reinisch[3], Frank Edenhofer[4], Federico Marini[5,6], Gregor Sturm[7,8], Markus List[1,9,†], Francesca Finotello[2,†]

[1]Data Science in Systems Biology, TUM School of Life Sciences, Technical University of Munich, 85354 Freising, Germany
[2]Department of Molecular Biology, Digital Science Center (DiSC), University of Innsbruck, 6020 Innsbruck, Austria
[3]Institute for Informatics, Ludwig-Maximilians-Universität München, 80333 München, Germany
[4]Department of Molecular Biology, Center for Molecular Biosciences Innsbruck (CMBI), University of Innsbruck, 6020 Innsbruck, Austria
[5]Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg University Mainz, 55131 Mainz, Germany
[6]Research Center for Immunotherapy (FZI), 55131 Mainz, Germany
[7]Biocenter, Institute of Bioinformatics, Medical University of Innsbruck, 6020 Innsbruck, Austria
[8]Boehringer Ingelheim International Pharma GmbH & Co KG, 88397 Biberach, Germany
[9]Munich Data Science Institute (MDSI), Technical University of Munich, 85748 Garching, Germany
*Equal contribution
[†]Equal contribution

Corresponding authors: Markus List (markus.list@tum.de), Francesca Finotello (francesca.finotello@uibk.ac.at).

Abstract

In silico cell-type deconvolution from bulk transcriptomics data is a powerful technique to gain insights into the cellular composition of complex tissues. While first-generation methods used precomputed expression signatures covering limited cell types and tissues, second-generation tools use single-cell RNA sequencing data to build custom signatures for deconvoluting arbitrary cell types, tissues, and organisms. This flexibility poses significant challenges in assessing their deconvolution performance. Here, we comprehensively benchmark second-generation tools, disentangling different sources of variation and bias using a diverse panel of real and simulated data. Our study highlights the strengths, limitations, and complementarity of state-of-the-art tools shedding light on how different data characteristics and confounders impact deconvolution performance. We provide the scientific community with an ecosystem of tools and resources, *omnideconv*, simplifying the application, benchmarking, and optimization of deconvolution methods.

Tissues and organs comprise various cell types, determining their structure and function. Characterizing the cellular composition of tissues is essential for studying cell development, homeostasis, and disease. In recent years, several in silico deconvolution methods have been developed to estimate the cellular composition of tissue samples profiled with bulk RNA sequencing (RNA-seq). Deconvolution algorithms consider gene expression profiles of a heterogeneous sample as the weighted sum of the gene expression profiles of the admixed cells, and estimate the unknown cell fractions leveraging cell-type-specific transcriptomic signatures[1]. While single-cell RNA-seq (scRNA-seq) enables studying the transcriptomes underlying cellular identities at unprecedented resolution and granularity[2], it is not suited to accurately quantify the cellular composition of tissues. This is mainly due to differences in single-cell dissociation efficiency, which can bias cell-type proportions[3]. Moreover, single-cell protocols entail considerable costs and technical challenges, making their application unattractive for profiling large sample collections. Thus, bulk transcriptome profiling remains popular, motivating further research into in silico cell-type deconvolution.

Earlier deconvolution tools are based on precomputed signatures covering a few cell types. In the past, the focus was mainly on human anticancer immunity[4], where these methods have been validated extensively[5]. The need for more flexible methodologies and the rapid pace of development of single-cell omics have motivated the development of a second generation of deconvolution tools[6] that directly learn cell-type-specific signatures, namely expression signatures or models, from annotated (i.e. cell-type labeled) scRNA-seq data. These methods allow, in principle, the deconvolution of any cell type across arbitrary tissues and organisms, as long as reference single-cell data is available. As second-generation methods derive deconvolution signatures “on the fly”, depending on the user-specified data, characterizing their accuracy and robustness in different contexts requires systematic and comprehensive benchmarking that differs from previous studies focused on first-generation methods. While some second-generation algorithms have been previously tested[7–14], major challenges in deconvolution benchmarking remain unaddressed[14–16]. These include assessing the methods’ ability to quantify rare or closely related cell types, and determining the impact of biological and technical biases on deconvolution performance.

In this study, we carry out a comprehensive benchmarking study of second-generation deconvolution tools leveraging a balanced and rationally designed set of simulated and experimental ground-truth data, while ensuring reproducibility and reusability. To disentangle and systematically assess the impact of various biological and technical confounders on methods performance, we used our previously developed simulator SimBu[17], which allows for the efficient generation of synthetic bulk RNA-seq datasets, i.e. ‘pseudo-bulks’ generated by controlled aggregation of single-cell expression profiles. SimBu allows the modeling of cell-type-specific mRNA levels, an important bias that deconvolution methods have to account for[1], and which was disregarded in previous benchmarking studies. We complemented our set of pseudo-bulk data with real RNA-seq samples from different tissues and organisms with matching ground-truth cell fractions. Overall, we assembled a compendium of more than 1,400 real and simulated RNA-seq samples and matched ground-truth cell fractions to systematically test method performance in different scenarios.

Benchmarking studies typically represent a snapshot of available methods in a fast-evolving field, making it challenging to consider additional tools and datasets in follow-up investigations[18]. To overcome this limitation, we assembled a freely available ecosystem of tools and resources called *omnideconv* (omnideconv.org), which enables the simplified use and assessment of cell-type deconvolution methods (Figure 1A). Besides SimBu and our collection of validation datasets (*deconvData*), it includes: 1) *omnideconv,* a novel R package that offers uniform access to several second-generation deconvolution methods, 2) *deconvBench*, a Nextflow[19] pipeline to reproduce and extend the presented benchmarking study, and 3) *deconvExplorer,* a web app to interactively investigate deconvolution signatures and results. We envision that the omnideconv ecosystem will aid researchers in deconvolving RNA-seq samples more easily and offer guidance for method choice in different scenarios. In addition, the flexibility of our applications allows for easy extension, which is a necessary feature to include upcoming deconvolution methods, as well as to benchmark and optimize them for specific applications.

Results

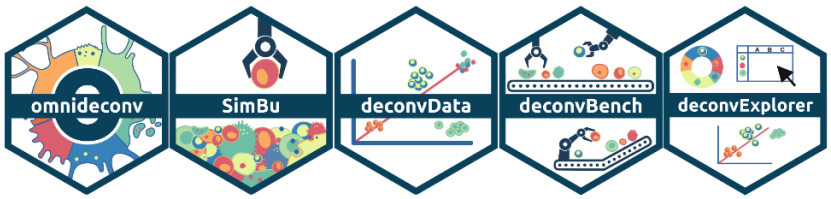
The omnideconv ecosystem enables the simplified application and benchmarking of second-generation deconvolution methods

Second-generation deconvolution methods differ in programming language, workflow, input data types, and processing. This complicates the simultaneous execution of multiple methods and the comparison of their estimates. To simplify the usage of second-generation deconvolution methods, we have developed *omnideconv* (https://github.com/omnideconv/omnideconv/), an R package providing a unified interface to multiple R- and Python-based methods. *omnideconv* currently supports twelve methods: AutoGeneS[20], BayesPrism[21], Bseq-SC[22], Bisque[23], CDseq[24], CIBERSORTx[25], CPM[26], DWLS[27], MOMF[28], MuSiC[29], SCDC[30], and Scaden[31]. It unifies the methods' workflows, input and output data, and semantics, allowing deconvolution analysis using one or two simple commands.

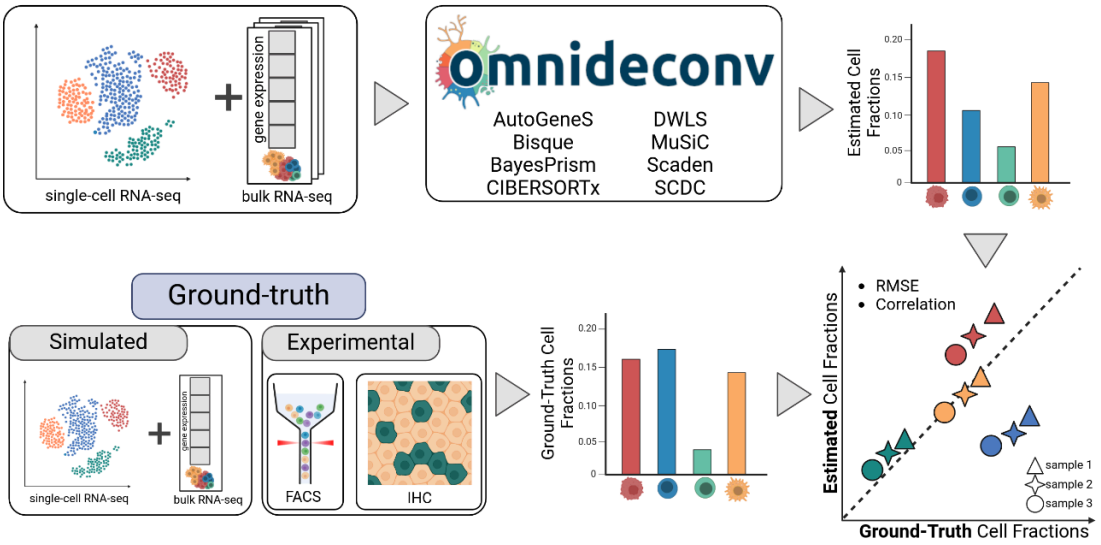
For this study, we selected eight methods (AutoGeneS, BayesPrism, Bisque, CIBERSORTx, DWLS, MuSiC, SCDC, and Scaden) that: 1) leverage annotated scRNA-seq data to directly perform cell-type deconvolution (rather than reference-free deconvolution followed by a posteriori annotation of derived cell phenotypes); 2) do not strictly require context-specific marker genes; and 3) provide deconvolution results in the form of relative cell fractions. We implemented and evaluated an optimized version of DWLS in *omnideconv,* with considerably more effective usage of computational resources ("DWLS optimized", Methods). In contrast to earlier studies assessing normalization and parameter optimization[9,18], we consulted the methods' developers for guidance on optimal parameter settings and data type usage. In the absence of feedback, we ran the corresponding method with default parameters and considered counts and transcripts per million (TPM) as single-cell and bulk RNA-seq input data, respectively (Methods).

Our benchmarking study (Figure 1B) comprises human and mouse bulk RNA-seq datasets for which ground-truth estimates of cell fractions were available from fluorescence-activated cell sorting (FACS) or immunohistochemistry (IHC), as well as pseudo-bulk datasets generated using SimBu[17] (*deconvData*). SimBu was used to systematically test methods' performance in various scenarios, evaluating their robustness to different confounders. In particular, we assessed the impact of different factors that can bias cell-type estimates: 1) cell-type-specific mRNA bias, since cells with higher overall mRNA abundance can be overestimated, and vice versa; 2) unknown cellular content, i.e. cell types present in the bulk RNA-seq data to be deconvolved but not in the scRNA-seq reference used for method training; 3) transcriptional similarity between closely-related cell types; 4) technology and tissue/disease context of the single-cell reference data (Figure 1C). Moreover, we assessed the impact of the resolution of cell type annotations, and how deconvolution performance and computational scalability are impacted by the size of the reference single-cell data (Figure 1D).
All the analyses performed in this benchmarking study were implemented in a well-documented, reusable, and extensible Nextflow[19] pipeline called *deconvBench* (https://github.com/omnideconv/deconvBench).

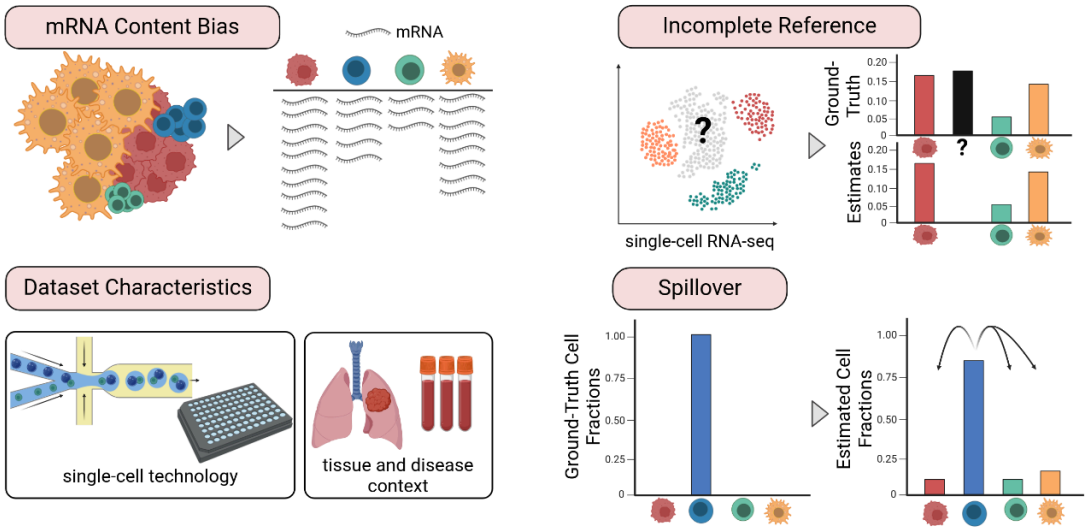
A) Omnideconv ecosystem



B) Benchmarking framework



C) Challenges addressed



D) Single-cell reference characteristics

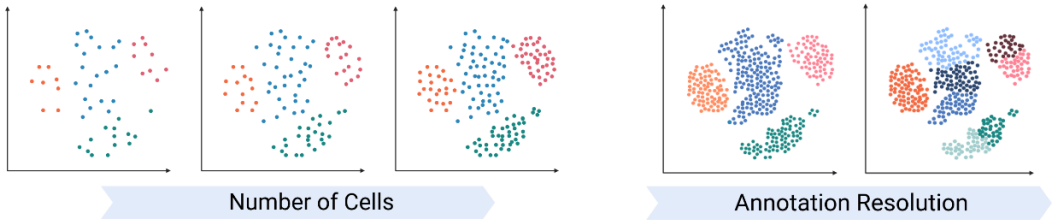


Figure 1: omnideconv ecosystem and benchmark
(A) The omnideconv benchmarking ecosystem offers five tools (from left to right): the R package *omnideconv* providing a unified interface to deconvolution methods, the pseudo-bulk simulation method SimBu, the *deconvData* data repository, the *deconvBench* benchmarking pipeline in Nextflow and the web-app *deconvExplorer*. **(B)** Outline of the benchmark experiment: scRNA-seq and bulk RNA-seq data is used as input for several methods, and a unified output of estimated cell-type fractions for each bulk sample is calculated. We compare the estimated fractions to ground-truth fractions (known from pseudo-bulks or FACS/IHC experiments) and compute performance measures per method and cell type. **(C)** Several challenges in cell-type deconvolution are addressed in this benchmark: (1) cell types show total mRNA bias; (2) scRNA-seq datasets vary by technology, tissue and disease; (3) a fraction of the cells may be of unknown type, since the scRNA-seq reference does not necessarily contain all cell types present in the bulk mixture; (4) some cell types are more similar on a transcriptomic level, leading to “spillover” towards similar cell types. **(D)** We evaluated two major parameters that can often be adapted by users of deconvolution methods and affect estimation quality: (1) the number of cells for each annotated cell type in the scRNA-seq reference dataset and (2) the level of annotation granularity. This figure was created with Biorender.com.

Deconvolution performance on real and pseudo-bulk RNA-seq datasets

We evaluated the methods’ performance on immune-cell type quantification, a key task in deconvolution analysis[5,6]. We selected two blood-cell-derived bulk RNA-seq datasets (*Finotello*[32]: n=9 and *Hoek*[33]: n=8) for which ground-truth cell fractions were available from FACS. For method training, we selected a CITE-seq single-cell dataset from human peripheral blood mononuclear cells[34] (PBMC) (Methods, Suppl. Figure S1). We fed 10% of the data to SimBu[17], maintaining the original cellular composition (*HaoSub* dataset, with 15,314 cells encompassing eleven cell types). The generated pseudo-bulk datasets mirror the cell-type composition and sequencing depth of the *Finotello* and *Hoek* RNA-seq samples, while using only cell types that existed in both datasets (*FinotelloSim* and *HoekSim* pseudo-bulk datasets, Methods). This collection of real and pseudo-bulk data allowed the comparison of predicted and ground truth cell-type fractions in easy and complex scenarios, respectively, and the evaluation of deconvolution performance in terms of Pearson correlation and root-mean-square error (RMSE) (Methods).

On human pseudo-bulk samples, all methods exhibited high correlation and low RMSE (Figure 2A, C). Using the same single-cell dataset for simulation and signature building can be considered the simplest scenario for deconvolution, especially for methods that create pseudo-bulks to train their internal model[31]. Despite the overall good performance, a systematic estimation bias for B cells, natural killer (NK) cells, and monocytes was evident for BayesPrism and Bisque.

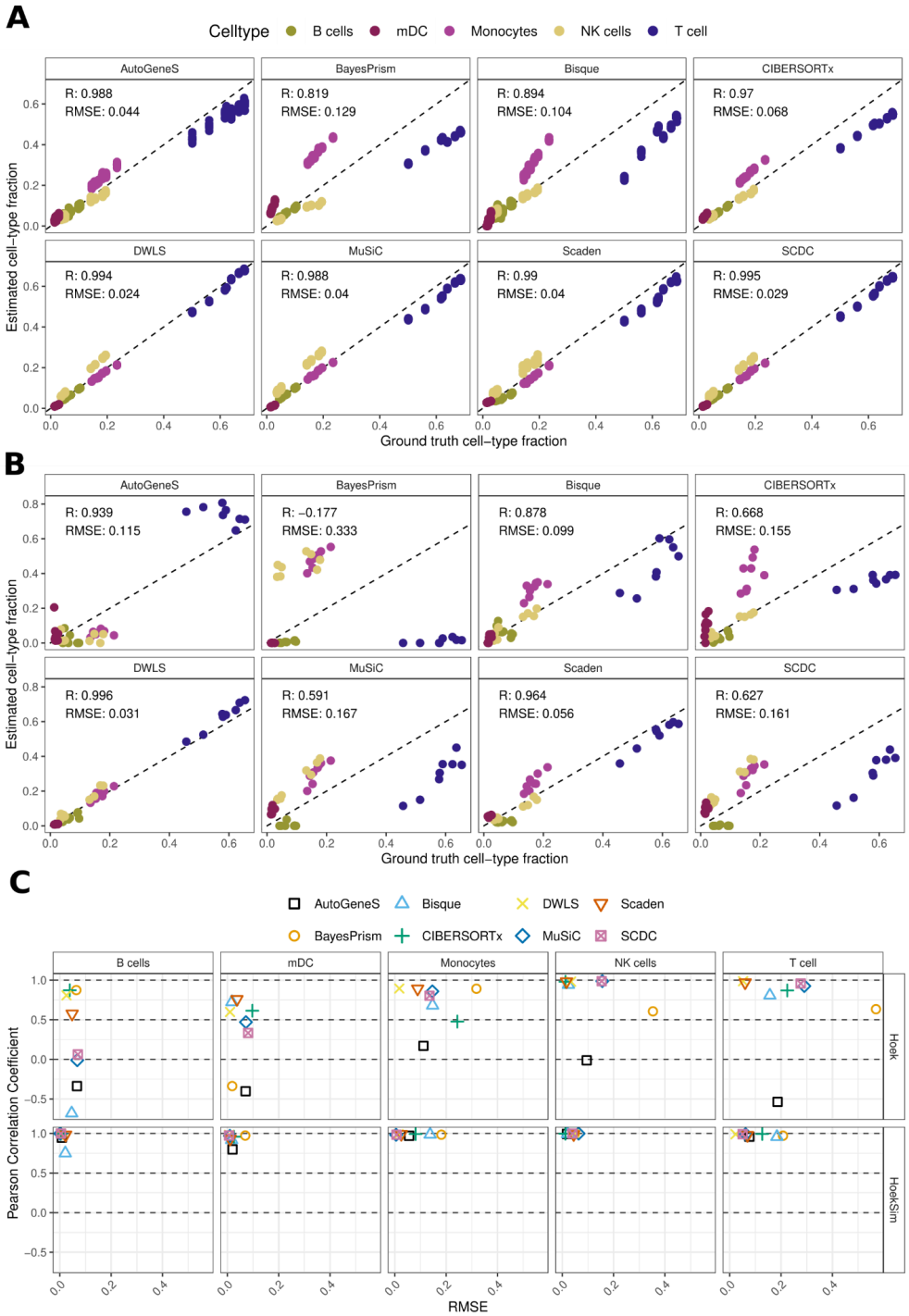


Figure 2: Deconvolution performances of eight second-generation methods

Predicted cell-type fractions of eight methods are compared against the ground truth fractions (either FACS or simulated). The performances on bulk (**A**, *Hoek*[33], n=8) and pseudo-bulk (**B**, *HoekSim,* n=80) datasets are displayed. The scRNA-seq reference dataset is *HaoSub*[34]. Pearson's correlation coefficients (R) and RMSE values are given for each method and dataset, aggregated by mean across cell types. Each point corresponds to one bulk sample (cell type is color-coded); the dashed line represents a perfect prediction. **C** shows the cell-type-specific RMSE and correlation values for all methods and the same two datasets. See details on simulation setup in Methods.
Results for three other datasets can be found in Suppl. Figures S2 and S3.

The real bulk RNA-seq datasets represented a more challenging scenario, with more diverse methods' performance. On the *Hoek* dataset (Figure 2B,C), Scaden and DWLS outperformed all other methods based on *global* correlation (i.e. computed considering all cell types together), surpassing coefficients of 0.95, followed by AutoGeneS and Bisque. When additionally considering RMSE, Scaden showed slightly worse performance than DWLS due to a systematic *estimation bias* (under- or over-estimation) for monocytes and T cells, an issue that also affected all other methods to varying extents. BayesPrism, MuSiC, and Scaden overestimated NK cells, increasing their global RMSE. All methods apart from AutoGeneS and BayesPrism displayed high cell-type-specific correlations for T and NK cells, whereas they showed extensive spread in RMSE, indicating less precise estimates (Figure 2C). Interestingly, the opposite behavior could be seen for B cells and myeloid dendritic cells (mDCs), which are rare in the *Hoek* dataset. Their low abundance challenged the methods resulting in low correlation values and their RMSE values were intrinsically limited compared to the other cell types. This example, together with the opposite case of monocyte overestimation (i.e. high correlation and high RMSE), shows how both performance metrics complement each other in method validation, providing complementary insights.

On the *Finotello* dataset (Suppl. Figures S2A, S3), Scaden again obtained a high correlation (R=0.903). DWLS performance significantly declined due to T regulatory (T_{REG}) cell overestimation and, consequently, conventional $CD4^+$ T cell underestimation. AutoGeneS had a similar problem, showing a large overestimation of $CD8^+$ T cells. As on the *Hoek* dataset, BayesPrism displayed cell-type-specific estimation bias*:* it over-estimated monocytes and NK cells and under-estimated $CD4^+$ and $CD8^+$ T cells. Between the two real bulk datasets, *Finotello* and *Hoek*, Scaden showed the best performance overall. Except for Scaden, in the *Finotello* data set, all methods showed false negative predictions (i.e. lack of detection) for $CD4^+$, $CD8^+$ T cells, and B cells.

To evaluate the methods' performance on a different organism, we considered two publicly available mouse bulk RNA-seq datasets with matched immune-cell fractions from FACS (*Chen*[35]: n=12 and *Petitprez*[36]: n=14), as well as their pseudo-bulk counterparts (*ChenSim* and *PetitprezSim*). We used a subset of the spleen Tabula Muris scRNA-seq dataset for training (*TM* with 9,083 cells, Suppl. Figure S1). Performance again differed substantially between simulated (correlation coefficients close to 1; only Bisque dropped to R=0.796) and real datasets (Suppl. Figures S2B,C and S3), where Scaden (R=0.771, RMSE=0.164), DWLS (R=0.768, RMSE=0.166), and CIBERSORTx (R=0.752, RMSE=0.158) showed the best overall performance.

Impact of single-cell reference size

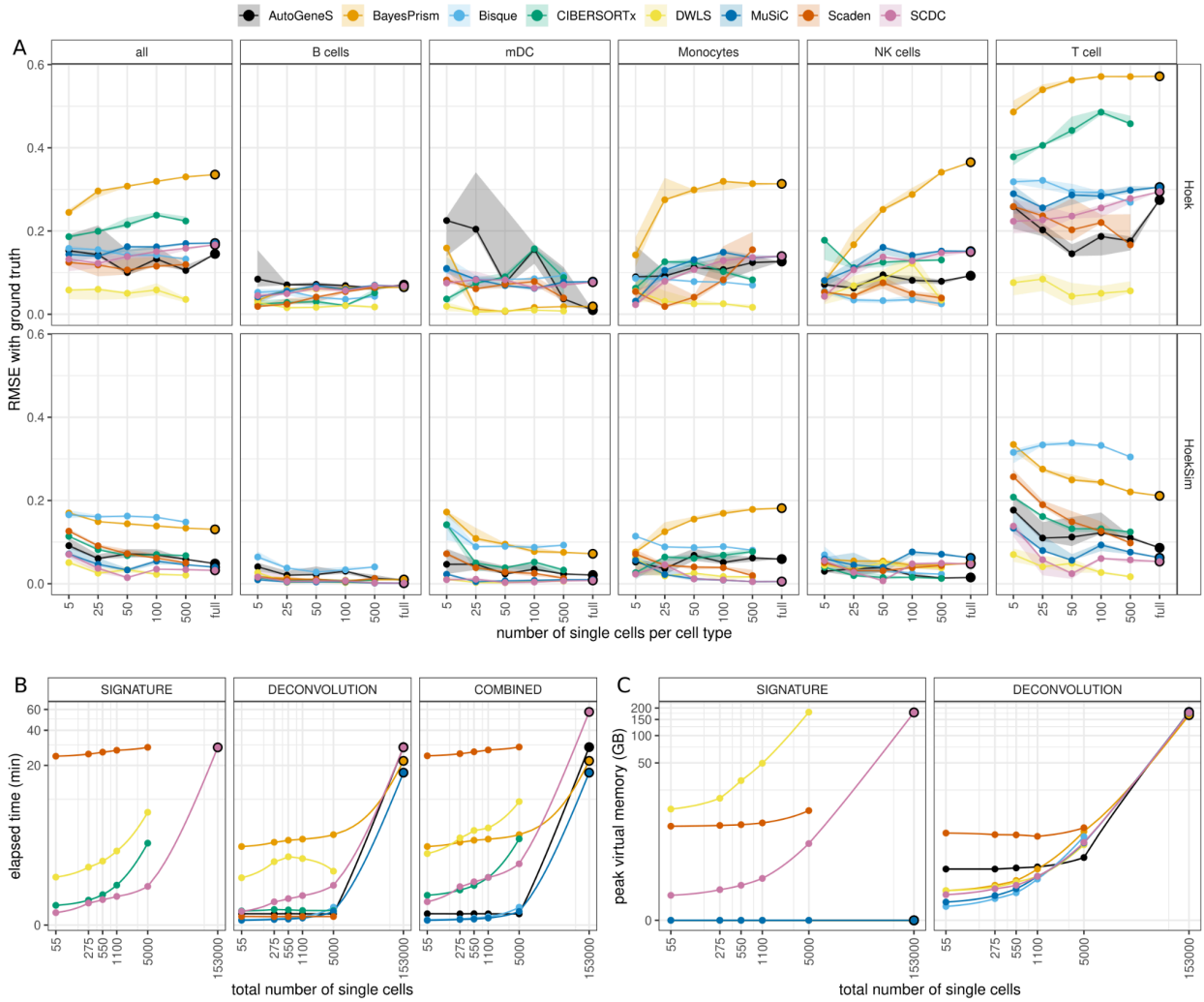


Figure 3: Increasing the reference size shows minor deconvolution performance improvement with more available cells.

Performance and resource consumption for different deconvolution methods applied on *Hoek* and *HoekSim* datasets. Methods were given an increasing number of cells (randomly downsampled) from the *Hao* scRNA-seq reference (Methods). We calculated the median RMSE (**A**) for five technical replicates. The shaded areas show the replicates' 25% and 75% quantiles (n=5). The black outlined points show the result on the full *Hao* reference; four methods (Bisque, DWLS, CIBERSORTx, Scaden) did not finish calculations on the full reference. The boxes in the first column ('all') display the RMSE values over all samples and cell types, while the following boxes show cell-type-specific RMSE values. Additionally, the computational resources of each method were tracked in terms of elapsed time (**B**) and maximal memory usage (**C**). BayesPrism, Bisque, and MuSiC do not build a separate signature in omnideconv; therefore, their runtime and memory usage for this step is 0 in all cases. AutoGeneS consumes most of its runtime to save the created signature to a file (Methods). For CIBERSORTx, we could not track memory usage due to its dockerized design.

Single-cell datasets are constantly increasing in size[37], but it is unclear to what degree second-generation deconvolution methods benefit from larger reference data sets and how they scale in terms of runtime and memory demands. Thus, we subsampled single-cell profiles from each annotated cell type in the *Hao* reference dataset, gradually increasing the reference size

while keeping the proportions of cell types uniform. The different subsets (with five technical replicates each) were then used to train the methods for the deconvolution of the real and pseudo-bulk versions of the *Hoek* and *Finotello* datasets (Methods).

On pseudo-bulk data, almost all methods showed the expected improvement in performance, visible in decreasing cell-type-specific and overall RMSE with increasing reference size (Figure 3A and Suppl. Figure S4). Similarly, predictions of cell-type fractions become more robust, as indicated by the narrower interquartile range. While this trend could be spotted for all tools, its extent differed between cell types. B and NK cells were accurately quantified with only five cells, and a larger reference set provided limited gain. T cells showed a more pronounced performance gain with increasing sizes across all methods, as the reference consisted of more diverse subtypes. The differences between predictions and true fractions flattened out beyond 25 cells per cell type in most cases (Suppl. Figure S4B). BayesPrism was the only method showing increased RMSE for monocytes with increasing reference size.
On real datasets, prediction quality was much lower overall and on a cell-type level (Figure 3A and Suppl. Figure S4). RMSE values for B cells were best for all methods, followed by similar performances on mDCs, monocytes, and NK cells. T cells showed the widest spread of RMSE performance between methods. The prediction quality also differed between datasets, as methods such as SCDC and MuSiC showed a considerably lower correlation in the *Finotello* dataset (R=0.25 and R=0.25, respectively; 500 cell subset over all cell types) compared to the *Hoek* one (R=0.64 and R=0.58, respectively). Overall, the decrease in RMSE for larger single-cell reference sizes seen in pseudo-bulk settings was recapitulated on real bulk samples.

For some tools, a larger reference size resulted in systematic estimation bias, most notably in the real bulk datasets (Suppl. Figure S4B). For instance, BayesPrism and CIBERSORTx consistently moved towards overestimation of monocytes. Scaden benefitted the most from a larger reference size, potentially due to a better optimization of its deep learning model. In general, methods affected by cell-type-specific bias (Figure 2A), did not resolve this issue with a larger reference.

We next assessed methods' scalability with respect to reference size (Figure 3B, C), considering signature creation and deconvolution separately wherever possible (i.e. for CIBERSORTx, DWLS, SCDC, Scaden). Only MuSiC, BayesPrism, AutoGeneS, and SCDC were able to use the full *Hao* dataset (~153,000 cells) for training within reasonable memory restrictions (Methods), underscoring the importance of reference data subsampling. For the full dataset, runtimes ranged from 12 to 50 minutes using between 150 and 200 GB of memory. MuSiC and Bisque had the fastest combined runtime (< 30s up until 5,000 cells), followed by SCDC (~120s for 5,000 cells). CIBERSORTx, BayesPrism, and AutoGeneS took about five minutes for 5,000 cells. Scaden was the slowest method (~27 min for 5,000 cells); this was mainly related to its signature calculation, which was computationally the most intensive task across methods. Scaden, Bisque, and MuSiC were the fastest tools for the deconvolution step. Interestingly, DWLS, AutoGeneS and, to a lesser extent, BayesPrism deconvolution runtimes did not increase with reference size –DWLS even got faster with more cells. DWLS was the most memory-intensive method, using up to 180 GB of memory for signature building (5,000

cells). Conversely, AutoGeneS had the lowest memory footprint, with only ~3.3 GB for the same number of cells.

Impact of reference atlas cellular resolution

Single-cell atlases can be annotated with cell labels representing coarse lineages or more fine-grained subsets and states. We considered two scRNA-seq reference datasets from lung and breast cancers (*Lambrechts*[3] and *Wu*[38], respectively) to test the impact of varying cellular resolutions on deconvolution performance (Methods, Table S1).

DWLS and MuSiC showed the most robust performance across all resolutions in both datasets (Figure 4A, and Suppl. Fig. S6), as their RMSE and correlation values remained in a similar range ($R > 0.75$, $RMSE < 0.1$) for almost all cell types. Scaden and CIBERSORTx showed less consistent results for T-cell subtypes and larger RMSE at coarse resolution. Bisque showed the weakest performance on fine resolution and the strongest performance increase towards coarse annotations. Conversely, BayesPrism and AutoGeneS showed decreased performance towards coarser resolutions, especially for T and NK cells.
Interestingly, AutoGeneS, BayesPrism, CIBERSORTx, and Scaden benefitted from running deconvolution at a more fine-grained resolution, and subsequently aggregating deconvolution results into coarse cell types (Figure 4B), suggesting that this agglomerative strategy could be routinely considered.

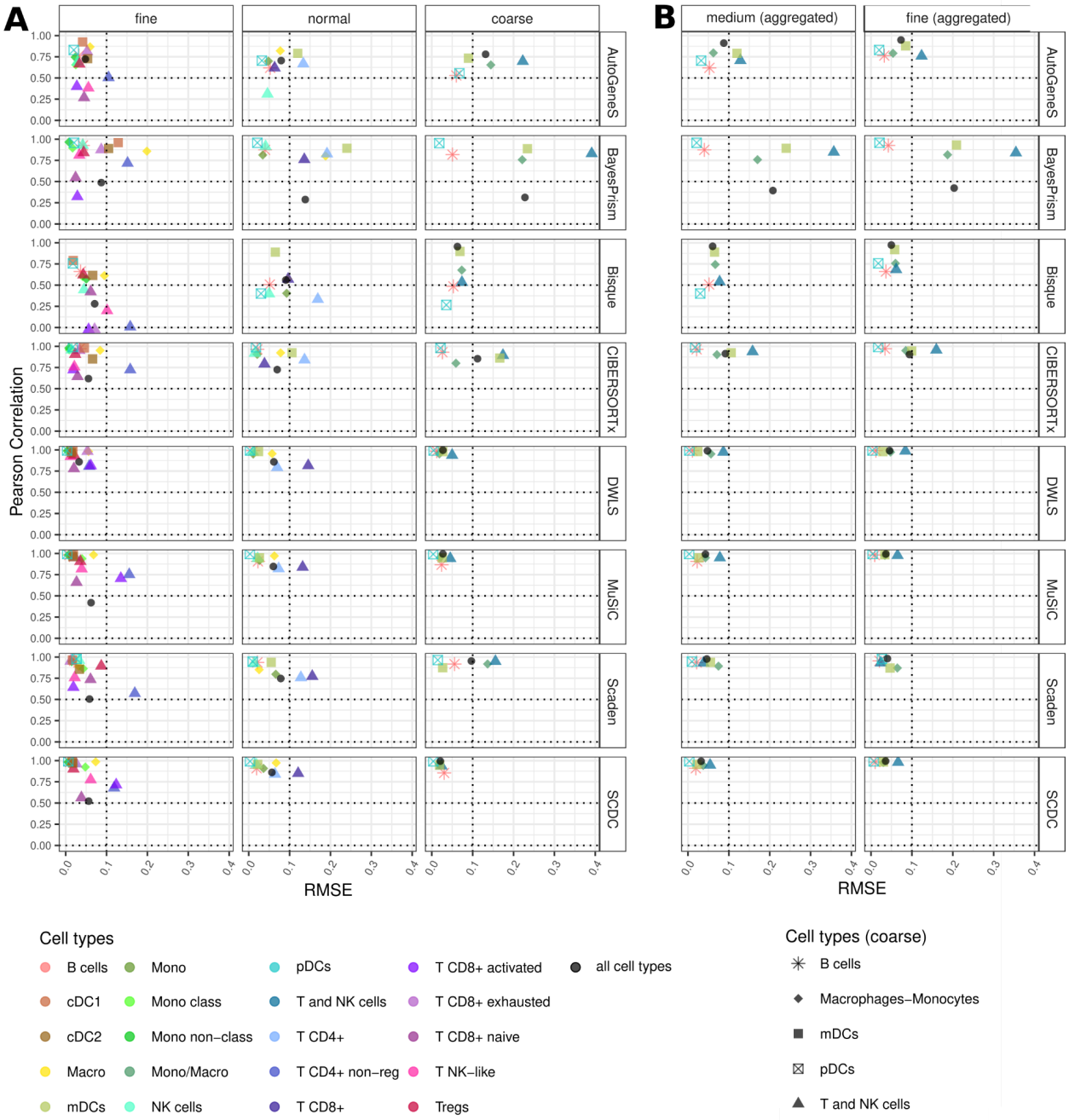


Figure 4: Methods performances with different annotation granularities of the reference, Hoek dataset.
(A) Pearson correlation coefficient and RMSE values computed for the cell-type-specific estimates obtained on pseudo-bulks (n=50) simulated from the Lambrechts datasets. **(B)** Cell-type fractions at finer resolution levels (fine, medium) were aggregated to obtain "coarse" estimates (Methods, Table S1).
Celltype abbreviations: $CD4^+$ T cells non-regulatory (T CD4+ non-reg), conventional dendritic cells 1 (cDC1), conventional dendritic cells 2 (cDC2), Macrophages (Macro), myeloid dendritic cells (mDCs), Monocytes (Mono), Monocytes classical (Mono class), Monocytes non-classical (Mono non-class), Monocytes/Macrophages (Mono/Macro), natural killer cells (NK cells), T cells NK-like (T NK-like), plasmacytoid dendritic cells (pDCs), T regulatory cells (Tregs).

Sources of systematic bias

We have seen that methods systematically over- and under-estimate specific cell types (Figures 2, S2-4). We investigated three aspects that potentially contribute to this bias: cell-type-specific mRNA content, unknown cellular content in the (pseudo-)bulk RNA-seq data, and the transcriptional similarity of cell types.

Cell types differ in their gene expression profiles but also in their size and metabolic activity[39], resulting in differences in the amount of mRNA they synthethize[40]. Methods not properly accounting for this bias may be prone to systematic estimation bias[1,15]. We used SimBu to create pseudo-bulk samples from the *Hao* dataset with and without cell-type-specific mRNA bias. As we previously showed[17], this bias can be modeled using the number of expressed genes per cell (Supplementary Figure S1). DWLS, MuSiC, and SCDC accounted for mRNA bias in most cell types (except $CD4^+$ T cells and NK cells), indicated by a lower RMSE on bias-aware compared to unbiased simulations (i.e. a negative difference in Suppl. Figure S5A), while Scaden obtained a less consistent pattern. AutoGeneS, BayesPrism, Bisque, and CIBERSORTx only sporadically increased their performance, accounting for mRNA bias in few cell types (Suppl. Figure S5B). BayesPrism, prone to over-estimations of NK cells and monocytes (Figure 2B), did not show significant differences between biased and unbiased simulations (Suppl. Figure S5B).

All the methods evaluated in this benchmarking assume that the provided single-cell dataset is an exhaustive reference of the cell types present in bulk RNA-seq to be deconvolved, and constrain the estimated cell fractions to sum up 100% of the sample. This is rarely the case in real applications, as publicly available single-cell datasets might not cover all the cell subsets present in the bulk RNA-seq data under investigation. The absence of a cell type from the training step might result in inaccurate signatures and estimation bias. To assess this, we repeatedly deconvolved the *Finotello* and *Hoek* datasets, each time removing one cell type from the *Hao* reference used for method training. We quantified the impact by calculating the difference in RMSE (Figure 5A, Suppl. Figure S7) obtained with the “incomplete” vs. the full reference. The removal of one cell type resulted in cell-type-specific estimation bias for all methods, although to varying extent. For most methods, the removal of monocytes resulted in the deterioration of mDC estimates (higher RMSE, Figure 5A). On the contrary, removing NK cells improved the estimates of T cell subsets. This effect was more evident in the *Hoek* dataset, since the T cells were not divided into subpopulations. AutoGeneS was strongly impacted by removing cell types from the reference dataset. Vice versa, Bisque was impacted the least but showed overall low accuracy.

Next, we assessed the presence of spillover-like effects, i.e. the overestimation of a cell type caused by cell types with similar transcriptional profiles (Figure 1C). We used SimBu to simulate pseudo-bulk samples composed exclusively of one cell type (Methods). DWLS was the best-performing method, with true fractions ranging in 92-100% per cell type (Figure 5C) and only 3% of total incorrect (“spillover”) predictions (Figure 5B). T-cell subpopulations posed a problem to BayesPrism (72-80%), CIBERSORTx (50-65%), MuSiC (75-84%), Scaden (79-84%),

and SCDC (75-85%), whereas other cell types were estimated robustly. AutoGeneS showed variable results, assigning an average of 76% cell fractions to the true cell type. Overall, the spillover effect was more prominent for $CD4^+$ T cells classified as T_{REG} cells and vice-versa, and NK cells classified as $CD8^+$ T cells (Figure 5B and Suppl. Figure S8). Other cell types showed only minor spillover effects across methods. Bisque failed this test, predicting equal fractions for all cell types.

Deconvolution has classically been applied to characterize immune cell infiltration in bulk-tumor RNA-seq samples, which is a key biomarker for clinical decision-making in oncology[1,41]. Immune-cell signatures can be derived from public data. Conversely, tumor cells show large variability between and even within patients[42], and tumor-matching scRNA-seq references for method training are scarce. We, hence, evaluated the impact of unknown tumor content in bulk RNA-seq data using pseudo-bulks simulated from the *Lambrechts* lung cancer dataset[3]. We mixed single-cell profiles from B cells, macrophages, $CD4^+$ T cells, and stromal cells, and introduced increasing fractions of tumor-cell profiles (Methods). Single-cell data from the same cell types, except tumor cells, was used for training. For each level of tumor content, we evaluated the deconvolution performance using Pearson correlation (Figure 5D). CIBERSORTx, DWLS, and Scaden showed very stable results, with correlations close to 1 up to 80% unknown content for the individual cell types. On global correlation (gray points), DWLS performed similarly well, followed closely by CIBERSORTx and Scaden. MuSiC and SCDC showed similar patterns, though correlation decreased earlier, at 30-50% unknown content. Overall, global correlation was more impacted than the correlations for the individual cell types, indicating an estimation bias due to the failed quantification of the tumor-cell content. Macrophages seemed the easiest cell population to be quantified for most methods. For BayesPrism and Scaden, stromal cell identification was the most problematic, while AutoGeneS struggled with $CD4^+$ T cells.

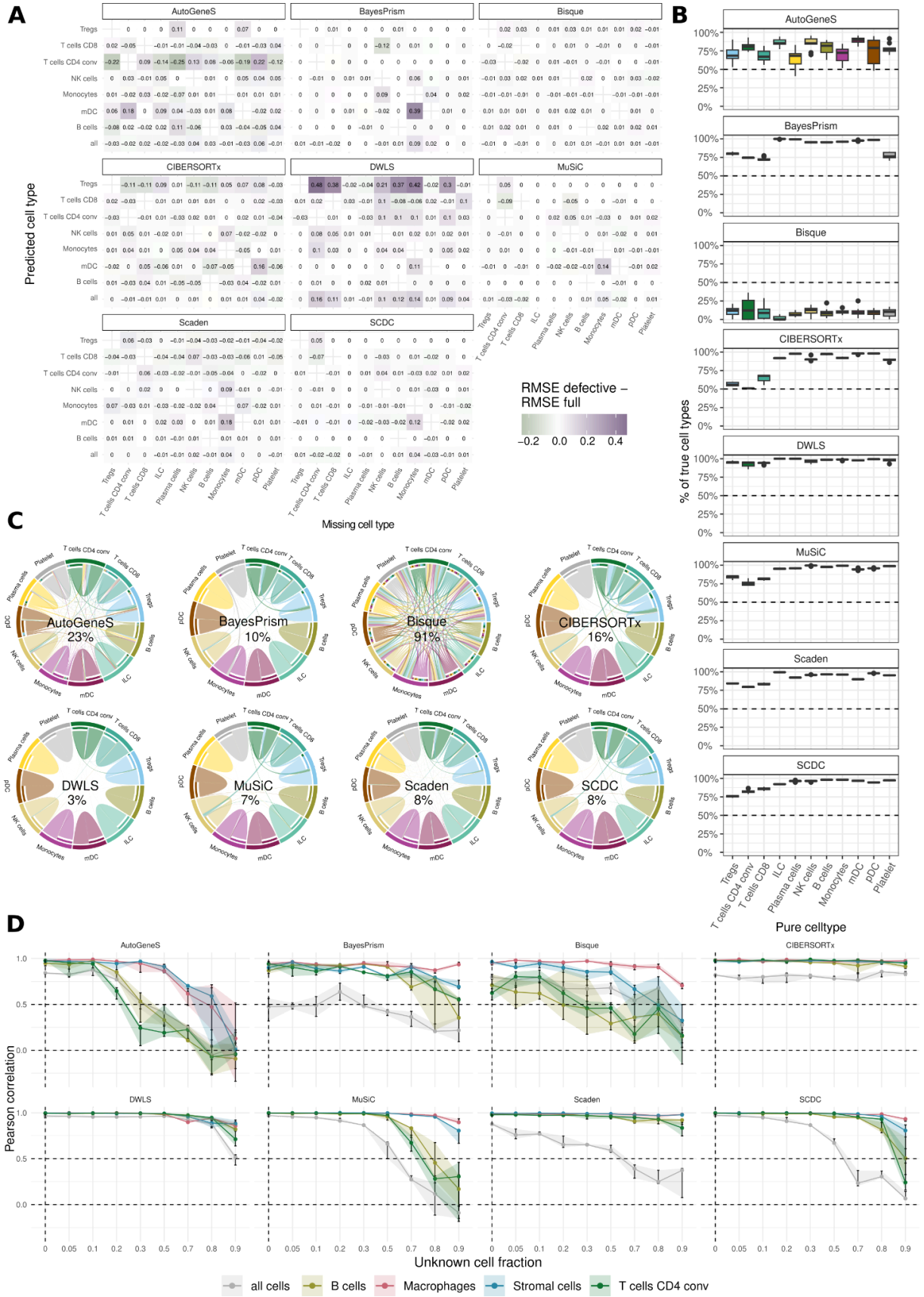

A
AutoGeneS
BayesPrism
Bisque
CIBERSORTx
DWLS
MuSiC
Scaden
SCDC
Predicted cell type
Missing cell type
Tregs
T cells CD8
T cells CD4 conv
NK cells
Monocytes
mDC
B cells
all
ILC
Plasma cells
pDC
Platelet
RMSE defective – RMSE full
−0.2 0.0 0.2 0.4
B
% of true cell types
Pure celltype
C
AutoGeneS 23%
BayesPrism 10%
Bisque 91%
CIBERSORTx 16%
DWLS 3%
MuSiC 7%
Scaden 8%
SCDC 8%
D
Pearson correlation
Unknown cell fraction
all cells
B cells
Macrophages
Stromal cells
T cells CD4 conv

Figure 5: Exploration of different sources for over-estimation include missing cell types, spillover and unknown content in the pseudo-bulk.
(A) Difference between the Root mean squared error (RMSE) between cell type predictions obtained with methods trained on the defective *Hao* reference (i.e. with one missing cell type) and cell type ground truth fractions, and RMSE obtained with the methods trained on the full reference and cell type ground truth fractions for the *Finotello* bulk dataset (n=9). A positive value means that the removal of a specific cell type worsens the deconvolution performance, and, vice versa, a negative value indicates improved performance. (**B)** Percentage of correctly predicted cell type abundance in the spillover analysis on the *Hao* reference while deconvolving pure pseudo-bulk samples (n=10 per cell type). The presented values correspond to the sum of the correctly predicted cell fractions for each method. **(C)** Chord diagrams showing the cell type predictions for the spillover analysis. The outer circle indicates the different cell types, while the chords represent the cell fractions predicted for the various samples. A connection starting and leading to the same segment represents a correct prediction, while a connection leading to a different section indicates the presence of spillover between the cell types. The values in the center indicate the total percentage of predictions attributed to wrong cell types. **(D)** Performance of the different deconvolution methods in the simulation with unknown tumor content using the *Lambrechts* scRNA-seq reference. Pseudo-bulks (n=50) were simulated using the same single-cell dataset, for each level of unknown tumor fractions. The mean Pearson correlation coefficient was computed for the predictions of each cell type. The shaded areas show the interquartile range as five technical replicates have been generated for each level of tumor content.

Impact of single-cell technology and tissue context

The performance of scRNA-seq-informed deconvolution methods might depend on the tissue and disease context, as well as on the adopted single-cell technology of the reference dataset. To assess the impact on deconvolution performance of scRNA-seq characteristics, we simulated two pseudo-bulk datasets from lung-cancer scRNA-seq datasets generated with different technologies: *Lambrechts*[3] (10x Genomics) and *Maynard*[43] (Smart-seq2). Of note, we leveraged harmonized cell-type annotations provided by the Lung Cancer Cell Atlas[44] to minimize discrepancies between cell-type labels. To deconvolve these data, we considered either one of these single-cell datasets for training. As tissue-specific references may not always be available, we used the *Hao* PBMC dataset as a reference (10x, PBMC). For both pseudo-bulk simulation and methods training, only the cell types in common across all three single-cell datasets were considered.

Overall, the most accurate results were obtained with matching pseudo-bulk and single-cell reference (Figure 6 A,B). When using training data from a different single-cell technology, the deconvolution performance deteriorated. The worst results were obtained when the training data came from a different tissue context (i.e. *Hao* PBMC data), with some T-cell subsets being undetected in the pseudo-bulks. B cells, monocytes, plasma cells and, in some settings, NK cells seemed less affected by the impact of technology and tissue context. Similarly, some methods (Scaden, DWLS, and CIBERSORTx) proved to be more robust to the impact of technology.

For signature-based (rather than model-based) methods, some determinants of deconvolution performance can be investigated in the characteristics of the derived signature matrix. Although signature assessment (only possible for half of the methods considered in this benchmarking) is beyond the scope of this study, we show an example of how our interactive web app *deconvExplorer* (https://exbio.wzw.tum.de/deconvexplorer/) can be used to compare the

signatures derived by DWLS from the two lung-cancer scRNA-seq datasets (Suppl. Figure S9, Methods). Both signatures contained similar numbers of genes (*Lambrechts*: 1112, *Maynard*: 1331), though only 359 genes were in common. The lower condition number of the Lambrechts signature (357.52) compared to the Maynard (502.28) indicates a higher discriminative power, in line with the obtained deconvolution results (Figure 6). Finally, visual inspection of the signatures represented as z-scored heatmaps shows clearly the unspecificity of various genes across T cell subtypes, additionally quantified by a lower gene-wise Gini index.

In addition to the pseudo-bulk analysis, we used all three single-cell reference datasets (this time using all available cell types) to deconvolve bulk RNA-seq data from lung tumors for which $CD4^+$ and $CD8^+$ T cell estimates were available from IHC (*Vanderbilt* dataset[32], Figure 6C). Except for Bisque, methods generally showed the poorest results using the *Hao* PBMC data set, confirming the importance of matching tissue/disease context. The impact of the single-cell technology was less clear in this assessment. Most methods performed better on the *Lambrechts* dataset, with the exception of DWLS and CIBERSORTx. This real-life application also shows that all methods struggled to robustly estimate tumor-infiltrating lymphocytes, which are transcriptionally similar and rare compared to the overall sample composition ($CD4^+$ T cells: 0.03±0.07%; $CD8^+$ T cells: 0.01±0.06%).

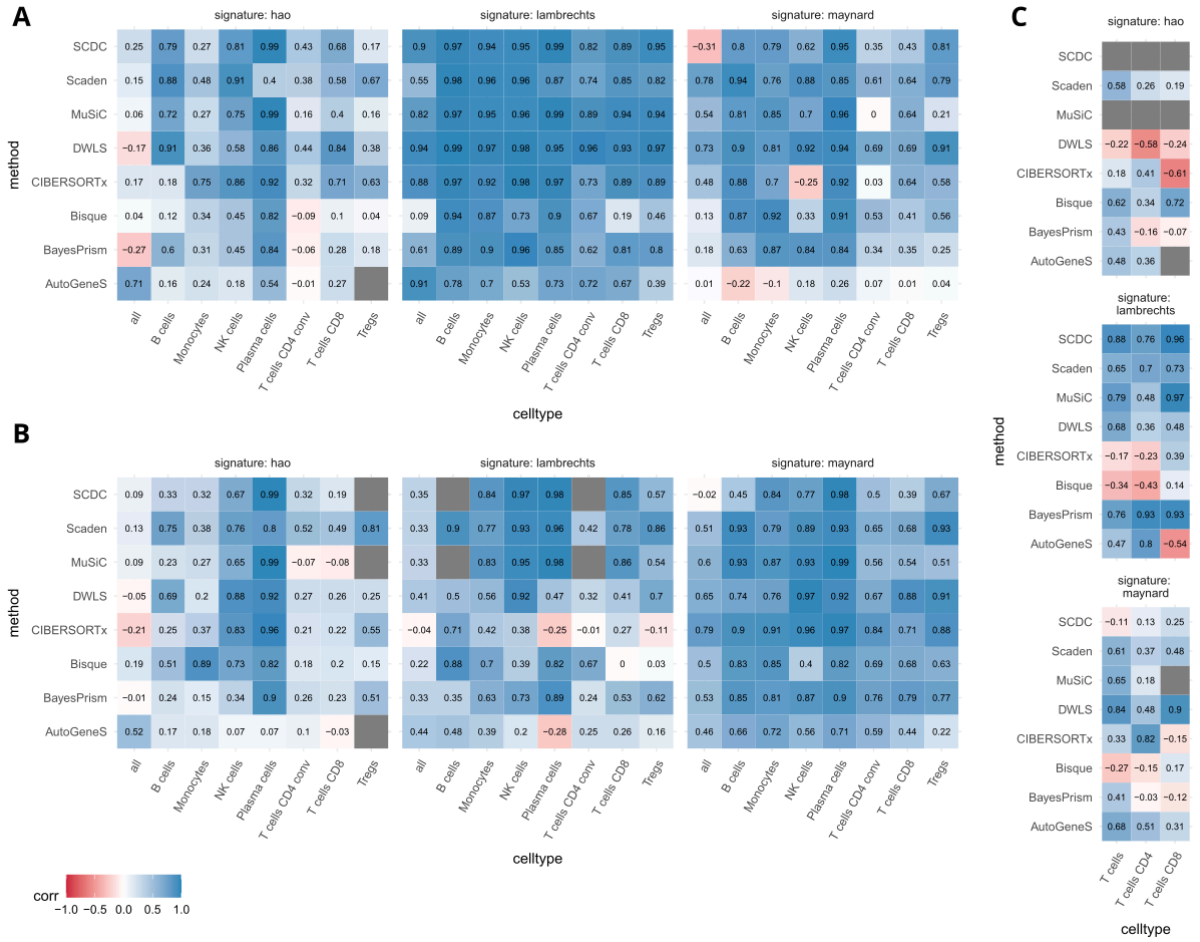


Figure 6: Single-cell references of different tissues and assays for deconvolution.
Deconvolution of pseudo-bulks derived from *Lambrechts* **(A)** or *Maynard* **(B),** and the *Vanderbilt* bulk dataset (**C**) with methods trained on different single cell datasets (indicated in headers of each heatmap). The heatmaps show the Pearson correlation coefficient between the ground truth cell fractions and the estimates obtained training the methods with different signatures. Grey tiles correspond to NA correlation values, due to all the estimates for cell type fractions being zero.

Discussion

Second-generation deconvolution is a powerful technique to investigate the complexity of tissues, coupling the high resolution of single-cell data[37,44–47] with the cost-effectiveness of bulk RNA-seq[6]. In principle, second-generation deconvolution methods can be applied to any cell type, tissue, and organism, but their inherent flexibility challenges their validation and the understanding of the determinants of deconvolution performance. In this study, we benchmarked a selection of second-generation deconvolution methods. Instead of focusing on previously assessed aspects like data normalization[9], we addressed a set of open challenges in second-generation deconvolution[14,16], which guided us in the rational design of a compendium of real and simulated bulk RNA-seq datasets across different tissues and organisms (Fig. 1A). Bulk datasets with reliable cell composition information are scarce and, due to different biases introduced by experimental protocols, the measured FACS proportions might not reflect those present in the bulk RNA-seq[14,16,48]. However, they still serve as the current gold-standard in the field. On the other hand, some of our simulated datasets might not be representative of real-life situations, but they nonetheless help to disentangle the impact of various sources of variability and understand methods behavior under a variety of controlled scenarios, which would be unfeasible to reproduce experimentally.

When testing deconvolution performance on pseudo-bulk data, all methods showed good performance, while on real datasets, the performance was lower across all methods. Overall, DWLS and Scaden obtained the most accurate results across datasets and organisms, followed by CIBERSORTx and AutoGeneS.

We showed how large single-cell references computationally challenge many methods to the point that it becomes infeasible to obtain a result. By training with subsampled reference data, we revealed that the accuracy of the results increased with the number of cells per cell type but plateaued early. DWLS and MuSiC provided the most stable results across reference subsets already for small-sized datasets, while the deep learning-based method Scaden performed better after training with large single-cell datasets. Both peak memory and elapsed time were increasing beyond a linear rate with the number of cells, with signature creation being the most resource-intensive step. As the deconvolution step was fast for most methods, reusing signatures (facilitated by *omnideconv*) can speed up deconvolution of similar bulk datasets. Notably, our analytical setup differs from previous ones[8], as we created subsampled reference datasets with uniform cell-type proportions, not mirroring the different imbalances of cell types in scRNA-seq datasets.

Another feature of the input scRNA-seq data that influences method performance is the resolution of cell-type annotations. A goal of cell type deconvolution methods, especially in biomedical research, is to discern not just broad cell types but also disentangle subtypes or functional states. In agreement with a previous study[49], fine-grained cell types, especially T-cell subsets, were more difficult to quantify for most methods. DWLS and MuSiC provided comparable performances across different resolution levels. BayesPrism and AutoGeneS performed better when the coarse estimates were the result of aggregating the fine-grained ones, in agreement with previous studies[21,24].

No previous benchmarking studies assessed the robustness of deconvolution methods to cell-type-specific mRNA bias, which can result in systematic estimation bias of cell fractions[1]. Leveraging our pseudo-bulk simulator SimBu[17], we showed clearly that only DWLS, MuSiC, and SCDC robustly corrected for mRNA bias.

A major challenge for deconvolution methods is an incomplete reference. scRNA-seq data are affected by cell-dissociation biases that distort the true cellular composition[3], with potential loss of specific cell types such as neutrophils[44,50]. In our evaluation, the absence of a specific cell type severely impacted the estimates, especially in the presence of transcriptionally similar cell types –a finding recapitulated by our “spillover” analysis. The robustness to unknown cellular content is key in ensuring between-sample comparability, especially in tumor-bulk deconvolution. In the absence of appropriate tumor signatures, first-generation methods for immune-cell type deconvolution, such as quanTIseq[32] and EPIC[51], enable the quantification of the percentage of “uncharacterized cells”, which in bulk-tumor deconvolution can be regarded as a proxy for tumor content[32]. None of the considered second-generation deconvolution methods offer this feature, but our results indicate that tools like CIBERSORTx and DWLS can provide estimates that are positively correlated with the true cell fractions even when the unknown tumor content reaches 80-90% of the sample. We point out, however, that the estimates have to be considered as systematically inflated due to the failed quantification of the unknown content, as all the assessed methods normalize cell type fractions to sum up to 100% of the sample.

Finally, our findings suggest that the tissue/disease context of the single-cell data used for method training can have an impact on deconvolution results, while effects of the sequencing technology were more variable across methods and datasets. When deconvolving lung cancer pseudo-bulks with PBMC signatures some cell types were often undetected, emphasizing tissue-specific differences in the transcriptomic profiles of various cell types[52] and the need for biologically similar background of the single-cell and bulk RNA-seq data. Our results also suggest that tag-based sequencing data (e.g. 10x) represents the most appropriate reference, though certain methods produce more accurate results when trained with full-transcript data (e.g. Smart-seq2). We speculate that this may be related to the data type used for method development and validation.

In summary, we recommend DWLS and Scaden for single-cell-informed deconvolution, and, while both require large computational resources and extensive reference, they have shown the

most robust results in every scenario; especially noteworthy is the amount of unknown content and spillover they can handle. In this study, we refrained from parameter tuning, and used instead developer-recommended or default options, to avoid giving any method an inherent advantage in a specific task. However, parameter optimization as well as unique properties affecting method performance could be easily explored in future studies, also thanks to the ecosystem of tools we make available to streamline deconvolution analysis and benchmarking (https://omnideconv.org/). For instance, some methods could possibly improve their deconvolution performance through tailored data and deconvolution approach: like multi-level cell-type annotation (BayesPrism), matching bulk and single-cell RNA-seq data (Bisque), multiple scRNA-seq datasets (Scaden and SCDC). All methods included in *omnideconv* can be easily used to their full potential and, together with our additional novel resources, we envision researchers to have better possibilities to thoroughly characterize and optimize deconvolution results, extending this powerful technique to an increasing panel of applications and domains.

Acknowledgments

This work was supported by European Cooperation in Science and Technology (COST) Action “Mye-InfoBank” (CA20117, supported by the EU Framework Program Horizon 2020), by the de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI), and by ELIXIR-DE (Forschungszentrum Jülich and W-de.NBI-001, W-de.NBI-004, W-de.NBI-008, W-de.NBI-010, W-de.NBI-013, W-de.NBI-014, W-de.NBI-016, W-de.NBI-022). FF was supported by the Austrian Science Fund (FWF) (no. T 974-B30 and FG 2500-B) and by the Oesterreichische Nationalbank (OeNB) (no. 18496). FE was supported by the Austrian Science Fund (FWF) (Special Research Program F7804-B and I5184). ML and AD were supported by the German Federal Ministry of Education and Research (BMBF) within the framework of the CompLS research and funding concept [031L0294B (NetfLID)]. The computational results presented here have been achieved in part using the LEO HPC infrastructure of the University of Innsbruck. We thank Nicolas Goedert and Yuyu Liang for their active development of a first version of the omnideconv package.

Competing interests

FF consults for iOnctura. GS is an employee of Boehringer Ingelheim International Pharma GmbH & Co KG, Biberach, Germany. ML consults for mbiomics GmbH.

Online Methods

Public data accession and processing

Raw FASTQ files were retrieved from the Gene Expression Omnibus (GEO) using the accession codes GSE107572 (*Finotello*[32] dataset, n=9) and GSE64655 (*Hoek*[33] dataset, n=8), as well as ArrayExpress using the accession codes E-MTAB-9271 (*Petitprez*[36] dataset, n=14) and E-MTAB-6458 (*Chen*[35] dataset, n=12). The datasets were processed in the same way, using the nf-core RNA-seq framework[53] and the human reference genomes GRCh38 with genome annotation v41, and murine genome GRCm39 with genome annotations vM30, from the Genome Reference Consortium. Briefly, FASTQ files were first aligned to the reference genome with STAR[54], and then summarized with Salmon[55] to obtain both read counts and transcripts per millions (TPMs). The *Vanderbilt* dataset was provided by the Vanderbilt Institute for Infection, Immunology and Inflammation (USA)[32]. The flow cytometry estimates for the various cell types were obtained from the corresponding authors.
The count matrix and the cell type annotations for the *Hao*[34] (number of cells = 152,135) and the *Wu*[38] (number of cells = 88,571) single cell dataset were retrieved from GEO, with accession numbers GSE164378 and GSE176078. Both the *Lambrechts*[3] (number of cells = 64,135) and the *Maynard*[43] (number of cells = 17,546) datasets were retrieved from the LuCA atlas (https://luca.icbi.at/, V. 2022.10.20)[44], and the cells classified as doublets ('doublet_status') were removed.
The Tabula Muris (*TM*, number of cells = 9,083) spleen dataset was retrieved from the atlas website (https://tabula-muris.ds.czbiohub.org/)[45].
The single cell datasets were analyzed with Seurat[56]. Quality control was performed according to the current best practices[57] considering the number of total counts, the number of expressed features (genes), and the fraction of mitochondrial genes per cell. We considered the MAD (Median Absolute Deviation), computed as $MAD = median(|X_i - median(X)|)$, with X_i being the respective metric of an observation. Cells that have $X_i < n * |MAD|$, with $n = 5$ for the number of total counts and the number of expressed genes, or $n = 3$ for the fraction of mitochondrial RNA were classified as outliers. Single cell datasets were CPM-normalized through Seurat. For the *Maynard* dataset, transcript-length-normalized values were retrieved from the LuCA atlas ('counts_length_scaled') and were then CPM-normalized.
The cell type annotation considered was the 'cell_type_tumor' column. We removed all cell types that were labeled as 'dividing' and 'transitional'. For the *Lambrechts* and *Maynard* dataset, the medium and fine cell type annotations considered were the 'cell_type_tumor' and 'ann_coarse' columns, while the coarse level was obtained aggregating the dendritic cells, macrophages /monocytes and T/NK cells populations.
The coarse and fine cell type annotations for the *Wu* dataset were obtained from the metadata provided with the cell counts ('celltype_major' and 'celltype_minor' fields, respectively).
The T cell subpopulation in the *TM* dataset was manually annotated to identify the $CD4^+$, $CD8^+$ and Tregs subpopulation.

Benchmarking pipeline *deconvBench*

We implemented a Nextflow pipeline called *deconvBench* that helped us to perform all of the simulation scenarios described above in an efficient and reproducible way: https://github.com/omnideconv/deconvBench. This pipeline was executed on a cloud computing cluster provided by the German Network for Bioinformatics Infrastructure (de.NBI), and included two worker nodes with 28 vCPUs and 256Gb of RAM each. These values were used as given limits for all deconvolution methods and we did not provide results in cases where a method would exceed this amount of memory and crashed.

Performance metrics

We considered two primary metrics for the performance of the benchmarked methods: Pearson correlation and root mean square error (RMSE). Both were calculated by comparing the available ground truth cell-type fraction with the estimations of the benchmarked methods. Metrics were calculated across all cell-type fractions and samples (*global correlation*) and separately for each individual cell type. This allows us to compare methods on a cell-type-specific level.

For each sample *s* with vectors *x* (ground truth cell-type fractions) and *y* (estimated fractions by a deconvolution method) of same length *C* (number of cell types), we calculated sample-wise RMSE as $RMSE_s = \sqrt{\frac{1}{C}\sum_{s=1}^{C}(x_s - y_s)^2}$. Similarly, we calculated RMSE for each cell-type *c* across all available samples *S* as $RMSE_c = \sqrt{\frac{1}{S}\sum_{c=1}^{S}(x_c - y_c)^2}$.

RMSE together with correlation allows us to quantify the *estimation bias* of methods, i.e. the systematic deviation of predictions from the ground truth in a positive (over-estimation) or negative (under-estimation) direction.

Simulating pseudo-bulk RNA-seq datasets and mRNA bias

Using the *Hao* scRNA-seq dataset, we simulated pseudo-bulk datasets, in which we mirrored sample-specific features of real bulk datasets (*Finotello* and *Hoek*, PBMC) with matching flow cytometry derived cell fractions. For each real bulk sample *n* we used its cell-type proportion and total number of read counts as basis for one pseudo-bulk sample and created 10 technical replicates with these features. Similarly, we used two mouse datasets (*Chen* and *Petitprez*) for which the corresponding cell fractions were available, along with the Tabula Muris (*TM*) spleen scRNA-seq reference. The simulations were carried out with the R package SimBu[17] with the following settings:

Each pseudo-bulk sample sampled 10,000 cells (with replacement) from the scRNA-seq datasets to have the same cell type composition as the bulk sample *n* (derived from FACS data). Only cells present in both the real bulk and scRNA-seq datasets were utilized for simulation. The pseudo-bulks were simulated to have the same sequencing depth as bulk

sample *n*. Additionally, we scaled the expression of genes in each cell by scaling factors, derived by the total number of expressed genes per cell in the scRNA-seq reference dataset (Supplementary Figure S1). This approach has been established as 'silver-standard' for accounting for mRNA content bias in cells[17].
As there are missing values in the FACS annotation in four samples of the *Petitprez* dataset, we left out these samples in the described setup. The *Hoek* real bulk dataset only has FACS annotations for T cells, but the *Hao* reference includes annotations for three subtypes (CD4+ T cells, CD8+ T cells and T regulatory cells), so we summed up the estimated fractions that methods produced for these three subtypes in order to compare this sum to the simulated ground truth. For the pseudo-bulk samples in return that are based upon the *Hoek* real bulk dataset (called *HoekSim*), we sampled from all three T cell subtypes in *Hao* during simulation.
While the *Hoek* dataset does have FACS annotations for T cells, the *Hao* reference distinguishes T cells into three subtypes. Therefore we did not simulate T cells in the *HoekSim* dataset, as we would not know in which proportions to mix the three T cell subtypes.

For the simulations used to compare pseudo-bulk samples with and without mRNA content bias, the simulation setup described above was repeated, but the `scaling_factor` parameter in SimBu was set to '`NONE`'. Additionally, to ensure comparability between samples, we used the same fixed seed for each sample with and without mRNA bias, such that the same cells are sampled for both samples and the only difference is due to mRNA bias modelling. A pseudo-bulk sample with both biased and unbiased versions can be considered a pair. We can define a $delta$ value for each methods estimation of a cell-type fraction as the difference between estimated and true fraction values:
$delta = fraction_{estimate} - fraction_{true}$. This value quantifies how different an estimate is from a true cell-type fraction and is calculated for all cell types in a sample. Further, we define $delta_{bias}$ as this $delta$ value in samples that were simulated with mRNA bias and $delta_{nobias}$ in samples without bias. We would expect, that a method that does account for mRNA content bias, will be 'further away' from the true fraction in samples without bias, i.e. $|delta_{nobias}| > |delta_{bias}|$. Note, that we use the the absolute values of $delta$ in this case, in order to to account for negative values of $delta$. As the values of $delta$ are not distributed normally, we can perform a one-sided Wilcoxon Rank Sum test, where the null hypothesis would assume $|delta_{nobias}| = |delta_{bias}|$ and our alternative hypothesis is $|delta_{nobias}| > |delta_{bias}|$. Resulting p-values were FDR-corrected for multiple testing. A significant result would therefore tell us that a method accounts for mRNA bias in a cell type to a considerable degree. Additionally, we can check wether $delta_{bias}$ is above or below 0 in order to see if a method over- or under-estimates a certain cell type and can potentially link this behavior back to mRNA bias.

Subsampling single-cell RNA-seq references

We systematically quantified the influence of a scRNA-seq reference by gradually increasing the number of cells from a reference and using them as input for the different methods. For each cell type present in the *Hao* scRNA-seq dataset, we sampled 5, 25, 50, 100 and 500 cells

without replacement. We repeated this five times for each subset size to create technical replicates. To ensure the reproducibility of the random sampling over different pipeline runs, we generated a random seed for each run composed of a combination of subset size, replicate number, and reference name. In cases where fewer cells per type than desired are present in the single-cell dataset, we used all cells of the corresponding type. The resulting references were applied to the real and artificial bulk datasets *Hoek*, *HoekSim*, *Finotello* and *FinotelloSim*.

Measuring computational efficiency

The setup we used to track deconvolution performance with different sizes of the scRNA-seq reference, could also be used by us to measure computational efficiency related to the reference. As we had eleven cell types present in the Hao dataset, this would lead to input references of sizes 55, 275, 550, 1100, 5500 and finally the full reference of about 153,000 cells. We measured memory for methods with the Nextflow trace functionality, which - among others - stores information on consumed memory in the 'peak_vmem' column. Each method was used with a single vCPU and a maximum available RAM of 240Gb on a virtual machine cluster provided by the de.NBI SimpleVM environment (https://simplevm.denbi.de/wiki/).
We calculated the runtime of methods by the difference in the end and start time of the core *omnideconv* functions 'build_model' and 'deconvolute'. In the case of omnideconv deconvolution methods reliant on Python, an overhead may arise as they are invoked through the *reticulate* interface rather than directly. Furthermore, for these methods, R data types must be first converted to formats compatible with Python, adding an additional computational step.

Simulation and deconvolution of pseudo-bulk RNA-seq datasets for spillover analysis

The pseudo-bulk datasets for the spillover analysis were simulated using the *Hao* scRNA-seq dataset. The cell types considered were: B cells, mDC, pDC, $CD4^+$ and $CD8^+$ T cells, T_{reg} cells, monocytes, NK cells, ILC, platelets, and plasma cells. For each cell type, a pseudo-bulk with 10 replicates was simulated with SimBu using the "pure" scenario and 100 cells, meaning only cells from this one cell type were sampled. These pseudo-bulks were then deconvolved with the *Hao* reference including all the cell types mentioned above, downsampled to 500 cells per cell type.
We quantified the amount of spillover as the sum of cell-type estimates that correspond to the pure, simulated cell type (giving a cell-type specific percentage value, ideally 100%) and the sum of cell-type estimates that correspond to all other predicted cell types (giving a method specific percentage value, ideally 0%).

Simulation of pseudo-bulk RNA-seq datasets with unknown content

The pseudo-bulk datasets for the unknown content analysis were simulated using the *Lambrechts* scRNA-seq dataset. The cell types considered were: B cells, CD4+ T cells, stromal cells and macrophages, in addition to the tumor cells. The pseudo-bulks were simulated with

SimBu using the “weighted” scenario, and increasing concentration of tumor cells: 0%, 5%, 10%, 20%, 30%, 50%, 70%, 80%, 90%. The remaining four cell types were sampled with random fractions. Five replicates with ten pseudo-bulk samples each, were simulated for each tumor concentration, resulting in 9*50=450 pseudo-bulk samples for the deconvolution, the methods were trained with the *Lambrechts* dataset with the four above-mentioned cell types, excluding the tumor cells, downsampled to 500 cells per cell type.

Simulation of pseudo-bulk RNA-seq datasets for the impact of cell type resolution analysis

The pseudo-bulk datasets to study the impact of cell type annotation resolution were simulated using the *Lambrechts*[3] and the *Wu*[38] scRNA-seq dataset. For the Lambrechts dataset, the cell types considered at the “coarse” resolution were: B cells, dendritic cells, monocytes/macrophages, monocytes and T/NK cells. Except for the B cells, the other cell types possess two additional levels of annotation, “normal” and “fine” (Table S1).
Five replicates of ten pseudo-bulk samples each were simulated with SimBu using the ‘mirror_db’ scenario (resembling the cell-type proportions from the scRNA-seq dataset in the pseudo-bulk dataset), considering the “fine” level of annotations. The simulated ground truth estimates of all subtypes of a certain cell type could then be summed up to get the corresponding fractions of this cell type, respectively for the “normal” and “coarse” annotation level.
The same was done for the Wu dataset considering (at the coarse resolution) B cells, Cancer associated fibroblasts (CAFs), Myeloid and T cells (Table S1).
This simulation scenario therefore provided us with ground truth cell fractions for each pseudo-bulk sample, at three (Lambrechts) and two (Wu) resolution levels. For the training of the methods, we only considered the cell types that were also used for simulations and downsampled each cell type to 500 cells, considering the fine cell type annotation. The training datasets were therefore composed of the same count matrix, and the different cell type annotations. The pseudo-bulks were then subsequently deconvolved providing a set of three (*Lambrechts*) or two (*Wu*) cell type fraction estimates.
Finally, we could compare the estimates for cell types that were calculated on each resolution to the corresponding ground truth resolution.

Simulation of pseudo-bulk RNA-seq datasets for impact of single-cell technology and tissue context analysis

The pseudo-bulks for the impact of cell technology and tissue context analysis were simulated using the *Lambrechts*, *Maynard* and *Hao* datasets, considering only the common cell types between the three (B cells, NK cells, T cells $CD4^+$, $CD8^+$ and Tregs, Monocytes and Plasma cells). This was done in order to minimize the variability between the pseudo-bulks and to isolate the effect of the tissue/technology of the dataset. Five replicates of ten pseudo-bulks each were simulated starting from the three single cell datasets. The same datasets,

subsampled at 500 cells per cell type, were then used to train the methods for the deconvolution of the pseudo-bulks. Each pseudo-bulk was then deconvolved with both its corresponding reference and the references from the other two datasets. The Vanderbilt samples, on the other hand, were deconvolved training the methods with the datasets including every cell type (i.e. also those not used in the pseudo-bulk simulation), subsampled at 500 cells per cell type.

DeconvExplorer

deconvExplorer is a R shiny web app that aims to provide easy access to analyze deconvolution signatures and results. Signatures can be explored using interactive heatmaps and compared to signatures of other references or methods via upset plots or signature-specific scores such as entropy, condition number and number of included genes. Finally, deconvolution results can be compared to results of other methods or to user-provided ground-truth data by correlation and RMSE metrics and fitting visualizations. The mentioned scores are calculated as such:
We calculate gene-wise entropy for each gene using the function *entropySpecificity* from the BioQC package[58]. It quantifies the amount of specificity a gene has for a certain cell type, therefore giving information on its potential as a cell-type marker. Genes with a high entropy value (close to 1) will therefore be the best markers, while genes with low entropy values (close to 0) will show a uniform expression level across cell types.
Similarly, the Gini Index is calculated using the same package and its *gini* function. It measures the dispersion in a list of values, where a value of 0 will appear in case that all values are exactly the same. It is maximized (gini_index=1) when the values are maximally different.
The condition number of the signature matrix is calculated using the *kappa* function in base R. It quantifies the stability of the signature matrix towards changes and errors in the input data and is defined as the product of the 2-norm of a matrix *M* and the 2-norm of its inverse:
$\kappa(M) = ||M||\,||M^{-1}||$.
A low condition number (close to 1) suggests that the signature matrix is stable towards changes in the bulk RNA-seq input data. This means that, if the bulk RNA-seq is perturbed by errors, the estimated cell-type fractions will not dramatically differ.

Details on Method settings

The following section covers the versions, parameter settings (if other than default) and data input normalizations we used for each benchmarked method. For some methods we forked the original repository and had to manually introduce changes to the source code or dependency versions, so that we can reliably install them via *omnideconv*. None of these changes will however influence the results a method produces.

AutoGeneS

We cloned the Python package AutoGeneS version 1.0.4 from https://github.com/theislab/AutoGeneS and installed it via the omnideconv package. The method offers an option to only keep highly variable genes for subsequent signature building and deconvolution, however, in order to have the same set of input genes for all methods, we

decided not to use this option. Default parameters in omnideconv were used for signature building and deconvolution, only the 'max_iter' parameter was fixed to 1,000,000 in order to limit the number of iterations in case no earlier convergence could be found. Also, we set the 'ngen' parameter to 5,000, which fixes the number of generations that the algorithm will run. We CPM-normalized scRNA-seq data and TPM-normalized bulk RNA-seq samples, as indicated in the original publication of the method.
AutoGeneS would be able to follow *omnideconv's* two-step design of separating signature building and deconvolution. However, by far the most amount of runtime is spent for saving the signature as a pickle file. Therefore, we opted to implement it as a one-step method and have the signature creation step as an optional separate step. For the runtime analysis, we benchmarked the one-step version of AutoGeneS in omnideconv, to not artificially increase runtime due to unnecessary IO steps.

BayesPrism

We forked the R package BayesPrism v2.0 from https://github.com/Danko-Lab/BayesPrism and installed it via the omnideconv package. BayesPrism would offer an option to distinguish between cell types and more detailed cell states in a cellular annotation. As it is the only method in this benchmark with this option, we chose to only use their predictions on cell types in order to keep results comparable. BayesPrism can apply a final round of Gibbs sampling to update the initial cell-type fractions. As recommended by the authors, we did not use this setting in the case that scRNA-seq and bulk RNA-seq originate from similar assays. This is the case when we created pseudo-bulks from the same scRNA-seq dataset that was also used as reference for the method. As we used data from two different organisms, we adapted the 'species' parameter accordingly. Other parameters remained with their default values as implemented in omnideconv. As recommended by the authors, we used un-normalized count values for scRNA-seq and bulk RNA-seq samples.

Bisque

The R package BisqueRNA v1.0.5 was forked from https://github.com/cozygene/bisque and installed via the omnideconv package. We kept all parameters at their default as implemented in omnideconv and used un-normalized count values for scRNA-seq and bulk RNA-seq samples.

CIBSERSORTx

We used the Docker image of CIBERSORTx, which can be downloaded from the https://cibersortx.stanford.edu/ website. CIBERSORTx has two options for batch correction, S-mode and B-mode; after discussion with the authors, the following parameter settings were suggested: S-mode in the case of deconvolution of real bulk RNA-seq samples using scRNA-seq reference data assayed by the 10x protocol or in case scRNA-seq and pseudo-bulk samples originate from different single-cell protocols. B-mode in the case of deconvolution of real bulk RNA-seq samples using scRNA-seq data assayed by the Smart-seq2 protocol. No batch correction if scRNA-seq and pseudo-bulk samples originate from the same protocol. We tried to follow these suggestions, however when using CIBERSORTx with S-mode on true bulk

samples with a 10x scRNA-seq reference, we had worse results compared to a run without batch correction. We therefore proceeded to use results without batch correction in the mentioned settings. Other parameters remained with their default values as implemented in omnideconv. As recommended by the authors, we used TPM normalized values for bulk RNA-seq samples and CPM normalization for the scRNA-seq samples.

DWLS

We originally cloned the R package DWLS v0.1 from https://bitbucket.org/yuanlab/dwls/src/master/ and installed it via omnideconv. The original DWLS exhibited prolonged runtimes during the construction of the signature matrix. A primary reason for this was the absence of parallel processing in the original DWLS, rendering it incompatible with multicore machines. To address this, we introduced a new parameter, 'ncores,' which allows users to specify the number of cores to be utilized. This parameter activates parallelization for MAST[59] functions like MAST::zlm, and enables us to leverage parallelized versions of iterative functions (e.g., replacing 'lapply' with 'mclapply') for enhanced parallel utilization.
In addition to parallelization, we conducted profiling to pinpoint performance bottlenecks within the DWLS program. Our analysis identified three key functions responsible for extended runtimes:

1. The internal method 'stat.log2.'
2. MAST::fromFlatDF, which was subsequently replaced with MAST::fromMatrix for streamlined data handling and improved speed.
3. The internal method 'm.auc,' which, upon evaluation, was found to be unused for signature creation and was subsequently removed.

These optimizations not only significantly improved runtimes but also simplified the program structure. For example, reducing unnecessary dataframe creations in the 'stat.log2' function minimized memory assignments, resulting in reduced code complexity and faster execution. The overall outcome of this improved DWLS implementation is a 3x speedup on a multicore machine with 10 cores, and a 2x improvement running DWLS with a single core.
The optimized version of DWLS can be executed in *omnideconv* with the parameter *dwls_method* set to *mast_optimized*, which is the default for all results presented in this work.

MuSiC

The R package MuSiC v0.3.0 was forked from https://github.com/xuranw/MuSiC and installed via omnideconv. While a version 1.0.0 is already available, the only difference between the versions is the additional support for SingleCellExperiment data objects. The method includes information of patients in its deconvolution algorithm, which we added via the ‘batch_ids’ parameter of the ‘deconvolute()’ function in omnideconv. All parameters were set to default as implemented in omnideconv, scRNA-seq expression values were left un-normalized and bulk RNA-seq values were TPM normalized.

Scaden

The python package Scaden v1.1.2 was forked from https://github.com/KevinMenden/scaden. All parameters were set to default as implemented in omnideconv, scRNA-seq expression values were left un-normalized and bulk RNA-seq values were TPM normalized.

SCDC

The R package SCDC v0.0.0.9 was forked from https://github.com/meichendong/SCDC and installed via omnideconv. Parameters for deconvolution were set to default as implemented in omnideconv, scRNA-seq expression values were left un-normalized and bulk RNA-seq values were TPM normalized. In contrast with other methods, SCDC allows the usage of multiple single cell RNAseq references. In this study, the method was used with just one single cell dataset as reference to keep the benchmarking as close as possible across methods.

Data availability

All bulk datasets that are part of this study can be accessed via *deconvData* at https://figshare.com/projects/deconvData/197794; it includes raw and TPM normalized gene expression matrices and FACS annotations for each sample. In addition, we give access to the cell-type annotations of every scRNA-seq dataset that was used in this study. The corresponding count matrices should be accessed from the original data source (see data access codes in previous chapter) and can be preprocessed in the same way as we described it earlier.
A Docker image that includes an installed version of omnideconv and its dependencies can be accessed via Dockerhub at https://hub.docker.com/repository/docker/alexd13/omnideconv_benchmark/general and a compressed file of the same image (version 1.2) is located in the *deconvData* repository to provide also long-term support.

Code availability

The omnideconv ecosystem and all its included software packages are available at https://omnideconv.org/. It includes the *omnideconv* R package (https://github.com/omnideconv/omnideconv), the *deconvExplorer* web server (https://exbio.wzw.tum.de/deconvexplorer/) and the *deconvData* data access (https://figshare.com/projects/deconvData/197794). The code to run *deconvBench*, reproduce the figures and simulation scenarios and a detailed documentation of *deconvBench* can be found at https://github.com/omnideconv/deconvBench.

References

1. Finotello, F. & Trajanoski, Z. Quantifying tumor-infiltrating immune cells from transcriptomics data. *Cancer Immunol. Immunother.* (2018) doi:10.1007/s00262-018-2150-z.
2. Jovic, D. *et al.* Single-cell RNA sequencing technologies and applications: A brief overview. *Clin. Transl. Med.* **12**, e694 (2022).
3. Lambrechts, D. *et al.* Phenotype molding of stromal cells in the lung tumor microenvironment. *Nat. Med.* **24**, 1277–1289 (2018).
4. Finotello, F., Rieder, D., Hackl, H. & Trajanoski, Z. Next-generation computational tools for interrogating cancer immunity. *Nat. Rev. Genet.* **20**, 724–746 (2019).
5. Sturm, G. *et al.* Comprehensive evaluation of transcriptome-based cell-type quantification methods for immuno-oncology. *Bioinformatics* **35**, i436–i445 (2019).
6. Merotto, L., Zopoglou, M., Zackl, C. & Finotello, F. Next-generation deconvolution of transcriptomic data to investigate the tumor microenvironment. in *International Review of Cell and Molecular Biology* (Academic Press, 2023). doi:10.1016/bs.ircmb.2023.05.002.
7. Tran, K. A. *et al.* Performance of tumour microenvironment deconvolution methods in breast cancer using single-cell simulated bulk mixtures. *Nat. Commun.* **14**, 5758 (2023).
8. Hippen, A. A. *et al.* Performance of computational algorithms to deconvolve heterogeneous bulk ovarian tumor tissue depends on experimental factors. *Genome Biol.* **24**, 239 (2023).
9. Avila Cobos, F., Alquicira-Hernandez, J., Powell, J. E., Mestdagh, P. & De Preter, K. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nat. Commun.* **11**, 5650 (2020).
10. Sutton, G. J. *et al.* Comprehensive evaluation of deconvolution methods for human brain gene expression. *Nat. Commun.* **13**, 1358 (2022).
11. Jin, H. & Liu, Z. A benchmark for RNA-seq deconvolution analysis under dynamic testing

environments. *Genome Biol.* **22**, 102 (2021).

12. Hu, M. & Chikina, M. Heterogeneous pseudobulk simulation enables realistic benchmarking of cell-type deconvolution methods. *bioRxiv* 2023.01.05.522919 (2023) doi:10.1101/2023.01.05.522919.
13. Pournara, A. V. *et al.* CATD: A reproducible pipeline for selecting cell-type deconvolution methods across tissues. *bioRxiv* 2023.01.19.523443 (2023) doi:10.1101/2023.01.19.523443.
14. Nguyen, H., Nguyen, H., Tran, D., Draghici, S. & Nguyen, T. Fourteen years of cellular deconvolution: methodology, applications, technical evaluation and outstanding challenges. *Nucleic Acids Res.* (2024) doi:10.1093/nar/gkae267.
15. Maden, S. K. *et al.* Challenges and opportunities to computationally deconvolve heterogeneous tissue with varying cell sizes using single-cell RNA-sequencing datasets. *Genome Biol.* **24**, 288 (2023).
16. Garmire, L. X. *et al.* Challenges and perspectives in computational deconvolution of genomics data. *Nat. Methods* (2024) doi:10.1038/s41592-023-02166-6.
17. Dietrich, A. *et al.* SimBu: bias-aware simulation of bulk RNA-seq data with variable cell-type composition. *Bioinformatics* **38**, ii141–ii147 (2022).
18. Vathrakokoili Pournara, A. *et al.* CATD: a reproducible pipeline for selecting cell-type deconvolution methods across tissues. *Bioinform Adv* **4**, vbae048 (2024).
19. Di Tommaso, P. *et al.* Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* **35**, 316–319 (2017).
20. Aliee, H. & Theis, F. J. AutoGeneS: Automatic gene selection using multi-objective optimization for RNA-seq deconvolution. *Cell Syst* **12**, 706–715.e4 (2021).
21. Chu, T., Wang, Z., Pe’er, D. & Danko, C. G. Cell type and gene expression deconvolution with BayesPrism enables Bayesian integrative analysis across bulk and single-cell RNA sequencing in oncology. *Nat Cancer* **3**, 505–517 (2022).

22. Baron, M. *et al.* A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Syst* **3**, 346–360.e4 (2016).
23. Jew, B. *et al.* Accurate estimation of cell composition in bulk expression through robust integration of single-cell information. *Nat. Commun.* **11**, 1971 (2020).
24. Kang, K., Huang, C., Li, Y., Umbach, D. M. & Li, L. CDSeqR: fast complete deconvolution for gene expression data from bulk tissues. *BMC Bioinformatics* **22**, 262 (2021).
25. Newman, A. M. *et al.* Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nat. Biotechnol.* **37**, 773–782 (2019).
26. Frishberg, A. *et al.* Cell composition analysis of bulk genomics using single-cell data. *Nat. Methods* **16**, 327–332 (2019).
27. Tsoucas, D. *et al.* Accurate estimation of cell-type composition from gene expression data. *Nat. Commun.* **10**, 2975 (2019).
28. Sun, X., Sun, S. & Yang, S. An Efficient and Flexible Method for Deconvoluting Bulk RNA-Seq Data with Single-Cell RNA-Seq Data. *Cells* **8**, (2019).
29. Wang, X., Park, J., Susztak, K., Zhang, N. R. & Li, M. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat. Commun.* **10**, 380 (2019).
30. Dong, M. *et al.* SCDC: bulk gene expression deconvolution by multiple single-cell RNA sequencing references. *Brief. Bioinform.* **22**, 416–427 (2021).
31. Menden, K. *et al.* Deep learning-based cell composition analysis from tissue expression profiles. *Sci Adv* **6**, eaba2619 (2020).
32. Finotello, F. *et al.* Molecular and pharmacological modulators of the tumor immune contexture revealed by deconvolution of RNA-seq data. *Genome Med.* **11**, 34 (2019).
33. Hoek, K. L. *et al.* A cell-based systems biology assessment of human blood to monitor immune responses after influenza vaccination. *PLoS One* **10**, e0118528 (2015).
34. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587.e29 (2021).

35. Chen, Z. *et al.* Inference of immune cell composition on the expression profiles of mouse tissue. *Sci. Rep.* **7**, 40508 (2017).

36. Petitprez, F. *et al.* The murine Microenvironment Cell Population counter method to estimate abundance of tissue-infiltrating immune and stromal cell populations in murine samples using gene expression. *Genome Med.* **12**, 86 (2020).

37. Svensson, V., da Veiga Beltrame, E. & Pachter, L. A curated database reveals trends in single-cell transcriptomics. *Database* **2020**, (2020).

38. Wu, S. Z. *et al.* A single-cell and spatially resolved atlas of human breast cancers. *Nat. Genet.* **53**, 1334–1347 (2021).

39. Monaco, G. *et al.* RNA-Seq Signatures Normalized by mRNA Abundance Allow Absolute Deconvolution of Human Immune Cell Types. *Cell Rep.* **26**, 1627–1640.e7 (2019).

40. Islam, S. *et al.* Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Res.* **21**, 1160–1167 (2011).

41. Kuksin, M. *et al.* Applications of single-cell and bulk RNA sequencing in onco-immunology. *Eur. J. Cancer* **149**, 193–210 (2021).

42. Nguyen, A., Yoshida, M., Goodarzi, H. & Tavazoie, S. F. Highly variable cancer subpopulations that exhibit enhanced transcriptome variability and metastatic fitness. *Nat. Commun.* **7**, 11246 (2016).

43. Maynard, A. *et al.* Therapy-Induced Evolution of Human Lung Cancer Revealed by Single-Cell RNA Sequencing. *Cell* **182**, 1232–1251.e22 (2020).

44. Salcher, S. *et al.* High-resolution single-cell atlas reveals diversity and plasticity of tissue-resident neutrophils in non-small cell lung cancer. *Cancer Cell* **40**, 1503–1520.e8 (2022).

45. Tabula Muris Consortium *et al.* Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* **562**, 367–372 (2018).

46. Regev, A. *et al.* The Human Cell Atlas. *Elife* **6**, (2017).

47. Sikkema, L. *et al.* An integrated cell atlas of the lung in health and disease. *Nat. Med.* **29**, 1563–1577 (2023).

48. Cobos, F. A. *et al.* Effective methods for bulk RNA-seq deconvolution using scnRNA-seq transcriptomes. *Genome Biol.* **24**, 177 (2023).

49. White, B. S. *et al.* Community assessment of methods to deconvolve cellular composition from bulk gene expression. *bioRxiv* 2022.06.03.494221 (2022) doi:10.1101/2022.06.03.494221.

50. Wigerblad, G. *et al.* Single-Cell Analysis Reveals the Range of Transcriptional States of Circulating Human Neutrophils. *J. Immunol.* **209**, 772–782 (2022).

51. Racle, J. & Gfeller, D. EPIC: A Tool to Estimate the Proportions of Different Cell Types from Bulk Gene Expression Data. *Methods Mol. Biol.* **2120**, 233–248 (2020).

52. Szabo, P. A. *et al.* Single-cell transcriptomics of human T cells reveals tissue and activation signatures in health and disease. *Nat. Commun.* **10**, 4706 (2019).

53. Ewels, P. A. *et al.* The nf-core framework for community-curated bioinformatics pipelines. *Nat. Biotechnol.* **38**, 276–278 (2020).

54. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).

55. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* **14**, 417–419 (2017).

56. Hao, Y. *et al.* Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat. Biotechnol.* (2023) doi:10.1038/s41587-023-01767-y.

57. Heumos, L. *et al.* Best practices for single-cell analysis across modalities. *Nat. Rev. Genet.* **24**, 550–572 (2023).

58. Zhang, J. D. *et al.* Detect tissue heterogeneity in gene expression data with BioQC. *BMC Genomics* **18**, 277 (2017).

59. Finak, G. *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* **16**, 278

(2015).