On the Promises and Limits of Multi-omics Integration for Deconvolution: The HADACA3 Benchmark

Hugo Barbot*[,1], Elise Amblard*[,2] Nicolas Homberg*[,2], Lucie Lamothe[2], Morgane Térézol[3], HADACA3 Consortium[†], Mira Ayadi [4], Aurélia Baurès [4], Yasmina Kermezli [2], Carl Herrmann [6], Sebastien Dejean[7], Lionel Spinelli[8], David Causeur[1], Florent Chuffart[9] Anaïs Baudot[§,3], Yuna Blum[§,10], Magali Richard[§,2]

[1]IRMAR CNRS, Institut Agro Rennes Angers [2]Univ. Grenoble-Alpes, CNRS, LIG [3]Aix-Marseille Univ, INSERM, MMG [4]CIT, Ligue Nationale Contre le Cancer, Paris [6]BioQuant, IPMB, Heidelberg Univ. [7]IMT CNRS, Univ. Toulouse [8]Aix-Marseille Univ, CNRS, CIML [9]Univ. Grenoble-Alpes, INSERM, IAB [10]IGDR CNRS, Univ. Rennes

*Co-first authors [§]Co-senior authors [‡]Full consortium list available in appendix.

Abstract

Understanding the cellular composition of complex tissues, such as tumors, is a key challenge in biology and medicine. A common approach, known as deconvolution, aims to estimate the cellular composition from bulk molecular measurements. With the growing availability of multiple types of molecular data, it is often assumed that combining data sources should improve deconvolution performance. Here, we present HADACA3, a community-driven benchmark designed to evaluate this assumption. We conducted a four-day collaborative competition followed by a large-scale computational benchmark, testing more than 250,000 analysis pipelines across nine datasets with matched DNA methylation (DNAm) and RNA profiles, representing a wide range of biological and experimental conditions. Our framework jointly evaluates the impact of preprocessing, feature selection, modeling, and integration strategies. We find that DNAm alone achieves the highest median performance across datasets, making it the most stable and reliable single-modality approach. However, multi-omics integration strategies can regularly achieve higher top performance in specific datasets and pipeline configurations. Among the tested strategies, late integration based on error-weighted averaging provides a strong and reliable baseline, while non-linear early integration methods, such as optimal transport, show promising results on real biological datasets. Overall, our results show that multi-omics integration does not systematically improve average performance over DNAm alone, but can improve best-case performance in specific settings. This highlights a trade-off between robustness and peak performance, and emphasizes the importance of aligning integration strategies with the statistical properties of the data. All data, code, and evaluation tools are publicly available to support reproducible research and future method development.

1 Introduction

Estimating the cellular composition of complex tissues from bulk molecular measurements, a task known as deconvolution, is central to understanding diseases, particularly in cancer. Bulk samples are molecular measurements obtained from heterogeneous mixtures of cells. They can span multiple omic layers such as transcriptomics (e.g. gene expression, RNA) and epigenomics (e.g. DNA methylation, DNAm). Tumors are composed of heterogeneous mixtures of cell populations, including cancer cells, stromal cells, and immune cells, whose relative proportions strongly influence disease progression, prognosis, and response to therapy [20]. Supervised deconvolution methods estimate these proportions by fitting bulk profiles against purified cell-type references, and have been the subject of extensive methodological development, using transcriptomic or epigenomic data [10, 2].

However, key challenges remain: ensuring that technical performance metrics reflect biologically meaningful criteria, and that methods generalize across different types of biological tissues, with varying noise structures, reference quality, and cellular compositions [39]. The integration of multi-modal omic data, combining, for instance, RNA and DNAm profiles, has emerged as a promising direction to improve computational methods in biology by capturing complementary molecular signals [28]. Yet, systematic benchmarks have shown that no single integration approach consistently outperforms others across tasks and data types [7]. In particular the specific question of multi-omics integration for cell-type deconvolution remains largely unexplored [1]. Our contributions are the following:

- **A community-driven benchmark and competition analysis.** We introduce HADACA3 (Figure 1), a community-driven competition followed by a comprehensive benchmark, designed to evaluate and compare multi-omics deconvolution strategies. We analyze which pipeline components drive performance and under what conditions, yielding generalizable insights for the community.
- **An empirical analysis of integration strategies for bulk deconvolution.** We show that multi-modal integration does not systematically improve over the best uni-modal strategy. We identify the conditions under which integration fails or succeeds. This critical analysis directly informs how integration methods should be evaluated in computational biology.
- **A diverse collection of benchmark datasets.** HADACA3 focuses on pancreatic cancer, a disease characterized by a complex and heterogeneous tumor microenvironment [19]. We assemble a set of datasets with known cell-type compositions spanning *in silico*, *in vitro*, and *in vivo* settings, including domain shifts with varying data distributions. We explicitly characterize the assumptions underlying each dataset and make them publicly available as a resource.
- **A modular framework.** We develop a reproducible Nextflow pipeline decomposing deconvolution workflows into four modules: (i) preprocessing, (ii) feature selection, (iii) deconvolution, and (iv) multi-omics integration; enabling systematic evaluation of all compatible combinations. This framework is designed to be reused and extended to evaluate new methods, datasets, and integration paradigms beyond the scope of this work.
- **An open evaluation platform.** All datasets, pipelines, and benchmarking infrastructure are publicly available through the scientific competition platform Codabench, enabling reproducible evaluation, continuous benchmarking, and community-driven method development.

2 Related work

Supervised deconvolution methods estimate cell-type proportions by fitting bulk molecular profiles against purified cell-type references. These methods span a broad algorithmic spectrum, from constrained linear regression to Bayesian models and machine learning approaches [12]. Early influential methods applied to RNA deconvolution for immune cell populations include CIBERSORT [30], based on support vector regression, and MCP-counter [6], based on gene signatures. For DNAm data, the first deconvolution algorithm was published in 2012 [21], and approximately 25 methods are now available, relying on supervised or unsupervised approaches [12]. Across modalities, the accuracy of cell-type estimation is strongly influenced by the quality of the reference data, the choice of preprocessing pipelines, feature selection, and the co-linearity of reference profiles [38, 10]. In addition, the availability of realistic benchmark datasets, with domain shifts, remains a key challenge [1].

Integrating multiple omics modalities has been proposed as a way to improve performance across a range of computational biology tasks. Integration strategies are broadly categorized into three paradigms: early integration, where both modalities are combined at the feature level prior to analysis; intermediate integration, where the integration is tightly coupled to the task-specific model; and late integration, where modality-specific results are combined at the output level [7]. Algorithmic approaches range from covariance and matrix factorization methods to probabilistic models, kernel-based methods, and deep learning [3]. However, their systematic evaluation for bulk tissue deconvolution remains largely unexplored.

The HADACA initiative has progressively addressed the evaluation of deconvolution methods through community-driven challenges. The first edition focused on unsupervised approaches for DNAm data, providing initial guidelines and identifying key factors influencing performance [15]. The second edition expanded the framework to both supervised and unsupervised methods across RNA and DNAm data, leading to the DECONbench platform for continuous benchmarking [11]. However, neither edition addressed multi-omics integration for deconvolution, and both were limited in dataset

diversity and preprocessing exploration. HADACA3 addresses these gaps by introducing diverse benchmark datasets and a modular pipeline for systematic evaluation of method combinations. It focuses on early and late integration strategies, which cover most practical approaches but have not yet been systematically compared in this context.

3 HADACA3 benchmark datasets

3.1 Reference profiles used for supervised deconvolution

Participants were provided with reference profiles of pure cell-types, representing five major cell-types found in pancreatic tumors: immune cells, fibroblasts, endothelial cells, and two cancer-cell subtypes: classical and basal-like (Figure 1A and Appendix B Figure 5A). Three types of reference profiles were made available: pure-bulk RNA, pure-bulk DNAm, and scRNA, all sourced from publicly available datasets (Appendix B Tables 5, 6). For the scRNA reference, data were integrated from three studies [33, 4, 35], with up to 5,000 single-cell profiles provided per cell type. Full details on data sources and processing are provided in Appendix B.1.1, data are publicly available (see Appendix Table 5)

3.2 Benchmark datasets

We assembled nine benchmark datasets spanning three experimental settings (Figure 1B, Table 1, Appendix B Figure 5B), designed to evaluate both accuracy and robustness of deconvolution methods under realistic biological and technical conditions. The feature space was restricted to the intersection of RNA and DNAm features available across all benchmark datasets and the reference profiles, yielding a common set of 20,000 RNA and 23,000 DNAm features (Appendix B.1.3).

Table 1: Summary of the benchmark datasets. α: proportion sampling scheme (α_{real}: Dirichlet with realistic parameters; α_{rare}: rare cell type scenario). ε: noise model applied independently to each modality (—: not applicable for real datasets).

Name	Source	Samples	Cell types	α	ε^{RNA}	$\varepsilon^{\text{DNAm}}$
VITR	*in vitro*	30	5	—	—	—
VIVO	*in vivo*	47	2	—	—	—
SBN5	*in silico*	60	5	α_{real}	pseudo-bulk	pseudo-bulk
SDN5	*in silico*	60	5	α_{real}	χ^2	Gaussian
SDN4	*in silico*	60	4	α_{real}	χ^2	Gaussian
SDN6	*in silico*	60	6	α_{real}	χ^2	Gaussian
SDE5	*in silico*	60	5	α_{real}	EM	EM
SDEL	*in silico*	60	5	α_{rare}	EM	EM
SDC5	*in silico*	60	5	α_{real}	Copula+NB	Copula+Beta

***In vivo* and *in vitro* datasets.** The gold standard (VIVO) consists of 47 pancreatic ductal adenocarcinoma (PDAC) surgical samples with matched RNA and DNAm data, for which classical and basal-like tumor cell proportions were estimated from histology slides using PACpAInt [37]. This dataset is original and is released with the benchmark. The silver standard (VITR) is a previously published *in vitro* mixture dataset of nine purified cell types mixed in known proportions [1]. Further details are provided in Appendix B.1.2.

***In silico* datasets.** Six simulated datasets were generated using a linear convolution model, by sampling cell-type proportions from a Dirichlet distribution and combining them with pure cell-type profiles [1]: $Y_i^{(m)} = X^{(m)}A_i + \varepsilon^{(m)}$, where $Y_i^{(m)} \in \mathbb{R}^{F_m}$ denotes the bulk profile of sample i for modalities $m \in \{\text{RNA}, \text{DNAm}\}$, $A_i \in \mathbb{R}^k$ is the vector of cell-type proportions, and $X^{(m)} \in \mathbb{R}^{F_m \times k}$ is the reference profiles with F_m the number of features. The term $\varepsilon^{(m)}$ represents modality-specific noise. Matched RNA and DNAm profiles share the same proportion vector A_i. Simulation scenarios cover heteroscedastic noise (SDN5), rare cell types (SDEL), mismatched compositions (SDN4, SDN6), and noise with structured feature correlations modeled via EM (SDE5) or copula-based

approaches (SDC5). An additional pseudo-bulk dataset (SBN5) was generated by aggregating single-cell profiles, providing a more realistic noise structure. Importantly, the reference profiles provided to participants were entirely distinct from those used to generate the mixtures in the benchmark datasets, avoiding circularity. Full simulation details are provided in Appendix B.1.2.

Data availability. Anonymized code is provided [17], full anonymity cannot be guaranteed due to the use of public datasets [18] and online platforms [16], but a best-effort attempt was made. Complete resources will be released upon acceptance.

4 Competition setup

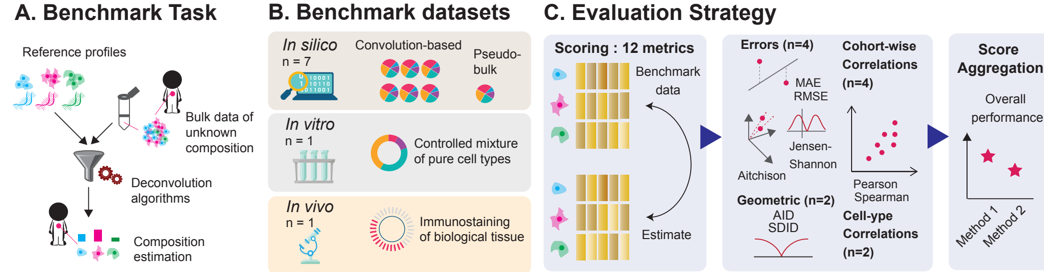


Figure 1: **Schematic representation of the competition setup.**

Task and data overview HADACA3 was a four-day on-site competition hosted on the Codabench platform (December 2–6 2024), which brought together ten teams of four participants [16]. The objective was to perform supervised deconvolution of bulk multi-omics data: given bulk profiles $Y_i^{(m)} \in \mathbb{R}^{F_m}$ and reference profiles $X^{(m)} \in \mathbb{R}^{F_m \times k}$ for modalities $m \in \{\text{RNA}, \text{DNAm}\}$, teams were asked to estimate a shared proportion vector $p_i \in \mathbb{R}^k$ satisfying:

$$Y_i^{(m)} \approx X^{(m)} p_i, \quad \forall m, \quad p_i \geq 0, \quad \sum_{c=1}^{k} p_{ic} = 1. \tag{1}$$

Participants had access to bulk RNA, DNAm, and scRNA reference profiles, and were free to use any deconvolution or integration strategy (Figure 1A). Datasets were split into public sets available during development and private sets used for final evaluation to prevent overfitting (Figure 1B). The competition ran in three phases: an introductory phase on a simple simulated dataset, a robustness phase on multiple datasets with domain shifts, and a generalization phase on private unseen data, with no feedback (Appendix B Table 7).

Evaluation strategy Deconvolution performance was assessed using twelve complementary metrics grouped into four families (Figure 1C): cohort-wise correlations (global and sample-wise, both Pearson and Spearman), cell-type correlations (cell-type-wise Pearson and Spearman), error metrics (RMSE, MAE, Aitchison distance, Jensen-Shannon divergence), and geometric metrics (AID, SDID). Each metric was normalized to $[0, 1]$ and combined into a single aggregate score via a weighted geometric mean, with each family contributing equally one quarter of the total score (Appendix B.2.1 and Appendix B Table 8).

Starting kit and baselines Participants received a starting kit with five baseline implementations (Supplementary Table 9), covering Non-Negative Least Squares (NNLS)-based deconvolution on RNA alone and on both modalities with late averaging, and utility scripts for package installation and external file loading. Baselines were implemented in both R and Python. Submissions were made through Codabench, which provided real-time feedback via a public leaderboard.

Best submission: the JOKER method Among the 627 submitted pipelines, the JOKER method achieved the best overall performance (results available on Codabench [16]). JOKER combined

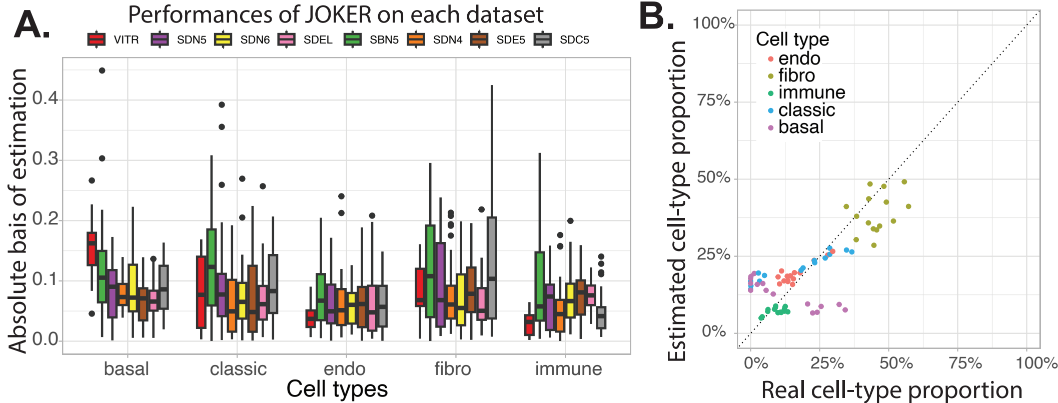


Figure 2: Performances of the JOKER method on cell type proportion estimation. **A.** Boxplots of absolute estimation bias across cell types and benchmark datasets. The absolute bias corresponds to the absolute difference between the estimated and ground-truth proportions for each cell type in each sample. **B.** Example scatterplot comparing estimated and ground-truth proportions on the VITR dataset, where each point corresponds to one sample and colors indicate cell types.

modality-specific preprocessing, independent deconvolution, and rule-based late integration (Appendix B.2.3). For RNA, data were normalized, variance-stabilized, and the 5,000 most variable genes were selected, followed by gene-wise scaling. For DNAm, low-variance CpG sites were removed. Deconvolution was performed independently on each modality using NNLS. Final estimates were obtained via rule-based aggregation: RNA and DNAm predictions were averaged for most cell types, while DNAm-only estimates were retained for the two cancer-cells subtypes (basal-like and classical). This choice was motivated by empirical observations that DNAm provided more stable and consistent estimates for these subtypes across datasets. The robustness of JOKER was supported by similar absolute bias distribution in cell-type estimation across benchmark datasets (Figure 2A), indicating reproducible performance patterns. At the cell-type level, however, estimation errors varied, with higher errors for basal and fibroblast cell types for instance for VITR dataset (Figure 2B).

Despite its top performance, JOKER is a monolithic pipeline whose design choices, preprocessing, feature selection, deconvolution algorithm, and integration strategy, are tightly coupled and cannot be evaluated independently. This raises a natural question: which of these choices actually drives performance, and do they generalize to other settings?

5 Benchmark of multi-omics data integration for bulk tissue deconvolution

Objective and design To fairly evaluate multi-omics integration strategies and identify which pipeline modules drive overall performance, we decomposed JOKER and other top submissions into four key modules: (i) preprocessing, (ii) feature selection, (iii) deconvolution, and (iv) multi-omics integration. We implemented a modular Nextflow pipeline to systematically evaluate all compatible combinations (Appendix B Figure 7), totaling 50,939 combinations for early integration and 219,093 for late integration. Each pipeline is a specific combination obtained by selecting one method for each of the four modules described in Table 2. This design enables quantifying the contribution of each module to the performance, identifying robust strategies across settings, and assessing whether multi-omics integration consistently improves over uni-modal approaches. Performance was evaluated across the 9 benchmark datasets (Table 1) using the aggregate score defined in Section 4. All experiments were conducted using the open-source Nextflow framework we developed and made available on GitHub [17], ensuring full reproducibility.

Early integration methods Early integration methods (Table 3) consist in combining RNA and DNAm data at the feature level prior to deconvolution. In this setting, a joint representation $\tilde{X}$ and $\tilde{Y}$ is constructed:

$$\tilde{Y}, \tilde{X} = \mathcal{F}_{\text{early}}\left(Y^{\text{RNA}}, Y^{\text{DNAm}}, X^{\text{RNA}}, X^{\text{DNAm}}\right), \tag{2}$$

Table 2: Methods available in each module of the benchmark pipeline. A complete pipeline is defined by selecting one method per module. Each method is fully described in Appendix B.3.

Module	Sub-module	Method	Brief description
Preprocessing	RNA DNAm	`ppID`	No preprocessing (identity)
		`Scale`	Column-sum normalization
		`LogNorm`	Log-normalization via Seurat
Feature selection	RNA	`fsID`	No selection (identity)
		`Toastbulknbfs`	Top 1,000 marker genes via TOAST on bulk reference
		`Toastvst`	Top 1,000 marker genes via TOAST on VST-transformed bulk reference
		`SCcluster`	Differential expression markers from clustered scRNA
		`scpseudobulk`	Pairwise t-test markers from scRNA
	DNAm	`fsID`	No selection (identity)
		`Toastpercent`	Top 80% probes via TOAST
		`mostmethylated`	Probes above 75th percentile per cell type
		`maxdiscriminant`	Maximally discriminant non-overlapping probes
		`splsda`	Sparse PLS-DA on logit-transformed reference
Deconvolution	–	`lm`	Ordinary least squares
		`nnls`	Non-negative least squares (NNLS)
		`nnlslargeref`	NNLS with iterative reference truncation
		`epic`	Constrained least squares with internal TPM normalization
		`RLR`	Robust linear regression via IRLS
		`RLRpoisson`	RLR with Poisson-inspired feature weights
		`RLRnnls`	Ensemble: RLR and NNLS selected by reconstruction RMSE
Integration	None	`onlyRNA`	RNA proportions only
		`onlyDNAm`	DNAm proportions only
	Early	`concatnoscale`	Raw feature concatenation
		`concatscale`	Normalized concatenation
		`omicade4bulk`	Latent linear embedding via MCIA
		`Kernel`	Non-linear kernel embedding via kernel PCA
		`OT`	Optimal transport-based representation via uniPort
	Late	`limean`	Uniform averaging
		`limeanRMSE`	Error-weighted aggregation
		`tunedJ`	Rule-based selective averaging

where $\mathcal{F}_{\text{early}}$ denotes a transformation function, that differs in the proposed methods. Deconvolution is then performed on the transformed data such that: $\tilde{Y}_i \approx \tilde{X} p_i$ with $p_i \geq 0$ and $\sum_{c=1}^{k} p_{ic} = 1$.

We categorize early integration methods into feature-level, latent, and transport-based families. All the methods ultimately aim to improve estimation of the shared latent variable p_i by reducing cross-modal discrepancies between RNA and DNAm (see Appendix B.3.4 for detailed explanations).

Table 3: Comparison of early integration strategies for multi-omics deconvolution. Each method defines an integration operator $\mathcal{F}_{\text{early}}$, which is applied prior to deconvolution.

Method	Principle	Key idea	Main limitation
concatnoscale	Raw feature concatenation	Stacks RNA expression and DNAm level without transformation; preserves original scale and structure	Sensitive to feature magnitude differences between modalities
concatscale	Normalized concatenation	Concatenation followed by sample-wise Gaussianization across features	Ignores cross-modal correlationsand dependancies
omicade4bulk	Latent linear embedding	Learns a low-dimensional representation via multiblock co-inertia analysis	Sensitive to non-linear cross-omic relationships
Kernel	Non-linear kernel embedding	Constructs a joint non-linear representation via kernel PCA over combined modality-specific kernels	Sensitive to kernel choice and computationally expensive on large datasets
OT	Optimal transport-based representation	Aligns RNA and DNAm distributions through optimal transport using uniPort [8], and learns a coupled latent space	High computational cost and dependence on distance metric quality

Late integration methods Late integration methods (Table 4) estimate cell-type proportions by performing deconvolution separately for each modality and then combining the resulting predictions:

$$Y_i^{\text{RNA}} \approx X^{\text{RNA}} \hat{p}_i^{\text{RNA}}, \quad Y_i^{\text{DNAm}} \approx X^{\text{DNAm}} \hat{p}_i^{\text{DNAm}}, \quad \hat{p}_i = \mathcal{F}_{\text{late}}\big(\hat{p}_i^{\text{RNA}}, \hat{p}_i^{\text{DNAm}}\big), \tag{3}$$

where $\mathcal{F}_{\text{late}}$ is an aggregation function that varies across methods. Late integration combines modality-specific estimates after independent deconvolution. All methods aim to improve robustness of the final estimate $\hat{p}_i$ by combining modality-specific predictions (see Appendix B.3.6 for detailed explanations).

Table 4: Comparison of late integration strategies for multi-omics deconvolution. Each method combines modality-specific deconvolution outputs via an aggregation operator $\mathcal{F}_{\text{late}}$.

Method	Principle	Key idea	Main limitation
limean	Uniform averaging	Averages RNA- and DNAm-based proportion estimates with equal weights	Assumes equal reliability of both modalities
limeanRMSE	Error-weighted aggregation	Weights modality-specific estimates using normalized reconstruction errors	Depends on accurate RMSE estimation; sensitive to reconstruction bias
tunedJ	Rule-based selective averaging	Selects or combines modalities depending on biological context (e.g., cancer vs normal)	Requires hand-crafted decision rules; requires prior biological knowledge

6 Results

A key question of this benchmark is whether integrating RNA and DNAm improves deconvolution over single-modality approaches. Our results show that multi-modal integration can match but rarely exceed DNAm alone, and that the benefit of integration is highly context-dependent.

DNAm-only dominates median performance while integration strategies show context-dependent gains Early and late integration achieve comparable peak scores overall (≈ 0.8), but median performances differ substantially. Across datasets, DNAm-only achieves the highest median performance and is rarely outperformed by multi-omics integration, although some integration strategies yield gains in specific pipelines or datasets (Figure 3A-C). For early integration, `concatscale` is the most consistently well-performing strategy, combining strong average performance, low computational cost, and robustness across pipeline configurations. The optimal transport-based method

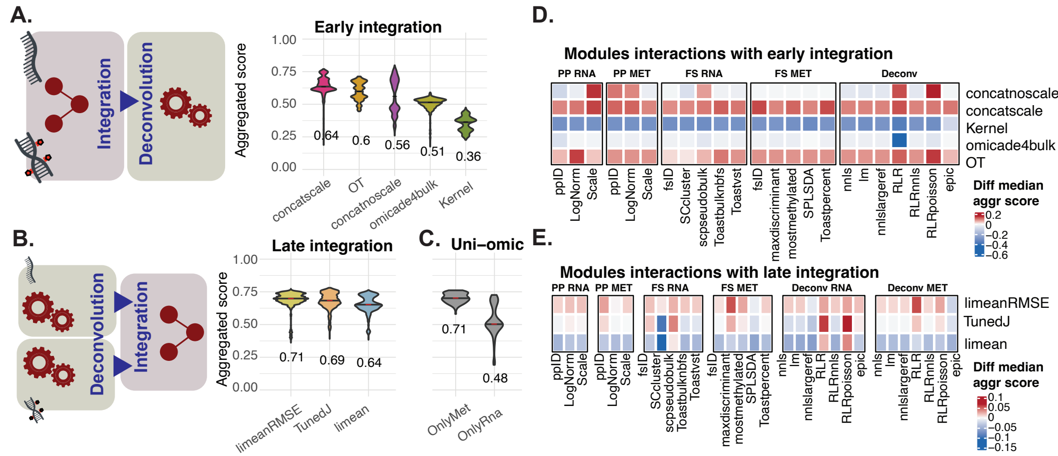


Figure 3: **Performance of integration strategies across preprocessing, feature selection, and deconvolution choices. A, B.** Median aggregated performance across integration strategies. Panel A shows early integration methods with $N \sim 10{,}000$ pipeline combinations per method, and panel B late integration methods with $N \sim 72{,}000$ combinations. **C** Uni-omic performs deconvolution using a single omic independently, with $N \sim 500$ combinations per omics. Median aggregated score are indicated. **D, E.** Pairwise interaction effects between integration strategies and other pipeline modules. Rows correspond to integration methods, while columns correspond to preprocessing (PP), feature selection (FS), and deconvolution (Deconv) modules, stratified by modality (RNA and DNAm). Panel D shows early integration, and panel E late integration. Colors indicate how the median aggregated score of each method combination deviates from the global median score, computed across all evaluated pipelines.

`OT` achieves competitive results, albeit at higher computational cost. In contrast, latent embedding methods (`omicade4bulk`, `Kernel`) exhibit high variance and weaker performances in several settings (Figure 3D). Among late integration strategies, `limeanRMSE` achieves comparable performance to DNAm-only while remaining agnostic to modality discriminative power, and may be employed as an integrative strategy when the relative reliability of each modality is not known in advance (Figure 3E).

Contribution and interaction of preprocessing, feature selection, and deconvolution. Preprocessing has little impact on late integration performance (Appendix B, Figure 8). In contrast, for early integration, RNA scaling or log-normalization consistently improves results, likely by reducing the scale mismatch between RNA counts and bounded DNAm values. Feature selection shows limited overall impact, but interacts strongly with the integration regime: strategies that perform well for early integration are not necessarily optimal for late integration (Appendix B, Figure 9). Consistent with previous benchmarks [1], RLR-based methods achieve the best deconvolution performance (Appendix B, Figure 10), with `RLRpoisson` further improving RNA-seq results. However, this advantage is reduced when using the `concatscale` early integration method, suggesting that normalization mitigates sensitivity to the choice of deconvolution model (Figure 3D).

Integration performance varies across datasets, with OT favored in realistic biological settings. Figure 4 shows that DNAm achieves the best average performance (median aggregated score, left), but is frequently outperformed by specific integration pipelines (top aggregated score, right). This indicates that, while integration is not consistently beneficial, it can yield gains in well-tuned settings and highlights promising directions for future methodological development. For early integration, method rankings based on median performance are largely consistent across simulation scenarios (Figure 4, left), regardless of the noise model (heteroscedastic-SDN5, EM-based-SDE5, or copula-SDC5), the number of cell types (missing-SDN4 or extra-SDN6), their abundance (rare cell types-SDEL), or the simulation paradigm (convolution-based or pseudo-bulk-SBN5). This suggests that early integration performance is relatively stable across simulation designs. In contrast, differences emerge on real datasets: the optimal transport-based method (`OT`) performs particularly well on average on the *in vitro* and *in vivo* data compared to uni-omic strategies, possibly reflecting the

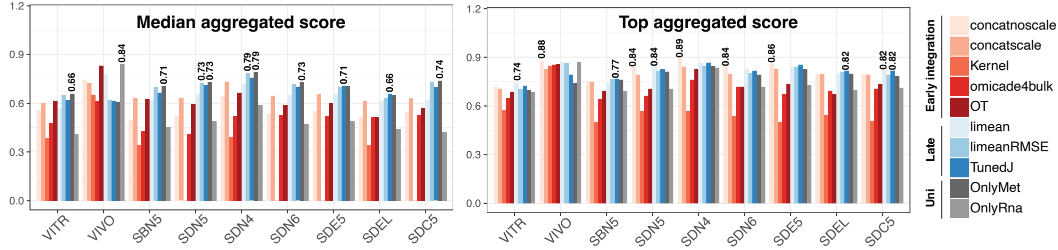


Figure 4: **Performance variability of integration strategies across datasets and metrics.** Median (left) and best (right) aggregated benchmark scores for each integration method across datasets. The best score achieved per dataset is reported above the corresponding bar. All top-performing pipelines are detailed in Appendix B, Figures 11 & 12 and Tables 14 & 15.

need for non-linear alignment under more complex and realistic biological distributions. When considering top-performing pipelines (Figure 4, right; Appendix B Figure 11), `concatnoscale` can achieve strong performance when combined with specific preprocessing strategies, highlighting the importance of pipeline interactions. For late integration, `limeanRMSE` is the most robust strategy on average, ranking first across 6 out of 9 datasets (Figure 4, left). `tunedJ` achieves strong top performance in datasets with complex noise structures (SDE5 and SDC5) (Figure 4, right; Appendix B Figure 12). Finally, metric-level analysis shows that cell-type correlation and Aitchison distance are the most discriminative components of the aggregate score (Appendix B, Figure 13). Lower correlations are observed in datasets with more complex noise structures, indicating increased difficulty in recovering cell-type-specific variation.

7 Limitations

Several limitations should be acknowledged. First, all datasets focus on pancreatic cancer. While this focus allowed us to assemble high-quality, matched multi-omics datasets with biologically validated ground truth, it limits the direct generalizability of our conclusions to other tissues or cancer types. In particular, the dominance of DNAm over RNA in our benchmark may reflect properties specific to pancreatic cancer, such as the strong epigenetic distinction between basal-like and classical tumor subtypes [23], rather than a universal property of multi-omics deconvolution. We encourage future work to evaluate whether these findings replicate in tissues with different cellular compositions, reference quality, and omic signal structures. The datasets and pipeline introduced here are designed to facilitate such extensions. Second, performance of supervised deconvolution is inherently dependent on the quality and representativeness of the reference profiles provided, a limitation shared by all supervised methods but not systematically evaluated here. Third, our benchmark covers early and late integration paradigms but does not evaluate intermediate integration strategies, where integration is co-designed with the deconvolution model and cannot be easily decoupled into modular components. While these approaches are not directly compatible with our modular evaluation framework, all datasets and the Nextflow pipeline are publicly available to enable their future extension within this setting. Fourth, RNA-only deconvolution underperforms across settings, potentially due to lower-quality reference profiles and to higher sparsity and zero-inflation, which may disproportionately affect metrics such as the Aitchison distance and bias the aggregate score. Fifth, given the large number of pipeline combinations evaluated (>250,000), even small differences in aggregate score may appear consistent without being statistically meaningful. Our primary analyses are based on median aggregate scores across pipeline combinations and datasets, which provides a robust summary but does not formally quantify uncertainty. Finally, while our simulation framework covers a range of noise models and compositional scenarios, it does not explicitly model complex cross-modal correlation structures, which may be relevant in real multi-omics settings.

8 Conclusion and future work

HADACA3 provides the first systematic, community-driven evaluation of multi-omics integration strategies for bulk tissue deconvolution, spanning over 250,000 method combinations across four key pipeline modules. Our main finding is that multi-modal integration does not consistently outperform the best uni-modal strategy: DNAm alone provides the most stable performance across datasets and pipeline configurations. However, integration can yield the best results in specific, well-tuned settings. In particular, optimal transport-based integration shows strong performance on real biological datasets, suggesting that non-linear alignment may better capture complex cross-modal relationships.

Mechanistic interpretations. Our findings can be explained by the complementary statistical properties of the modalities. DNAm profiles exhibit strong cell-type specificity and low variability, making them well suited for linear mixture modeling. In contrast, RNA is more affected by sparsity and technical variability, which may reduce its effectiveness under standard deconvolution metrics. Early integration is sensitive to scale differences and noise heterogeneity across modalities, which normalization only partially corrects. Latent representation methods may further deviate from the linear mixing assumption by optimizing cross-modal alignment rather than preserving the structure $Y \approx Xp$. Late integration avoids these issues by decoupling modality-specific inference from aggregation, resulting in improved robustness across heterogeneous settings. This highlights a trade-off between tight cross-modal coupling (early integration) and robustness (late integration).

Broader significance and future directions. From a practical perspective, our results suggest using `concatscale` or `OT` for early integration, and `limeanRMSE` for late integration, as promising directions that merit further investigation, while single-modality approaches remain strong baselines. Beyond deconvolution, our findings contribute to a broader understanding of multi-omics integration. Future work should extend this benchmark to other complex tissues and further investigate intermediate integration strategies. All datasets and pipelines are publicly available through Codabench, enabling reproducible evaluation and future method development.

References

[1] Elise Amblard, Vadim Bertrand, Luis Martin Pena, Slim Karkar, Florent Chuffart, Mira Ayadi, Aurelia Baures, Lucile Armenoult, Yasmina Kermezli, Jerome Cros, Yuna Blum, and Magali Richard. A robust workflow to benchmark deconvolution of multi-omic data, November 2024. Pages: 2024.11.08.622633 Section: New Results.

[2] Francisco Avila Cobos, José Alquicira-Hernandez, Joseph E. Powell, Pieter Mestdagh, and Katleen De Preter. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nature Communications*, 11(1):5650, November 2020. Number: 1.

[3] Ana R Baião, Zhaoxiang Cai, Rebecca C Poulos, Phillip J Robinson, Roger R Reddel, Qing Zhong, Susana Vinga, and Emanuel Gonçalves. A technical review of multi-omics data integration methods: from classical statistical to deep generative approaches. *Briefings in Bioinformatics*, 26(4):bbaf355, July 2025.

[4] Maayan Baron, Adrian Veres, Samuel L. Wolock, Aubrey L. Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K. Wagner, Shai S. Shen-Orr, Allon M. Klein, Douglas A. Melton, and Itai Yanai. A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Systems*, 3(4):346–360.e4, October 2016.

[5] Effie E. Bastounis, Yi-Ting Yeh, and Julie A. Theriot. Subendothelial stiffness alters endothelial cell traction force generation while exerting a minimal effect on the transcriptome. *Scientific Reports*, 9(1):18209, December 2019. Publisher: Nature Publishing Group.

[6] Etienne Becht, Nicolas A. Giraldo, Laetitia Lacroix, Bénédicte Buttard, Nabila Elarouci, Florent Petitprez, Janick Selves, Pierre Laurent-Puig, Catherine Sautès-Fridman, Wolf H. Fridman, and Aurélien de Reyniès. Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression. *Genome Biology*, 17(1):1–20, December 2016. Number: 1.

[7] Zhaoxiang Cai, Rebecca C. Poulos, Jia Liu, and Qing Zhong. Machine learning for multi-omics data integration in cancer. *iScience*, 25(2):103798, February 2022.

[8] Kai Cao, Qiyu Gong, Yiguang Hong, and Lin Wan. A unified computational framework for single-cell data integration with optimal transport. *Nature Communications*, 13(1):7419, December 2022.

[9] Tim H. H. Coorens, Amy Guillaumet-Adkins, Rothem Kovner, Rebecca L. Linn, Victoria H. J. Roberts, Amrita Sule, and Patrick M. Van Hoose. The human and non-human primate developmental GTEx projects. *Nature*, 637(8046):557–564, January 2025. Publisher: Nature Publishing Group.

[10] Kobe De Ridder, Huiwen Che, Kaat Leroy, and Bernard Thienpont. Benchmarking of methods for DNA methylome deconvolution. *Nature Communications*, 15(1):4134, May 2024.

[11] Clémentine Decamps, Alexis Arnaud, Florent Petitprez, Mira Ayadi, Aurélia Baurès, Lucile Armenoult, HADACA consortium, Sergio Escalera, Isabelle Guyon, Rémy Nicolle, Richard Tomasini, Aurélien de Reyniès, Jérôme Cros, Yuna Blum, and Magali Richard. DECONbench: a benchmarking platform dedicated to deconvolution methods for tumor heterogeneity quantification. *BMC bioinformatics*, 22(1):473, October 2021.

[12] Maísa R Ferro dos Santos, Edoardo Giuili, Andries De Koker, Celine Everaert, and Katleen De Preter. Computational deconvolution of DNA methylation data from mixed DNA samples. *Briefings in Bioinformatics*, 25(3):bbae234, May 2024.

[13] Julia Franzen, Anne Zirkel, Jonathon Blake, Björn Rath, Vladimir Benes, Argyris Papantonis, and Wolfgang Wagner. Senescence-associated DNA methylation is stochastically acquired in subpopulations of mesenchymal stem cells. *Aging Cell*, 16(1):183–191, 2017. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/acel.12544.

[14] Chloé Friguet, Kloareg , Maela, , and David Causeur. A Factor Model Approach to Multiple Testing Under Dependence. *Journal of the American Statistical Association*, 104(488):1406–1415, December 2009. Publisher: ASA Website _eprint: https://doi.org/10.1198/jasa.2009.tm08332.

[15] HADACA consortium, Clémentine Decamps, Florian Privé, Raphael Bacher, Daniel Jost, Arthur Waguet, Eugene Andres Houseman, Eugene Lurie, Pavlo Lutsik, Aleksandar Milosavljevic, Michael Scherer, Michael G. B. Blum, and Magali Richard. Guidelines for cell-type heterogeneity quantification based on a comparative analysis of reference-free DNA methylation deconvolution software. *BMC Bioinformatics*, 21(1):16, December 2020.

[16] HADACA3 Consortium. Hadaca3 benchmark: Multi-omic deconvolution challenge. `https://www.codabench.org/competitions/4714/`, 2024. Online benchmark hosted on Codabench.

[17] HADACA3 Consortium. Hadaca3 framework: Modular nextflow pipeline for multi-omic deconvolution benchmarking, 2026. `https://anonymous.4open.science/r/hadaca3_framework-08D7` - Will be released upon acceptance.

[18] HADACA3 Consortium. Hadaca3 in silico multi-omic benchmark datasets, 2026. Zenodo repository : `https://zenodo.org/records/19677979`.

[19] Christopher J. Halbrook, Costas A. Lyssiotis, Marina Pasca Di Magliano, and Anirban Maitra. Pancreatic cancer: Advances and challenges. *Cell*, 186(8):1729–1754, April 2023.

[20] Douglas Hanahan. Hallmarks of cancer—Then and now, and beyond. *Cell*, 189(8):2254–2277, April 2026.

[21] Eugene Andres Houseman, William P. Accomando, Devin C. Koestler, Brock C. Christensen, Carmen J. Marsit, Heather H. Nelson, John K. Wiencke, and Karl T. Kelsey. DNA methylation arrays as surrogate measures of cell mixture distribution. *BMC Bioinformatics*, 13(1):86, May 2012.

[22] Ziyi Li and Hao Wu. Toast: improving reference-free cell composition estimation by cross-cell type differential analysis. *Genome Biology*, 20(1):18, 2019.

[23] Gwen Lomberk, Yuna Blum, Rémy Nicolle, Asha Nair, Krutika Satish Gaonkar, Laetitia Marisa, Angela Mathison, Zhifu Sun, Huihuang Yan, Nabila Elarouci, et al. Distinct epigenetic landscapes underlie the pathobiology of pancreatic cancer subtypes. *Nature communications*, 9(1):1978, 2018.

[24] Kim-Anh Lê Cao, Simon Boitard, and Philippe Besse. Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, 12(1):253, June 2011.

[25] Jérôme Mariette and Nathalie Villa-Vialaneix. Unsupervised multiple kernel learning for heterogeneous data integration. *Bioinformatics*, 34(6):1009–1015, March 2018.

[26] Chen Meng, Bernhard Kuster, Aedín C. Culhane, and Amin Moghaddas Gholami. A multivariate approach to the integration of multi-omics datasets. *BMC bioinformatics*, 15:162, May 2014.

[27] Katharine M. Mullen and Ivo H. M. Van Stokkum. nnls: The Lawson-Hanson Algorithm for Non-Negative Least Squares (NNLS), October 2007. Institution: Comprehensive R Archive Network Pages: 1.6.

[28] Yonghyun Nam, Jaesik Kim, Sang-Hyuk Jung, Jakob Woerner, Erica H. Suh, Dong-gi Lee, Manu Shivakumar, Matthew E. Lee, and Dokyoon Kim. Harnessing AI in Multi-Modal Omics Data Integration: Paving the Path for the Next Frontier in Precision Medicine. *Annual review of biomedical data science*, 7(1):225–250, August 2024.

[29] NCBI Gene Expression Omnibus. Generation of multi-omic datasets using high-throughput molecular profiling of rna data in human pancreatic cancer (pdac). `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE328792`, 2026. NCBI Gene Expression Omnibus (GEO) GSE328792.

[30] Aaron M Newman, Chih Long Liu, Michael R Green, Andrew J Gentles, Weiguo Feng, Yue Xu, Chuong D Hoang, Maximilian Diehn, and Ash A Alizadeh. Robust enumeration of cell subsets from tissue expression profiles. *Nature Methods*, 12(5):453–457, May 2015.

[31] Rémy Nicolle, Yuna Blum, Pauline Duconseil, Charles Vanbrugghe, Nicolas Brandone, Flora Poizat, Julie Roques, Martin Bigonnet, Odile Gayet, Marion Rubis, Nabila Elarouci, Lucile Armenoult, Mira Ayadi, Aurélien de Reyniès, Marc Giovannini, Philippe Grandval, Stephane Garcia, Cindy Canivet, Jérôme Cros, Barbara Bournet, Vincent Moutardier, Marine Gilabert, Juan Iovanna, Nelson Dusetti, and Louis Buscail. Establishment of a pancreatic adenocarcinoma molecular gradient (PAMG) that predicts the clinical outcome of pancreatic cancer. *EBioMedicine*, 57:102858, July 2020.

[32] Rémy Nicolle, Yuna Blum, Laetitia Marisa, Celine Loncle, Odile Gayet, Vincent Moutardier, Olivier Turrini, Marc Giovannini, Benjamin Bian, Martin Bigonnet, Marion Rubis, Nabila Elarouci, Lucile Armenoult, Mira Ayadi, Pauline Duconseil, Mohamed Gasmi, Mehdi Ouaissi, Aurélie Maignan, Gwen Lomberk, Jean-Marie Boher, Jacques Ewald, Erwan Bories, Jonathan Garnier, Anthony Goncalves, Flora Poizat, Jean-Luc Raoul, Veronique Secq, Stephane Garcia, Philippe Grandval, Marine Barraud-Blanc, Emmanuelle Norguet, Marine Gilabert, Jean-Robert Delpero, Julie Roques, Ezequiel Calvo, Fabienne Guillaumond, Sophie Vasseur, Raul Urrutia, Aurélien de Reyniès, Nelson Dusetti, and Juan Iovanna. Pancreatic Adenocarcinoma Therapeutic Targets Revealed by Tumor-Stroma Cross-Talk Analyses in Patient-Derived Xenografts. *Cell Reports*, 21(9):2458–2470, November 2017. Publisher: Elsevier.

[33] Junya Peng, Bao-Fa Sun, Chuan-Yuan Chen, Jia-Yi Zhou, Yu-Sheng Chen, Hao Chen, Lulu Liu, Dan Huang, Jialin Jiang, Guan-Shen Cui, Ying Yang, Wenze Wang, Dan Guo, Menghua Dai, Junchao Guo, Taiping Zhang, Quan Liao, Yi Liu, Yong-Liang Zhao, Da-Li Han, Yupei Zhao, Yun-Gui Yang, and Wenming Wu. Single-cell RNA-seq highlights intra-tumoral heterogeneity and malignant progression in pancreatic ductal adenocarcinoma. *Cell Research*, 29(9):725–738, September 2019.

[34] Julien Racle, Kaat de Jonge, Petra Baumgaertner, Daniel E Speiser, and David Gfeller. Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data. *eLife*, 6:e26476, November 2017.

[35] Srivatsan Raghavan, Peter S. Winter, Andrew W. Navia, Hannah L. Williams, Alan DenAdel, Kristen E. Lowder, Jennyfer Galvez-Reyes, Radha L. Kalekar, Nolawit Mulugeta, Kevin S. Kapner, Manisha S. Raghavan, Ashir A. Borah, Nuo Liu, Sara A. Väyrynen, Andressa Dias Costa, Raymond W. S. Ng, Junning Wang, Emma K. Hill, Dorisanne Y. Ragon, Lauren K. Brais, Alex M. Jaeger, Liam F. Spurr, Yvonne Y. Li, Andrew D. Cherniack, Matthew A. Booker, Elizabeth F. Cohen, Michael Y. Tolstorukov, Isaac Wakiro, Asaf Rotem, Bruce E. Johnson, James M. McFarland, Ewa T. Sicinska, Tyler E. Jacks, Ryan J. Sullivan, Geoffrey I. Shapiro, Thomas E. Clancy, Kimberly Perez, Douglas A. Rubinson, Kimmie Ng, James M. Cleary, Lorin Crawford, Scott R. Manalis, Jonathan A. Nowak, Brian M. Wolpin, William C. Hahn, Andrew J. Aguirre, and Alex K. Shalek. Microenvironment drives cell state, plasticity, and drug response in pancreatic cancer. *Cell*, 184(25):6119–6137.e26, December 2021. Publisher: Elsevier.

[36] Naim U. Rashid, Xianlu L. Peng, Chong Jin, Richard A. Moffitt, Keith E. Volmar, Brian A. Belt, Roheena Z. Panni, Timothy M. Nywening, Silvia G. Herrera, Kristin J. Moore, Sarah G. Hennessey, Ashley B. Morrison, Ryan Kawalerski, Apoorve Nayyar, Audrey E. Chang, Benjamin Schmidt, Hong Jin Kim, David C. Linehan, and Jen Jen Yeh. Purity Independent Subtyping of Tumors (PurIST), A Clinically Robust, Single-sample Classifier for Tumor Subtyping in Pancreatic Cancer. *Clinical Cancer Research: An Official Journal of the American Association for Cancer Research*, 26(1):82–92, January 2020.

[37] Charlie Saillard, Flore Delecourt, Benoit Schmauch, Olivier Moindrot, Magali Svrcek, Armelle Bardier-Dupas, Jean Francois Emile, Mira Ayadi, Vinciane Rebours, Louis de Mestier, Pascal Hammel, Cindy Neuzillet, Jean Baptiste Bachet, Juan Iovanna, Nelson Dusetti, Yuna Blum, Magali Richard, Yasmina Kermezli, Valerie Paradis, Mikhail Zaslavskiy, Pierre Courtiol, Aurelie Kamoun, Remy Nicolle, and Jerome Cros. Pacpaint: a histology-based deep learning model uncovers the extensive intratumor molecular heterogeneity of pancreatic adenocarcinoma. *Nature Communications*, 14(1):3459, June 2023. Number: 1 Publisher: Nature Publishing Group.

[38] Anna Vathrakokoili Pournara, Zhichao Miao, Ozgur Yilimaz Beker, Nadja Nolte, Alvis Brazma, and Irene Papatheodorou. CATD: a reproducible pipeline for selecting cell-type deconvolution methods across tissues. *Bioinformatics Advances*, 4(1):vbae048, 2024.

[39] Maik Wolfram-Schauerte, Thomas Vogel, Hanati Tuoken, Maria Fälth Savitski, Eric Simon, and Kay Nieselt. Approaching the holistic transcriptome—convolution and deconvolution in transcriptomics. *Briefings in Bioinformatics*, 26(4):bbaf388, July 2025.

[40] Shijie C Zheng, Charles E Breeze, Stephan Beck, Danyue Dong, Tianyu Zhu, Liangxiao Ma, Wei Ye, Guoqing Zhang, and Andrew E Teschendorff. EpiDISH web server: Epigenetic Dissection of Intra-Sample-Heterogeneity with online GUI. *Bioinformatics*, 36(6):1950–1951, March 2020.

9 Acknowledgments

We thank all supporting organizations: GDR BIMMM, Région Auvergne-Rhône-Alpes, M4DI PEPR Santé Numérique, ITMO Cancer Aviesan, La Ligue Française contre le cancer, LabEx PERSYVAL-2, RT Math Bio Santé (CNRS), CLARA, RIS, EFELIA-MIAI, and the GRICAD mesocenter at UGA. We thank the CAES Paul Langevin facility in Aussois for hosting, and Isabelle Guyon, Franck Picard and Charles Lecellier for fruitful discussions. This work was supported by ANR (CauseHet, ANR-22-CE45-0030), France 2030 (ANR-22-PESN-0013, ANR-23-IACL-0006), and ITMO Cancer of Aviesan / Inserm (ACACIA, AAP-MIC-2021).

A HADACA3 Consortium Members

We thank all members of the HADACA3 consortium for helpful discussion and contributions during the data challenge (December 2024, Aussois, France).

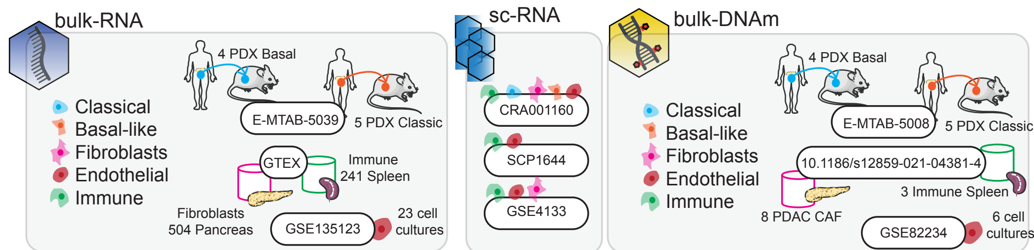
HADACA3 collaborating authors:

Al-Shahrour Fatima (Bioinformatics Unit, CNIO, Madrid, SP); Amblard Elise (Univ. Grenoble Alpes, CNRS, LIG, Grenoble, FR); Appé Guillaume (Epigene Labs, Paris, FR); Agonkoui Christelle (Université de Lille / Inria MODAL, FR); Barbot Hugo (IRMAR CNRS, Institut Agro Rennes Angers, FR); Baudot Anaïs (Aix Marseille Univ, INSERM, MMG, Marseille, FR); Beaufils Audrey (Université Paris Cité, CRI, INSERM U1149, FR); Becht Etienne (Université Paris Cité, CRI, INSERM U1149, FR); Benoit Clément (Institut de Biologie et Pathologie, CHUGA, FR); Blum Yuna (IGDR CNRS, Université de Rennes, FR); Burban Ewen (Univ Rennes, Inserm, Irset UMR_S 1085, FR); Carpentier Océane (Univ Rennes, Inria, CNRS, IRISA, FR); Causeur David (IRMAR CNRS, Institut Agro Rennes Angers, FR); Chassagnol Bastien (Aix Marseille Univ, INSERM, MMG, Marseille, FR); Chepeleva Maryna (Luxembourg Institute of Health, LU); Chuffart Florent (Univ. Grenoble Alpes, INSERM, CNRS, IAB, FR); Dejean Sebastien (IMT CNRS, Université de Toulouse, FR); De Wit Renske (Netherlands Cancer Institute / Utrecht University, NL); Deligne Margaux (CentraleSupélec / MICS, FR); Derouin Margot (INSERM UMR1141 NeuroDiderot, Paris, FR); Dhar Gaurav (Toulouse School of Economics, FR); Essabbar Abdelmounim (CRCT, Université de Toulouse, INSERM, CNRS, FR); Garreau Jules (IGDR CNRS, Université de Rennes, FR); Ghestem Florence (Université Paris-Saclay, UVSQ, Inserm, CESP, FR); Gorse Marine (Univ Rennes, Inserm, Irset UMR_S 1085, FR); Herrmann Carl (BioQuant, IPMB, Heidelberg University, DE); Homberg Nicolas (Univ. Grenoble Alpes, CNRS, LIG, FR); Hughes Arthur (Bordeaux Population Health, INSERM U1219, FR); Hurtado Marcelo (CRCT, Université de Toulouse, INSERM, CNRS, FR); Kubica Jędrzej (Univ. Grenoble Alpes, CNRS, TIMC, FR); Lamothe Lucie (Univ. Grenoble Alpes, CNRS, LIG, FR); Laval Quentin (INSERM, Bordeaux Population Health, FR); Le Clech Renée (HeKa UMR1346, Inserm, Inria, Université Paris Cité, FR); Loire Benjamin (Aix Marseille Univ, INSERM, MMG, FR); Louistisserand Juliette (Univ. Grenoble Alpes, CNRS, LIG, FR); Lukic Vesna (CentraleSupélec / MICS, FR); Malleval Inès (CentraleSupélec / MICS, FR); Marcou Quentin (Inria–Inserm COMPO, CRCM, Marseille, FR); Menard Thomas (CentraleSupélec / MICS, FR); Najm Matthieu (Institut Pasteur, Université Paris Cité, FR); Nosirov Bakhtiyor (Luxembourg Institute of Health, LU); Pal Annesh (Bordeaux Population Health, INSERM U1219, FR); Pignolet Camille (Université Paris Cité, CRI, INSERM U1149, FR); Privé Florian (Aarhus University, DK); Rau Andrea (Université Paris-Saclay, INRAE, AgroParisTech, GABI, FR); Richard Magali (Univ. Grenoble Alpes,

CNRS, LIG, FR); Spinelli Lionel (Aix-Marseille Univ, CNRS, INSERM, CIML, FR); Tabbakh Danial (Luxembourg Institute of Health, LU); Térézol Morgane (Aix Marseille Univ, INSERM, MMG, Marseille, FR); Toffano Antoine (Université de Montpellier, LIRMM UMR5506, FR); Valentin Lorena (Aix Marseille Univ, INSERM, MMG, Marseille, FR); Ventre Elias (Inria–Inserm COMPO, CRCM, Marseille, FR); Vignes Matthieu (Massey University, NZ); Von Grafenstein Klaus (Univ Rennes, Inserm, Irset UMR_S 1085, FR); Weill Solene (Epigene Labs, Paris, FR).

B Appendix : Supplementary Material, Tables and Figures

B.1 HADACA3 datasets



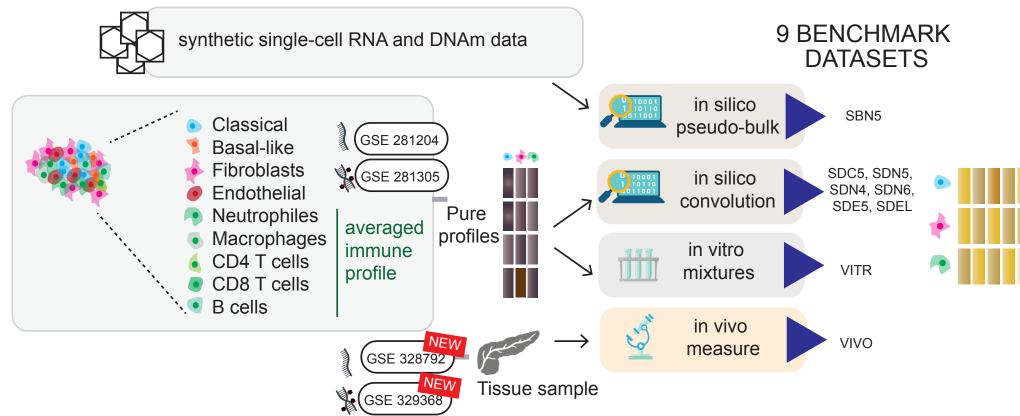


Figure 5: **A.** Schematic representation of reference data construction. **B.** Schematic representation of benchmark data construction.

B.1.1 Reference data used for supervised deconvolution

Bulk DNAm et RNA Participants were provided with reference profiles of five major cell types found in pancreatic tumors: immune cells, fibroblasts, endothelial cells, and two tumor subtypes: classical and basal-like (Supplementary Figure 5A). All reference data were sourced from publicly available datasets. For pure-bulk RNA and pure-bulk DNAm, replicates corresponding to a given cell type were aggregated into a single meta-reference profile by averaging all available cell-type-specific profiles (Supplementary Table 5).

- Transcriptomic profiles of immune cells and fibroblasts were obtained from the GTEx Analysis [9], endothelial cell profiles from GEO [5], and tumor subtype profiles from patient-derived xenografts (PDX) [32]. Only genes shared across all datasets were retained, resulting in a final set of 27,786 genes.
- For DNAm profiles, immune and fibroblast profiles were obtained from Decamps et al. [11], endothelial cell profiles from GEO [13], and tumor subtype profiles from the same

PDX study as the bulk RNA-seq reference [32]. All datasets were generated using the Illumina Infinium HumanMethylation450 BeadChip platform. CpG sites were restricted to the intersection across datasets (n = 416,830).

Table 5: Bulk reference datasets used to construct cell-type signatures for PDAC deconvolution.

Omic	Cell type	Samples (n)	Source	Description
RNA-seq	Endothelial	23	[5] (GSE135123)	Cultured endothelial cells (HMEC-1 or HUVEC) grown on soft or stiff matrices. One empty sample removed from the original 24 samples.
RNA-seq	Basal tumor	4	[32]	Patient-derived xenograft (PDX) samples (PDAC001T, PDAC004T, PDAC028T, PDAC024T). Extreme basal samples selected according to the PAMG gradient [31].
RNA-seq	Classical tumor	5	[32]	PDX samples (PDAC018T, PDAC011T, PDAC026T, PDAC006T, PDAC030T) selected according to the PAMG gradient.
RNA-seq	Immune	260	GTEx (v8)	"Spleen" samples from the GTEx portal (RNASeQCv1.1.9).
RNA-seq	Fibroblasts	527	GTEx (v8)	"Cells – Cultured fibroblasts" samples from the GTEx portal.
DNAm	Endothelial	6	[13] (GSE82234)	Cultured HUVEC donor samples.
DNAm	Basal tumor	4	[32]	PDX samples corresponding to the RNA-seq reference dataset.
DNAm	Classical tumor	5	[32]	PDX samples corresponding to the RNA-seq reference dataset.
DNAm	Fibroblasts/ Immune	8 / 3	[11]	Aggregated beta values obtained without batch correction.

scRNA-seq. The scRNA-seq reference integrated data from three studies: Peng et al., which includes immune, fibroblast, endothelial, and cancer cells [33]; Baron et al., which includes immune, fibroblast, and endothelial cells [4]; and Raghavan et al., which includes immune, endothelial, and cancer cells [35]. In the Peng and Raghavan datasets, cancer cells were further stratified into basal-like and classical subtypes using the `PurIST` classifier [36]. Briefly, tumor cells were normalized using `SCTransform`, subtype predictions were obtained using a pre-trained PurIST model, and cells with ambiguous ("lean") subtype assignments were excluded. For each cell type present in these datasets, 5,000 single-cell profiles were provided (Supplementary Table 6).

B.1.2 Benchmark datasets

Generation of the original *in vivo* dataset The *in vivo* multi-omics dataset comprises transcriptomic and DNA methylation (DNAm) profiles from 47 tumor samples, that were generated exclusively for this study. RNA and DNA were simultaneously extracted from formalin-fixed, paraffin-embedded (FFPE) samples using the Manual-Quick DNA/RNA FFPE Miniprep Kit (Zymo Research). Transcriptomic profiling was performed using a $3'$ RNA-seq protocol, and DNA methylation was assessed using the Infinium MethylationEPIC 850K array (Illumina). For this dataset, ground truth information is only partially available. Specifically, we used estimates of the relative proportions of classical and basal-like tumor cells as inferred by PACpAInt [37], a deep learning-based tool for automated histological subtype classification from H&E-stained slides .

$3'$ RNA-seq protocol. The protocol was applied as follows. Total RNA was fragmented, and the $3'$ end of mRNA molecules was captured using a poly(T) reverse transcription primer containing a Unique Molecular Identifier (UMI). During reverse transcription, Illumina adapters were incorporated via a template-switching mechanism. The resulting fragments were amplified through two rounds of

Table 6: Single-cell RNA-seq reference datasets used to derive cell-type profiles.

Study	Samples (n)	Data source	Cell-type mapping used in this study
[33]	35	PRJCA001063 (CRA001160)	*Mapping:* Endothelial → endothelial; B cell, T cell, Macrophage → immune; Stellate, Fibroblast → fibroblasts; Ductal cell type 2 → cancer cells. *Tumor subtypes:* basal/classical inferred using `PurIST` [36].
[35]	7	Broad Institute Single Cell Portal (SCP1644)	*Mapping:* Endothelial → endothelial; B_Cells, Macrophage, T_NK, T_Regs, DC, pDC_cell → immune; Tumor → cancer cells. *Note:* no fibroblasts available.
[4]	4	GSE84133	*Mapping:* endothelial → endothelial; macrophage, mast, t_cell → immune; activated_stellate, quiescent_stellate → fibroblasts. *Note:* no cancer cells available.

PCR to complete the adapter sequences and introduce sample-specific indices. The read structure consisted of paired-end 100 bp reads sequenced on an Illumina NovaSeq platform. Read 1 included a 26 bp UMI followed by the start of the poly(T) region. Read 2 began with a GGG motif (to be trimmed), followed by a variable-length $3'$ tag insert and a poly(A) tail. Image analysis and base calling were performed using Illumina Real-Time Analysis (RTA) software version 3.4.4 with default settings. The data were preprocessed as follows: Fastq file were aligned using STAR (2.7.1a) on UCSC hg38 genome, STAR –genomeDir STAR.index –readFilesCommand gunzip -c –readFilesIn input.rev –runThreadN threads –sjdbGTFfile params.gtf –outFilterMismatchNoverLmax 0.08 –outSAMtype BAM SortedByCoordinate –genomeLoad NoSharedMemory. BAM file where counted using featureCounts (v2.0.0) with options -T 15 -Q -t exon -g gene_name. Gene counts were normalized using standard DESeq2 procedure

DNAm protocol. The library and data were generated using standard Illumina Protocol(Infinium Methylation EPIC BeadChip). The raw DNA methylation intensity data files (IDAT) were processed with the lumi and methylumi R packages. We performed pre-normalization filtering (removing probes containing SNP, high intensity probes, not detected probes). We performed normalization using color balance adjustment and between sample normalisation by the "quantile" method.

*Data accessibility.*The gene expression data and the DNA methylation data have been deposited on GEO under accession codes GSE328792 and GSE329368.

In vitro dataset We used a previously published *in vitro* multi-omic dataset (GSE281305 DNA methylation MethEPIC 850K, GSE281204 RNA-seq) [1]. This dataset contains 30 samples of 9 pure cell types commonly found in pancreatic ductal adenocarcinoma (PDAC): classical-like tumor cells, basal-like tumor cells, cancer-associated fibroblasts, endothelial cells and immune cells (B cells, $CD4^+$ cells, $CD8^+$ cells, neutrophils and M2-macrophages). In order to reduce the number of less abundant cell types, all immune cell types were grouped in a single population termed Immune. Cells were mixed in known proportions that are coherent with proportions in human PDAC samples.

Simulated datasets We first simulated proportion matrices $A \in \mathbb{R}^{k \times n}$, distributed according to a Dirichlet distribution, with k the number of cell types and n the number of samples:

$$A_i \sim \text{Dir}(\alpha),$$

where A_i is the vector of cell types proportion for the sample i. The vector of parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$ is the Dirichlet parameter. We used two values for the α parameter in our simulations: the first one α_{real} is based on proportions found in the *in vitro* dataset, and the other one α_{rare} is such that the immune type is rare. The parameter value that has been used for the simulations is α_{real} unless otherwise specified.

Then, we multiplied the reference matrix of the matching pure cell types $X^{(m)} \in \mathbb{R}^{F_m \times k}$, with F_m the number of features for modalities $m \in \{\text{RNA}, \text{DNAm}\}$, by the proportions A; and we added a noise $\varepsilon^{(m)}$ to obtain realistic simulated bulk data:

$$Y^{(m)} = X^{(m)} \times A + \varepsilon^{(m)}$$

In each simulation procedures, we simulate $n = 60$ samples and designed several noises to simulate heterogenous datasets (see Table 1). Finally, A and $Y^{(m)}$ were divided in two for the train and test phases, with challenge participants having access to the train datasets and not to the test datasets on which they were evaluated.

Simulations SDN4, SDN5 and SDN6. The first strategy is to add an heteroscedastic χ^2 noise for RNA data and a Gaussian noise for DNAm data. This strategy has been used in three simulations: in the case of matching cell types (simulation SDN5) between the references used for the simulations and the references used for the deconvolution, one less (simulation SDN4) or one more (simulation SDN6) cell type in the simulation references compared to the deconvolution references. In the SDN4 simulation, we removed the basal-like tumor type in the simulation references and the corresponding proportion in A was transferred to the classical-like tumor type to keep the same proportion of tumor cells. In SDN6, the extra cell type was generated by randomly sampling, for each feature, the value of one of the existing cell types according to a uniform distribution. We set at 10% the proportion of this additional cell type, subtracting 2% to each of the original cell types.

Simulations SDE5 and SDEL. The second strategy takes into account patterns of dependence across genes, as gene regulatory networks induce a particular dependence structure. We inferred the dependence structure conditionally on the cell types references. For RNA data, we took the residuals $\varepsilon_{vitro}^{\text{RNA}}$ from the *in vitro* dataset:

$$\varepsilon_{vitro}^{\text{RNA}} = Y^{\text{RNA}} - X^{\text{RNA}} \times A_{vitro},$$

where A_{vitro} is the true proportions matrix of the *in vitro* dataset.

For DNAm data, we first reduced the number of features: all CpG probes in the promoter region of each gene are averaged into a gene-level measurement. We obtained reduced references with $m = 18735$ gene-level features.

We inferred the conditional co-expression network from the correlation matrix of the residuals $\varepsilon_{vitro}^{\text{(m)}}$; the correlation matrix of $\varepsilon_{vitro}^{\text{(m)}}$ is approximated with a low-rank factor decomposition based on an Expectation-Maximisation procedure [14].

For RNA data, noise $\varepsilon_{\text{EM}}^{\text{RNA}}$ is generated from this decomposition, centered and scaled. Each row is multiplied by the corresponding standard deviation from the residuals $\varepsilon_{vitro}^{\text{RNA}}$ to replicate the heteroscedasticity of each feature.

To obtain the noise at the probe level for DNAm data, we duplicated the noise of each gene by the number of matching probes while adding a small centered noise uniformly distributed. For intergenic probes, we obtained their noise by sampling uniformly the noise of one of the $m = 18735$ gene-level features. Then each row of $\varepsilon_{vitro}^{\text{DNAm}}$ is multiplied by the probe-specific standard deviation to replicate the heteroscedasticity of each feature.

Based on this strategy, we generated two datasets: dataset SDE5 with proportions simulated with the parameter α_{real}, and dataset SDEL with α_{rare}.

Simulation SDC5 We also inferred the conditional dependence structure with the empirical Copula on the scaled and centered residuals $\varepsilon_{vitro}^{\text{(m)}}$, using the R package `copula`. From the empirical copula, we generated a dependent noise, with respectively a negative binomial for RNA data and a Beta for DNAm data, for each feature: $\varepsilon_{\text{Copula}}^{\text{(m)}}$. Copulas were used for the simulation SDC5.

Pseudo-bulk generation from simulated single-cell data

To generate an additional benchmarking dataset independent from the convolution-based simulations described above, we simulated pseudo-bulk samples from synthetic single-cell RNA-seq and DNA methylation profiles.

1. Simulation of pseudo single-cell RNA-seq data.
   For each gene g and cell type c, RNA counts at the single-cell level were simulated using a negative binomial distribution:
   $$X_{gj}^{(c)} \sim \text{NB}(\mu_{gc}, \theta_{gc}),$$
   where $X_{gj}^{(c)}$ denotes the expression count of gene g in cell j from cell type c, μ_{gc} is the expected expression level, and θ_{gc} is the dispersion parameter.
   The mean expression μ_{gc} was derived from bulk RNA-seq reference profiles:
   $$\mu_{gc} = \frac{E_{gc}^{\text{bulk}}}{N_{\text{bulk}}},$$
   where E_{gc}^{bulk} denotes the bulk expression of gene g in cell type c and N_{bulk} is a scaling factor corresponding to the average number of transcripts per bulk sample.
   Dispersion parameters θ_{gc} were estimated from real single-cell RNA-seq data [33] by fitting a gene-wise negative binomial generalized linear model:
   $$X_{gi} \sim \text{NB}(\mu_{gc}, \theta_{gc}),$$
   using cells belonging to the corresponding cell type.
2. Simulation of pseudo single-cell DNA methylation data
   For DNA methylation data, probe-level methylation values were simulated using a discrete distribution over three possible states representing unmethylated, partially methylated, and fully methylated probes:
   $$M_{pj}^{(c)} \in \{0,\ 0.5,\ 1\},$$
   where $M_{pj}^{(c)}$ denotes the methylation value for probe p in cell j of cell type c.
   Let μ_{pc} be the mean methylation level observed in the bulk reference dataset for probe p and cell type c. Probabilities p_0, $p_{0.5}$ and p_1 were defined such that
   $$p_0 + p_{0.5} + p_1 = 1 \quad 0 \cdot p_0 + 0.5 \cdot p_{0.5} + 1 \cdot p_1 = \mu_{pc}.$$
   Pseudo single-cell methylation values were then sampled as
   $$M_{pj}^{(c)} \sim \text{Categorical}(p_0, p_{0.5}, p_1).$$
3. Simulation of mixture proportions
   For each pseudo-bulk sample i, cell-type proportions were generated using a Dirichlet distribution:
   $$A_i \sim \text{Dir}(\alpha_{\text{real}}),$$
   where $A_i = (a_{1i}, \ldots, a_{ki})$ represents the vector of proportions for the k cell types in sample i.
4. Generation of pseudo-bulk samples
   Pseudo-bulk expression profiles were obtained by aggregating simulated single cells according to the sampled proportions. For each sample i, a total of $K = 100$ cells were allocated across cell types according to the proportion vector $A_i = (a_{1i}, \ldots, a_{ki})$. The number of cells assigned to cell type c in sample i is:
   $$n_{ic} = \lfloor a_{ci} \cdot K \rfloor,$$

with a rounding correction applied to ensure $\sum_{c=1}^{k} n_{ic} = K$. The set S_{ic} of simulated cells assigned to cell type c in sample i was then obtained by sampling n_{ic} cells independently from the pool of simulated single cells of type c.

The bulk signal for feature f in sample i and modality $m \in \{\text{RNA}, \text{DNAm}\}$ was then computed by summing the modality-specific signals across all selected cells:

$$Y_{fi}^{(m)} = \sum_{c=1}^{k} \sum_{j \in S_{ic}} X_{fj}^{(m,c)},$$

where $X_{fj}^{(m,c)}$ denotes the value of feature f in simulated cell j of type c for modality m. This procedure produces synthetic bulk datasets whose statistical properties closely resemble those observed in real bulk RNA-seq and DNA methylation experiments while preserving realistic single-cell variability.

B.1.3 Feature space harmonization.

To ensure comparability across benchmark datasets and reference profiles, we restricted the feature space to the intersection of genes and CpG probes present in all datasets. Specifically, the set of authorized genes was defined as the intersection of genes available in the bulk RNA-seq reference and in all mixture datasets. The same procedure was applied to CpG probes for DNA methylation data. This yielded a common set of F_{RNA} genes and F_{DNAm} CpG authorized probes shared across all datasets and the reference.

To limit the size of the distributed datasets and facilitate the execution of computationally intensive methods, an additional restricted probe set was defined as the intersection of the authorized probes with the Illumina HumanMethylation27k annotation, yielding $F_{\text{DNAm}} = 23,724$ probes. This subset retains probes consistently covered across dataset while preserving biologically informative methylation variation. All deconvolution methods were evaluated on this harmonized feature space to ensure fair comparison.

B.2 Competition setup

Table 7: Overview of the competition phases.

Phase	Key characteristics
1 – Introduction	Single dataset: 5 cell types; matched RNA-seq and DNAm + baseline scripts
2 – Robustness	Multiple datasets: Noise, missing/extra cell types, pseudo-bulk, *in vitro/vivo* mixtures
3 – Generalization	Unseen data; no feedback; single final submission

B.2.1 Evaluation strategy

Individual metrics. Let $A \in \mathbb{R}^{k \times n}$ denote the ground-truth proportion matrix and $\hat{A} \in \mathbb{R}^{k \times n}$ the predicted proportion matrix, where k is the number of cell types and n the number of samples. Twelve metrics are computed, grouped into three families (Supplementary Table 8).

We measured the quality of the deconvolution based on various metrics. We used error metrics such as the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the Aitchison distance and the Jensen-Shannon divergence (JSD) between the predicted and the real proportions matrix. We looked also at the angle between predicted and real samples and derived two metrics from this angle: the Angular Inequality/Disproportionality (AID) and the Sine-Diagonal ID (SDID). Finally, we included Pearson's and Spearman's correlations on the whole vectorized matrix (global), on the rows (cell types) or on the columns (samples). For the columns' and the rows' correlations, we did the arithmetic mean along the respective axis. In the case of an extra cell type in the reference matrix used for the deconvolution compared to the ground truth, we added a row of zeroes in the ground truth matrix to match the number of cell types in the predicted matrix. In the case of a missing cell type in the reference, we kept in the ground truth matrix only the cell types that have been estimated.

Table 8: Summary of the twelve evaluation metrics. Dir. indicated directionality of best score.

Family	Metric	Dir.	Weight	Formula
Cohort Correlation	Pearson (global)	↑	1/16	$\rho_P(A, \hat{A})$
	Spearman (global)	↑	1/16	$\rho_S(A, \hat{A})$
	Pearson (sample)	↑	1/16	$\frac{1}{n}\sum_i \rho_P(A_{\cdot i}, \hat{A}_{\cdot i})$
	Spearman (sample)	↑	1/16	$\frac{1}{n}\sum_i \rho_S(A_{\cdot i}, \hat{A}_{\cdot i})$
Cell-type correlation	Pearson (cell type)	↑	1/8	$\frac{1}{k}\sum_c \rho_P(A_{c\cdot}, \hat{A}_{c\cdot})$
	Spearman (cell type)	↑	1/8	$\frac{1}{k}\sum_c \rho_S(A_{c\cdot}, \hat{A}_{c\cdot})$
Error	RMSE	↓	1/16	$\sqrt{\frac{1}{kn}\sum_{c,i}(A_{ci} - \hat{A}_{ci})^2}$
	MAE	↓	1/16	$\frac{1}{kn}\sum_{c,i} \lvert A_{ci} - \hat{A}_{ci}\rvert$
	Aitchison	↓	1/16	$\frac{1}{n}\sum_i d_A(A_{\cdot i}, \hat{A}_{\cdot i})$, values clipped at 10^{-9}
	JSD	↓	1/16	$\frac{1}{n}\sum_i \mathrm{JSD}(A_{\cdot i} \Vert \hat{A}_{\cdot i})$
Geometric	AID	↓	1/8	$\frac{1}{n}\sum_i \frac{90}{\pi/2} \arccos\left(\frac{A_{\cdot i} \cdot \hat{A}_{\cdot i}}{\lVert A_{\cdot i}\rVert_2 \lVert \hat{A}_{\cdot i}\rVert_2}\right)$
	SDID	↓	1/8	$\frac{1}{n}\sum_i \sqrt{\sin\left(\arccos\left(\frac{A_{\cdot i} \cdot \hat{A}_{\cdot i}}{\lVert A_{\cdot i}\rVert_2 \lVert \hat{A}_{\cdot i}\rVert_2}\right)\right)}$

Correlation between metrics are presented in Supplementary Figure 6

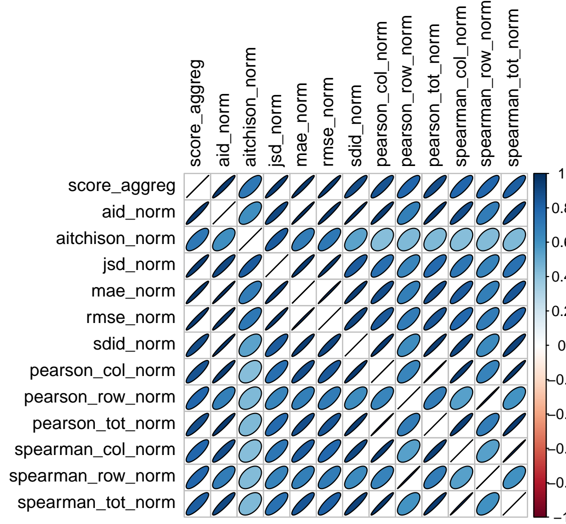


Figure 6: Correlations observed across the 12 metrics included in the benchmark after center-scaling transformation for all deconvolution methods evaluated, as well as the aggregate score.

Normalization. Each metric m is normalized to $[0, 1]$ using a linear shift between its worst-case and best-case values:

$$\tilde{m} = \frac{m - m_{\text{worst}}}{m_{\text{best}} - m_{\text{worst}}},$$

where m_{best} and m_{worst} are computed from an ideal prediction ($\hat{A} = A$) and a deliberately bad prediction that assigns all probability mass to the least abundant cell type, respectively. All normalized metrics are oriented so that $\tilde{m} = 1$ is the best possible score. For the correlations metrics, the best score is 1 and the worst is -1. For the angle-based metrics, the best and worst scores are defined by the best and worst angles: the best angle is 0 and the worst is $\frac{\pi}{2}$ since we have compositional data. For the error metrics, the best score is 0 and the worst has been computed based on the worst possible prediction. The worst prediction is a prediction with a proportion of 1 for the lowest abundant cell

type and zeroes everywhere else. After the center-scaling procedure, we transformed the error and angle-based metrics by subtracting them from 1, such that the best score is 1 and the worst is 0 for all metrics.

Aggregate score. The twelve normalized metrics are combined into a single aggregate score via a weighted geometric mean:

$$s = \prod_m \tilde{m}^{w_m}, \quad \sum_m w_m = 1,$$

with the following weight scheme: cell-type correlations metrics (2 total) account for one fourth of the score, other correlation metrics (4 total) for one fourth error metrics (4 total) for fourth, and geometric metrics (2 total) for one fourth. Within each group, weights are equal.

Special case: partial ground truth. For the *in vivo* dataset (VIVO), ground truth is only available for two cell types (basal-like and classical). In this case, only cell-type-wise Pearson and Spearman correlations are computed, each weighted equally at $w = 1/2$.

B.2.2 Starting kit and baselines

Table 9: Baseline scripts provided in the starter kit.

Script	Description
`submission_script.R`	NNLS deconvolution on RNA-seq data.
`_nnlsmultimodal.R`	NNLS on each modality independently, with late averaging.
`_nnlsmultimodalSource.R`	Demonstrates loading external R scripts or `.rds` files.
`_installpkgcran.R`	Demonstrates CRAN package installation within a submission.
`Submission_script.py`	Python equivalent of the NNLS baseline.

B.2.3 Description of the winning method JOKER

The proposed method follows a omic-specific preprocessing strategy, combined with independent deconvolution and late integration of estimates derived from RNA-seq and DNAm data.

RNA-seq preprocessing. Raw RNA-seq counts were first normalized for sequencing depth using counts per million (CPM), obtained by dividing each column by its total counts and scaling by 10^6. To reduce noise and focus on informative features, the 5,000 most variable genes were selected based on their variance after applying a variance-stabilizing transformation defined as $\log_2(x + c)$, where c is the median of non-zero CPM values. To ensure comparability between mixture and reference profiles, gene expression values were further scaled by the mean expression across both datasets. Specifically, for each gene g, values were divided by $\mu_g = \text{mean}(X_g^{\text{mix}}) + \text{mean}(X_g^{\text{ref}}) + \epsilon$, where ϵ is a small constant to avoid division by zero.

DNA methylation preprocessing. For DNAm data, CpG sites were filtered based on their variability across the reference profiles. Only sites with variance greater than 0.1 were retained, resulting in a subset of informative features used for deconvolution.

Deconvolution model. For each modality, cell-type proportions were estimated independently using a non-negative least squares (NNLS) model. For a given sample i, the bulk profile Y_i is modeled as a linear combination of reference profiles X weighted by a vector of proportions p_i:

$$\hat{p}_i = \arg\min_{p_i \geq 0} \|Y_i - X p_i\|_2^2,$$

followed by normalization to enforce the sum-to-one constraint:

$$\tilde{p}_i = \frac{\hat{p}_i}{\sum_{c=1}^{k} \hat{p}_{ic}}.$$

Late integration of modalities. Final cell-type proportions were obtained using a late integration strategy. For most cell types, proportions estimated from RNA-seq and DNAm were averaged:

$$p_i^{\text{final}} = \frac{1}{2}\left(p_i^{\text{RNA}} + p_i^{\text{DNAm}}\right).$$

However, for basal-like and classical tumor cell types, RNA-seq-based estimates were deemed less reliable and were therefore replaced by DNAm-based estimates. The resulting proportions were finally renormalized to ensure that they sum to one for each sample.

B.3 Benchmark

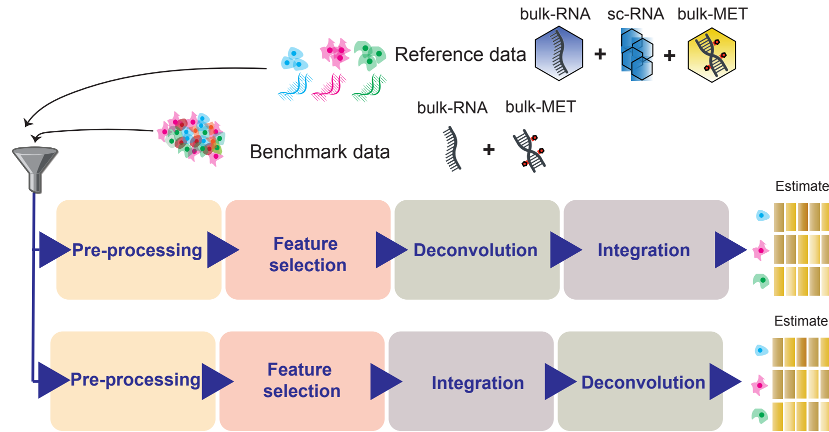


Figure 7: Overview of the modular benchmark framework. The competition submissions were decomposed into four main modules: preprocessing, feature selection, deconvolution, and integration (early or late). All compatible combinations of module methods were systematically evaluated.

B.3.1 Preprocessing

Three types of preprocessing were tested (Supplementary Table 10. Preprocessing has no measurable impact on late integration performance (Supplementary Figure 8), which is expected since each modality is processed independently before the proportions are combined. For early integration, however, RNA-seq scaling or log-normalization globally improves median aggregate scores, suggesting that reducing the magnitude gap between RNA-seq counts and DNAm beta values is beneficial prior to joint feature-level integration. By contrast, preprocessing of DNAm data has negligible impact in both integration regimes, likely because beta values are already bounded in $[0, 1]$ and exhibit less dynamic range variability than RNA-seq counts.

B.3.2 Feature selection

Feature selection consists in identifying a subset of genes or CpG probes that are most informative for distinguishing cell types, prior to deconvolution. We evaluated diverse strategies for both RNA-seq and DNAm data, described in Supplementary Tables 11 and 12.

Overall, feature selection yields only marginal improvements in median aggregate score compared to the identity baseline (Supplementary Figure 9). However, its impact is strongly modulated by the integration strategy. First, certain feature selection methods are particularly detrimental in combination with specific early integration approaches, as evidenced by the bimodal distributions observed in the violin plots. Second, the optimal feature selection strategy in median differs between integration regimes. For early integration, the best results are obtained by combining cell-type-specific DNAm probes selected by sparse PLS-DA (`splsda`) with the top 1,000 cell-type-specific RNA-seq genes (`Toastbulknbfs`). For late integration, maximally discriminant non-overlapping DNAm probes (`maxdiscriminant`) combined with scRNA-seq-based pseudo-bulk references (`scpseudobulk`) yield the best results. Notably, feature selection strategies that perform well under early integration

Table 10: Comparison of preprocessing methods, applied to both RNA-seq and DNA methylation data. All methods transform the mixture and reference profiles prior to feature selection and deconvolution.

Method	Principle	Key idea
`ppID`	Identity (no preprocessing)	Returns the data unchanged. Serves as a baseline to assess the benefit of preprocessing. Applied identically to RNA-seq and DNAm data.
`Scale`	Column-sum normalization	Normalizes each sample by its total count, so that all samples sum to 1: $$\tilde{X}_{gi} = \frac{X_{gi}}{\sum_{g'} X_{g'i}}.$$ Makes samples comparable regardless of sequencing depth or global methylation level differences. Applied identically to RNA-seq and DNAm data.
`LogNorm`	Log-normalization	Applies Seurat's `LogNormalize` function, which scales each sample by its total count, multiplies by a scale factor of 10^4, and applies a log transformation: $$\tilde{X}_{gi} = \log\left(1 + \frac{X_{gi}}{\sum_{g'} X_{g'i}} \times 10^4\right).$$ For bulk RNA-seq and DNAm data, an exponential transformation is applied post hoc to recover a linear scale: $\hat{X}_{gi} = \exp(\tilde{X}_{gi})$, which is equivalent to a 10^4-scaled column-sum normalization without log transformation. For scRNA-seq data, the log-transformed values are retained directly.

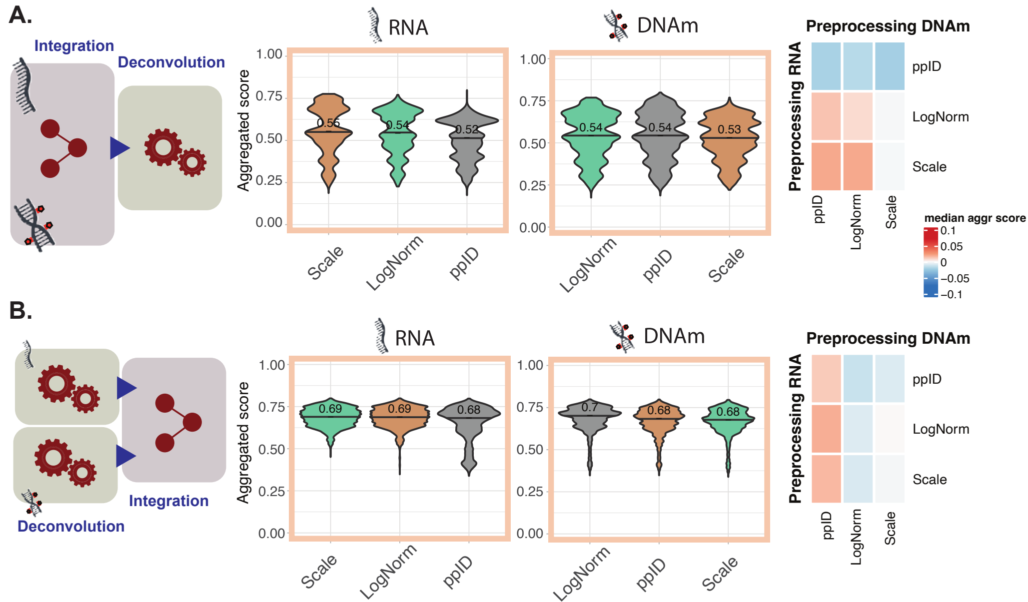


Figure 8: Aggregate score distribution across all combinations for pre-processing methods on bulk RNA-seq (left) and DNA methylation (right) used for early (panel A) and late (panel B) integration. Pairwise interaction effects between pre-processing methods for RNA and DNAm are shown at rightmost of each, according to the integration strategy. Rows correspond to pre-processing methods for RNA and columns correspond to pre-processing methods for DNAm.

tend to underperform under late integration and vice versa, suggesting that the optimal feature set is not independent of the integration paradigm.

Table 11: Comparison of RNA-seq feature selection methods. All methods select a subset of genes from the mixture and reference profiles prior to deconvolution.

Method	Principle	Key idea
`fsID`	Identity (no selection)	Returns the data unchanged. All available genes are passed to the deconvolution module. Serves as a baseline to assess the benefit of feature selection.
`Toastbulknbfs`	Marker gene selection from bulk reference	Applies `TOAST::findRefinx` [22] to the bulk RNA-seq reference matrix to select the top $n = 1{,}000$ marker genes. Marker genes are defined as genes with high cell-type specificity in the reference profiles. Both mixture and reference are then restricted to this gene set.
`Toastvst`	Marker gene selection from VST-transformed bulk reference	Extends `Toastbulknbfs` by applying a variance-stabilizing transformation (VST) to the bulk reference prior to marker selection. `TOAST::findRefinx` is then applied to select the top $n = 1{,}000$ marker genes. The VST reduces the influence of highly expressed genes on marker selection.
`SCcluster`	Differential expression markers from clustered scRNA-seq	Uses Seurat's `FindAllMarkers` (Wilcoxon rank-sum test) on clustered scRNA-seq data to identify cell-type-specific marker genes. Genes are retained if they satisfy: adjusted p-value $<$ 0.05, fraction expressed in the target cluster > 0.6, and fraction expressed in other clusters < 0.3. This combines statistical significance with specificity criteria.
`scpseudobulk`	pseudo-bulk references using scRNA-seq with a cell-type specific gene set	For each pair of cell types (c, c') and each scRNA-seq dataset d, computes a t-statistic for each gene g. The top $n_{\text{top}} = 20$ genes with highest and lowest t-statistics are retained per pair, across all datasets. The final gene set is the union of genes consistently selected across all pairwise comparisons. A pseudo-bulk reference is then constructed by averaging single-cell profiles per cell type over the selected genes.

Table 12: Comparison of DNA methylation feature selection methods. All methods select a subset of CpG probes from the mixture and reference profiles prior to deconvolution.

Method	Principle	Key idea
`fsID`	Identity (no selection)	Returns the data unchanged. All available CpG probes are passed to the deconvolution module. Serves as a baseline to assess the benefit of feature selection.
`Toastnbfs`	Full probe ranking via TOAST	Applies `TOAST::findRefinx` [22] to the DNAm reference matrix with $n_{\text{marker}} = p$ (all probes), returning all probes ranked by cell-type specificity. Equivalent to `fsID` in terms of probe set, but reorders probes by discriminative power.
`Toastpercent`	Partial probe selection via TOAST	Applies `TOAST::findRefinx` to the DNAm reference matrix retaining the top $\lfloor 0.8 \times p \rfloor$ probes, where p is the total number of probes. Removes the least cell-type-specific 20% of probes.
`mostmethylated`	Biologically informed probe selection	Retains probes with high methylation levels based on biological prior knowledge that cancer cells exhibit high methylation. For each cell type c, probes above the 75th percentile of methylation are selected.
`maxdiscriminant`	Maximally discriminant non-overlapping probes	For each cell type c, probes are ranked by their absolute deviation from the global mean. The largest n such that the top-n probe sets across cell types are mutually non-overlapping is determined iteratively (up to $n_{\max} = 100$ per cell type). This ensures maximal cell-type specificity with no redundancy across types.
`splsda`	Sparse PLS-DA on logit-transformed reference	First applies `TOAST::findRefinx` to retain the top $10{,}000$ most variable probes. A logit transformation is then applied to stabilize beta-valued methylation data. Sparse partial least squares discriminant analysis (sPLS-DA) [24] is then fitted on reference data with cell type as the response, retaining $n = 1{,}000$ probes per component over 2 components. The final probe set is the union of selected variables from both components.

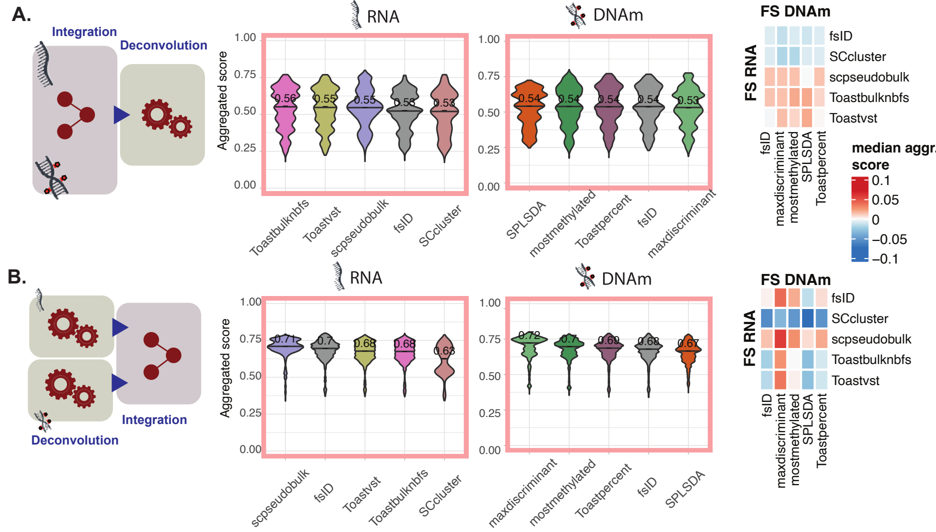


Figure 9: Aggregate score distribution across all combinations for feature selection methods on bulk RNA-seq (left) and DNA methylation (right) used for early (panel A) and late (panel B) integration. Pairwise interaction effects between feature selection methods for RNA and DNAm are shown at rightmost of each, according to the integration strategy. Rows correspond to feature selection methods for RNA and columns correspond to feature selection methods for DNAm.

B.3.3 Deconvolution

All deconvolution methods solve a variant of the following problem: given a mixture profile $Y_i \in \mathbb{R}^p$ and a reference matrix $X \in \mathbb{R}^{p \times k}$, estimate a proportion vector $p_i \in \mathbb{R}^k$ such that:

$$Y_i \approx X p_i, \quad p_i \geq 0, \quad \sum_{c=1}^{k} p_{ic} = 1.$$

Methods differ in how they handle noise, outliers, and reference misspecification (Supplementary Table 13). The methods evaluated here span a spectrum from standard least squares (`lm`, `nnls`) to robust regression (`RLR`, `RLRpoisson`) and ensemble approaches (`RLRnnls`), differing primarily in how they handle noise, outlier features, and the simplex constraint. Deconvolution algorithms tested are described in Supplementary Table 13. Consistent with previous benchmarks [1], RLR-based methods achieve the highest deconvolution performances (Supplementary Figure 10), suggesting that down-weighting outlier features via iteratively reweighted least squares improves deconvolution outcomes. A variant of RLR incorporating Poisson-inspired feature weights (`RLRpoisson`) further improves performance on RNA-seq data, consistent with the approximately negative binomial distribution of read counts, where variance scales with the mean. Notably, the advantage of RLR-based algorithms is attenuated when combined with normalized concatenation early integration (`concatscale`), suggesting that this preprocessing step may reduce sensitivity to the choice of deconvolution algorithm. The ensemble approach `RLRnnls`, which selects between RLR and NNLS based on global reconstruction RMSE, does not consistently improve over RLR alone, possibly because the global RMSE criterion does not reflect sample-level accuracy. As expected, `epic` performs poorly on DNAm data: its internal TPM normalization is designed for RNA-seq count data and is incompatible with beta-valued methylation profiles bounded in $[0, 1]$. Finally, `nnlslargeref`, which iteratively removes one reference cell type and retains the truncated model if it yields lower reconstruction error, does not improve overall median performance. This strategy may however be beneficial in settings where the reference contains cell types absent from the mixtures, leading to reference over-specification.

Table 13: Comparison of deconvolution methods. All methods estimate cell-type proportions p_i from a mixture profile Y_i and a reference matrix X.

Method	Principle	Key idea
`lm`	Ordinary least squares	Minimizes $\|Y_i - Xp_i\|_2^2$ without intercept; negative coefficients are set to zero and proportions are renormalized
`nnls`	Non-negative least squares	Minimizes $\|Y_i - Xp_i\|_2^2$ subject to $p_i \geq 0$, enforcing non-negativity by construction (Lawson-Hanson algorithm) [27], non-negativity by construction; proportions are renormalized post hoc to satisfy $\mathbf{1}^\top p_i = 1$.
`nnlslargeref`	NNLS with reference truncation	Extends NNLS by iteratively removing one reference cell type at a time and retaining the truncated model if it yields lower reconstruction RMSE; handles potential over-specification of the reference.
`epic`	Constrained least squares with internal normalization	Solves a simplex-constrained least squares problem [34]: $$\hat{p}_i = \arg \min_{\substack{p \geq 0 \\ \mathbf{1}^\top p = 1}} \left\| \bar{Y}_i - \bar{X}p \right\|_2^2,$$ where $\bar{Y}_i$ and $\bar{X}$ denote internally TPM-normalized versions of the mixture and reference profiles. The simplex constraint $\mathbf{1}^\top p = 1$ is enforced during optimization, unlike NNLS where it is applied post hoc. No uncharacterized cell population is included (`withOtherCells=F`). In the original EPIC formulation, gene-specific weights $w_g \propto 1/V_{gc}$ down-weight genes with high variability across reference replicates. However, as our reference consists of a single meta-profile per cell type (no replicates available), all weights are equal ($w_g = 1$), reducing the weighted objective to a standard least squares criterion.
`RLR`	Robust linear regression	Applies robust regression via the RPC algorithm implemented in EpiDISH [40]; down-weights outlier features via iteratively reweighted least squares (IRLS).
`RLRpoisson`	Robust linear regression with Poisson weights	Extends RLR by weighting features inversely proportional to their mean reference expression, mimicking Poisson variance stabilization; $w_g \propto 1/\bar{x}_g$. Weighting scheme assumes Poisson-like mean-variance relationship, which may not hold for all feature types (e.g., DNA methylation beta values).
`RLRnnls`	Ensemble: RLR and NNLS	Runs both RLR and NNLS independently and selects the estimate with lower reconstruction RMSE after column-sum normalization. This strategy aims to combine the robustness of RLR with the non-negativity guarantees of NNLS, defaulting to NNLS when RLR fails.

B.3.4 Early integration methods

Raw feature concatenation (concatnoscale). In the case of raw concatenation, the integration operator $\mathcal{F}_{\text{concat}}$ is defined as a simple feature-level stacking of both omics without any transformation:

$$\mathcal{F}_{\text{concat}}: \quad \begin{cases} \tilde{Y} = \begin{bmatrix} Y^{\text{RNA}} \\ Y^{\text{DNAm}} \end{bmatrix}, \\ \tilde{X} = \begin{bmatrix} X^{\text{RNA}} \\ X^{\text{DNAm}} \end{bmatrix}, \end{cases} \qquad \tilde{Y}, \tilde{X} \in \mathbb{R}^{(p_{\text{RNA}} + p_{\text{DNAm}}) \times n},$$

where $n = n_{\text{mix}}$ or $n = k$ depending on whether mixture or reference samples are considered, and $p_{\text{RNA}}, p_{\text{DNAm}}$ denote the number of features in each modality. No normalization, scaling, or latent transformation is applied beyond the concatenation itself. The resulting representations $\tilde{Y}$ and $\tilde{X}$ are directly used as mixture and reference inputs for downstream deconvolution.

Normalized concatenation (concatscale). In this case, the integration operator $\mathcal{F}_{\text{scale}}$ performs feature-level concatenation of RNA-seq and DNAm data, followed by sample-wise normalization via a Gaussian CDF mapping. The concatenation step stacks both modalities along the feature dimension:

$$\tilde{Y} = \begin{bmatrix} Y^{\text{RNA}} \\ Y^{\text{DNAm}} \end{bmatrix}, \quad \tilde{X} = \begin{bmatrix} X^{\text{RNA}} \\ X^{\text{DNAm}} \end{bmatrix}, \quad \tilde{Y}, \tilde{X} \in \mathbb{R}^{(p_{\text{RNA}} + p_{\text{DNAm}}) \times n},$$

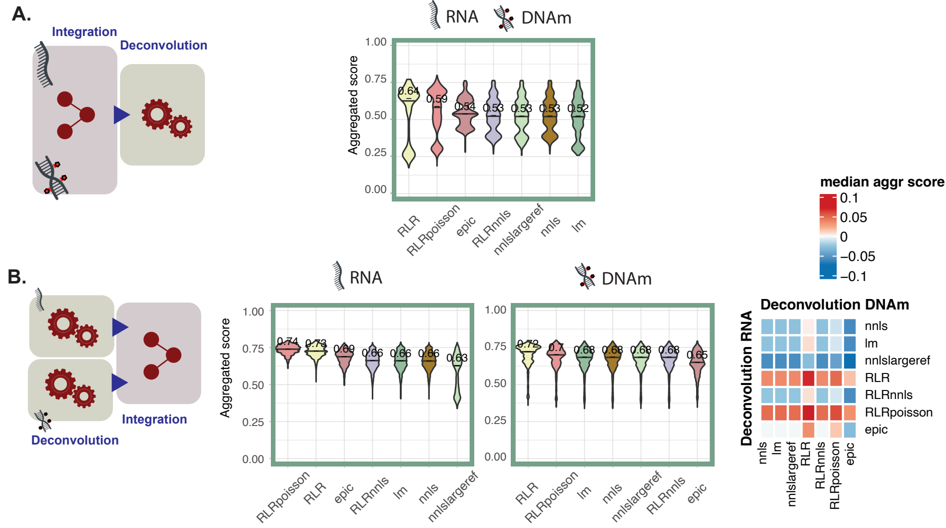


Figure 10: Aggregate score distribution across all combinations for deconvolution methods used for early integration (panel A) and on bulk RNA-seq (left) and DNA methylation (right) used for late integration (panel B). Pairwise interaction effects between deconvolution methods for RNA and DNAm in late integration strategie are shown at rightmost of panel B. Rows correspond to deconvolution methods for RNA and columns correspond to deconvolution methods for DNAm.

where $n = n_{\text{mix}}$ or $n = k$ depending on whether mixture or reference samples are considered, and $p_{\text{RNA}}, p_{\text{DNAm}}$ denote the number of features in each modality. Each sample i is then independently normalized across its features and mapped to a Gaussian CDF scale: for each feature j of sample i,

$$\mathcal{F}_{\text{scale}}(\tilde{y}_{ij}) = \Phi\left(\frac{\tilde{y}_{ij} - \mu_i}{\sigma_i}\right),$$

where μ_i and σ_i denote the empirical mean and standard deviation of sample i computed across all features, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. The same sample-wise transformation is applied to $\tilde{X}$, using the mean and standard deviation of each sample computed across its own features. The resulting representations $\tilde{Y}$ and $\tilde{X}$ are directly used as mixture and reference inputs for downstream deconvolution.

Latent linear embedding (omicade4bulk). In this case, the integration operator $\mathcal{F}_{\text{mCIA}}$ constructs a joint latent representation of RNA-seq and DNAm data using Multiple Co-Inertia Analysis (MCIA), implemented via the **omicade4** package [26]. This method performs a feature-to-latent space transformation by seeking a consensus synthetic variable that maximizes the co-inertia between modality-specific projections. First, mixture and reference samples are concatenated within each modality:

$$W^{\text{RNA}} = \left[Y^{\text{RNA}}, X^{\text{RNA}}\right], \quad W^{\text{DNAm}} = \left[Y^{\text{DNAm}}, X^{\text{DNAm}}\right] \in \mathbb{R}^{p \times (n_{\text{mix}}+k)},$$

where columns correspond to samples, p denotes the number of features, and k denotes the number of reference cell types, with one reference sample per cell type (i.e. $n_{\text{ref}} = k$). MCIA finds a d-dimensional synthetic variable $Z_s \in \mathbb{R}^{d \times (n_{\text{mix}}+k)}$ and modality-specific projection matrices $A^{\text{RNA}}, A^{\text{DNAm}} \in \mathbb{R}^{p \times d}$ by solving:

$$\max_{A^{\text{RNA}},\, A^{\text{DNAm}},\, Z_s} \sum_{m \in \{\text{RNA}, \text{DNAm}\}} \text{cov}^2\left({A^{(m)}}^\top W^{(m)},\, Z_s\right),$$

where the maximization is performed iteratively over d orthogonal components. In practice, $d = 10$ components are computed and retained, a value fixed empirically and not tuned per dataset. The latent coordinates are shifted to ensure non-negativity ($Z \geq 0$ coordinate-wise):

$$Z = Z_s - \min_{i,j}(Z_s)_{ij}, \quad Z \in \mathbb{R}^{d \times (n_{\text{mix}}+k)},$$

where $\mathcal{F}_{\text{mCIA}}$ returns the synthetic variable Z_s. The latent representation is then partitioned into mixture and reference components:

$$Z = [Z_{\text{mix}}, Z_{\text{ref}}], \quad Z_{\text{mix}} \in \mathbb{R}^{d \times n_{\text{mix}}}, \; Z_{\text{ref}} \in \mathbb{R}^{d \times k}.$$

Deconvolution is performed sample-wise in the aligned latent space: for each mixture sample $i \in \{1, \ldots, n_{\text{mix}}\}$,

$$z_{\text{mix},i} \approx Z_{\text{ref}}\, p_i, \quad p_i \in \mathbb{R}^k, \quad p_i \geq 0, \quad \sum_{c=1}^{k} p_{ic} = 1,$$

where $z_{\text{mix},i}$ is the i-th column of Z_{mix} and p_i is the vector of cell-type proportions for sample i.

Non-linear kernel embedding (Kernel). In this case, the integration operator $\mathcal{F}_{\text{kernel}}$ constructs a non-linear joint representation of RNA-seq and DNAm data using kernel-based feature mapping, implemented via the **mixKernel** package [25]. First, mixture and reference samples are concatenated within each modality:

$$W^{\text{RNA}} = \left[Y^{\text{RNA}}, X^{\text{RNA}}\right], \quad W^{\text{DNAm}} = \left[Y^{\text{DNAm}}, X^{\text{DNAm}}\right] \in \mathbb{R}^{p \times (n_{\text{mix}}+k)},$$

where columns correspond to samples, p denotes the number of features, and k denotes the number of reference cell types, with one reference sample per cell type (i.e. $n_{\text{ref}} = k$). An abundance kernel is then computed independently for each modality:

$$K_{ij}^{\text{RNA}} = \kappa\big(w_i^{\text{RNA}}, w_j^{\text{RNA}}\big), \quad K_{ij}^{\text{DNAm}} = \kappa\big(w_i^{\text{DNAm}}, w_j^{\text{DNAm}}\big), \quad K^{\text{RNA}}, K^{\text{DNAm}} \in \mathbb{R}^{(n_{\text{mix}}+k) \times (n_{\text{mix}}+k)},$$

where $w_i^{(\cdot)}$ denotes the i-th column of the corresponding matrix and $\kappa(\cdot,\cdot)$ denotes the abundance kernel function. The modality-specific kernels are then combined into a single multi-omic kernel via STATIS-UMKL optimal weighting [25]:

$$K = \mathcal{C}\left(K^{\text{RNA}}, K^{\text{DNAm}}\right), \quad K \in \mathbb{R}^{(n_{\text{mix}}+k) \times (n_{\text{mix}}+k)},$$

where $\mathcal{C}(\cdot)$ denotes the STATIS-UMKL kernel aggregation operator. A low-dimensional latent representation is then obtained via kernel PCA:

$$Z = \mathcal{F}_{\text{KPCA}}(K) - \min_{i,j}\left[\mathcal{F}_{\text{KPCA}}(K)\right]_{ij}, \quad Z \in \mathbb{R}^{d \times (n_{\text{mix}}+k)},$$

where d is the number of retained components and the shift ensures $Z \geq 0$ coordinate-wise. The latent representation is finally partitioned into mixture and reference components:

$$Z = [Z_{\text{mix}}, Z_{\text{ref}}], \quad Z_{\text{mix}} \in \mathbb{R}^{d \times n_{\text{mix}}}, \; Z_{\text{ref}} \in \mathbb{R}^{d \times k}.$$

Deconvolution is performed sample-wise in the aligned latent space: for each mixture sample $i \in \{1, \ldots, n_{\text{mix}}\}$,

$$z_{\text{mix},i} \approx Z_{\text{ref}}\, p_i, \quad p_i \in \mathbb{R}^k, \quad p_i \geq 0, \quad \sum_{c=1}^{k} p_{ic} = 1,$$

where $z_{\text{mix},i}$ is the i-th column of Z_{mix} and p_i is the vector of cell-type proportions for sample i.

Optimal transport-based representation (OT). In this case, the integration operator $\mathcal{F}_{\text{OT}}$ aligns RNA-seq and DNAm profiles using uniPort [8], a variational autoencoder (VAE) that enforces cross-modal alignment via entropic optimal transport in the latent space. We consider concatenated RNA and DNAm datasets:

$$W^{\text{RNA}} = \left[Y^{\text{RNA}}, X^{\text{RNA}}\right], \quad W^{\text{DNAm}} = \left[Y^{\text{DNAm}}, X^{\text{DNAm}}\right] \in \mathbb{R}^{p \times (n_{\text{mix}}+k)},$$

where columns correspond to samples, p denotes the number of features, and k denotes the number of reference cell types, with one reference sample per cell type (i.e. $n_{\text{ref}} = k$).

Each column w_i^{RNA} of W^{RNA} (resp. w_j^{DNAm} of W^{DNAm}) is encoded by the VAE encoder q_ϕ into a latent vector:

$$h_i^{\text{RNA}} = q_\phi(w_i^{\text{RNA}}), \quad h_j^{\text{DNAm}} = q_\phi(w_j^{\text{DNAm}}).$$

Let μ_{RNA} and μ_{DNAm} denote the empirical distributions induced by $\{h_i^{\text{RNA}}\}$ and $\{h_j^{\text{DNAm}}\}$. The entropic optimal transport plan is computed in this latent space:

$$\pi^\star = \arg \min_{\pi \in \Pi(\mu_{\text{RNA}}, \mu_{\text{DNAm}})} \sum_{i,j} \pi_{ij}\, d\big(h_i^{\text{RNA}}, h_j^{\text{DNAm}}\big) + \varepsilon \sum_{i,j} \pi_{ij} \log \pi_{ij}.$$

The regularization parameter $\varepsilon = 0.5$ is set to the default value of uniPort [8], and the model is trained for 500 iterations with a batch size adapted to the number of available samples (batch_size $= \min(50, n_{\text{mix}}, n_{\text{ref}})$). The transport plan $\pi^\star$ is then used to realign the encoder outputs across modalities, yielding a joint latent representation:

$$Z = \mathcal{F}_{\text{OT}}\left(W^{\text{RNA}}, W^{\text{DNAm}}\right), \quad Z \in \mathbb{R}^{d \times (n_{\text{mix}}+k)},$$

where Z aggregates the OT-aligned encodings of all samples from both modalities.

The latent representation is then partitioned into mixture and reference components:

$$Z = [Z_{\text{mix}}, Z_{\text{ref}}], \quad Z_{\text{mix}} \in \mathbb{R}^{d \times n_{\text{mix}}},\ Z_{\text{ref}} \in \mathbb{R}^{d \times k},$$

where k denotes the number of reference cell types. Deconvolution is performed sample-wise in the aligned latent space: for each mixture sample $i \in \{1, \ldots, n_{\text{mix}}\}$,

$$z_{\text{mix},i} \approx Z_{\text{ref}}\, p_i, \quad p_i \in \mathbb{R}^k, \quad p_i \geq 0, \quad \sum_{c=1}^{k} p_{ic} = 1,$$

where $z_{\text{mix},i}$ is the i-th column of Z_{mix} and p_i is the vector of cell-type proportions for sample i.

B.3.5 Detailed results for early integration

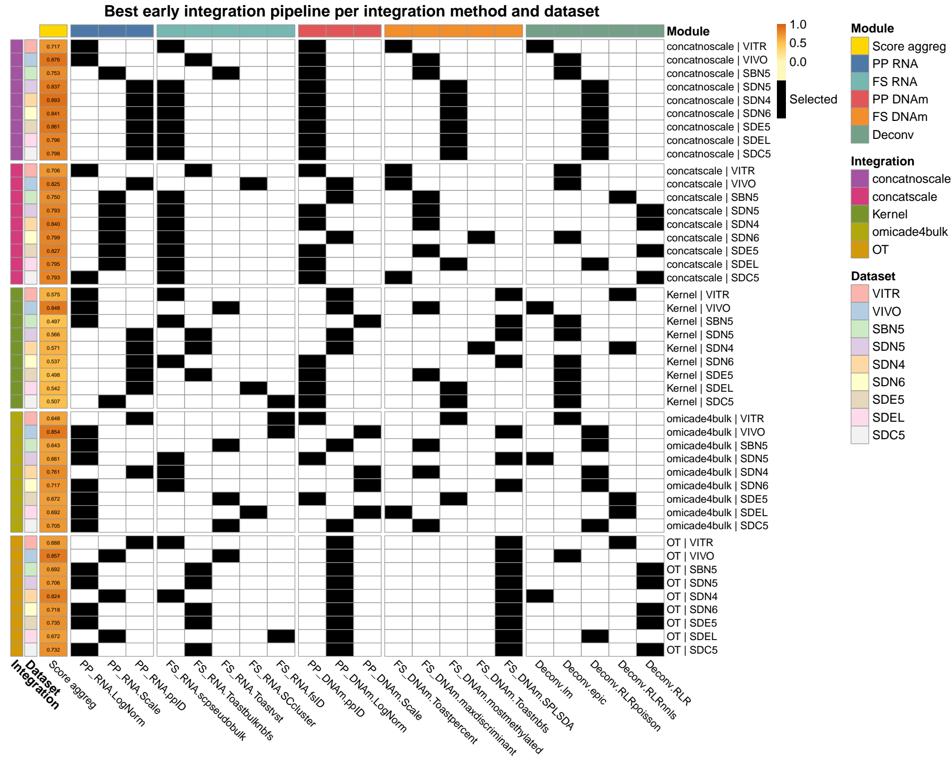


Figure 11: Best early integration pipeline combination for each integration method and dataset. Each row corresponds to one integration method and dataset pair. The first column reports the best aggregate score achieved (yellow-to-orange gradient, range $[0, 1]$). Remaining columns indicate the selected method in each pipeline module (black: selected; white: absent), grouped by module type (color-coded column annotations): preprocessing (PP RNA, PP DNAm), feature selection (FS RNA, FS DNAm), and deconvolution (Deconv). Horizontal gaps separate integration methods; vertical gaps separate modules.

B.3.6 Late integration methods

DNAm-only aggregation (onlyMet). The final estimate is obtained by selecting the DNAm-based proportions only:

$$\hat{p}_i = \hat{p}_i^{\text{DNAm}}.$$

RNA-only aggregation (onlyRna). The final estimate is obtained by selecting the RNA-based proportions only:

$$\hat{p}_i = \hat{p}_i^{\text{RNA}}.$$

Uniform averaging (limean). Cell-type proportions are first estimated independently from RNA-seq and DNAm data, yielding $\hat{p}_i^{\text{RNA}}$ and $\hat{p}_i^{\text{DNAm}}$ for sample i.

The final estimate is obtained by averaging the two modality-specific predictions:

$$\hat{p}_i = \frac{1}{2}\left(\hat{p}_i^{\text{RNA}} + \hat{p}_i^{\text{DNAm}}\right).$$

This strategy assumes equal contribution and reliability of both modalities.

Error-weighted aggregation (limeanRMSE). For each modality $m \in \{\text{RNA}, \text{DNAm}\}$, the observed bulk signal and its reconstruction are first normalized by their respective column sums:

$$\tilde{Y}^{(m)} = \frac{Y^{(m)}}{\mathbf{1}^\top Y^{(m)}}, \quad \hat{\tilde{Y}}^{(m)} = \frac{X^{(m)}\hat{P}^{(m)}}{\mathbf{1}^\top X^{(m)}\hat{P}^{(m)}},$$

where $\hat{P}^{(m)} = [\hat{p}_1^{(m)}, \ldots, \hat{p}_{n_{\text{mix}}}^{(m)}]$ denotes the matrix of estimated proportions for all samples. A global reconstruction error is then computed over all samples and features:

$$\text{RMSE}_m = \sqrt{\frac{1}{G_m \cdot n_{\text{mix}}} \sum_{i=1}^{n_{\text{mix}}} \left\| \tilde{Y}_i^{(m)} - \hat{\tilde{Y}}_i^{(m)} \right\|_2^2},$$

where G_m denotes the number of features in modality m and $\tilde{Y}_i^{(m)}$ is the i-th column of $\tilde{Y}^{(m)}$. Normalized weights are then computed inversely proportional to each modality's reconstruction error:

$$w_{\text{RNA}} = \frac{\text{RMSE}^{\text{DNAm}}}{\text{RMSE}^{\text{RNA}} + \text{RMSE}^{\text{DNAm}}}, \quad w_{\text{DNAm}} = \frac{\text{RMSE}^{\text{RNA}}}{\text{RMSE}^{\text{RNA}} + \text{RMSE}^{\text{DNAm}}}.$$

The final estimate is obtained as a weighted combination of the two modalities:

$$\hat{p}_i = w_{\text{RNA}}\hat{p}_i^{\text{RNA}} + w_{\text{DNAm}}\hat{p}_i^{\text{DNAm}}.$$

This strategy assigns higher weight to the modality with lower reconstruction error, reflecting its greater reliability.

Rule-based selective averaging (tunedJ). A baseline estimate is obtained by averaging both modalities:

$$\tilde{p}_i = \frac{1}{2}\left(\hat{p}_i^{\text{RNA}} + \hat{p}_i^{\text{DNAm}}\right).$$

To account for modality-specific reliability, a subset of cell types $\mathcal{C}_{\text{tumor}} = \{\text{basal}, \text{classic}\}$ is estimated using DNAm data only. The final estimate is thus defined as:

$$\hat{p}_{ic} = \begin{cases} \hat{p}_{ic}^{\text{DNAm}} & \text{if } c \in \mathcal{C}_{\text{tumor}}, \\ \tilde{p}_{ic} & \text{otherwise.} \end{cases}$$

Finally, proportions are renormalized to satisfy the sum-to-one constraint:

$$\hat{p}_i = \frac{\hat{p}_i}{\sum_{c=1}^{k} \hat{p}_{ic}}.$$

B.3.7 Detailed results for late integration

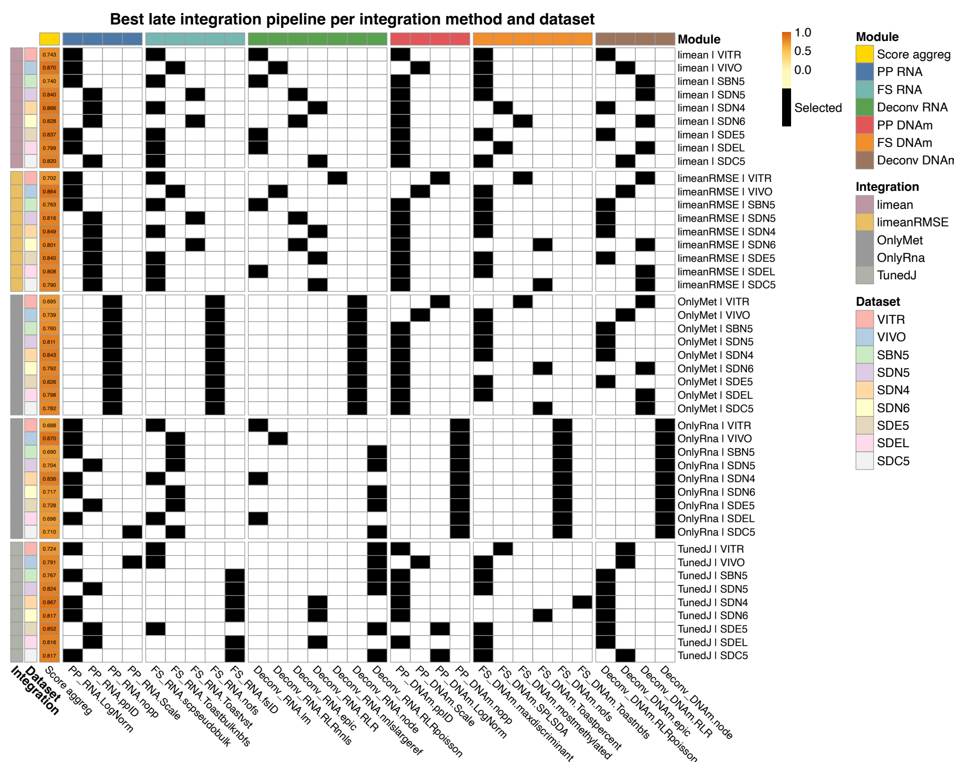


Figure 12: Best late integration pipeline combination for each integration method and dataset. Each row corresponds to one integration method and dataset pair. The first column reports the best aggregate score achieved (yellow-to-orange gradient, range $[0, 1]$). Remaining columns indicate the selected method in each pipeline module (black: selected; white: absent), grouped by module type (color-coded column annotations): preprocessing (PP RNA, PP DNAm), feature selection (FS RNA, FS DNAm), and deconvolution applied independently to each modality (Deconv RNA, Deconv DNAm). Horizontal gaps separate integration methods; vertical gaps separate modules.

B.3.8 ANOVA modelisation

To quantify the contribution of each module to overall performance, we fitted the following linear model:

$$s_{ijkl} = \mu + \alpha_i^{\text{prep}} + \beta_j^{\text{fs}} + \gamma_k^{\text{deconv}} + \delta_l^{\text{int}} + \varepsilon_{ijkl}, \tag{4}$$

where s_{ijkl} denotes the aggregate score of the combination defined by preprocessing method i, feature selection method j, deconvolution method k, and integration method l, and ε_{ijkl} is a residual term.

This linear decomposition of aggregate scores by module (Supplementary Methods B.3.8, Figure 3) reveals that the dominant factor differs between regimes: the deconvolution algorithm drives late integration performance, while the integration method dominates early integration, largely because latent embedding approaches (`omicade4bulk`, `Kernel`) perform poorly and inconsistently.

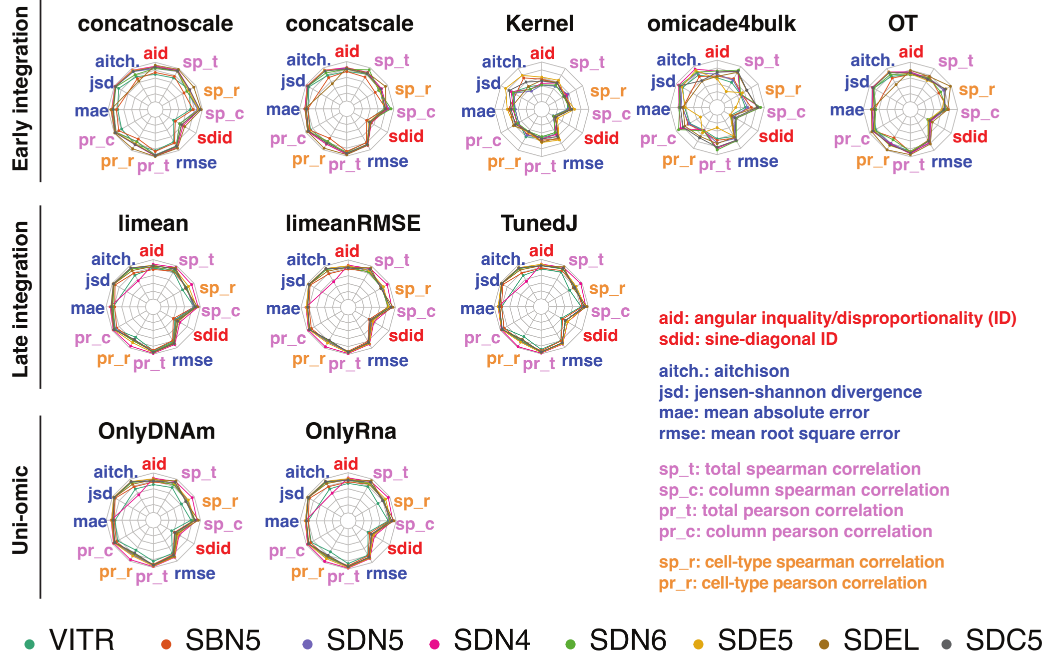


Figure 13: **Performance variability of integration strategies across datasets and metrics.** Radar plots showing per-metric scores of the best-performing pipeline for each integration strategy, on each dataset (except VIVO, which follows a specific scoring procedure, see section B.2.1). All top-performing pipelines are detailed in Tables 14 and 15.

B.3.9 Top-performing combination for each integration method

Table 14: Top combinations for each early integration method.

	RNA		DNAm		Deconv.
Integration	PP	FS	PP	FS	
`concatnoscale`	ppID	scpseudobulk	ppID	mostmethylated	RLRpoisson
`concatscale`	LogNorm	scpseudobulk	ppID	Toastpercent	RLR
`Kernel`	ppID	Toastbulknbfs	Scale	splsda	epic
`omicade4bulk`	Scale	Toastvst	Scale	splsda	lm
`OT`	LogNorm	Toastbulknbfs	LogNorm	fsID	RLR

Table 15: Top combinations for each late integration method.

	RNA			DNAm		
Integration	PP	FS	Deconv.	PP	FS	Deconv.
`limean`	LogNorm	scpseudobulk	lm	ppID	maxdiscriminant	RLRpoisson
`tunedJ`	Scale	fsID	RLRpoisson	ppID	maxdiscriminant	RLRpoisson
`limeanRMSE`	LogNorm	Toastbulknbfs	RLRpoisson	ppID	maxdiscriminant	RLRpoisson
`onlyDNAm`	—	—	—	ppID	maxdiscriminant	RLRpoisson
`onlyRNA`	LogNorm	Toastbulknbfs	RLRpoisson	—	—	—

B.4 Computational costs

All experiments were run on a high-performance computing cluster. Details will be provided upon acceptance. The computational pipeline is structured as a collection of discrete tasks that vary considerably in their resource requirements. Tasks are heterogeneous by design: at one extreme, trivial operations such as returning an identity matrix require negligible computation; at the other, memory-intensive dataframe operations can demand significant CPU time and RAM. This heterogeneity must be kept in mind when interpreting raw task counts. The minimal pipeline (the smallest complete set of tasks required to reproduce the core results of this paper) consists of 1,073,340 tasks. This figure is a strict lower bound, assuming no failures, reruns, or exploratory work. In practice, the total number of tasks executed over the course of this project was an order of magnitude larger, reaching at least 11,881,512 tasks. This inflation reflects failed jobs requiring resubmission, parameter sweeps, prototyping runs, and results that were ultimately not included in the paper. To estimate CPU consumption, we rely on the empirical throughput of the main production runs, in which approximately 6,000,000 tasks were processed over three weeks (504 hours of wall time) across 3 nodes of 192 cores each (a total of 290,304 core-hours). This yields an average of approximately 2.9 CPU-minutes per task (0.048 core-hours), taken across the full heterogeneous task mix. Applying this rate, the minimal pipeline requires an estimated 51,000 core-hours, while the full scope of computation (including failures, prototyping, and non-reported results) accounts for upwards of 575,000 core-hours.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are supported by a large-scale empirical benchmark covering more than 250,000 method combinations across nine datasets. Our conclusions, in particular that multi-omic integration does not consistently outperform the best uni-modal strategy and that performance is context-dependent, are directly derived from these results and are carefully qualified within the scope of the evaluated datasets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: see Section 7

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: the paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes. All datasets, preprocessing steps, and evaluation procedures are described in the paper Section 4 and 5 and supplementary materials. In addition, the full benchmarking pipeline is implemented in Nextflow and publicly available, enabling exact reproduction of the experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The datasets are described in Sections 3.2 and Supplementary Material B.1 and available in Zenodo [18] and GEO NCBI [29]. The code used to simulate datasets and generate all figures presented in the paper and to reproduce the benchmark is also publicly available on GitHub [17].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The full modular pipeline used to generate and evaluate all method combinations is publicly available as a reproducible Nextflow workflow [17], along with the implementations of all evaluated methods. A detailed description of each method and its configuration is provided in the Supplementary Material (Section B.3).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our evaluation is based on an exhaustive benchmark covering all method combinations across multiple datasets, rather than stochastic training runs or repeated sampling experiments. As such, performance variability arises from systematic differences

between pipeline configurations rather than random noise, and is already captured through the distribution of results (e.g., median and top scores) reported in Section 6. We therefore do not report classical error bars, which are not meaningful in this exhaustive combinatorial setting. Statistical testing is also discussed in the Section 7.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are described in Section B.4. All experiments were run on a high-performance computing cluster. Details will be provided upon acceptance. The computational pipeline consists of heterogeneous tasks with widely varying costs, from negligible operations to memory- and CPU-intensive analyses.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: see Sections 7 and 8.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No],

    Justification: This work does not involve models or datasets with a high risk of misuse, such as generative models or sensitive scraped data. The benchmark relies on publicly available biological datasets and simulated data. We do not identify plausible misuse scenarios beyond standard scientific use.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external datasets and resources used in this work are publicly available and have been appropriately cited and used in accordance with their respective licenses and standard academic practice.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The datasets and code are made available on GEO NCBI, Codabench, GitHub and Zenodo (links provided upon acceptance).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:This paper does not involve human subjects or crowdsourcing experiments in the sense of human participation in behavioral or annotation tasks. The "competition" refers to a computational benchmarking challenge where participants submit algorithms rather than providing human annotations or judgments. Therefore, no participant instructions, compensation schemes, or human-subject protocols are applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.