Milite *et al. Genome Biology* (2025) 26:90
https://doi.org/10.1186/s13059-025-03530-9

Genome Biology

BRIEF REPORT

Open Access

MIDAA: deep archetypal analysis for interpretable multi-omic data integration based on biological principles

Salvatore Milite[1*], Giulio Caravagna[2*] and Andrea Sottoriva[1*]

*Correspondence: salvatore.milite@fht.org; gcaravagna@units.it; andrea.sottoriva@fht.org

[1] Computational Biology Research Centre, Human Technopole, Milan, Italy
[2] Department of Mathematics, Informatics and Geosciences, University of Trieste, Trieste, Italy

Abstract

High-throughput multi-omic molecular profiling allows the probing of biological systems at unprecedented resolution. However, integrating and interpreting high-dimensional, sparse, and noisy multimodal datasets remains challenging. Deriving new biological insights with current methods is difficult because they are not rooted in biological principles but prioritise tasks like dimensionality reduction. Here, we introduce a framework that combines archetypal analysis, an approach grounded in biological principles, with deep learning**.** Using archetypes based on evolutionary trade-offs and Pareto optimality, MIDAA finds extreme data points that define the geometry of the latent space, preserving the complexity of biological interactions while retaining an interpretable output. We demonstrate that these extreme points represent cellular programmes reflecting the underlying biology. Moreover, we show that, compared to alternative methods, MIDAA can identify parsimonious, interpretable, and biologically relevant patterns from real and simulated multi-omics.

Background

Fundamental processes in cellular biology, such as cell differentiation, development, and carcinogenesis, are inherently driven by multiple interacting molecular layers. Those encode the information that orchestrates the intricate regulatory networks of proteins, transcription factors, and signaling molecules [1] that give rise to biological phenomena. Any attempt to look at a single molecular layer at a time will miss crucial biological insights. High-throughput multi-omics technologies that can measure many concurrent molecular layers in the same cell or sample promise to help gain a more comprehensive picture of biological phenomena [2]. However, integrating and extracting patterns from these high-dimensional, noisy, and sparse data is a significant statistical and algorithmic challenge [3]. The biggest problem is that current state-of-the-art methods are based on something other than biological principles but merely focus on the issue of

dimensionality reduction in a data-driven fashion, which makes the output of those approaches hard to interpret from a biological perspective.

Moreover, some methods might also have limiting assumptions when applied to biological data. For instance, probabilistic multi-omics factor analysis (MOFA), a technique successfully used to find patterns in multiple omics data (e.g., gene expression, protein abundances) [5–7], posits that the observed data can be linearly reconstructed from the latent factors and their loadings. Latent factors from linear models are interpretable; for instance, a factor with high loadings from genes involved in a specific metabolic pathway might be interpreted as representing that pathway. However, linear models miss complex non-linear interactions typical of real biological systems, such as the non-proportional relationship between gene expression and metabolite concentrations [8], threshold-dependent effects of epigenetic modifications on gene activity [9], cooperative transcription factor binding [10], and general environmental factors [11].

To overcome the limits of linear models, a popular non-linear dimensionality reduction framework is the variational autoencoder (VAE) architecture [12, 13]. VAEs can model arbitrarily complex interactions between the input variables via an encoding/decoding mapping parameterised by a deep neural network. The latent space provided by VAEs is more powerful and expressive than linear ones. Yet, it is no longer interpretable, making VAEs like "black-box" compression machines [14]. This is a limitation for biological applications where we want to understand the system. In particular, in biology, we need generative models with an interpretable latent space that can be used to analyze specific system perturbations. For this reason, we argue that we need to inject biological principles into data integration approaches.

Results

Archetypal analysis (AA) [15] is a matrix factorization algorithm designed to decompose the input data as a convex (i.e., linear) combination of extreme data points called archetypes. Contrasted with other methods, AA forces strong constraints on the geometry of the latent space and recovers a set of bases that are expressed only in terms of the relative distances from the archetypes. AA is grounded in the biological principles of evolutionary trade-offs and Pareto optimality, where extreme geometrical points in the space of biological "states" represent phenotypic programs cells or organisms converge to [16]. AA is a promising alternative for dimensionality reduction because, by construction, its coordinate system is trivially interpretable in the same domain of the data. Moreover, in the last years, AA has enjoyed active development of efficient algorithms and tools (ParTi [17], PCHA [18], gradient-based methods [19]) and has already been shown to be able to recover relevant patterns from single-modality high-throughput biological data [20–22].

The linear latent space of AA can be turned into a non-linear manifold by combining AAs with deep neural networks for archetypal decomposition [23]. In this way, it is possible to retain the interpretability of the latent space while leveraging the power of non-linear dimensionality reduction. Building on this idea, we developed MIDAA, an open-source Python framework that integrates multi-omics data using deep archetypal analysis. MIDAA supports different input types and neural network architectures, adapting seamlessly to the high complexity of modern biological data,

which ranges from counts in sequencing assays to binary values in CpG methylation assays. In principle, the model could be extended to combine data from non-omics sources (e.g., text and images) when combined with embeddings from other deep-learning models. MIDAA is implemented on a PyTorch [24] backend that leverages GPU acceleration, scaling to thousands of cells (e.g., 100,000 cells in ~ 5 min for 500 epochs training, Additional file 1: Fig. S1).

Using synthetic data (Methods), we tested if MIDAA could decipher two relevant biological processes: cellular differentiation (Fig. 1B, C) and evolutionary dynamics on a fitness landscape (Fig. 1E, F). In both cases, we generated synthetic data from 1000 cells (20 datasets, 3 noise levels) with matched ATAC/RNA sequencing (chromatin accessibility and gene expression measurements) and compared MIDAA to a pipeline where we first performed dimensionality reduction with both linear and non-linear models, followed by canonical AA. To ensure a broad comparison, we selected a set of methods that use different statistical techniques for multi-omics data integration: JIVE [25] based on PCA, intNMF [26] based on non-negative matrix factorization, MOFA [6] based on factor analysis, and a vanilla VAE [27]. In the latent space produced by these methods (Supplementary Figs. 2–4), we ran linear archetypal analysis as implemented in the R package *archetypes* [28]. In the differentiation test, MIDAA vastly outperformed competing methods, on average reducing the RNA and ATAC reconstruction error by 15% and 55%. In the evolutionary dynamics test dataset, where we simulate a branching cellular differentiation process, MIDAA decreased the reconstructor error for the latent space by 13% (average) across all noise levels (Fig. 1D–H). Notably, in the latter test, a clear performance difference was observed between linear and non-linear statistical models (Fig. 1H), with MIDAA being the top performer on average. Interestingly, in the oversimplified case of a linear generative latent space (Additional file 1: Fig. S5), while linear models achieved the lowest reconstruction error, MIDAA was the best non-linear model, suggesting its geometrical constraints regularize the model.

A critical problem that involves complex multimodal interactions is the differentiation of hematopoietic stem cells (HSC) into mature blood cells, known as hematopoiesis. We used MIDAA to extract biologically interpretable insights from single-cell multi-omics data (whole genome CpG methylation status and transcriptional activity) of CD34 + positive cells, a type of hematopoietic progenitor cell [29]. First, we calculated the level of commitment for specific lineages in each cell by computing a score for a particular gene signature [29]. In this dataset, we found a group of hematopoietic stem and progenitor cells (HSPC) differentiating first into immature myeloid progenitors (IMP) and then into erythroid progenitors (EP) and neutrophil progenitors (NP). MIDAA analyzed 512 cells to find four optimal archetypes (Additional file 1: Fig. S6 and Fig. 2A, B), producing a latent space that recapitulates lineage commitment in this dataset. In particular, the archetype weights were strongly associated with all the terminal states in the adopted gene signature (Additional file 1: Fig. S7), suggesting that the latent geometry matches the differentiation landscape. In comparison, MOFA and VAE embeddings failed to extract the patterns of EP and NP cells, with the most relevant MOFA factors driven by highly variable samples. Overall, none of the competing methods fully recapitulated these cells' differentiation features (Supplementary Figs. 8–10).

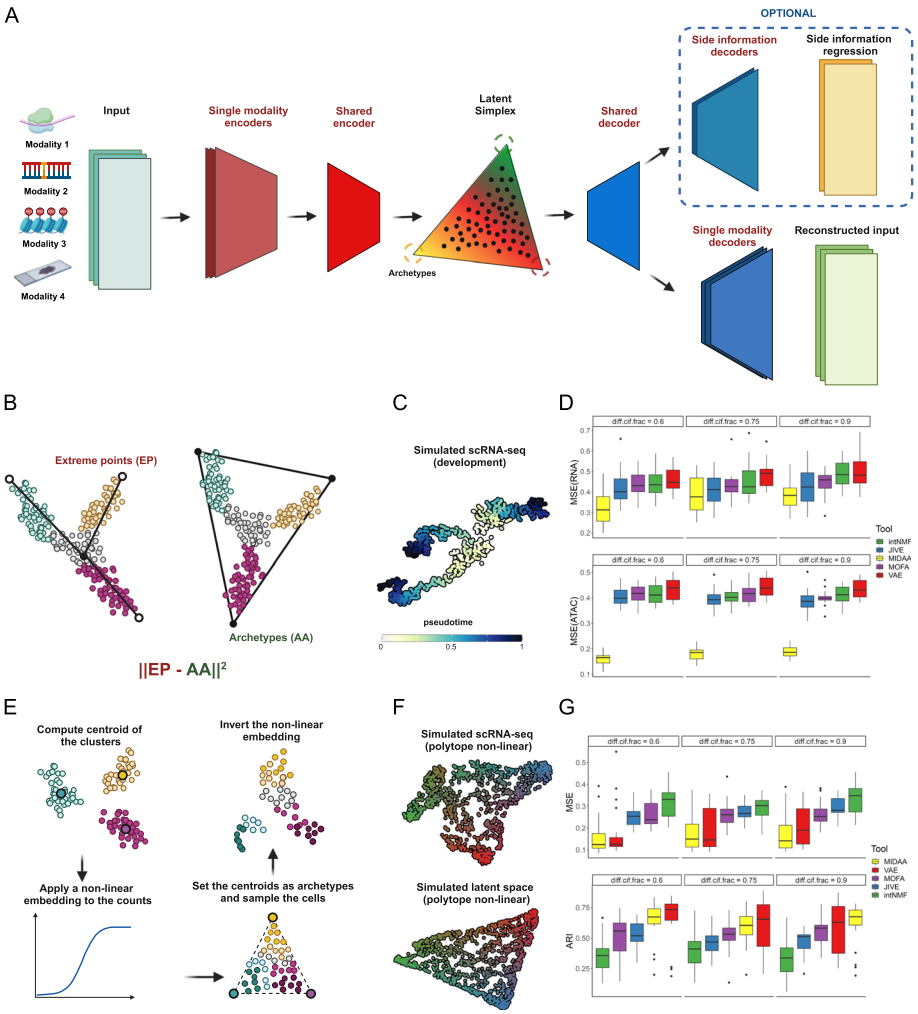

Fig. 1 Performance of MIDAA on a multiomics benchmark dataset. **A** Schematic representation of the model. We allow an arbitrary number of modalities in input, the model then encodes each modality using a private encoder. The last layer of these modality-specific encoders is concatenated and given as input to a shared encoder that learns the latent space and the simplex structure. The decoding part is exactly the reverse with the addition of an optional decoding branch for regression/prediction tasks. **B**–**C** We simulate a branching differentiation process. The process is indexed continuously by a pseudotime value that roughly recapitulates the differentiation level of a cell. We model the differentiation starting from a stem center population with pseudotime 0 differentiating towards 3 different states terminal states with high pseudotime. Our goal here is to understand if the terminal (i.e., low and high pseudotime) state of differentiation is recapitulated correctly by the archetypes **D** We measured the mean squared error (MSE) between the aggregated expression (top panel) or peak counts (bottom panel) of cells at terminal states (bottom 15% and top 75% percentile of pseudotime) and the reconstructed archetypes **E**–**F** For the second test, we sample from a simplex structure in a non-linear latent space, the non-linearity is parameterized by a neural network. **G** Here we measure how well the tools reconstruct the original latent space. As error measures, we computed the MSE of the true and inferred archetype weights (top) and the Adjusted Rand Index (ARI) for the true and inferred highest archetype assignments. In all the plots "diff.cif.fraction" controls the fraction of divergence among archetypes or populations in the development trajectory, a lower number implicates a higher noise

We then investigated whether MIDAA's latent space reproduced known differentiation lineages for these cells. We compared our archetypes (recapitulated by cells with weight > 80%) to a k-means clustering in MOFA and VAE latent spaces to answer this. Our archetypes identified clear progenitor cells, whereas the standard

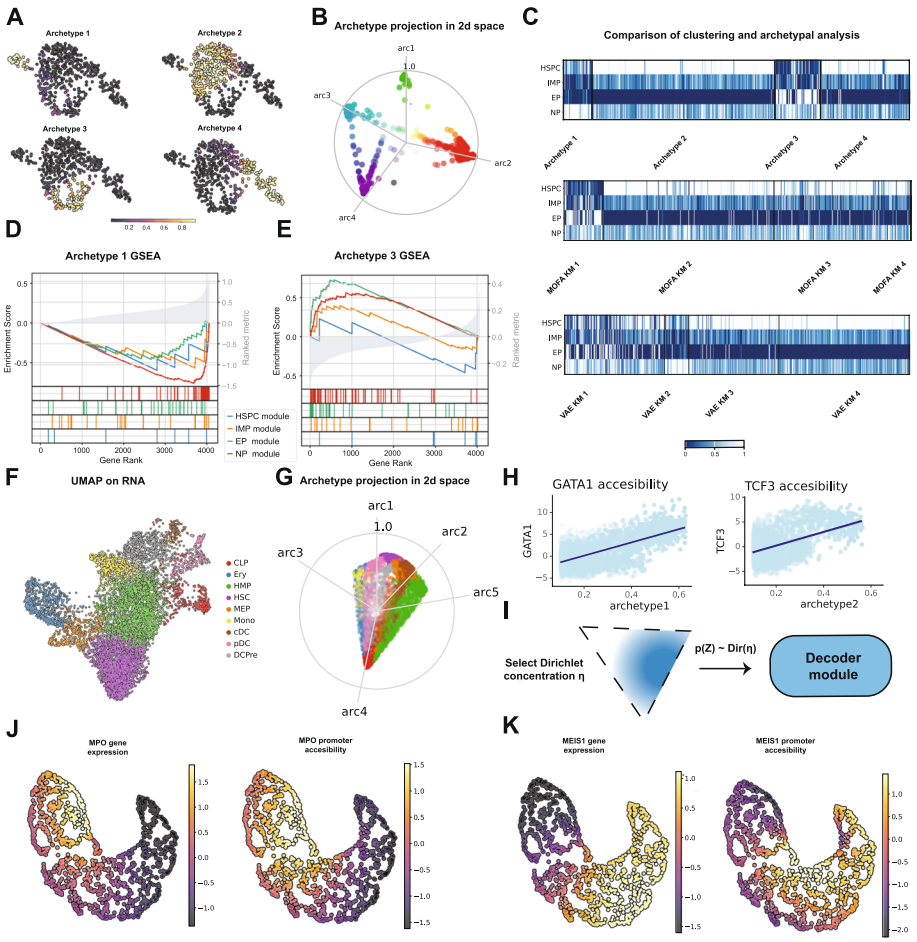

Fig. 2 Multimodal deep archetypal analysis reconstructs an efficient and biologically meaningful latent space. **A** Archetype distribution plotted over the RNA UMAP. **B** A 2d projection of the simplex latent space. Here weights vectors are plotted in 2d polar coordinates. Cells that closely resemble archetypes are far from the center and close to the specific archetype on the outer circle, point on the inside are a mixture of different archetypes. The weights components can be identified by considering the direction of each point in the space as a mixture of unitary vectors pointing at the text labels on the outer circle. A detailed mathematical description of the projection can be found in the Methods section **C** Heatmap of normalized [0–1] cell progenitor scores for cells with archetype probability $\geq$ 80% and K-means clustering in VAE and MOFA space. **D**–**E** GSEA enrichment analysis for archetypes 1 and 3 using the cell progenitor gene sets from [29]. **F**–**G** UMAP and 2d simplex projection of the dataset in [30]. **H** Correlation of transcription factor motif deviation and archetype weights. GATA 1 is an erythropoietic commitment marker and TCF3 is enriched in dendritic progenitors. **I** The generative nature of the model makes it easy to produce synthetic datasets from the latent space. First of all the user can sample from a Dirichlet distribution specifying the concentration parameter and from that the decoder generates realistic multi-modal data. **J**–**K** Concordance of gene expression and promoter accessibility in a synthetic dataset consisting mainly of the erythropoietic and stem archetypes

achieved the worst separation (MIDAA silhouette score increased by ~ 90%) (Fig. 2C and Additional file 1: Fig. S11). Interestingly, this analysis highlighted that a single MIDAA archetype did not represent immature myeloid progenitors (IMP). Instead, we observed that a combination of archetypes represented IMPs. This is consistent with IMP cells being in a transition state from HSPC to EP and NP cells.

Finally, we tested whether archetypes could be used in an unsupervised way for the discovery of biological programs. To investigate this, we ran a gene set enrichment

analysis on the expression reconstructed for each archetype, using as input gene sets the cellular programs of hematopoietic progenitors identified by MIDAA. We found one archetype enriched for genes characteristic of EP and one positively enriched for NP genes but negatively enriched for HSPC genes (Fig. 2D, E), consistent with our previous clustering analysis. This suggests that MIDAA's archetypes can be easily associated with well-defined biological characteristics, which can be used for downstream analysis to represent realistic data measurements.

To further confirm MIDAA's flexibility and potential to adapt to different input data types, we analyzed a distinct cohort of CD34 + cells, this time generated with $10 \times$ GEX + ATAC libraries [30]. MIDAA found an optimal number of five archetypes (Fig. 2F, G).

Taking advantage of the ATAC measurement, we first ran chromVar [31] to calculate the dataset's transcription factor (TF) motif deviations. We then correlated the inferred values for some critical TFs of hematopoietic development with the archetype weights, observing a significant positive correlation trend (Fig. 2H). This shows again how MIDAA's latent space recapitulates the known biological processes in the data.

To demonstrate the clear advantages of archetype analysis (AA) in biological discovery, we compared the expression of known cell markers across the top 15% of cells with the highest scores for each archetype and cluster. We focused on SPINK2, a marker of hematopoietic stemness, and CA1, a marker of erythropoietic lineage commitment. In the case of SPINK2, several clusters exhibit high SPINK2 expression, while a single archetype demonstrates a clearer, stronger enrichment in the marker expression (Additional file 1: Fig. S11A and B). A similar pattern is observed for CA1 (Additional file 1: Fig. S11C and D). This evidence suggests that archetypes provide a more specific and less ambiguous signal of biological processes, offering stronger, more consistent insights for downstream analyses. Similarly, we plotted the cell type composition for both AA and clusters (Additional file 1: Fig. S11E and F). As anticipated, the clusters exhibit significant heterogeneity, containing a mixture of multiple cell types. In contrast, archetypes are predominantly composed of a single cell type, particularly those representing extreme states within the differentiation process.

Thanks to its generative architecture, MIDAA makes it possible to simulate multiomics data from the latent space in a biologically informed fashion. To achieve this, the user can sample from the archetypes simplex and, from these samples, the decoder will generate realistic data measurements (Fig. 2I). To show this, we sampled a synthetic dataset consisting mainly of archetypes 3 and 5 associated with HSC and dendritic cell (DC) progenitors. The output recapitulates the expected HSC to DC transition, as evidenced by the MPO and MEIS1 markers. Notably, this effect is observed at the gene expression level and as chromatin accessibility in the promoter, proving how MIDAA can produce realistic, consistent synthetic data across distinct data modalities.

The fit quality of the deep archetypal analysis model depends highly on the overall geometry of the latent space of the input data. Specifically, as we try to approximate the convex hull of the latent space, the result will be less reliable if it is highly non-convex, as it would produce a polytope with low density regions. At the same time, even when convex, a space that is poorly approximated by a polytope (i.e., a circle in 2d) will produce a very large number of archetypes and thus will be harder to interpret.

Care must be also taken when interpreting the results in the light of evolutionary trade-offs and Pareto optimality [16]. The setting described in [16] is indeed a sufficient condition for the data to be arranged in a convex polytope structure, but it is not a necessary condition. Consequently, it is possible for some biological systems to have archetypes that do not correspond to optimal phenotypic programs. Nevertheless, their interpretation as extreme points still holds and, as such, can be useful for modeling purposes.

Conclusions

In this paper, we demonstrated that MIDAA generates interpretable, biologically coherent, and expressive embeddings for multi-omics data. Moreover, thanks to its generative architecture MIDAA can also be used to simulate new synthetic data.

Methods

The matrix factorization problem

Omics data is commonly represented in the form of high-dimensional, sometimes sparse, numerical matrices. In this context, dimensionality reduction becomes essential not only to make subsequent analysis feasible from a computational point of view, but also to filter out technical noise and minor sources of variability. Indeed, the most common analysis pipeline for single cell RNA and ATAC assays first involves dimensionality reduction using PCA or similar methods, and then graph modularity clustering to extract relevant groups in the dataset.

The general definition of the problem is quite simple: given an input matrix $X^{N\times M}$ with $N \in \mathbb{N}$ samples and $M \in \mathbb{N}$ features, we wish to find a two-matrix decomposition of X . In other words, after fixing an $R \in \mathbb{N} < M$, our decomposition writes as:

$$\mathbf{X} \approx \mathbf{HW}$$

Here $\mathbf{H}$ is an $N \times R$ matrix, and $\mathbf{W}$ is an $R \times M$ matrix. This formulation describes an extremely broad family of methods, depending on the specific constraints and properties we force on the two matrices H and W and on the metrics we optimize for the reconstruction. For instance, when we constrain the W matrix to be orthogonal and to explain the maximum amount of variance by component we obtain the PCA. On the other hand constraining both matrices to be positive while minimizing some generic cost function corresponds to the learning formulation of NMF.

Trivially, if we have multiple input modalities, if we index them by $g = [1, \ldots, G]$, we can naturally reframe the problem as:

$$\mathbf{X}_g \approx \mathbf{HW}_g$$

In this case, we allow the number of features to differ by modality so that we have $\mathbf{X}_g$ and $\mathbf{W}_g$ specific for each modality, with dimension $N \times M_g$ and $R \times M_g$ where M_g is the number of features for modality g .

Archetypal analysis

Archetypal analysis (AA) is a dimensionality reduction method that solves the matrix factorization problem by enclosing the data into a convex polytope [15]. The vertices of this polytope, which span its convex hull, are called archetypes and are generally interpreted as extreme or ideal samples in the dataset. Differently from clustering, where centroids are mean or prototypic representation of a given class, in AA the archetypes represent the farthest point in the data cloud and as such can be seen as the most extreme points in a dataset. Another important difference with clustering is in the interpretation of archetypes and centroids. While in clustering we use the centroids as representative for all the points belonging to a cluster, and thus we discretize the data; in AA each point is always seen as a continuous mixture of archetypes, this property has a clear advantage in non-discrete settings.

More formally, let us fix the number of vertices (or, equivalently, archetypes) to $K \in \mathbb{N}$, and introduce the matrices $\mathbf{A} = (\mathbf{a}_{nk})$ and $\mathbf{B} = (\mathbf{b}_{kn})$ with sizes respectively of $N \times K$ and $K \times N$. Moreover, let us constraint these matrices to be row stochastic, namely:

$$\sum_{n=1}^{N} a_{nk} = 1 \; and \; a_{nk} \geq 0$$

$$\sum_{k=1}^{K} b_{kn} = 1 \text{ and } b_{kn} \geq 0$$

In this setting, by assuming again multiple input modalities our AA decomposition reads as:

$$\mathbf{X}_g \approx \mathbf{A}\mathbf{B}_g\mathbf{X}_g$$

which reduces to the original matrix factorization problem if we set $R = K$, $\mathbf{H} = \mathbf{A}$, and $\mathbf{W} = \mathbf{B}_g\mathbf{X}_g$.

The original algorithm to solve AA was introduced by [15] and formulated as an alternating least square problem on the two matrices. Faster approaches have been developed such as the principal convex hull method [18] and the Frank-Wolfe method [32] gradient. Nevertheless, also those former optimized methods still need to perform computations using the full input matrix, making AA generally slow for datasets of millions of points. Archetypal analysis has been successfully used in modeling single-modality data in biology [17, 33, 34]; our goal here is to extend it to multimodal data and provide a unified framework in the context of deep latent variable models. All of this is conveniently packed in a user-friendly Python package that easily adapts to the plethora of omics data currently available.

Deep multi-omics archetypal analysis

We started from the deep learning extension of the archetypal analysis proposed in [23] to build our MIDAA model. Our main goal is to perform amortized inference over the

two matrices **A** and **B** in some latent space **Z**, now treated as random matrices. The main idea of amortized inference is to spread out (or amortize) the computational cost of inference over multiple input data points by learning a reusable inference model. Instead of performing inference from scratch for each new data point, a neural network is trained to quickly approximate posterior distributions or latent variables, making inference more efficient.

Ideally, we want a reduced latent representation of the input in some non-linear shared space **Z** and then learn the convex polytope. Indeed, our method performs joint inference over the polytope and the latent space. To reduce the degrees of freedom and avoid optimizing both over the number of archetypes and the dimensionality of the hidden space, we fix the polytope shape to be a simplex and set the number of dimensions of the hidden space as the number of archetypes—1, as in [23].

Moreover, we will use an encoder-decoder to encode our latent space and project back the AA results. Formally, let us define the number of latent dimensions as $K - 1$ and the latent space representation as **Z** with dimensions $K \times N$. Then, we can define the simplex reconstruction in latent space $\mathbf{Z^*}$ as:

$$\mathbf{Z^*} = \mathbf{ABZ}$$

Unlike [23], we do not fix **BZ** to be the standard simplex; rather, we explicitly learn and compute both the factor **BZ** in one passage. We do this for two reasons: first, we have fewer parameters to tweak, and second, we observed that, in this configuration, our latent space formulation achieves better average scores on synthetic tests (Additional file 1: Fig. S12). In our model, we constrain **Z** to be in $[0, 1]^{K-1}$ instead of the standard isotropic Gaussian used in VAEs [27]. This choice regarding inference will be made more evident in the next sections.

In MIDAA, we use the standard encoder-decoder inference approach of VAE [27]. In our specific case, the parameters of the **ABZ** distributions are amortized by a neural network f_g^θ, referred as the encoder.

To simplify the notation, we will write the encoder as a single function for the rest of this section. However, it is important to note that the first step of the encoding process is specific to each modality, meaning that each modality has its own independent function and network. The outputs of these individual modality encoders are then concatenated and passed into a shared encoder, as shown in Fig. 1A.

To then compute the training loss, we project the simplex reconstruction $\mathbf{Z^*}$ back to the original space using another neural network, called the decoder, which reconstructs the input features using as input $\mathbf{Z^*}$. In addition to the input reconstruction loss (RHS of the equation below) of standard autoencoders, we allow the network to optionally classify side data **Y** that we index with $s \in \mathbb{N}$ (LHS of the equation). This is useful when we want our archetypes to also reflect some additional variables that we do not want to include in the encoding phase. For instance, in a scRNA-seq experiment we might want the latent space and archetype model to reflect previously annotated cell types, without however using this information to inform the gene expression reconstruction.

In particular, given a likelihood distribution with its parameter set, the total likelihood reads as:

$$p\left(\mathbf{X},\ \mathbf{Y}|\mathbf{A},\ \mathbf{B},\ \mathbf{Z}\right) = \sum_{g=1}^{G} \lambda_g p\left(\mathbf{X}_g | f_g^{\psi}\left(\mathbf{Z}^*\right)\right) + \sum_{s=1}^{S} \lambda_s p\left(\mathbf{Y_s} | f_s^{\xi}\left(\mathbf{Z}^*\right)\right)$$

where we define $\mathbf{X} = [\mathbf{X_1}, \ldots, \mathbf{X}_\mathrm{G}]$ and $\mathrm{Y} = [\mathrm{Y}_1, \ldots, \mathrm{Y}_\mathrm{G}]$.

Here, f_g^{ψ} and f_s are the decoding network for the side and input data. Again, for simplicity, we omit that there is a shared part and a modality-specific part and refer to Fig. 1A for a full representation of the network used. As different modalities can have different numbers of features, we allow the user to specify constants λ_g and λ_s to normalize the likelihood; by default, they are set to respectively $\frac{1}{M_g}$ and $\frac{1}{M_s}$ in order to give the same importance to each modality (where M_g and M_s are the number of features for each input and side modality).

Model inference and formulation

We define the learning objective in a way akin to a standard VAE [27] but with some significant differences regarding the form of the distribution involved in the latent space. The loss function that we optimize is, however, the same, as well the evidence lower bound (ELBO) that we maximize throughout training using stochastic gradient descent with the Adam optimizer:

Our variational distributions q are defined over the matrices $\mathbf{A}$, $\mathbf{Z}$ and $\mathbf{B}$ and we assume the following factorization $\prod_n q(\mathbf{a}_n) \prod_k q(\mathbf{b}_\mathrm{k}) \prod_{nm} q(z_{nm})$. To keep the notation consistent, here we multiply and over respectively the rows and the columns (i.e., we assume independence among archetypes, latent dimensions, and samples).

The choice of the variational distributions comes naturally from the constraint of AA:

$$q(\mathbf{a}_n) = Dirichlet\left(f_{0,n}^{\theta}(\mathbf{X})\right)$$

$$q(\mathbf{b}_k) = Dirichlet\left(f_{1,k}^{\theta}(\mathbf{X})\right)$$

$$q(z_{nk}) = Uniform\left(f_{2,n,k}^{\theta}(\mathbf{X})\right)$$

Here with we index the output dimensions of the encoder. Priors have the same functional forms as the variational posteriors and have equal unitary concentration for the Dirichlet, while the Uniform has a range $[-1, 1]$. Regarding the distribution for $\mathbf{z}$, we departed from the standard isotropic Gaussian as a prior as it tends to concentrate probability density on the shell of a hypersphere (in high dimension) or push towards the center (in lower dimensions) and, as such, makes the space particularly badly suited for learning a simplex representation of the data [35].

Regarding the likelihood distributions, we allow flexibility and currently support a broad range of distributions as valid likelihoods:

- Beta, for variables distributed in the [0, 1] range such as allelic frequencies.
- Poisson and negative binomial for counts data with and without overdispersion, such as those produces by scRNA-seq or scATAC-seq experiments.
- Gaussian and gamma for respectively real and positive continuous values, like normalized gene expression or PCA components.
- Categorical for discrete and binary classes, like the absence or presence of a mutation or the cell cycle phase.

Benchmark on simulated data

We used the scMultisim tool [36] to generate synthetic single-cell multi-omics data, which uses real-data inferred gene regulation networks to sample both trajectory-like and clustered gene expression and chromatin accessibility data.

Here we give a brief mathematical description of the other tools used in the benchmark; we refer the reader to the original papers for further details.

- JIVE aims at decomposing the set of input matrices as $\mathbf{X}_{\mathrm{g}} = \mathbf{J} + \mathbf{A}_{\mathrm{g}} + \boldsymbol{\epsilon}_{\mathrm{g}}$, where $\mathbf{J}$ is a common matrix among the modalities, while $\boldsymbol{A}_{\mathbf{g}}$ models modality-specific factors and $\boldsymbol{\epsilon}_g$ is a noise term. Importantly, the rows of joint and individual structures are constrained to be orthogonal.
- intNMF is a method that extends non-negative matrix factorization (NMF) to the multimodal setting. More precisely, it tries to solve the problem $\mathrm{argmin}_{\mathbf{H_g},\,\mathbf{W}} \parallel \mathbf{X}_{\mathrm{g}} - \mathbf{W}\mathbf{H}_{\mathrm{g}} \parallel_2 \quad \mathrm{i} = 1, 2, \ldots, m$ such that all the entries $\mathbf{H}_g,\ \mathbf{W}_g \geq 0$.
- MOFA is a popular tool for solving factor analysis problems in multi-omics settings. The problem setting is similar to the one above: namely, we model $\mathbf{X}_g = \mathbf{H}\mathbf{W}_g + \boldsymbol{\epsilon}_g$. Where $\mathbf{H}$ is a common loadings matrix and $\mathbf{W}_g$ is a modality-specific factors matrix $\boldsymbol{\epsilon}_g$ is an error term. MOFA is a Bayesian model and, instead of specifying hard constraints in the optimization problem, it assigns prior distributions to the matrix entries, a likelihood distribution for each modality, and computes a posterior distribution for the two matrices.
- VAE is a non-linear generative model that learns a probabilistic mapping from data modalities $\mathbf{X}_g$ to a latent space $\mathbf{Z}$ and back to the original space. Both the encoding$\left(p\left(\mathbf{Z}|\mathbf{X}_{\mathrm{g}}\right)\right)$ and the decoding $(p(\mathbf{X}|\mathbf{Z}))$ distributions are parameterized by a neural network. It maximizes a lower bound on the likelihood called the evidence lower bound (ELBO).

We analyzed two main case studies: one in which the latent space is a simplex and one in which it is instead a differentiation trajectory. In the first case, we generated a cohort in which the mapping function from the space of observables to the latent space is linear and another one in which it is non-linear.

For each of these cases, we simulated 20 datasets of 1000 cells.

We also repeated the experiments for three values of the parameter *diff.cif.fraction* in the *sim_true_counts* function, namely [0.6, 0.75, 0.9] to simulate different amounts of noise (lower values correspond to higher noise).

To generate the datasets in the latent simplex case, we first sampled three clusters and took their centroids as archetypes. The single cells were then simulated by sampling a matrix of archetype weights from a Dirichlet distribution $\mathbf{A}$ and multiplying it with the observation centroid for each modality $\mathbf{C}_g$. In the non-linear case, we first learn a latent space with a variational autoencoder $\mathbf{AC}$, compute the centroids in this latent space, and then feed $\mathbf{A}$ to the decoder (note that this time centroid are modality agnostic). We tested how well the methods reconstructed the archetype distribution $\mathbf{A}$. If we call $\tilde{\mathbf{A}}$ the inferred the score we computed is:

$$MSE\left(\mathbf{A},\ \tilde{\mathbf{A}}\right) = \frac{1}{NK}\sum_{nk}\left(a_{nk} - \tilde{a}_{nk}\right)^2$$

We also computed the Adjusted Rand Index (ARI) between the inferred and true highest archetype defined as $h_i = \text{argmax}\ \mathbf{a_i}$.

For the trajectory cohorts, we were interested in comparing the archetypes to the terminal points of the trajectory. In this case, we define the terminal points as those having the lowest and highest pseudotime values. We computed a set of trajectory endpoints $\mathbf{t}_k$ by aggregating the expression of the bottom 15% percentile and the top 75% percentile of pseudotime for each terminal branch. We did the same to get and aggregate the 75% percentile of cells with the highest weight for each archetype to $\mathbf{h}_k$. We matched each archetype index $\hat{k}$ to the differentiation branch k with the lowest Euclidean distance and then computed:

where M is the number of features. We computed this score for both the RNA and the ATAC reconstruction.

Real data analysis: G&T

For the methylation and expression CD34 + dataset, we first filtered the CpG data by keeping only those with sites with less than 65% missing cells. We then filled the NA with 0 (unmethylated CpG). For the RNA, we used as input the batch-corrected latent representation of Scanorama [37] already computed by the authors in the original work (Additional file 1: Fig. S13A). We then run our model with a Gaussian likelihood for the RNA and a Bernoulli likelihood for the methylation. We set a batch size of 300, a learning rate of 0.0001 with an exponential decaying schedule with a rate of 0.1, and run the inference for 1000 epochs using the Adam [38] optimizer. We run the model for a number of archetypes ranging from 2 to 12 and choose the best value of 4 based on plateaus in the ELBO plot (Additional file 1: Fig. S7).

Scores for the different progenitor cells were computed using the function *score_genes* of Scanpy [39] from the gene sets in [29].

To compare the representation power of the different methods, we set the number of latent dimensions in both the VAE and MOFA to 4 and correlate the gene scores to the latent coordinates. For the K-mean clustering, we again chose 4 as the number of clusters, but this time we learned a MOFA model with 30 factors to simulate a more realistic scenario. GSEA [40] was computed for archetypes 1 and 3 on the cell progenitor gene sets using the Python packages [41].

Real data anaylsis: 10 ×

The input matrices for RNA and ATAC where generated by taking the highly variable genes and 10,000 peaks and then log transform and scale them after a library size normalization using Scanpy [39] and SnapATAC [42] (Additional file 1: Fig. S13B). We then run the model with a Gaussian likelihood for 3000 steps, exponential decay of 0.1. The best number of archetypes 6 was selected by again running the model in a range of [2, ..., 12] and looking at the negative ELBO decrease.

We confirmed the relation between archetype weights and cell fate commitment by first running chromVAR [31] to obtain transcription factor deviation scores and then correlating marker TFs with archetype weights. We used the model learned from this dataset to generate some synthetic data. We sampled archetype weights for each cell from a Dirichlet with concentrations [1e − 16, 1e − 16, 2, 1, 2] that were then fed to the decoder. Clustering was performed on the RNA portion using the Leiden algorithm implemented in the Scanpy [39] function *scanpy.tl.leiden*, with a resolution of 0.2, producing a number of clusters comparable in size to the archetypes. Cell markers were sourced from the CellTypist [44] annotation tool.

Projection of a multidimensional simplex in a 2d space

A convenient way of plotting archetypes in a lower dimensional 2d space that captures the space's salient feature is a projection to polar coordinates in a polytope bounded by a unit circle.

We describe the procedure for a single point in the space.

This point is described by a vector $\boldsymbol{a} \in [0.1]^K$ *s.t.* $\sum_k \alpha_k = 1$ with respect to the archetype basis. The archetypes have coordinates in the latent space $\mathbf{Q} = \mathbf{BZ}$. The first step is to find the relative positions of the archetypes on the circle (outer labels in the plots); we ideally want archetypes close in the latent space to be close on our circle.

To do that we compute the Euclidean distance among all pairs of archetypes $\sqrt{< \mathbf{q}\mathrm{k},\ \mathbf{q}\mathrm{j} >}$ and then solve a traveling salesman problem (TSP) to find the path optimizing the pairwise distances. Once we have the optimal order and distances d_k, we normalize them to sum $\hat{d}_k = \frac{d_k}{\sum_k d_k}$ and divide the circle accordingly to get the label position or, equivalently, vertices of the polytope $l_k = 360 \times \sum_{j=1}^{k} \hat{d}_j$.

We can then write the angle θ and the norm of the vector ρ as:

$$\rho = \sqrt{\left(\sum_k \alpha_k \cos\left(l_k\right)\right)^2 + \left(\sum_k \alpha_k \sin\left(l_k\right)\right)^2}$$

$$\theta = arc\tan 2\left(\sum_k a_k \cos\left(l_k\right), \sum_k a_k \sin\left(l_k\right)\right)$$

We end up with a representation where the mixture of archetypes is represented as the angle of the vector representing a point and the amount of purity by its norm. Note also that only pure archetypes have norm one and live in the circle's perimeter,

while the other points shape the polytope. It is important to note that this representation is not unique, and as such the greater the number of archetypes the less reliable the plot becomes. A similar idea appears in [43].

Interpreting the model results

Translating the final model to interpretable biological knowledge is crucial in all machine learning analyses in life science. The advantage of generating synthetic data that resembles actual measurements opens the possibility of extracting meaningful biological information. To achieve this, one can follow at least three distinct approaches.

First, one can assign the archetypes with features from the original data. For example, from a single-cell transcriptomics assay, one could compute log-fold change estimates or gene set enrichment scores between reconstructed archetypes. This would characterize the archetypes (but not necessarily the potentially non-linear weights), at least for the data modality of interest to the user. Second, one can compute the correlation between features and archetype weights to determine which features are related to the weights. This straightforward operation can be implemented by using non-parametric correlation measures that capture non-linear relations, such as Spearman's correlation coefficient. Third, one can compute how important each data modality or feature is to determine archetypes by adopting a leave-one-out approach. In particular, one could remove either a feature or a data type and re-run the model, measuring the distance between the full and reduced models.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-025-03530-9.

Additional file 1: Supplementary figures

Additional file 2: Supplementary analysis of intestinal enterocytes

Acknowledgements
We thank Konstantin Winter and Anastasiia Romanova for their invaluable support to the Computational Biology Research Centre at Human Technopole.

Peer review information
Tim Sands was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contribution
S.M conceived the method, wrote the software and, performed the analysis. A.S and, G.C. supervised the work. S.M., A.S and, G.C. wrote the manuscript text. All authors reviewed the manuscript.

Funding
This work was supported by the CRUK/AIRC Accelerator Award (A26815) to A.S. and has received funding from AIRC under MFAG 2020 - ID. 24913 project – PI. Caravagna Giulio. We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 1409 published on 14.9.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union – NextGenerationEU– CUP J53D23015060001.

Data availability
All the data used in this paper is publically available the CD34+ methylation and RNA datasets are stored on GEO and has accession number GSE158057 while the CD34+ 10x multiome can be downloaded from the Human Cell Atlas portal (https://explore.data.humancellatlas.org/projects/091cf39b-01bc-42e5-9437-f419a66c8a45)

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 31 May 2024 Accepted: 6 March 2025
Published online: 08 April 2025

References

1. Barabási A-L, Oltvai ZN. Network biology: understanding the cell's functional organization. Nat Rev Genet. 2004;5:101–13.
2. Karczewski KJ, Snyder MP. Integrative omics for health and disease. Nat Rev Genet. 2018;19:299–310.
3. Tarazona S, et al. Harmonization of quality metrics and power calculation in multi-omic studies. Nat Commun. 2020;11:3092.
4. Meng C, et al. Dimension reduction techniques for the integrative analysis of multi-omics data. Brief Bioinform. 2016;17:628–41.
5. Argelaguet R, et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. 2020;21:111.
6. Argelaguet R, et al. Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. Mol Syst Biol. 2018;14: e8124.
7. Velten B, et al. Identifying temporal and spatial patterns of variation from multimodal data using MEFISTO. Nat Methods. 2022;19:179–86.
8. Feist AM, Herrgård MJ, Thiele I, Reed JL, Palsson BØ. Reconstruction of biochemical networks in microorganisms. Nat Rev Microbiol. 2009;7:129–43.
9. Zuin J, et al. Nonlinear control of transcription through enhancer-promoter interactions. Nature. 2022;604:571–7.
10. Ibarra IL. Mechanistic insights into transcription factor cooperativity and its impact on protein-phenotype interactions. Nat Commun. 2020;11:124.
11. Igler C, Rolff J, Regoes R. Multi-step vs. single-step resistance evolution under different drugs, pharmacokinetics, and treatment regimens. Elife. 2021;10:e64116.
12. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15:1053–8.
13. Ashuach T, et al. MultiVI: deep generative model for the integration of multimodal data. Nat Methods. 2023;20:1222–31.
14. Montavon G, Samek W, Müller K-R. Methods for interpreting and understanding deep neural networks. Digit Signal Process. 2018;73:1–15.
15. Cutler A, Breiman L. Archetypal analysis. Technometrics. 1994;36:338–47.
16. Shoval O, et al. Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space. Science. 2012;336:1157–60.
17. Hart Y et al. Inferring biological tasks using Pareto analysis of high-dimensional data. Nat Methods. 2015;12:233–5. 3 p following 235.
18. Mørup M, Hansen LK. Archetypal analysis for machine learning and data mining. Neurocomputing. 2012;80:54–63.
19. NumPy/SciPy recipes for data science: archetypal analysis via Frank-Wolfe optimization. https://www.researchgate.net/profile/Christian-Bauckhage/publication/344671912_NumPy_SciPy_Recipes_for_Data_Science_Archetypal_Analysis_via_Frank-Wolfe_Optimization/links/5f885d9f458515b7cf824bd8/NumPy-SciPy-Recipes-for-Data-Science-Archetypal-Analysis-via-Frank-Wolfe-Optimization.pdf.
20. Korem Y, et al. Geometry of the gene expression space of individual cells. PLoS Comput Biol. 2015;11: e1004224.
21. Adler M, et al. Emergence of division of labor in tissues through cell interactions and spatial cues. Cell Rep. 2023;42: 112412.
22. Groves SM, et al. Archetype tasks link intratumoral heterogeneity to plasticity and cancer hallmarks in small cell lung cancer. Cell Syst. 2022;13:690-710.e17.
23. Keller SM, Samarin M, Wieser M, Roth V. Deep archetypal analysis. German Conference on Pattern Recognition. Dortmund: Springer; 2019. p. 171–85.
24. Paszke A. et al. PyTorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst. 2019. abs/1912.01703.
25. Lock EF, Hoadley KA, Marron JS, Nobel AB. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. Ann Appl Stat. 2013;7:523–42.
26. Chalise P, Fridley BL. Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm. PLoS ONE. 2017;12:e0176278.
27. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. *arXiv [stat.ML]* (2013).
28. Eugster M, Leisch F. From spider-man to hero—archetypal analysis in R. J Stat Softw. 2009;30:1–23.
29. Nam AS, et al. Single-cell multi-omics of human clonal hematopoiesis reveals that DNMT3A R882 mutations perturb early progenitor states through selective hypomethylation. Nat Genet. 2022;54:1514–26.
30. Setty M, et al. Characterization of cell fate probabilities in single-cell data with Palantir. Nat Biotechnol. 2019;37:451–60.
31. Schep AN, Wu B, Buenrostro JD, Greenleaf WJ. chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data. Nat Methods. 2017;14:975–8.
32. Frank M, Wolfe P, Others. An algorithm for quadratic programming. Nav Res Logist Q. 1956;3:95–110.
33. Persad S, et al. SEACells infers transcriptional and epigenomic cellular states from single-cell genomics data. Nat Biotechnol. 2023;41:1746–57.

34. Wang Y, Zhao H. Non-linear archetypal analysis of single-cell RNA-seq data by deep autoencoders. PLoS Comput Biol. 2022;18: e1010025.
35. Davidson TR, Falorsi L, De Cao N, Kipf T, Tomczak JM. Hyperspherical variational auto-encoders. 2018. arXiv [stat.ML].
36. Li H, Zhang Z, Squires M, Chen X, Zhang X. scMultiSim: simulation of single cell multi-omics and spatial data guided by gene regulatory networks and cell-cell interactions. Res Sq. 2023. https://doi.org/10.21203/rs.3.rs-3301625/v1.
37. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nat Biotechnol. 2019;37:685–91.
38. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. *arXiv [cs.LG]* (2014).
39. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19:15.
40. Subramanian A, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A. 2005;102:15545–50.
41. Fang Z, Liu X, Peltz G. GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. Bioinformatics. 2023;39:btac757.
42. Fang R, et al. Comprehensive analysis of single cell ATAC-seq data with SnapATAC. Nat Commun. 2021;12:1337.
43. Seth S, Eugster MJA. Probabilistic archetypal analysis. Mach Learn. 2016;102:85–113.
44. Domínguez Conde C, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. Science. 2022;376:eabl5197.
45. Salvatore Milite, Multiomics integration via deep archetypal analysis (MIDAA), sottorivalab/midaa, https://github.com/sottorivalab/midaa
46. Salvatore Milite, MIDAA reproducibility, sottorivalab/midaa_reproducibility, https://github.com/sottorivalab/midaa_reproducibility.

Publisher's Note