ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF FARMACY AND BIOTECHNOLOGY

SECOND CYCLE DEGREE IN
BIOINFORMATICS

MULTIOMICS ANALYSIS OF THE GENE EXPRESSION AND EPIGENETIC DYNAMICS ACROSS CARDIAC DIFFERENTIATION WITH VARIATIONAL AUTOENCODERS

Internal Supervisor

Prof. Castrense Savojardo

Defended by

Mario Esposito

External Supervisors

Enrique Blanco García, PhD

Luciano Di Croce, PhD

Graduation Session III / December 12th, 2024

Academic Year 2023/2024

Table of Contents

Abstract

Gene regulatory circuits must be particularly configured for each cell type. The appropriate deployment of specific gene expression programs is key throughout development to terminate and maintain successfully each class of cells. For instance, cardiac differentiation represents a suitable model for understanding gene regulation and epigenetic control in development. Over the past decades, significant advancements in next-generation sequencing technologies have dramatically accelerated data production. Recently, the increasing application of deep learning (DL) models further is demonstrating the ability to extract meaningful insights from this kind of information that are not reported by classical data mining techniques. Here, we evaluated DL models, including variational autoencoders (VAEs) and autoencoders (AEs), alongside other dimensionality reduction techniques for their ability to compress and reconstruct experimental features collected along cardiac differentiation, by considering genes as data points. For this, we have constructed a comprehensive multi-omics dataset from public repositories, combining gene expression and multiple posttranslational histone modifications in four stages of mouse cardiac differentiation. By adopting the VAE model as feature extractor, genes were mapped into a lower dimensional space. Remarkably, clustering of the VAE latent code has revealed distinct groups with characteristic expression and epigenetic patterns throughout differentiation. To sum up, this study presents a VAE-based clustering pipeline to capture gene expression and epigenetic data dynamics in cardiac development, serving as a proof of concept for a flexible framework that could be generalized to other biological scenarios.

1 Introduction

1.1 Chromatin and histone modifications

In eukaryotic cells, DNA is tightly packed within the nucleus by associating with histone proteins to form a structured complex called chromatin (Millán-Zambrano *et al.*, 2022). This organization allows DNA to be highly condensed while remaining accessible for critical processes such as transcription, replication, repair, and recombination. The fundamental unit of chromatin is the nucleosome, where DNA is wrapped around an octamer of histones (two each of H2A, H2B, H3, and H4) (Luger *et al.*, 1997). Chromatin compaction is further stabilized by linker H1 histones, which bind at the DNA entry and exit points of each nucleosome (Fyodorov *et al.*, 2018).

DNA methylation, core histone posttranslational modifications (PTMs), and binding of non-histone architectural proteins have been shown to affect the intrinsic properties of nucleosomes, and thus have important functions in regulating nucleosome dynamics (Li and Reinberg, 2011). Histone PTMs serve as signals that influence chromatin compaction and, consequently, gene expression. For instance, histone modifications like tri-methylation at lysine 4 of histone H3 (H3K4me3) and acetylation at lysine 27 of histone H3 (H3K27ac) are associated to transcriptionally active states. H3K4me3 is associated with active promoters, while H3K27ac is enriched at both active promoters and enhancers. Histone PTMs are dynamically regulated by enzymes categorized as "writers" (which deposit the modifications), "readers" (which interpret these modifications), and "erasers" (which remove them). Numerous factors are involved in the deposition of PTMs on histones, most of them acting within a protein complex, such as the Trithorax complexes responsible for gene activation, and the Polycomb complexes, implicated in gene repression (Schuettengruber *et al.*, 2017). In mouse embryogenesis, Polycomb group (PcG) proteins are key PTM modifiers, essential for regulating transcriptional programs during development (Aranda *et al.*, 2015). PcG proteins silence lineage-specific genes in pluripotent cells and repress pluripotency genes in differentiated cells (Pasini *et al.*, 2007). PcG proteins are organized into Polycomb Repressive Complexes 1 and 2 (PRC1 and PRC2), catalyzing H2A monoubiquitination (H2Aub) and H3 lysine 27 methylation (H3K27me1/2/3), both correlated with transcriptionally silent chromatin (Aranda *et al.*, 2015). Interestingly, some promoters and enhancers display both activating (H3K4me3) and repressive (H3K27me3) marks in a "bivalent" state (Bernstein *et al.*, 2006). This unique chromatin configuration maintains genes in a poised state, ready to activate or

repress as developmental cues arise. This bivalency is especially important in stem cells and during development, where rapid shifts in gene expression are essential (Blanco *et al.*, 2020). Chromatin is a highly dynamic structure shaped by the interplay between histone PTMs, chromatin remodelers, and transcription factors, rather than being a static scaffold. These factors regulate chromatin accessibility and transitions between open and closed states, enabling precise control of gene expression. Adding another layer of complexity, various species exhibit distinct histone variants that can undergo unique modifications (Martire and Banaszynski, 2020). The significance of these variants is highlighted by findings showing that key residues in histones, especially those near critical regulatory PTM, are frequently mutated in certain cancers. Additionally, the machinery responsible for writing, reading, and erasing PTMs is frequently altered in cancer, and in many cases these mutations are oncogenic drivers or contributors to tumor progression (Shen and Laird, 2013).

1.2 Epigenetic control in heart development

Understanding the regulatory networks controlling heart development have led to significant insights into its lineage origins and morphogenesis, illuminating important aspects of mammalian embryology. The morphogenesis of the mouse heart resembles that of the human, and thus has been critical for understanding human congenital heart disease (Bruneau, 2013). Cardiac differentiation is a key model for studying gene regulation and epigenetic control in development, as it transforms pluripotent cells into specialized cardiac cells through tightly coordinated gene expression programs. This process involves the simultaneous differentiation of cell types like cardiomyocytes, endothelial cells, and smooth muscle cells, essential to heart structure. Precise timing in gene activation and repression is critical; disruptions can lead to human congenital heart disease (CHD), highlighting the role of transcriptional and epigenetic regulation (Akerberg and Pu, 2020).

Significant research has been invested to map the cardiomyocyte epigenome throughout stem cell to ultimate differentiation (Wamstad *et al.*, 2012; Paige *et al.*, 2012) and normal heart development (Nord *et al.*, 2013; Gilsbach *et al.*, 2018). These studies reveal a highly dynamic epigenome with stage-specific changes in histone modifications at promoters and enhancers, shaping transcriptional networks during cardiac differentiation. Transcription factors implicated in CHD, such as Tbx5 and Nkx2-5, interact with histone modifying enzymes to regulate gene expression (Miller *et al.*, 2010). The Polycomb H3K27 methyltransferase Ezh2, regulates gene expression programs that are important for heart development and homeostasis, and deletion of *Ezh2* in cardiac progenitors caused postnatal myocardial pathology (Delgado-Olguín *et al.*, 2012). Furthermore, Ezh2 directly methylates Gata4, a key regulator in heart development,

attenuating its transcriptional activation ability in both mouse and human (He *et al.*, 2012). By understanding these regulatory networks, we will gain insight into cardiac lineage formation, heart morphogenesis, and the genetic mechanisms underlying CHD.

1.3 Chromatin Immunoprecipitation coupled with high-throughput sequencing (ChIP-seq)

Chromatin immunoprecipitation coupled with high-throughput sequencing (ChIP-seq) has become the primary method for mapping genome-wide protein-DNA interactions and histone PTMs. Unlike previous techniques, such as ChIP-chip, ChIP-seq directly sequences DNA fragments of interest, providing higher resolution, fewer artefacts, broader coverage, and a larger dynamic range (Park, 2009). In a typical ChIP-seq experiment, chromatin is first crosslinked to stabilize protein-DNA interactions and then sheared into fragments. Antibodies specific to the target, such as transcription factors, chromatin-associated proteins, or histone PTMs, are applied to selectively precipitate and enrich these fragments. The purified DNA is then sequenced on next-generation sequencing platforms, where adaptors are ligated, clonally clustered amplicons are generated, and enzyme-driven extensions occur in parallel. High-resolution imaging detects fluorescent labels after each extension, enabling precise, comprehensive profiling of DNA-protein interactions and epigenetic marks across the genome. The general computational pipeline involves quality control, read alignment to a reference genome, removal of duplicates, and identification of enriched regions using peak-calling algorithms. The output typically includes genomic coordinates and signal intensity values of ChIP-seq enriched regions, which can be visualized and annotated for downstream analyses of chromatin states and gene regulation. The ENCODE Consortium has conducted extensive ChIP-seq experiments (Moore *et al.*, 2020), leading to the development of standards and guidelines. However, due to the variety of cell types, conditions, and modifications involved, universal guidelines are challenging to establish. The success of ChIP experiments largely relies on the specificity of antibodies used for target proteins or histone PTMs. Further challenges include the need for many cells and prior knowledge of DNA-binding proteins or histone modifications (Furey, 2012). Control experiments in ChIP are essential to address artifacts raised by uneven DNA fragmentation and repetitive sequence enrichment (Park, 2009). To validate ChIP-seq peaks, they should be compared with a matched control sample, common controls include input DNA (pre-IP sample) also known as Whole Cell Extract (WCE), mock IP DNA (IP without antibodies), and non-specific IP DNA (using an antibody against a protein unrelated to DNA binding or histone PTMs). Peak calling remains a computational

challenge, as manual inspection and data visualization are often required to confirm true binding sites, despite advancements in peak-calling tools. Recently, two peak-callers based on Convolutional Neural Networks showed improvements over classical statistical methods, though their benchmarking was limited in sample diversity, and they have not been widely adopted and tested in the literature. (Oh *et al.*, 2020; Hentges *et al.*, 2022).

1.4 RNA sequencing (RNA-seq)

RNA sequencing (RNA-seq) is a powerful tool developed over a decade ago that has transformed molecular biology by enabling detailed analysis of gene expression and transcriptomic profiles (Emrich *et al.*, 2007; Lister *et al.*, 2008). Primarily used for differential gene expression (DGE) analysis, canonical RNA-seq workflow involves extracting RNA, enriching or depleting specific RNA types (such as mRNA or rRNA), synthesizing cDNA, preparing sequencing libraries, and high-throughput sequencing. The computational steps include aligning reads to a reference transcriptome, quantifying and normalizing gene expression levels, and applying statistical models to identify significant changes in expression between conditions (Stark *et al.*, 2019).

Bulk RNA-seq has significantly advanced biological research, but it lacks the ability to analyze gene expression at single-cell resolution, thereby missing insights into individual cell types and spatial organization within tissues (Piwecka *et al.*, 2023). Single-cell RNA-seq, developed in 2009 (Tang *et al.*, 2009), addresses this limitation by enabling detailed profiling of individual cells, which reveals cellular diversity and rare cell populations in complex tissues. Spatial transcriptomics further enhances this by preserving information about each cell's location within the tissue, helping to understand how the expression of the same gene is different depending on the specific domain of the sample that is scrutinized (Ståhl *et al.*, 2016).

While bulk RNA-seq remains a key tool for quick gene expression screens, single-cell and spatial RNA-seq methods are rapidly becoming essential for high-resolution analysis, together providing a more comprehensive understanding of cellular complexity and tissue architecture.

1.5 Deep learning applications in gene expression regulation

The vast accumulation of multi-omics data from NGS has fueled the expansion of deep learning (DL) into gene regulation research. DL algorithms excel due to their ability to capture complex and nonlinear features relationship from high volumes of information. Supervised learning approaches in DL, which rely on labeled data, have proven useful for tasks like predicting regulatory regions, gene expression, and chromatin state classification (Li *et al.*,

2023). In contrast, unsupervised approaches are particularly valuable when data lacks labels, as they are autonomously capable in discovering patterns and relationships within the data (Kang *et al.*, 2022). Autoencoders (AEs) are neural network models that learn efficient representations of data by compressing it into a lower-dimensional space and then reconstructing the original data (self-supervised learning). Unlike other dimensionality reduction techniques like Principal Component Analysis (PCA) and Uniform manifold approximation and projection (UMAP) which reduce dimensions through methods such as eigen decomposition or neighborhood embeddings, AEs rely on DL. Variational Autoencoders (VAEs) are the probabilistic extension of AEs that encode instances as a continuous distribution rather than a single data point, allowing also to generate new instances. Generally, VAEs have demonstrated better performance compared to AEs and other dimensionality reduction methods (e.g. PCA), although the differences were not substantial. VAEs and their variations have shown promise as powerful tools for analyzing data coming from single-cell RNA sequencing (Lopez *et al.*, 2018; Lotfollahi *et al.*, 2019; Eraslan *et al.*, 2019; Grønbech *et al.*, 2020) and bulk RNA sequencing (RNA-seq) (Way and Greene, 2018; Eltager *et al.*, 2023; Mora *et al.*, 2022).

VAEs have been employed for multi-omics (bulk) data integration to investigate gene regulation during mouse CNS development, where the model captured transcriptomic and epigenomic gene patterns over time and across tissues (Mora *et al.*, 2022). Here, VAEs were applied for the first time by transposing the data matrix representing genes as individual data points defined by experimental features collected across different times, tissues and conditions. This transformation effectively increased the number of instances (genes) and reduced the number of features, directly addressing the High-dimension and low-sample-size (HDLSS) data sets problem. This method underscores the flexibility of VAEs in restructuring data to uncover biologically meaningful patterns across complex, dynamic systems. In another study, a VAE model was applied on DNA methylation data from lung cancer samples, uncovering latent features that distinguish cancer subtypes despite using an unsupervised approach (Wang and Wang, 2019). Another piece of work introduced the roadmap-ENCODE Variational Auto-Encoder (RE-VAE), which compresses ChIP-seq signals from promoter and enhancer regions across 935 samples from various tissues and histone modification targets. Analysis of the RE-VAE latent space revealed that most samples clustered by histone marks, but not by tissue or cell type, likely due to the HDLSS dataset problem (Hu *et al.*, 2021).

Clustering is crucial in bioinformatics for understanding gene functions, cell types, and regulatory processes. While traditional clustering methods (e.g. hierarchical clustering, K-means) can be effective in specific domains, they often struggle with the complexity and

variability inherent in biological datasets. Representation learning enables the extraction of meaningful patterns from data, creating representations that are easier to interpret and process (Karim *et al.*, 2020). DL-based approaches can model non-linear relationships between features, leading to more refined and clustering-friendly representations of complex data. Studies applying these techniques to multi-omics datasets, such as those from TCGA Breast Cancer and TARGET Neuroblastoma, have demonstrated the effectiveness of DL-driven representations for generating well-balanced and biologically meaningful clusters (Viaud *et al.*, 2022).

1.6 Introducing dimensionality reduction techniques

1.6.1 Principal component analysis (PCA)

Principal Component Analysis (PCA) is a widely adopted linear dimensionality reduction technique (Pearson, 1901; Hotelling, 1933). PCA seeks to transform high-dimensional data into a lower-dimensional space by identifying orthogonal directions, or principal components, that capture the maximum variance within the data. Given an input matrix of dimensions $(n \times d)$, where n is the number of samples and d is the number of features, PCA starts by centering the input matrix (subtracting the mean feature-wise) to obtain a zero-centered matrix X. The covariance matrix C of the matrix X is computed as follows:

$$C = \frac{1}{n-1} X^T X \tag{1}$$

An eigen decomposition is performed on the covariance matrix C, yielding eigenvalues and eigenvectors. The eigenvectors, also known as principal components (PCs), represent directions of maximum variance, while the correspondent eigenvalues indicate the magnitude of variance explained by each PC. To project the data into a lower-dimensional subspace, PCs are ranked by their eigenvalues (explained variance), the top k PCs are selected, with $k < d$ and the transformed matrix Z_k is given by:

$$Z_k = XW_k \tag{2}$$

where W_k is a $(d \times k)$ matrix of eigenvectors corresponding to the k largest eigenvalues. The resulting matrix Z_k represents the data in a k-dimensional space. While PCA is effective for datasets with linear relationships, it is limited in capturing non-linear patterns, for which non-linear dimensionality reduction techniques may be more suitable.

It is possible to obtain the inverse transformation in PCA, or reconstruction, recovering the original feature space from the reduced-dimensional representation Z_k. This is accomplished by reversing the projection. In other words, the matrix Z_k is multiplied by the transpose of the matrix of the selected eigenvectors W_k^T, to bring it back to the original feature space:

$$\hat{X} = Z_k W_k^T \tag{3}$$

where $\hat{X}$ is the reconstruction of the original data matrix X and since only the top k components are employed, some information is lost in the reconstruction process.

1.6.2 Uniform manifold approximation and projection (UMAP)

Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimensionality reduction technique, that is widely implemented to visualize high-dimensional data in 2D or 3D (McInnes *et al.*, 2018). UMAP is known for its speed and scalability, often requiring fewer neighbors and less computation time than similar techniques like t-Stochastic Neighbor Embedding (Maaten and Hinton, 2008), while allowing flexibility to balance local and global structure preservation through its tunable parameters. UMAP begins by constructing a k-nearest neighbor (kNN) graph to capture local relationships in the data, then projects this graph into a lower-dimensional space by minimizing a cross-entropy objective function. For each data point x_i, it considers its k nearest neighbors, denoted as $\mathcal{N}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,k}\}$. The Gaussian or Radial Basis Function (RBF) kernel is employed to measure the similarity between points in the input space, thus, the probability $p_{j|i}$ of x_j being a neighbour of x_i is determined using the following formula:

$$p_{j|i} = exp\left(-\frac{||x_i - x_j||_2 - \rho_i}{\sigma_i}\right) \tag{4}$$

where $||x_i - x_j||_2$ denotes the L2 norm (Euclidean distance), ρ_i is the distance from x_i to its nearest neighbor:

$$\rho_i = min\{||x_i - x_{i,j}||_2 : x_{i,j} \in \mathcal{N}_i\} \tag{5}$$

σ_i is a local scaling parameter for x_i, chosen so that:

$$\sum_{j=1}^{k} exp\left(-\frac{||x_i - x_{i,j}||_2 - \rho_i}{\sigma_i}\right) = log_2(k) \tag{6}$$

This formula (4) calculates $p_{j|i}$ directly for the case where x_j is within the neighborhood $\mathcal{N}_i$ of x_i , otherwise, in the case x_j is not in $\mathcal{N}_i$, $p_{j|i}$ is set to 0. To ensure symmetry in neighborhood relationships, UMAP calculates a final pairwise similarity, p_{ij} between points x_i and x_j as:

$$p_{ij} \coloneqq p_{j|i} + p_{i|j} - p_{j|i}\, p_{i|j.} \tag{7}$$

This ensures that $p_{ij} = p_{ji}$, making the similarity measure symmetrical. In the low-dimensional embedding space, UMAP represents each high-dimensional point x_i by a corresponding point y_i in a p-dimensional space, with $p \ll d$. The probability of two points y_i and y_j being neighbors in this low-dimensional space is modeled as:

$$q_{ij} = \frac{1}{1 + a||y_i - y_j||_2^{2b}} \tag{8}$$

where a and b are hyperparameters that shape the similarity curve in the embedding space, commonly set to $a \approx 1.929$ and $b \approx 0.7915$. UMAP aligns the high-dimensional graph with the low-dimensional embedding by minimizing the fuzzy cross-entropy cost function, c_1:

$$c_1 = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \left(p_{ij}\, ln\left(\frac{p_{ij}}{q_{ij}}\right) + \left(1 - p_{ij}\right) ln\left(\frac{1 - p_{ij}}{1 - q_{ij}}\right)\right) \tag{9}$$

The cost function includes two terms: an attraction term and a repulsion term. When p_{ij} is high the attraction term $ln(p_{ij}/q_{ij})$ encourages q_{ij} to be large, pulling y_i and y_j closer together. This term should only appear when $p_{ij} \neq 0$, which means either x_j is a neighbour of x_i, or x_i is a neighbour of x_j, or both. When p_{ij} is low the repulsion term $ln(1 - p_{ij}/1 - q_{ij})$ encourages q_{ij} to be small, pushing y_i and y_j apart. The result is a low-dimensional embedding that preserves both the local and global structure of the original data.

1.6.3 Autoencoders (AE)

An autoencoder (AE) is a type of neural network architecture designed to firstly efficiently compress (encode) input data down to its essential features, and secondly, reconstruct

(decode) the original input from this compressed representation (Ballard, 1987). Being the target label the input itself, AEs employ a form of self-supervised learning, without needing external labels. The simplest form of an autoencoder, often referred to as a vanilla AE, consists of an input layer, one or more hidden layers, and an output layer, with no additional complexities or architectural variations. Autoencoders architecture can be divided in two parts: the encoder and the decoder. The encoder learns to extract non-linear feature relations mapping the input into a lower dimensional space (code or latent space), while the decoder learns to reconstruct the input from the latent space. The encoder and decoder mapping functions are respectively:

$$z = f_\theta(x) = s(W_e x + b_e) \tag{10}$$

$$\hat{x} = g_\theta(z) = s(W_d z + b_d) \tag{11}$$

where s is a non-linear activation function, W_e and W_d are the encoder and decoder weight matrices, b_e and b_d are the encoder and decoder bias vectors. The encoding function f_θ maps the vector x (input) into a lower dimensional space obtaining the vector z (code). The decoding function g_θ maps the vector z (code) back to the output vector $\hat{x}$ (output) (Figure 1). During the training process the network's weights (W and b) are adjusted with the objective of minimizing a loss function (x) which penalize differences between the input vector x and the reconstructed vector $\hat{x}$.

$$L(x, \hat{x}) = L(x, g_\theta(f_\theta(x))) \tag{12}$$

When the inputs are real numbers, the Mean Squared Error (MSE) is typically adopted as loss function to measure the reconstruction error. The MSE computed per vector is defined as follow:

$$MSE(x, \hat{x}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 = \frac{1}{N}\|x - \hat{x}\|_2^2 \tag{13}$$

where N is the number of components (dimensions) of the vector x and $\|x - \hat{x}\|_2^2$ denotes the squared L2 norm (Squared Euclidean distance), The square root of the MSE yields the Root Mean Squared Error (RMSE), which is often employed for easier interpretation by maintaining the same unit as the input data.

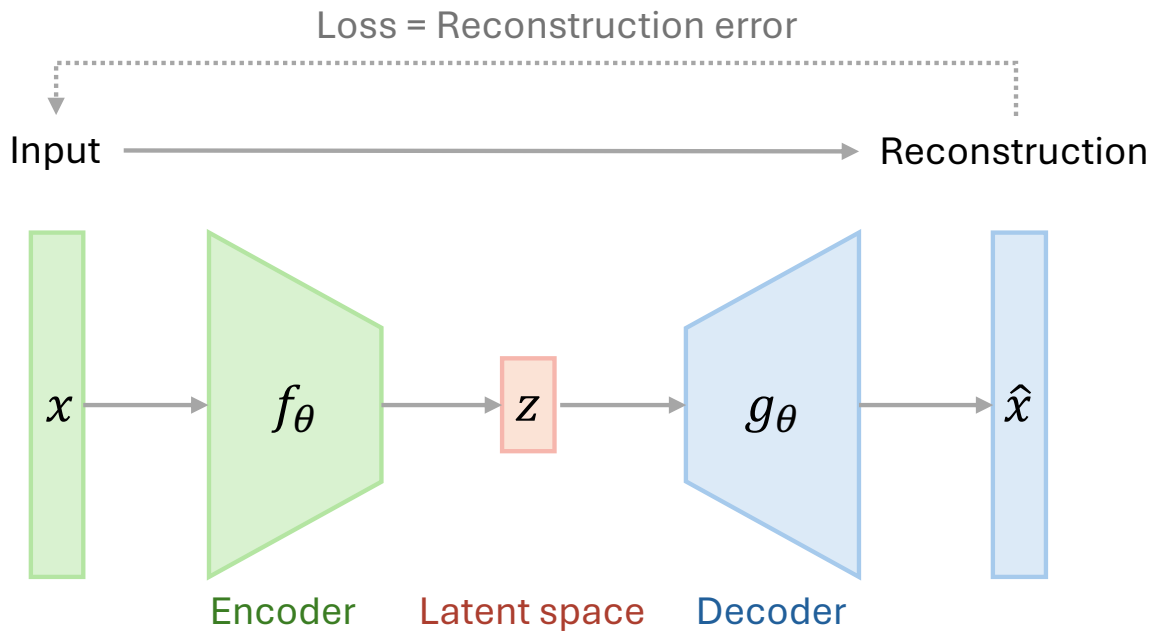

Figure 1. A schematic representation of the AE architecture, highlighting its main components.

1.6.4 Variational autoencoder (VAE)

Variational Autoencoders (VAEs) are a probabilistic extension of autoencoders that aim to model the latent space as a continuous distribution often a Gaussian (Kingma and Welling, 2013). This allows for both dimensionality reduction and the generation of new samples by sampling from the latent distribution. The network encoder is connected with two output vectors, the mean vector μ and the standard deviation vector σ, which defines a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. The latent vector z is sampled from the latent vectors sigma and mu and to enable backpropagation for the sampling step, the reparameterization trick is applied according to the following equation:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0,1) \tag{14}$$

where ϵ is a random variable sampled from a unit Gaussian distribution $\mathcal{N}(0,1)$.

The VAE objective function comprises two main terms: the reconstruction loss and the Kullback–Leibler Divergence (KLD). As in AEs the reconstruction loss measures the difference between the original input x and its reconstruction $\hat{x}$. The KLD serves to regularize the latent space by minimizing the divergence between the encoder's learned distribution $\mathcal{N}(\mu, \sigma)$ and a unit Gaussian distribution $\mathcal{N}(0,1)$. This regularization term ensures that the

learned latent space is continuous and prevents overfitting by encouraging the latent space to follow a Gaussian distribution. The overall loss function L_{VAE} to be minimized is defined as:

$$L_{VAE} = -E_{q(z|x)}[log\, p\,(x|z)] + D_{KL}\big(q(z|x)|p(z)\big) \tag{15}$$

where $E_{q(z|x)}[\log p\,(x|z)]$ is the expectation of the log-probability of the reconstruction, and $D_{KL}\big(q(z|x)|p(z)\big)$ is the KLD between the learned latent distribution $q(z|x) = \mathcal{N}(\mu, \sigma)$ and the prior $p(z) = \mathcal{N}(0,1)$. This latter can be expanded as follows:

$$D_{KL}\big(\mathcal{N}(\mu,\sigma)|\mathcal{N}(0,1)\big) = -\frac{1}{2}\sum_{j=1}^{D}\Big(1 + log\big(\sigma_j^2\big) - \mu_j^2 - e^{log\left(\sigma_j^2\right)}\Big) \tag{16}$$

Where *D* represents the dimensionality of the latent space. By taking the logarithm of the variance, the network is forced to output the range of the natural numbers rather than just positive values (variance would only have positive values). This allows for smoother representations for the latent space. In β-VAE, the loss function is modified by introducing a weighting parameter β which controls the trade-off between the reconstruction loss and the KLD term (Higgins *et al.*, 2017). This adjustment allows for greater control over the balance between the fidelity of data reconstruction and the regularization of the latent space.

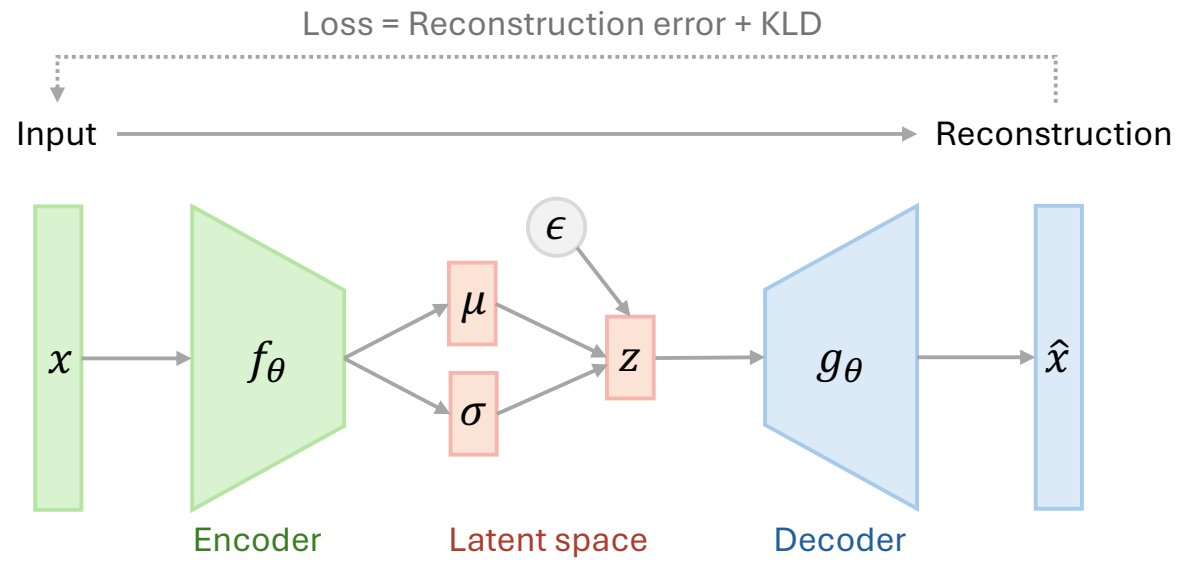

Figure 2. A schematic representation of the VAE architecture, highlighting its main components. Unlike a standard AE, the VAE integrates a probabilistic latent space and a KLD term within the loss function.

1.7 Brief description of the workflow of our study

Here, we have compiled a RNA-seq and ChIP-seq dataset from publicly available repositories (Wamstad *et al.*, 2012), covering four stages of mouse cardiac differentiation and multiple histone PTMs. We separately preprocessed and normalized RNA-seq and ChIP-seq data before concatenating them as set of experimental features. By this way, each gene was represented as a data point, defined by a set of features collected across different time points during cardiac differentiation. Next, we have evaluated data reconstruction performance after dimensionality reduction using VAEs, AEs, PCA, and UMAP. The dataset was split into training, validation, and testing sets to optimize DL hyperparameters and assess generalization capabilities. Among the models, VAE delivered the best reconstruction scores, with AE and PCA scoring similarly, and UMAP the worst. In the final step, we have utilized the VAE encoder to map genes in a unified latent space and then we have performed clustering on the resulting gene representations, identifying interesting groups of genes previously not documented to share certain transcriptomic and epigenomic patterns along cardiac differentiation.

2 Methods

2.1 Dataset construction

2.1.1 RNA-seq and ChIP-seq raw data fetching

Transcriptomics and epigenomics raw data were downloaded as FASTQ files from the Sequence Read Archive (SRA), by the following Gene Expression Omnibus (GEO) (Clough *et al.*, 2024) project accessions: GSE47948 and GSE47949 (Wamstad *et al.*, 2012). To manage data programmatically, metadata tables were downloaded from the European Nucleotide Archive (ENA) projects repositories (Leinonen *et al.*, 2010). RNA-seq data were available as eight experiments comprising of two replicates in four cardiac differentiation stages in mouse: Embryonic Stem Cells (ESC), Mesoderm (MES), Cardiac Precursors (CP), and Cardiomyocytes (CM). ChIP-seq data were available for the same cell types (CTs), with two or three replicates, targeting three histone posttranslational modifications (HMs): H3K4me3, H3K27me3, H3K27ac and whole cell extract (WCE) as a control. For consistency on the number of replicates across different ChIP-seq targets, only the first two replicates were utilized in this study. All FASTQ files were downloaded from SRA by adopting parallel-fastq-dump (https://github.com/rvalieris/parallel-fastq-dump, v0.6.7) which is a wrapper of fastq-

dump (from NCBI SRA toolkit 3.0.10) that allows to speed-up the downloading process with multi-threading (–threads = 12)(Leinonen *et al.*, 2011).

2.1.2 Gene expression levels and fold-changes

The RNA-seq experiments of our dataset consisted of paired-end reads of 150 base pairs (bp) and they were aligned to the reference mouse genome (GRCm38/mm10) using TopHat2 v2.0.14 with specific parameters (Kim *et al.*, 2013); the mean inner mate distance was 176 bp, and the standard deviation of the insert size was 11 bp, mimicking the configuration implemented in the original study; the library type was *fr-firststrand* and the *--no-coverage-search* option allowed to skip the exhaustive coverage based search for junctions. The alignment was further restricted to report a maximum of one alignment per read (*-g 1*) to exclude reads mapping to multiple locations. Gene annotations were provided using the GTF file, generated from the UCSC-RefSeq (refGene) gene annotations downloaded from University of California Santa Cruz (UCSC) Genome Browser (Kent *et al.*, 2002; Pruitt *et al.*, 2014). To increase computational efficiency TopHat2 was run using multi-threading (*-p 6*) and pigz (a parallel version of gzip) was employed to decompress the input FASTQ files. Each alignment file (SAM format) was split in two files, corresponding to the positive and negative strand alignment, by matching respectively the patterns “XS:A:+” and “XS:A:-“.
Gene expression was computed for all the genes (n=24919) in UCSC-RefSeq catalog with DESeq2 (Love *et al.*, 2014). Expression values were normalized from reads count to Fragments Per Kilobase Million (FPKM) to account for gene length and total number of mapped reads. For each gene, 6 expression fold-changes (FCs) were computed for all CT pairwise combinations (without repetition). FCs were expressed as $\log_2$ ratio between the average FPKMs in the two replicates of each CT.

2.1.3 ChIP-seq signals at gene promoter regions

The ChIP-seq experiments consisted of single-end reads and they were aligned to the reference mouse genome (GRCm38/mm10) (Church *et al.*, 2011), using Bowtie2 v2.5.3 with default parameters (Langmead and Salzberg, 2012). Unmapped reads and reads with a mapping quality (MAPQ) lower than 10 were removed by using SAMtools v1.19.2 (Danecek *et al.*, 2021). ChIP-seq reads were counted in all the promoter regions from -2500bp to +2500bp with respect to the Transcription Starting Site (TSS) of each gene. The reads counts were extracted for all the TSSs present in UCSC-RefSeq considering an average fragment size of 250 and normalizing by the total number of reads per sample, with recoverChIPlevels tool from SeqCode toolkit v1.0 (Blanco *et al.*, 2021)(in-lab developed). To assign a single

value at each gene, the average reads count across different TSSs was computed if a gene was annotated with more than one TSS.

2.1.4 Upload alignment on genome browser and gene filtering

For both RNA-seq and ChIP-seq, all the output alignment files in SAM format were compressed in BAM format with SAMtools, and they were converted in bedGraph format and normalized by sequencing depth, running buildChIPprofile tool from SeqCode toolkit. All the bedGraph files were uploaded as custom tracks on UCSC Genome Browser to visualize and integrate the reads of all the experiments in the genome.

To exclude genes with very low expression values, only genes having an average FPKM > 0.5 (across replicates) in at least one CT were retained. Genes having at least one promoter region intersecting with low mappability or high signal regions according to ENCODE blacklist v2 (Amemiya *et al.*, 2019) were excluded with SeqCode-matchpeaks. Small non-coding RNAs and genes mapping on alternative chromosome scaffolds (unlocalized and unplaced clone contigs) were excluded from the analysis too. In the end of the filtering, 14,996 of 24,919 mouse genes were retained.

2.1.5 Normalization and scaling of RNA-seq and ChIP-seq data

To normalize RNA-seq values, the FPKM values were log transformed and then they were normalized across samples adopting the Z-score normalization. For each gene i and for each sample j, z_{ij} was computed as following:

$$x_{ij} = \log_{10}(FPKM_{ij} + 1) \tag{17}$$

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{18}$$

where x_{ij} is the log10(FPKM+1) of the gene i in the sample j, μ_j and σ_j are the mean and the standard deviation in the sample j. To introduce the 6 RNA logFCs as features at the same scale, each logFC feature was divided by its standard deviation without mean centering, to not alter the directionality of the signs. To normalize ChIP-seq values across genes, for each gene i and for each target experiment j, the enrichment ratio e_{ij} (named ChIP-level) was computed as the log ratio between the number of reads (N) of the target experiment and the control of the same cell type.

$$e_{ij} = \log_{10} \frac{N_{ij} + 1}{N_i^{CTR} + 1} \quad (19)$$

Whole Cell Extract (WCE) was available as control experiment in two replicates, however, only the first replicate was adopted for normalization as the second replicate exhibited significantly higher background noise. To normalize across samples, the Z-score normalization was applied with same procedure explained in the equation 18 by replacing x_{ij} with e_{ij}.

2.1.6 Evaluating normalization effects through PCA and correlation analyses

To check how data transformation and normalization could affect samples similarities, distributions were plotted and compared as violin plots and Principal Component Analyses (PCAs) were performed in scikit-learn library in python. All the PCAs were run adopting as input features the genes and only the first 2 PCs were selected for plotting. A PCA was performed on the RNA-seq samples and three PCAs were independently performed by grouping ChIP-seq sample based on the target histone modifications (H3K4me3, H3K27me3 and H3K27ac).

To further inspect ChIP-seq sample, Kendall's rank correlation and hierarchical clustering were performed in Python implementing the SciPy library (Virtanen *et al.*, 2020). The correlation matrix (C) of the input data frame was computed by performing all-vs-all samples Kendall's rank correlation. After that the C matrix was converted to the dissimilarity matrix D by subtracting it from 1 (D=1-C). Hierarchical clustering was performed on the D matrix using average linkage with optimal ordering. The results were presented as correlation heatmaps with dendrograms. Both PCAs and correlation analyses were performed before and after normalization to determine the benefit of the corrections.

2.2 Models training and hyperparameters optimization

2.2.1 Dataset split

The input matrix X was created by concatenating as features for all the genes their FPKM values (n=8), ChIP enrichment signal (n=24) and RNA logFC (n= 6), all independently normalized (See section 2.1.5). The input matrix X consisted in 14,996 genes as instances and 38 features. To compare how the four approaches perform in compressing and reconstructing unseen data, the dataset X was randomly split in training set X_t (90%) and testing set X_{test} (10%), and X_{test} was adopted as hold-out set. The training set X_t was further

split randomly in training set X_{train} (80%) and validation set X_{val} (20%), and the X_{val} was implemented to optimize AE and VAE hyperparameters. To check whether the two random splits produce balanced feature values distributions, violin plots were generated after each split and visually compared (Figure S4). After the performance comparison between the four techniques, the best models were retrained on the whole dataset X for downstream analysis.

2.2.2 Custom loss functions for AE and VAE models

Two custom loss functions L_{AE} and L_{VAE} were designed respectively for the AE and VAE models. The general formulas of the loss functions are the following:

$$L_{AE} = MSE - S_c \tag{20}$$

$$L_{VAE} = MSE - S_c + \beta \cdot D_{KL} \tag{21}$$

Where MSE is the mean squared error (13), D_{KL} is the KL-divergence (16) and S_c is the cosine similarity. The S_c measures the cosine of the angle between two non-zero vectors. For two vectors x (input) and $\hat{x}$ (reconstructed output), the S_c is defined as:

$$S_c(x, \hat{x}) = \frac{x \cdot \hat{x}}{\|x\| \|\hat{x}\|} \tag{22}$$

Where $x \cdot \hat{x}$ is the dot product between vectors x and $\hat{x}$, while $\|x\|$ and $\|\hat{x}\|$ are the norms of vectors x and $\hat{x}$. The value of $S_c(x, \hat{x})$ spans between -1 and 1. If the two vectors point in the same direction, the S_c is 1; if they are orthogonal (perpendicular), the S_c is 0; and if they point in opposite directions, the S_c is -1. Since the objective of the training is to minimize the loss function and the S_c must be maximized, the S_c is subtracted from the total loss (20, 21), turning it into a minimization problem. This custom loss account for difference in magnitude and direction by including both the MSE and S_c. The MSE minimizes the magnitude difference between x and $\hat{x}$, ensuring numerical similarity. S_c, on the other hand, encourage the alignment of the vector's directions.

2.2.3 AE and VAE model optimization via random search

To optimize the hyperparameters of both the AE and VAE models, a random search with 500 trials was conducted using the Keras library with TensorFlow as backend in Python (Chollet, 2015; Abadi *et al.*, 2016). For consistency, the same approach and hyperparameter space

were applied to both AE and VAE models. In this section, any differences specific to AE or VAE will be explicitly noted, otherwise, the descriptions refer to both models collectively. The models consisted of an encoder-decoder structure with a six-dimensional (6D) bottleneck layer, designed to compress the 38 input features, into a lower-dimensional latent space representation. The encoder comprised dense layers with non-linear activation functions and the decoder's architecture mirrored the encoder one.

Key hyperparameters of this architecture were tuned to optimize performance: activation functions, number of activation layers, neuron scaling, batch normalization, batch size, and, for the VAE, the β parameter. Activation functions introduce non-linearity into the model, which allows the AE and VAE to capture complex patterns in the data. Three activation functions were explored here: the exponential linear unit (ELU), the scaled exponential linear unit (SELU) and the parametric rectified linear unit (PReLU). The ELU and SELU are expressed as following:

$$ELU(x) = \begin{cases} x & if\ x > 0 \\ \alpha e^x - \alpha & if\ x \leq 0 \end{cases} \quad with\ a\ =\ 1 \tag{23}$$

$$SELU(x) = \lambda \begin{cases} x & if\ x > 0 \\ \alpha e^x - \alpha & if\ x \leq 0 \end{cases} \quad with\ a\ =\ 1.6733\ and\ \lambda = 1.0507 \tag{24}$$

The PReLU is an advanced variation of the traditional ReLU and Leaky ReLU activation functions. As the Leaky ReLU, PReLU solves the gradient vanishing problem, but differently from it, it makes the slope of (α) a learnable parameter for negative input values.

$$PReLU(x) = \begin{cases} x & if\ x > 0 \\ \alpha x & if\ x \leq 0 \end{cases} \tag{25}$$

The depth of the network was varied by tuning the number of activation layers, which ranged from 2 to 8. This range allowed examination of different model complexities, where deeper networks could capture more intricate relations between the features, potentially enhancing the representation power at the cost of increased computational demand. Also, to control how the neuron count decreases with each successive layer, a neuron scaling factor between 1.0 and 2.0 was explored. Batch size can affect training time and performance; hence, different batch sizes were tested by sampling logarithmically between 16 and 512. The addition of batch normalization between activation layers was also tested. For the VAE, the β hyperparameter was tested for values lower than 1 to decrease the weight of the KLD term in the loss function

(Table 1), thus, leveraging the trade-off between reconstruction accuracy and the regularization of the latent space.

Table 1. AE and VAE hyperparameters optimized through the random search, with their description, search space and sampling distribution (*only for VAE).

Hyperparameter	**Description**	**Search space**	**Type**
Activation function	Non-linear activation function in the neurons	SELU, ELU, PReLU	Categorical
Num. activation layers	Number of activation layers in encoder (same in decoder)	From 2 to 8	Integer, Linear sampling
Num. neuron scaling factor	Factor by which neuron count scales per layer	From 1.0 to 2.0	Float, Linear sampling
Batch normalization	Whether to apply batch normalization between layers	True, False	Boolean
Batch size	Size of each training batch	From 16 to 512	Integer, Log sampling
β*	Weight applied to the KLD term in the loss	From 1×10^{-9} to 1×10^{-1}	Float, Log sampling

The optimization process was guided by the objective of minimizing the validation loss, with a fixed number of trials (n=500, ~7h of running time). The Adam (Adaptive Moment Estimation) optimizer was adopted as training algorithm (Kingma and Ba, 2017). During model training, early stopping was employed with a patience value of 14 epochs. If no improvement in validation loss was observed within this window, training was stopped, and the best-performing model weights (lowest validation loss), were restored. This ensured the model avoids overfitting and prevents from unnecessary training. Additionally, the learning rate was dynamically adjusted using a reduction on plateau strategy and the starting learning rate was set to 0.05. When the validation loss plateaued for 2 epochs $(\Delta L \leq 0.0001)$, the learning rate was reduced by a factor of 0.5. This reduction allowed for finer gradient updates, encouraging a stable convergence.

2.2.4 Comparative evaluation of dimensionality reduction approaches

To evaluate and compare the performance of different DR approaches in compressing and subsequently reconstructing input features, reconstruction error was implemented as scoring metric. For each data subset (X_{train}, X_{val}, X_{test} and X) both MSE and S_c were computed

vector-wise (gene-wise). The final scores were expressed as the mean, along with the 5th and 95th percentiles of their respective distributions, providing a robust summary of model performance across different data subsets. To make a fair comparison, the dimensionality of the latent space (or embedding) was heuristically set to 6 dimensions across all models. PCA and UMAP were trained using the scikit-learn and umap-learn libraries in Python, respectively (Pedregosa *et al.*, 2011; McInnes *et al.*, 2018). Both PCA and UMAP were trained on the X_{train} dataset, after which the X_{test} dataset, which had not been seen during training, was passed into the models for compression and subsequent reconstruction of the original features. For PCA, the reconstruction of features was possible by applying the inverse transformation function from the reduced latent space to the original space (See section 1.6.1). In the case of UMAP, the reconstruction was also achievable, but the process was computationally complex (scales exponentially with respect to the number of components). Specifically, the inverse transformation for the X_{test} dataset took approximately 0.5 hours to compute. Due to this complexity, hyperparameter optimization resulted impractical, and the default parameters were used instead. For AE and VAE, the best models identified through hyperparameter optimization were selected. These models were trained on the X_{train} dataset and later tested on the unseen X_{test} dataset, allowing for a consistent comparison with the PCA and UMAP. Additionally, the four models were retrained on the whole dataset X for downstream analysis and performance comparison from another perspective.

2.3 Clustering genes based on their VAE-based representation

2.3.1 Coefficient of variation on gene expression (stable and variable genes)

For each gene, its variation in expression across cardiac differentiation (different CTs) was measured as coefficient of variation (CV). The CV_i for each gene i was computed by dividing the standard deviation σ_i over the mean μ_i of the FPKM values across the different replicates and CTs (4).

$$CV_i = \frac{\sigma_i}{\mu_i} \tag{26}$$

The CV was computed only for genes having a mean FPKM > 0.5. The gene list was sorted by CV values and the top 4000 and bottom 4000 genes were defined respectively as the most “Variable” and “Stable” genes. Then for each gene was assigned the cell type with the highest FPKM (CT_{max}) and the CT_{max} frequencies were computed in both the “Stable” and “Variable” genes lists. To functional characterize the genes comprised in the two gene lists, term

enrichment analyses were conducted using GSEApy in Python (via the EnrichR API), utilizing three gene set libraries: 'KEGG_2019_Mouse', 'WikiPathways_2019_Mouse' and 'GO_Biological_Process_2023' (Kanehisa and Goto, 2000; Agrawal *et al.*, 2024; Ashburner *et al.*, 2000; The Gene Ontology Consortium *et al.*, 2023). The background list for each analysis included all genes from the dataset (n=14996). Only the top five significant gene sets were reported for each test, based on the adjusted p-value.

2.3.2 Selection of cell type marker genes across cardiac differentiation

A reference set of genes was selected to facilitate the latent space characterization and visually inspect reconstruction performance. Four well-known expression marker genes for each CT were chosen based on prior literature (Figure 3)(Wamstad *et al.*, 2012). Additionally, eight more marker genes per CT were selected based on their fold changes (FCs) and corresponding p-values from differential expression analysis between CT (Figure S1).

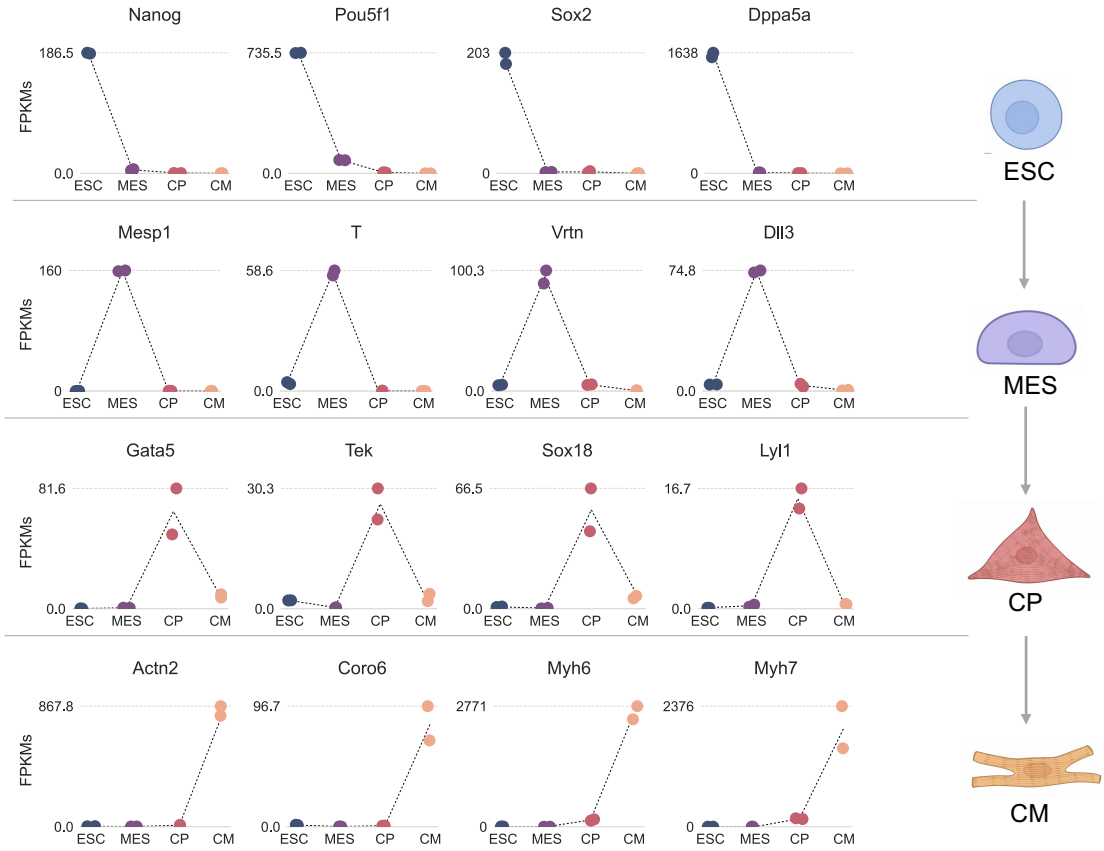

Figure 3. Gene expression profile of marker genes across cardiac differentiation stages (CT). Four marker genes for each CT.

2.3.3 Clustering genes in the latent spaces

Gaussian mixture model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown

parameters, where each Gaussian represents a distinct cluster (Reynolds, 2009). Unlike other clustering techniques that require clearly defined cluster boundaries, GMM is ideal for datasets where the boundaries between clusters are not well-defined, such as the continuous latent spaces produced by VAEs and AEs (Karim *et al.*, 2020).

In this study, GMM was employed to identify distinct gene clusters starting from the gene representation learned in the latent space of the VAE model. The GMM was implemented in scikit-learn library in Python (Pedregosa *et al.*, 2011). The number of clusters k and the covariance type were HPs to be optimized. Thus, a grid search was performed exploring all the value combinations of covariance types and k, with 4 types of covariance (full, tied, diagonal, spherical) and k spanning from 1 to 100 with steps of 5. To evaluate the unsupervised clustering performances and select the best model, three scores were adopted: Silhouette score (S-score), Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC). The Silhouette score measures how similar a data point is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 to 1, where a higher score indicates better-defined clusters. The Silhouette score s_i for each point i is calculated as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{27}$$

where a_i is the average distance between i and all other points in the same cluster, and b_i is the lowest average distance between i and points in a different cluster. A higher average Silhouette score (S-score) across all points reflects better clustering. Both BIC and AIC are information-theoretic metrics used to balance the trade-off between model fit and complexity. The AIC is defined as:

$$\text{AIC} = -2 \cdot log\, L + 2p \tag{28}$$

And the BIC is defined as:

$$\text{BIC} = -2 \cdot log\, L + p \cdot log\, n \tag{29}$$

where $log\, L$ is the log-likelihood of the model, p is the number of parameters in the model, and n is the number of data points. BIC penalizes model complexity more heavily, particularly in situations with large datasets. The best hyperparameter combination was selected by

analyzing the scores behavior in dependence of the HP values by looking at the score's curves (Figure 4). First, it was possible to notice that the optimal covariance type was the spherical one resulting in higher S-score for almost all k values and better BIC for k greater than 30. The optimal number of clusters was set to 80 preferring a higher number of clusters to meet the objective of the analysis. In the end the GMM assigned each gene to a different cluster, yielding to 80 clusters containing on average 187 genes each.

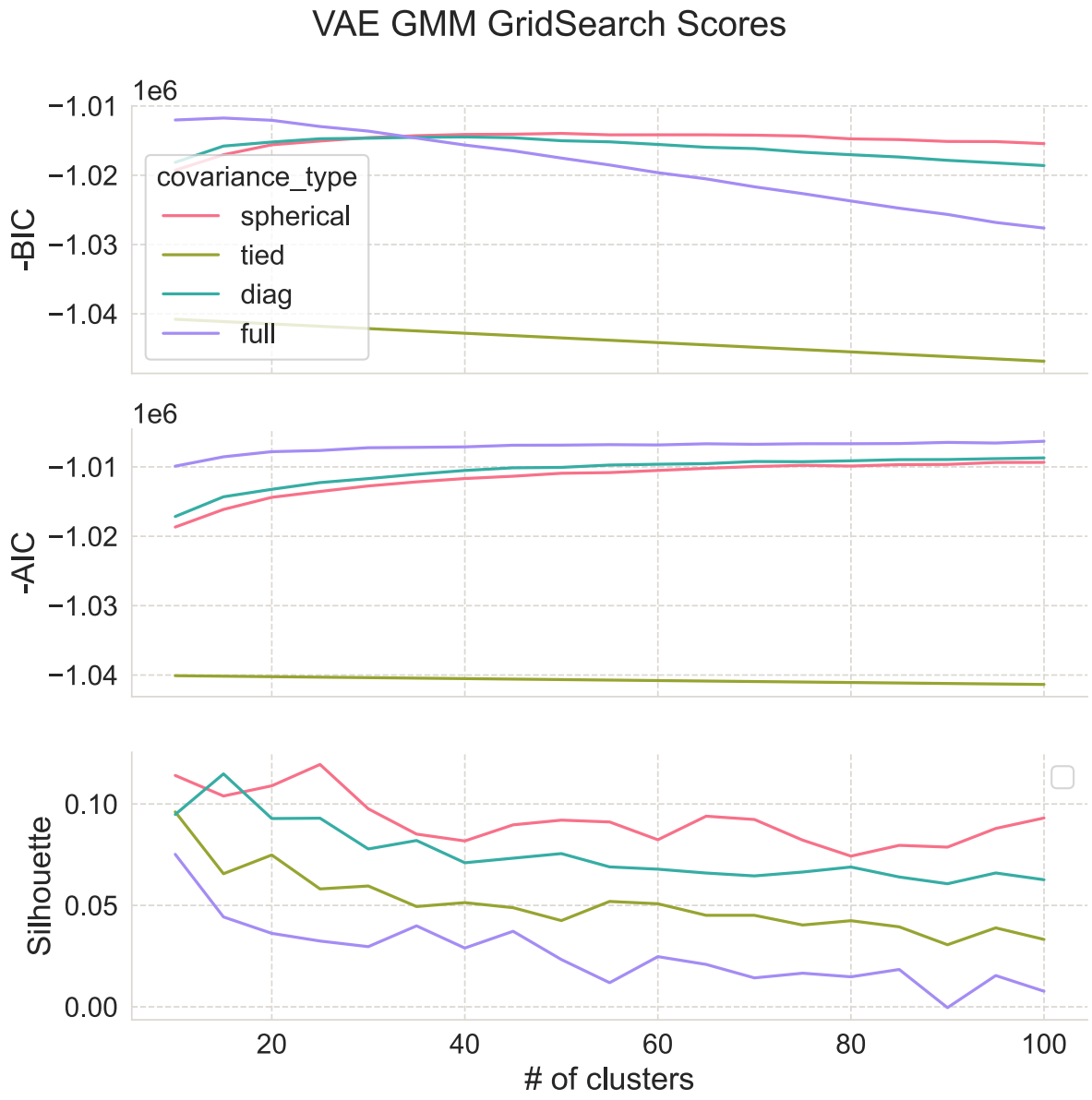

Figure 4. HPs grid-search results, displayed as one line plot for each score on the y-axis (-BIC, -AIC and S-score), with number of clusters (k) on the x-axis and the lines colored by covariance type.

2.3.4 Clusters characterization and visualization

Genes with similar expression and epigenetic patterns across differentiation were expected to be located close to each other in the latent space, thus grouping into the same cluster. Different analyses were conducted for each cluster**:** intersection with gene lists, ChIP-seq TSS-plots, term enrichment and features distribution.

For each cluster, the genes intersection with a gene list was calculated as the percentage of the gene in the cluster common to the gene list and visualized using tree maps, produced with squarify Python library (https://github.com/laserson/squarify). The gene lists were generated according to different biological basis. First, the overlap between the gene clusters and gene lists based on the coefficient of variation (CV) computed on gene expression (See section 2.3.1). Five lists were generated: the most variable genes (top 4,000 CV) stratified by their CT_{max} values (ESC, MES, CP, CM) and the most stable genes (bottom 4,000 CV). Similarly, to investigate the epigenetic state of the clusters, genes annotated as bivalent or active in ESC based on a previous study were intersected with the clusters (Mas *et al.*, 2018).

For each cluster, averaged ChIP-seq metaplots centered on the Transcription Start Site (TSS) were generated independently for each HM (and control) across all CTs, using SeqCode (produceTSSplots). A single replicate BAM file was used for each HM/CT combination. The fragment length was set to 250 bp, with a sliding window of 50 bp, covering a region from -2500 to +2500 bp relative to the TSS. Signal profiles were smoothed using a moving rolling mean with a window size of 200 bp. To make the plots scales as much comparable as possible the maximum y-axis value was set for each CT-HM combination as the maximum value among all clusters.

For each cluster, term enrichment analyses were conducted using GSEApy in Python using the same modalities described in section 2.3.1 with the additional gene set library of 'GO_Molecular_Function_2023'.

For each cluster, gene expression dynamics across cardiac differentiation was plotted as FPKM for a subset of 16 randomly selected genes. Violin and box plots were used to display the distribution of all features (replicates average) within each cluster, enabling comparisons across clusters to capture the general expression and epigenetic dynamics throughout differentiation.

2.4 Hardware and computational setup

All analyses were performed on a MacOS Sonoma 14 system installed on a MacBook M1 Pro with an 8-cores processor (arm64) and 16GB of RAM. To ensure reproducibility, the source

code is available on GitHub (https://github.com/espositomario/CardioDiff-VAE). The repository includes all necessary Python notebooks, Bash scripts and R scripts, as well as an Anaconda environment configuration file to generate the python environment with the same versions adopted here (https://anaconda.com). All plots were generated in Python employing matplotlib and seaborn python packages (Hunter, 2007; Waskom, 2021). For computational efficiency, we utilized GNU parallel to distribute the SeqCode tools processes (recoverChIPlevels, buildChIPprofile and produceTSSplots) in multiple jobs, reducing running time. To speed up model training, Keras and TensorFlow were configured to utilize the Metal API for GPU acceleration (https://developer.apple.com/metal/tensorflow-plugin/).

3 Results

We constructed a dataset of RNA-seq and ChIP-seq data from publicly available resources, covering four key cell types (CTs) in cardiac differentiation: Embryonic Stem Cells (ESC), Mesoderm (MES), Cardiac Precursors (CP), and Cardiomyocytes (CM) (See section 2.1.1). The RNA-seq data comprised eight experiments, with two replicates per CT. For ChIP-seq data, we included three histone modifications (HMs), H3K4me3, H3K27ac, and H3K27me3, each with two replicates per CT. Out of the 24,919 mouse genes annotated in UCSC-RefSeq, 14,996 genes met our filtering criteria. We excluded low-expression genes, small non-coding RNAs, genes mapping to alternative scaffolds, and those intersecting with ENCODE blacklist regions (See section 2.1.4). This curated dataset served as the foundation for all subsequent analyses.

3.1 RNA-seq data normalization reduced distributions skewness

We log-transformed gene expression values (FPKM) and subsequently applied Z-score normalization across samples (See section 2.1.2). The log-transformation reduced skewness in the FPKM distributions (Figure 5b), while the Z-score normalization, although not affecting much the distributions between samples, it adjusted the scale and mean to resemble those of the ChIP-seq features (Figure 5c). To explore the similarity between replicates of the same CT and between different CTs, we performed principal component analyses (PCAs). Although there were some differences between the PCAs, in all of them it was possible to clearly see a trajectory corresponding to the cardiac differentiation (Figure 5a, b, c). PCA performed on the log-transformed FPKM values (Figure 5e) showed more similarity between replicates of MES, CP and CM experiments, when compared with PCA produced on the FPKM (Figure 5d). Apparently, this was due to the log capabilities of mitigate the outliers' impact on the scale.

The PCA after the Z-score normalization did not show noticeable differences (Figure 5f), as PCA itself include a mean-centering step, only the division over the standard deviation makes the difference with respect to the previous PCA.

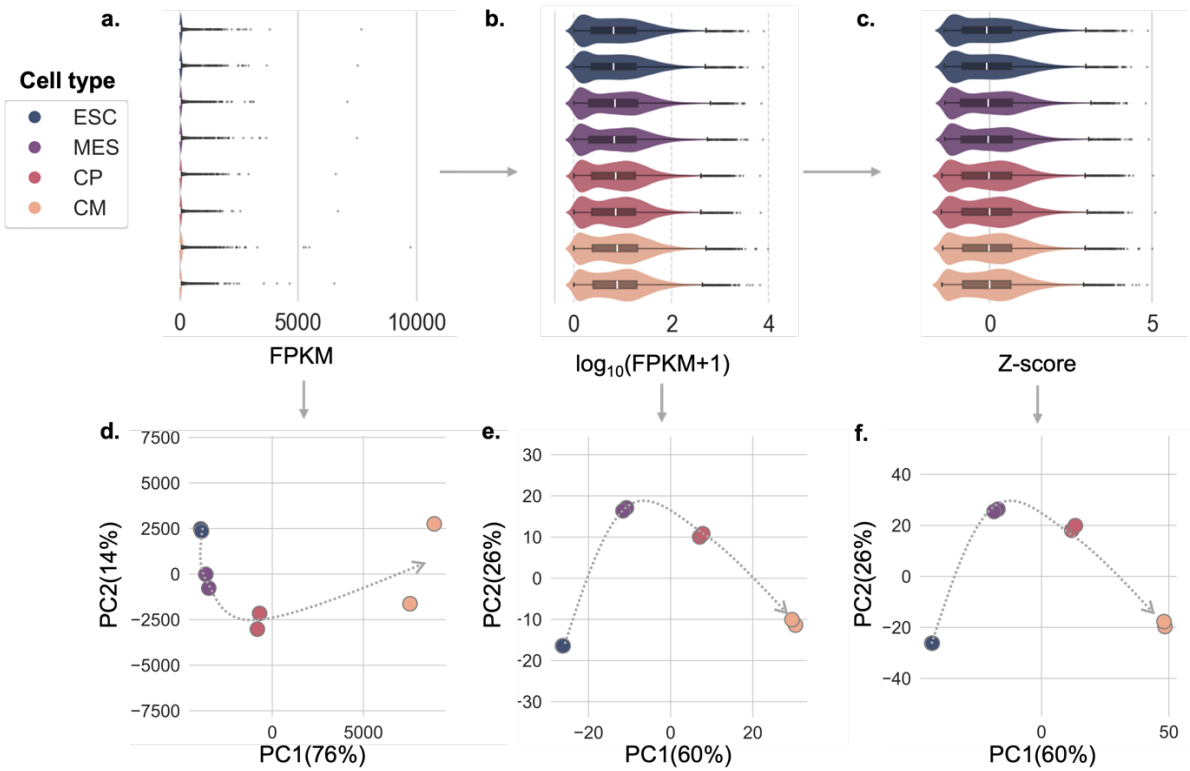

Figure 5. Gene expression values distributions (a, b, c) and PCA plots (d, e, f) before and after log transformation and Z-score normalization. For each replicate, gene expression distributions are shown as violin and box plots colored by CT (outliers as points). RNA-seq replicates are showed as dots in the PCA plot by their top two PCs (% of variance explained between brackets).

3.2 ChIP-seq level normalization balanced samples distributions

We examined the read count distributions (Figure 6a) to identify variations in overall ChIP levels across different HMs and between replicates. H3K4me3 exhibited the highest signal intensity, followed by H3K27ac, while H3K27me3 showed the weakest signal. This pattern aligns with expectations, likely reflecting the differences in antibody affinity for each target, which influences the signal ratio between promoter regions and background. We also observed variations in overall ChIP levels between replicates. After converting read counts to log enrichment ratios (See section 2.1.5), the distributions became less skewed (Figure 6b), though discrepancies between replicates and differences in overall ChIP levels across HMs

remained. Z-score normalization across samples helped reduce these differences substantially (Figure 6c).

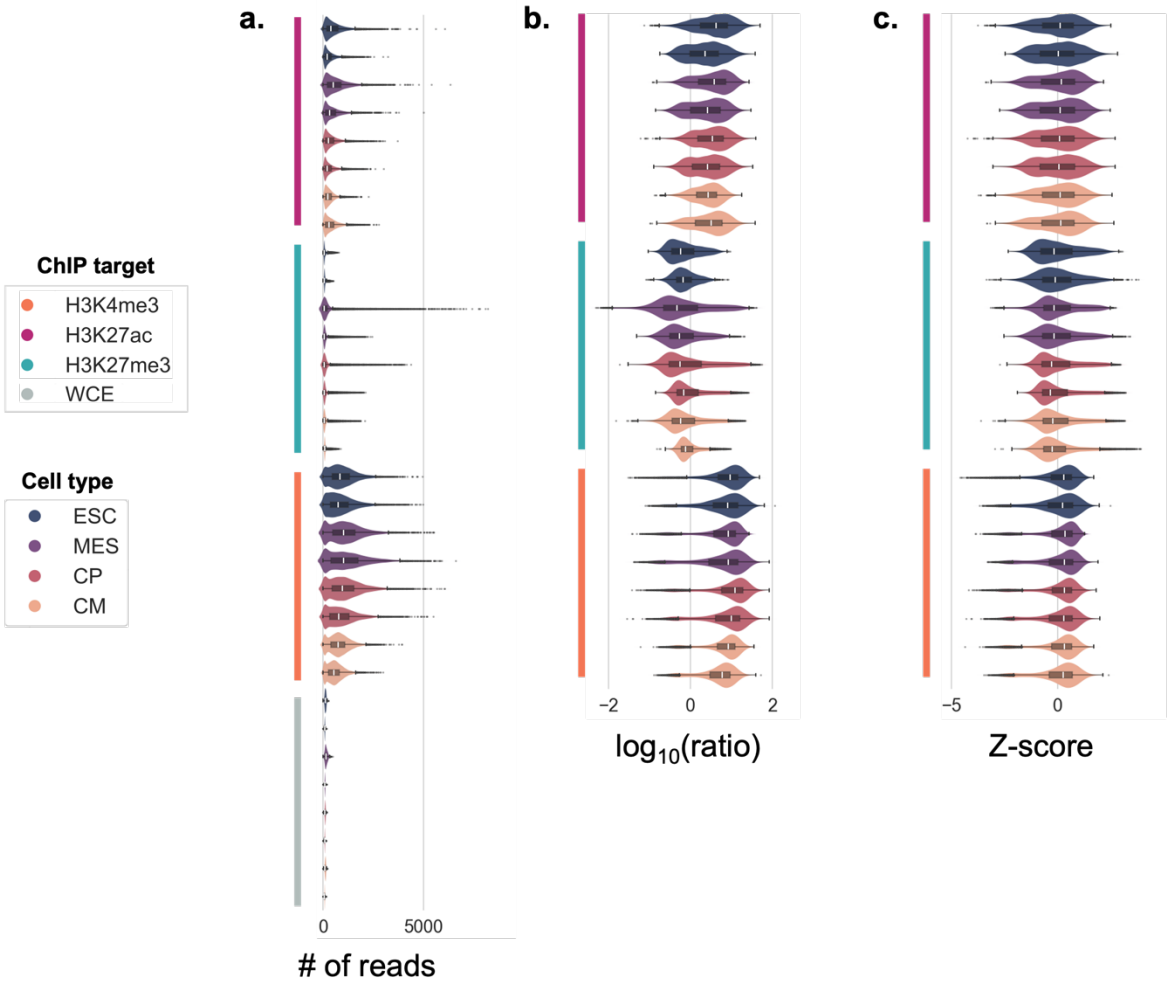

Figure 6. ChIP levels distribution before (a) and after log ratio over the control (WCE) (b) and Z-score normalization (c). Distributions are produced per replicates and grouped by cell type and ChIP target. Notice the WCE which is no longer used after computed the log ratio.

To further analyze the data, we applied Kendall's rank correlation and hierarchical clustering to the datasets, both before and after log ratio normalization relative to the control (Figure S2) (See section 2.1.6). In both heatmaps, we found a positive correlation across all comparisons of H3K4me3 and H3K27ac samples, as expected since both HMs are associated with active chromatin regions. In contrast, all H3K27me3 samples showed negative correlations when compared with the other HMs, consistent with its association with repressive chromatin.

The unsupervised clustering dendrograms revealed that one pair of H3K27me3 replicates in the MES group failed to cluster together both before and after normalization. After

normalization, however, all samples grouped into three distinct clades based on HM (Figure S2b), whereas, prior to normalization, two H3K27ac replicates in the ESC group formed a separate clade (Figure S2a).

We also conducted PCAs on samples grouped by HM, both before and after the normalization steps (Figure S3). In the PCAs following Z-score normalization (Figure S3c), the differentiation trajectory appeared more clearly defined than in the pre-normalization PCAs (Figure S3a, b), and replicate aggregation was notably improved. As anticipated, the H3K27me3 replicates displayed greater dispersion than the other HMs, likely due to the lower antibody affinity associated with this mark.

3.3 DL models optimization and comparison

We generated a dataset by combining FPKM values (n=8), ChIP enrichment levels (n=24), and RNA log-fold change (logFC) values (n=6) as 38 features across 14,996 genes. We assess and compare different dimensionality reduction approaches, including Deep Learning (DL) models, Autoencoder (AE) and Variational Autoencoder (VAE) as well as PCA and Uniform Manifold Approximation and Projection (UMAP), for their effectiveness in compressing and reconstructing gene expression and epigenetic patterns. First, we evaluated the generalization capabilities of each technique, specifically by splitting the data into training, validation, and testing sets (See section 2.2.1); optimize DL model Hyperparameters (HPs) on validation set (See section 2.2.3) and compare reconstruction performances of all methods on unseen testing data (See section 2.2.4). Secondly, we retrained the four models on the full dataset to obtain a compressed and informative representation of all genes, enabling further downstream analysis and reconstruction comparison from another perspective.

3.3.1 Optimal hyperparameters for DL models

HPs for AE and VAE were optimized using a 500-trial random search across a broad hyperparameter space (See section 2.2.3). The best hyperparameters for AE and VAE were nearly identical while in term of reconstruction scores, the VAE achieved a lower Root Mean Squared Error (RMSE) and the same cosine similarity (S_c) than AE on validation set (Table 2 and 3). The best VAE HPs resulted in the PReLU activation function across 3 activation layers, batch normalization enabled between activation layers, a neuron scaling factor of 1.0, batch size of 128 and beta equal to 1×10^{-9} (Figure 7). The best AE HPs were the same apart from a batch size of 64 and 2 activation layers. Similar scores were achieved for training and validation sets indicated no sign of overfitting in both architectures (Figure S6). Examining the

frequency of each HP value tested across 500 trials, revealed nearly a uniform distribution for all hyperparameters in both AE and VAE models, with no single value appearing significantly more often than others (Figure S5).

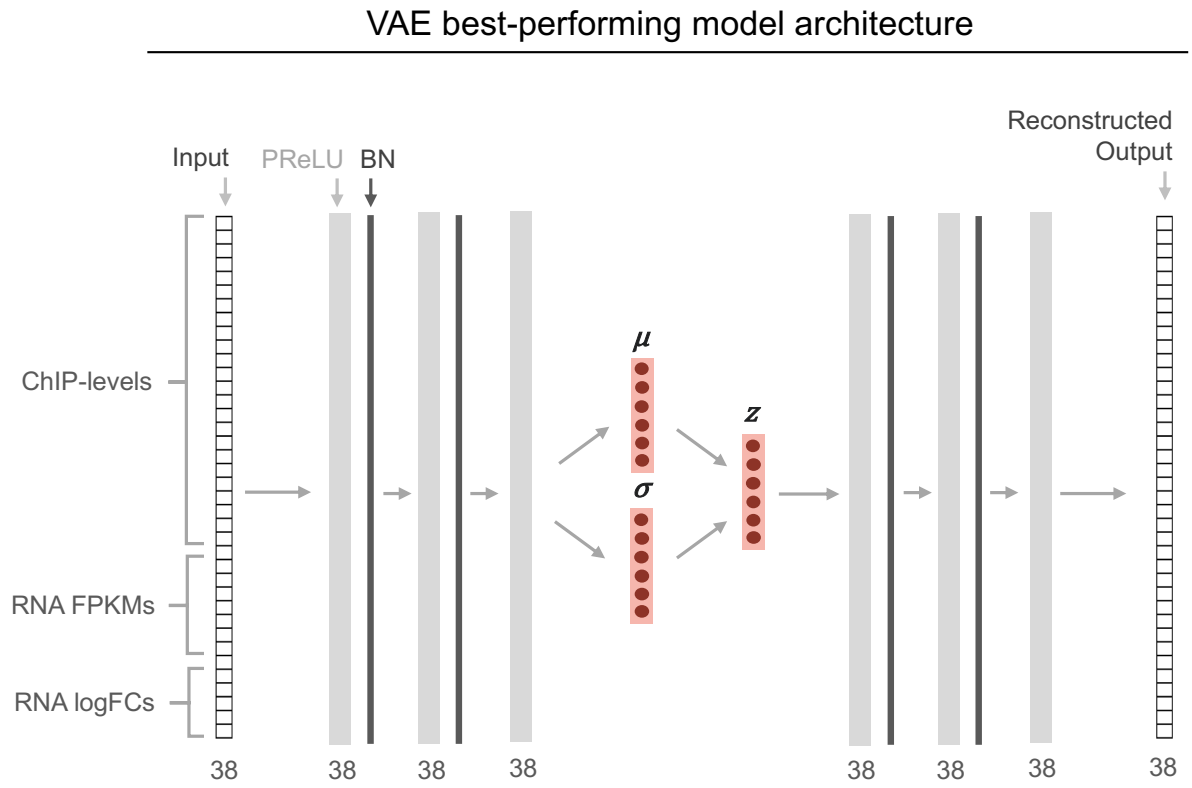

Figure 7. The variational autoencoder architecture with the best hyperparameters is depicted, consisting of three fully connected PReLU activation layers (shown in light grey) with batch normalization (BN) layers (shown in dark grey) positioned between them. The architecture includes a 6-dimensional mean and standard deviation layer, followed by a sampling layer, both highlighted in red.

3.3.2 VAE as best model in compressing and reconstructing features

On testing data, VAE outperformed the three other dimensionality reduction methods, achieving a RMSE of 0.26 [0.15–0.42] and a S_c of 0.94 [0.83–0.99] (Table 2 and 3). When models were trained on the whole dataset, VAE continued to show the lowest reconstruction error (Table 2 and 3 and Figure 8a), confirming its suitability as the best-performing architecture. When comparing PCA and AE, both showed similar averages and percentiles on the testing set and under whole-dataset training conditions, with PCA achieving a higher median score in the latter case. In contrast, UMAP consistently showed the lowest scores across all data subsets. Hence, the ranking of methods as VAE, PCA, AE, and UMAP was consistently observed for both RMSE and Sc (computed on the whole data), with significant

differences in score distributions confirmed by the Wilcoxon Signed-Rank test performed for all comparisons (p-value < 1×10^{-4}).

Table 2. Root Mean Squared Error (RMSE) expressed as average and 5th-95th percentile computed on data subsets and when the models are trained on the whole dataset. Lower values indicate more similarity between input and reconstructed data.

	VAE	AE	PCA	UMAP
Training	0.26 [0.15-0.43]	0.28 [0.16-0.49]	0.29 [0.16-0.48]	0.38 [0.19-0.7]
Validation	0.26 [0.15-0.42]	0.28 [0.16-0.49]	/	/
Testing	0.26 [0.15-0.42]	0.28 [0.15-0.5]	0.29 [0.16-0.48]	0.53 [0.2-1.3]
Whole	0.24 [0.14-0.42]	0.3 [0.17-0.52]	0.29 [0.16-0.48]	0.37 [0.19-0.68]

Table 3. Cosine similarity (S_c) expressed as average and 5th-95th percentile computed on data subsets and when the models are trained on the whole dataset. Higher values indicate more similarity (in term of vector directionality) between input and reconstructed data.

	VAE	AE	PCA	UMAP
Training	0.94 [0.83-0.99]	0.93 [0.81-0.99]	0.93 [0.79-0.99]	0.89 [0.69-0.98]
Validation	0.94 [0.83-0.99]	0.94 [0.83-0.99]	/	/
Testing	0.94 [0.83-0.99]	0.93 [0.81-0.99]	0.93 [0.8-0.99]	0.68 [-0.56-0.98]
Whole	0.95 [0.85-0.99]	0.93 [0.81-0.99]	0.93 [0.79-0.99]	0.89 [0.7-0.98]

In Figure 8b, we showed the training history of the VAE model trained on the entire dataset. The curves revealed a gradual reduction in loss over the training epochs, with a plateau forming toward the final epochs. The S_c curve reached a plateau earlier than the RMSE, reflecting their differences in measuring reconstruction error. Additionally, adjusting automatically the learning rate based on loss improvement allowed for fine-tuning of the parameters, facilitating convergence to an optimal local minimum. To visualize the VAE model's reconstruction capabilities, we generated heatmaps of the original and reconstructed values for a selected set of CT marker genes (Figure 8c)(See section 2.3.2). Comparing these

heatmaps, we observed that the model successfully captured the expression and epigenetic patterns for this subset of genes, with only minor discrepancies. We have examined the gene-wise reconstruction metrics to assess if the selected subset of genes stood out compared to all genes. The subset was randomly distributed along the metric range, rather than concentrated in the tail as the best-reconstructed genes, indicating they did not exhibit exceptional reconstruction quality (Figure S7).

3.4 Coefficient of variation to identify the most variable and stable genes in cardiac differentiation

For each gene in mouse, we computed its coefficient of variation (CV) in expression levels across cardiac differentiation (See section 2.3.1). Based on CV values, we selected the 4,000 genes with the lowest CV as the "stable genes" and the 4,000 with the highest CV as the "variable genes" (Figure 9a). We stratified these groups by the CT with maximum expression (CT_{max}) and for each group, visualized CT_{max} frequencies and FPKM distributions. Stable genes, expected to include consistently and highly expressed genes, showed higher FPKMs and more uniform CT_{max} distribution (Figure 9b), compared to variable genes, which had peak expression concentrated in ESC and CM stages rather than mid-stages (Figure 9c). To functionally characterize these groups, we performed term enrichment analyses. Stable genes were enriched for housekeeping functions, unrelated to specific CTs (e.g., cytoplasmic ribosomal proteins, ribosome, and translation) (Figure 9b). In contrast, variable genes were enriched in pathways and functions specific to cardiac differentiation stages, including cardiac function (e.g., striated muscle contraction, dilated cardiomyopathy), heart development (in WikiPathways and GO Biological Processes), and ESC-related pathways (e.g., Plurinetwork) (Figure 9c). To further characterize the four CTmax groups within the variable genes, we performed term enrichment analysis separately for each group and the resulting terms confirmed the CT specificity of each group (Figure S8).

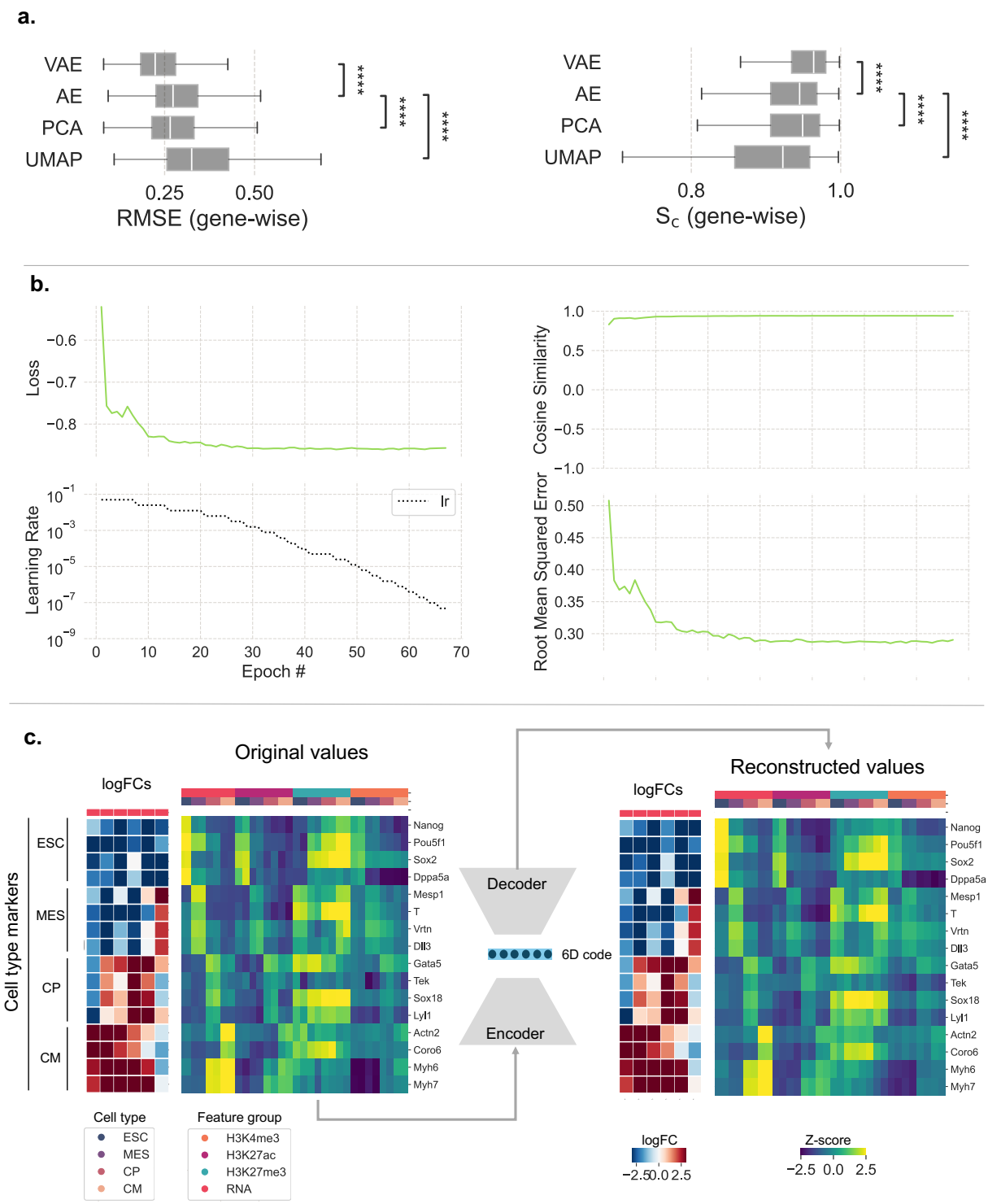

Figure 8. **a.** Reconstruction comparison of different dimensionality reduction methods, trained on the whole data, in term of gene-wise RMSE and Sc scores (**** Wilcoxon's p value < $1x10^{-4}$) **b.** VAE training history recorded at each epoch when training the model on the whole dataset, capturing the loss, dynamic learning rate, and reconstruction scores. **c.** Heatmaps of the original and reconstructed values are displayed for a selected set of CT marker genes. LogFC features (standardized) are displayed by a color map trimmed from -2.5 to 2.5. The logFCs are ordered by the following CT combinations: CM/CP, CM/MES, CM/ESC, CP/MES, CP/ESC, and MES/ESC.

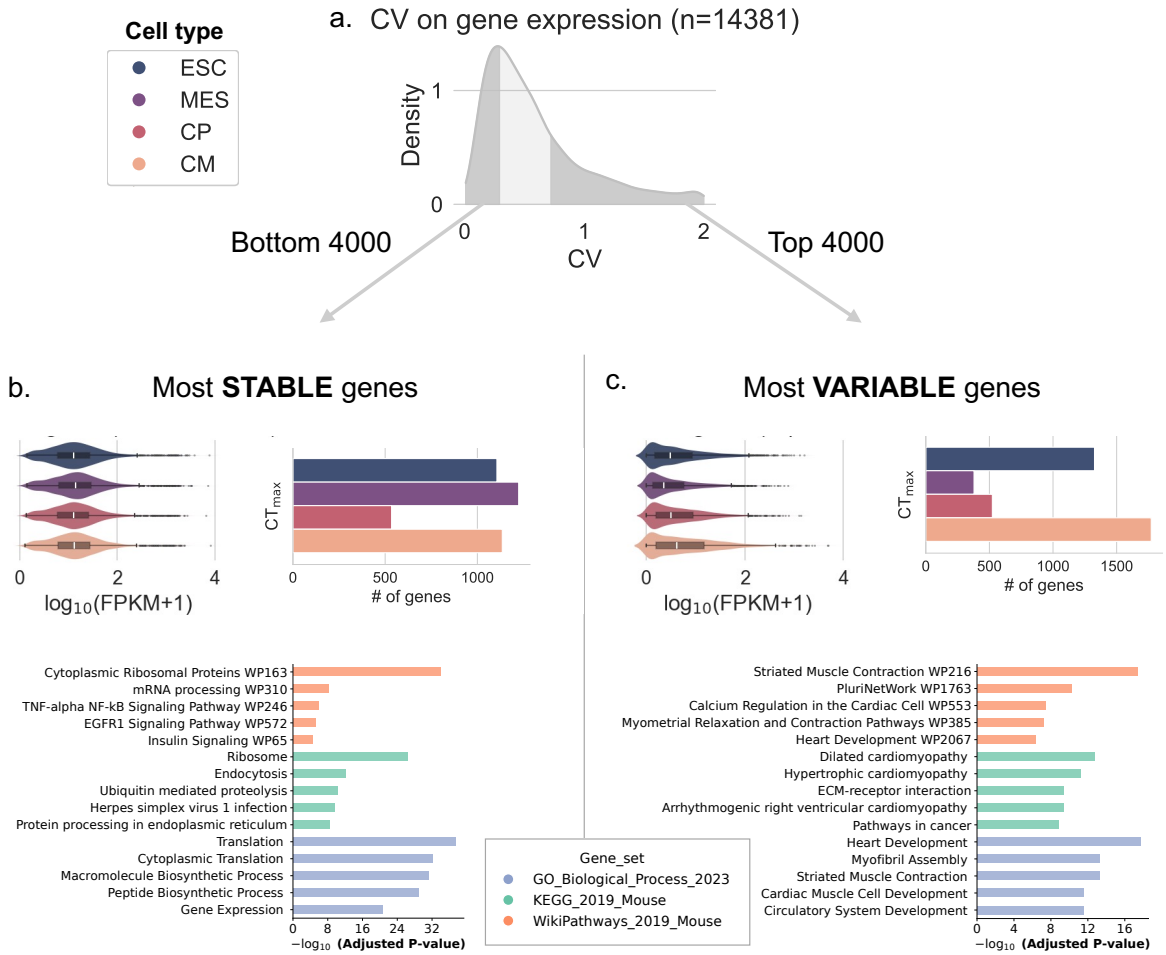

Figure 9. Expression CV distribution showing the most stable and variable genes in the tails of the distribution (**a**). Stable genes (**b**) and variable genes (**c**) FPKM distributions stratified per CT_{max} group, CT_{max} frequencies and term enrichment analysis results (top 5 term per gene set).

3.5 Expression and epigenetic patterns identified by clustering genes on the VAE-based representation

We have used the encoder of the best-performing VAE model as a feature extractor to map genes into a cluster-friendly 6D latent space. By applying unsupervised clustering on it, we aimed to capture novel groups of genes with similar epigenetic patterns and potentially co-regulated during the cardiac differentiation. In addition, by projecting the latent space in 2D with UMAP, we have visualized how genes distribute based on their features and attributes. A scheme depicting the entire workflow is displayed in Figure 10.

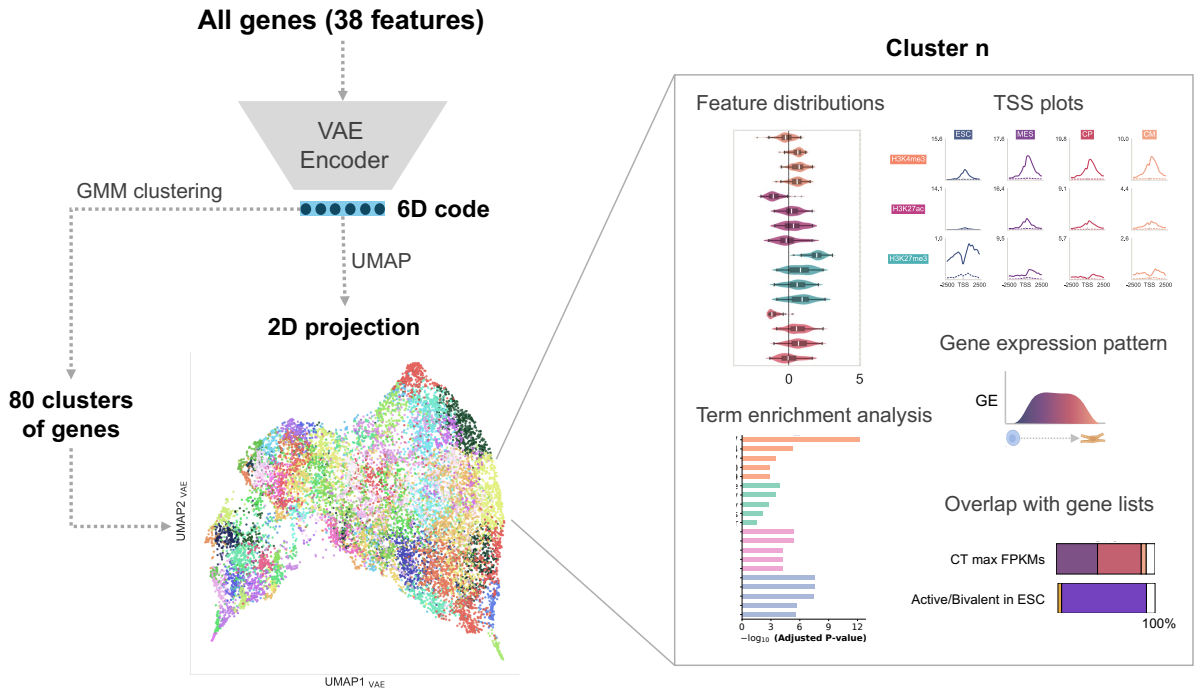

Figure 10. Schema depicting the approach of VAE-based clustering and visualization.

3.5.1 Visualize genes in the latent space

To visualize the latent space distribution of the genes, their corresponding 6D latent vectors μ were projected in 2D by UMAP. By coloring the genes according to their coefficient of variation (CV), calculated from expression values, particular regions in the UMAP (on the left side) that were enriched with genes exhibiting a high CV were revealed (Figure 11b). Cell type marker genes, which are expected to show high variation, were indeed concentrated in these regions (Figure 11a). Examining UMAP plots colored by gene expression levels during differentiation (Figure S9), we observed that genes with the greatest variation were predominantly located on the left side of the UMAP, concordantly with the CV distribution. On the other hand, the right side contained genes with stable expression throughout differentiation, although with differing FPKM levels within the same cell type. This stability was further supported by the low CV on the right side of the UMAP, as well as low fold-changes (Figure S10). In the bottom right region of the UMAP (Figure 11b, Figure S9), a cluster of points with very high expression and low CV likely represented constitutively expressed genes, such as ribosomal or histone genes. Interestingly, a distinct cluster emerged in the bottom left containing genes with moderate expression levels and lacking active chromatin marks (H3K4me3, H3K27ac).

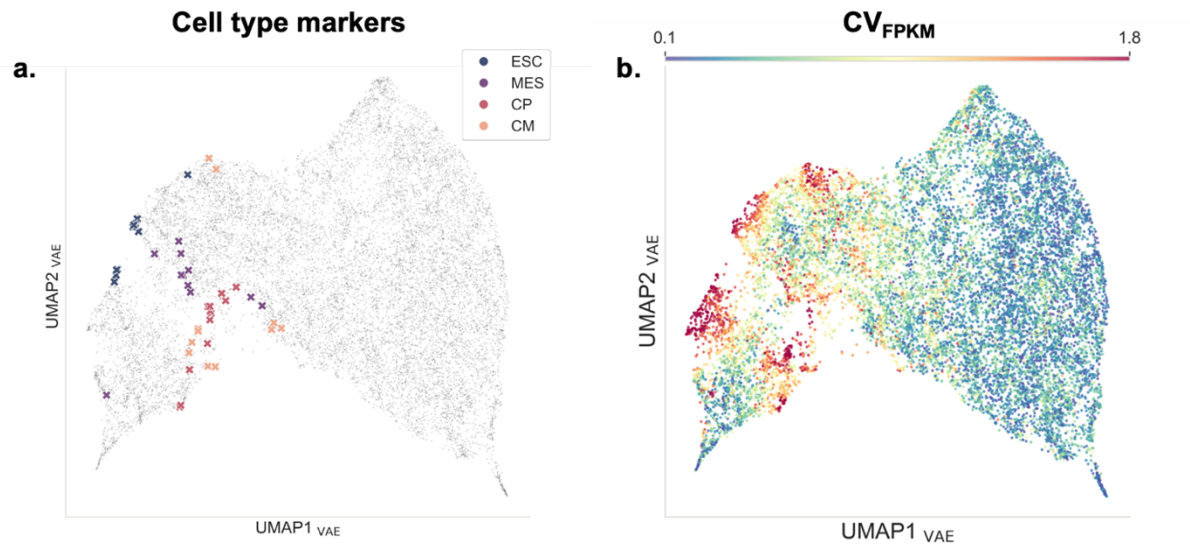

Figure 11. UMAP projection of the genes mapped in the 6-dimensional latent space of the VAE (genes as points) **a.** UMAP with CT marker genes colored according to the CT with maximum expression. **b.** UMAP with genes colored by their expression coefficient of variation across cardiac differentiation (color mapped from the 1st to the 99th percentile).

3.5.2 Genes clustering based on their VAE-based representation

We optimized and trained a Gaussian Mixture Model (GMM) to cluster genes based on their representation encoded in the VAE latent space, resulting in 80 clusters (on average with ~187 genes each). To have a general view of their composition, we computed the percentages of genes in each cluster belonging to some category lists (See section 2.3.4). In Figure 12, we observed that genes were distributed homogeneously by expression coefficient of variation (CV) category across most clusters. Genes primarily expressed in ESC were found in clusters C0, C5, C24, C28, C29, C39, and C77, while those with peak expression in MES clustered in C11, C48, and C83. Genes with maximum expression in CP were concentrated in C17, C42, and C49, whereas genes peaking in CM were primarily located in clusters C1, C32, C33, C68, and C70. Additionally, several clusters primarily contained stable genes (low CV), which likely represent genes with consistent expression levels. This aligned with the expectation that, during cardiac differentiation, the majority of the 14,996 genes would not exhibit significant changes in expression. Certain clusters exhibited a mix of expression peak categories, suggesting that they contained genes with expression trends showing multiple peak time points, such as, clusters like C48, C67, and C76 showing genes with higher expression in MES and CP than in ESC or CM.

In Figure 13, it was possible to observe the distribution of genes in clusters according to their chromatin state previously annotated in mouse ESC (Gonzalez et al., 2021). Several clusters

consisted mostly of bivalent or active genes, while some of them seemed to comprise both categories in comparable proportions. Due to the high number of clusters and resultant plots, an exhaustive analysis is difficult in practice. Moreover, several clusters displayed similar features, making the analyses complex. To overcome this, we heuristically selected some clusters by scanning the clusters composition in terms of expression CV (Figure 12) and chromatin state in ESC (Figure 13), for a more in-depth analysis.

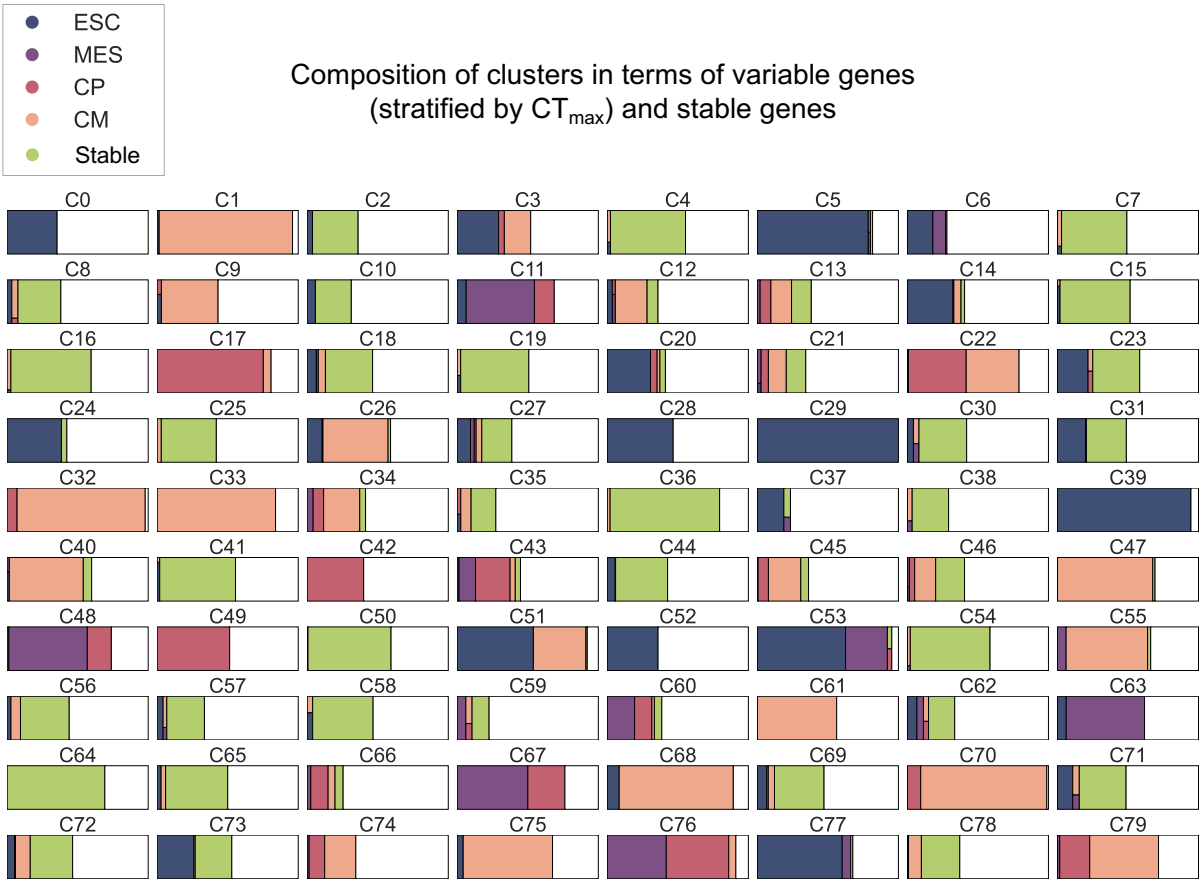

Figure 12. Tree maps depicting the percentage of genes in each cluster defined as most variable (stratified by CT$_{max}$) and most stable in term of expression CV. The max rectangle area corresponds to a percentage of 100%.

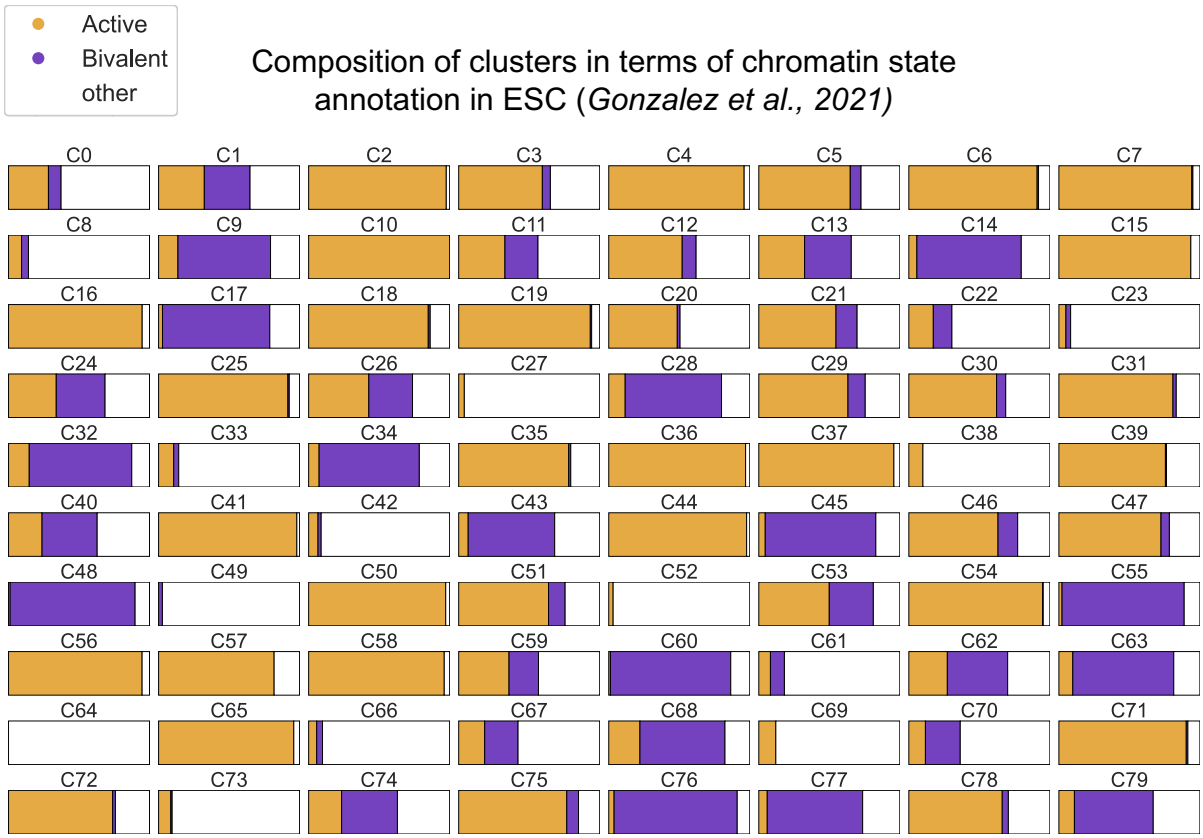

Figure 13 Tree maps depicting the percentage of genes in each cluster annotated with an active, bivalent or other chromatin state in another study (Gonzalez, 2021). The max rectangle area corresponds to a percentage of 100%.

3.5.3 C76: Developmental genes bivalent in ESC

Cluster 76 (C76) consisted of developmental genes marked by low expression in ESC but increasing expression in MES and CP (Figure 14a). In ESCs, these genes exhibited bivalent chromatin marks, with high H3K27me3 and moderate H3K4me3 at transcription start sites (TSSs), a characteristic feature of genes poised for later activation in developmental contexts (Figure 14b). Term enrichment analysis indicated functions in cardiac and neural differentiation, with top pathways associated with heart development and neural crest specification (Figure 18).

The genes in this cluster suggested key roles in early heart and neural development. *Gata4* and *Gata6*, important for various stages of heart development, help drive mesodermal patterning and cardiomyocyte differentiation (Maitra *et al.*, 2009). The Wnt pathway genes *Wnt2* and *Wnt5a* play roles in cardiac morphogenesis and skeletal development (Kwon *et al.*, 2007). *Hoxb2* and *Hoxb3* are critical for anterior posterior axis development in the embryo, with *Hoxb3* influencing neural crest cell differentiation (Tümpel *et al.*, 2009). *Hand1* and *Hand2*

transcription factors are essential for shaping the heart and neural crest-derived structures (George and Firulli, 2019).

To confirm the general expression trends assigned to this cluster, we randomly selected 16 genes from Cluster C76 and inspected their expression patterns during differentiation (Figure S13a). These genes exhibited no expression in ESC, expression peak in MES or CP and were less expressed in CM confirming the general pattern observed in Figure 14a. Examining the first gene in the selection, *Twist1*, in the genome browser it confirmed the cluster's epigenetic pattern, with bivalency in ESC and losing the repression marks in the MES (Figure S14). This gene codes for a transcription factor, critical regulator of valve development in the heart (Chakraborty *et al.*, 2010).

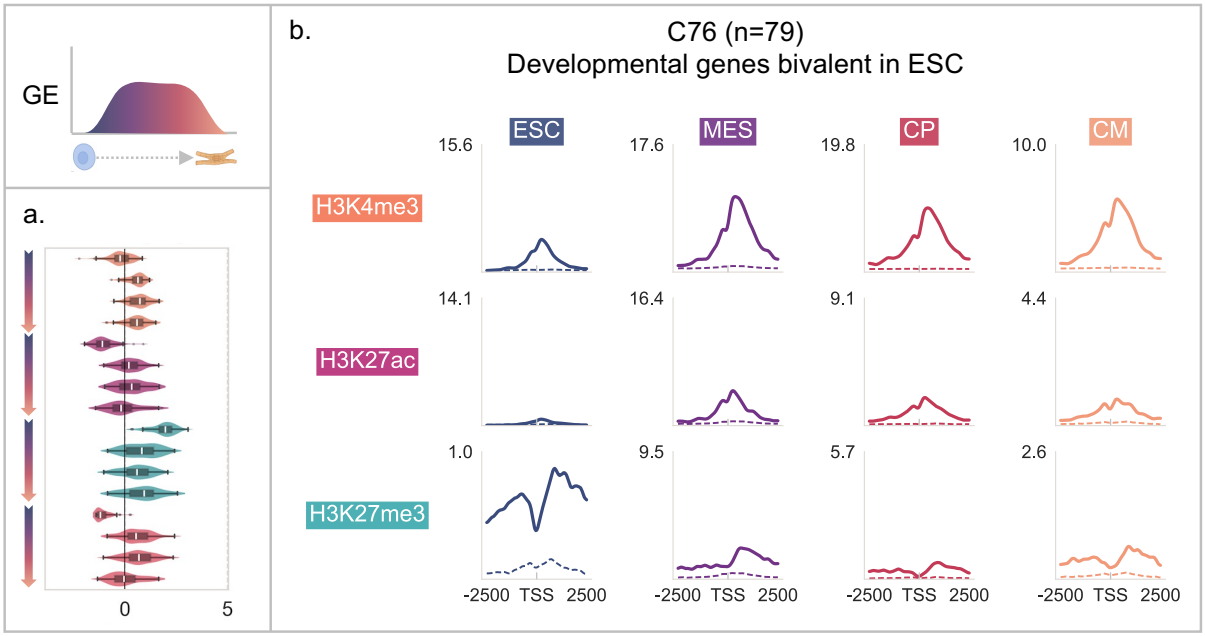

Figure 14. C76: Developmental genes bivalent in ESC. In the top-left, a pictogram illustrates the overall gene expression trends along differentiation. In panel (**a**) features distributions averaged across replicates, are grouped by HMs and gene expression (in red) levels, represented as Z-scores on the x-axis. In panel (**b**), TSS meta-plots display each HM across distinct CTs, with dashed lines representing the corresponding control (Whole Cell Extract) signals for each CT. The y-axis maximum is consistent among clusters, set to the maximum y-value among all clusters for each HM and CT combination.

3.5.4 C70: Silenced genes activated in CM that are non-targets of Polycomb

Cluster C70 consisted in genes with low expression in early cardiac differentiation stages, lacking H3K27me3 marks throughout. H3K4me3 and H3K27ac marks increased from CP to CM, corresponding with higher expression. The absence of H3K27me3 in ESC and MES

suggested that these genes were not regulated by Polycomb complexes but might be repressed through other mechanisms.

The overlap with active and bivalent genes showed that only a small subset of this cluster was annotated as bivalent or active in ESCs in Gonzalez et al. data (Figure 13). Term enrichment analysis highlighted associations with striated muscle contraction and other cardiac functions, concordantly to their higher expression in CM (Figure 18).

Key genes in this cluster related to heart function include *Myh6* and *Myh7*, which encode cardiac myosin heavy chains, and *Myl2*, *Myl3*, *Myl4*, and *Myl7*, which encode light chain subunits with distinct roles in atrial and ventricular function (Lu *et al.*, 2022; Sitbon *et al.*, 2020), and *Tnnt2* and *Tnni3*, encoding components of the cardiac troponin complex (Joyce *et al.*, 2023).

To validate the general expression pattern, we examined 16 randomly selected genes, all of which displayed the overall cluster trend of no expression in ESC and MES, with increased expression in CP and CM (Figure S13b). UCSC genome browser tracks for genes like *Myh7* and *Myh6* (located in tandem) confirmed the absence of the bivalent marks in early stages (Figure S15), supported by Gonzalez's annotation, indicating they are not bivalent in ESCs.

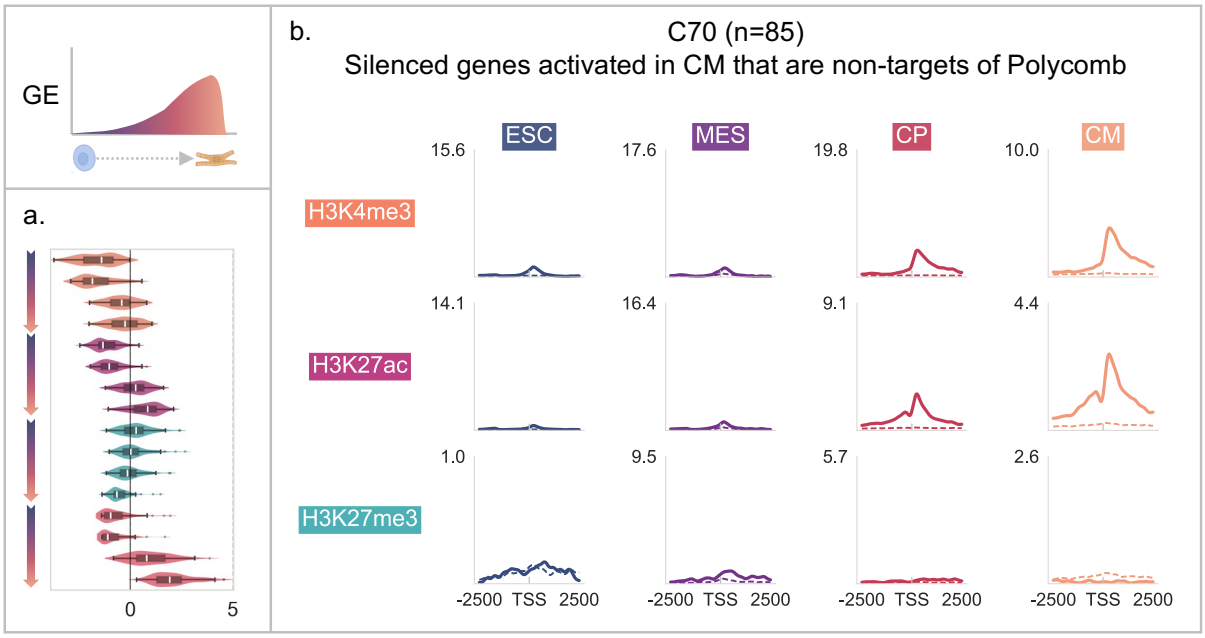

Figure 15. C70: Silenced genes activated in CM that are non-targets of Polycomb. General plot information is consistent with those described in Figure 14.

3.5.5 C48: Bivalent throughout the differentiation and expressed in mid-stages

Cluster C48 contained genes that appeared to remain bivalent throughout the differentiation process but showed expression in both MES and CP stages (Figure 16). The genes in this cluster were associated with neurodevelopmental processes, as indicated by the top terms in the enrichment analysis (Figure 18). However, the presence of heart development as fifth top term in WikiPathways suggested a role also in cardiac development.

Some example genes in C48 with neurodevelopmental associations included: *Alx4* and *Msx1*, required for osteogenesis in the cranial neural crest (Han *et al.*, 2007), and *Foxa1* and *Foxa2* which regulate gene networks in multiple organ systems, including differentiation of midbrain dopaminergic neurons (Ang, 2009).

From the random selection of 16 genes, a similar expression pattern emerged across the cluster, with moderate expression levels in the MES and CP stages (Figure S13c). When examining individual genes, distinct epigenetic patterns emerged. *Alx4*, *Foxa1* and *Foxa2* were low expressed in mid-stages, while also being covered by H3K27me3 marks (Figure S16-18). In contrast, *Vrtn* deviated from the overall trend by lacking the H3K27me3 mark at mid-stages (Figure S19).

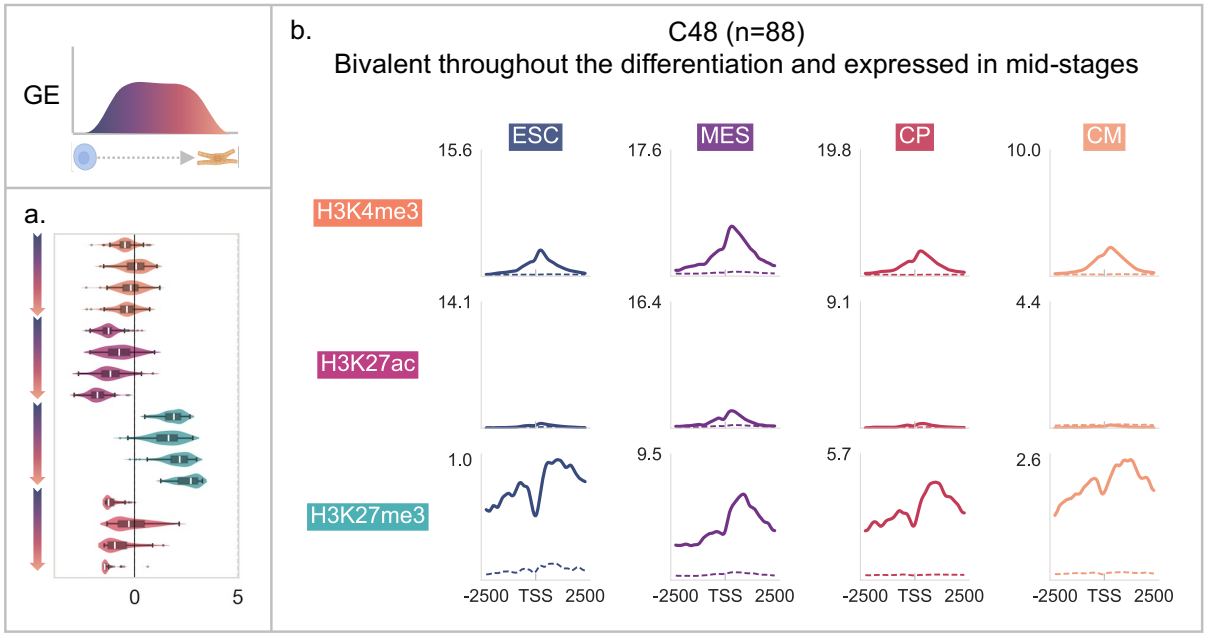

Figure 16. C48: Bivalent throughout the differentiation and expressed in mid-stages. General plot information is consistent with those described in Figure 14

3.5.6 C51: Valley-like expression pattern, bivalent in mid-stages

C51 was characterized by a distinctive "valley-like" pattern in gene expression, with high expression levels in the ESC and CM stages and a markedly reduced or nearly absent expression in the mid-stages (Figure 17).

The temporal dynamics of histone modifications supported this valley-shaped expression pattern. The repressive mark H3K27me3 was absent in ESC, peaked in MES and CP stages, and declined again in CM, suggesting a bivalent state in MES and CP. In contrast, the active marks H3K4me3 and H3K27ac showed on average higher peaks in ESC and CM, concordantly with the expression peaks at these stages (Figure 17). This alternating pattern of histone modifications indicated a shift from an active state in ESC to a bivalent state in the mid-stages, followed by reactivation in CM, depicting the "valley-like" expression profile.

Term enrichment analysis identified "plurinetwork" as the top enriched term in WikiPathways for C51 (Figure 18), with ten out of 102 genes belonging to this pathway: *Bcam*, *Klf5*, *Igfbp3*, *Perp*, *Esrrb*, *Pim1*, *Tfeb*, *Klf2*, *Icam1*, and *Tle5*.

Among the sixteen randomly selected genes in this cluster, all but one displayed the valley-like expression pattern (Figure S13). Examples illustrating this pattern included *Adgre5* and *Pim1*, both of which showed peaks of H3K27me3 only in MES or CP, with H3K4me3 persisting along the differentiation, confirming a transient bivalent state during these stages, as observed in the genome browser (Figure S20, S21). *Pim1* is notable for its role in the "plurinetwork", where it promotes cardiomyocyte survival by upregulating c-Kit protein expression (Ebeid *et al.*, 2020). However, *Pura* deviates slightly from this epigenetic pattern, being bivalent throughout differentiation despite its expression following a valley-like pattern (Figure S22).

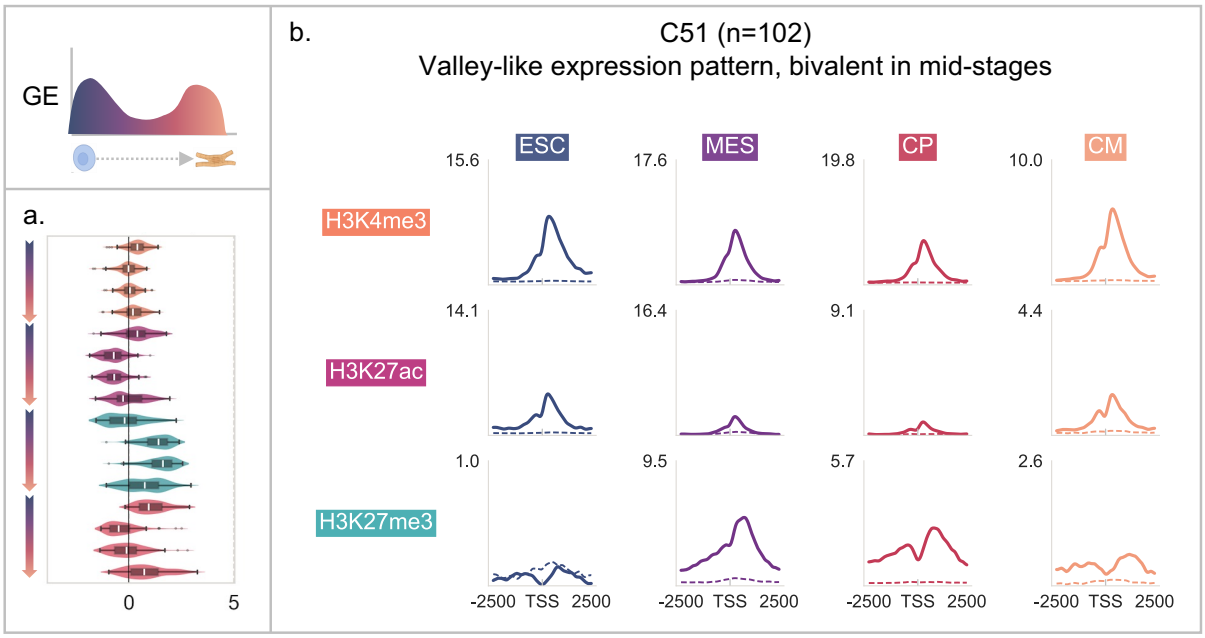

Figure 17. C51: Valley-like expression pattern, bivalent in mid-stages. General plot information is consistent with those described in Figure 14

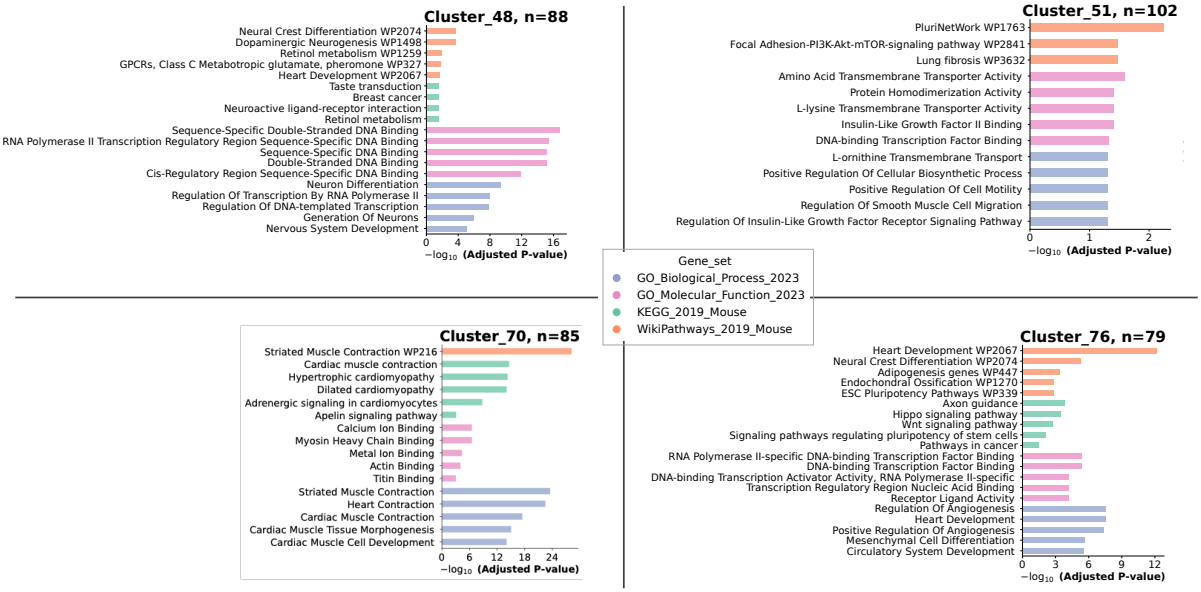

Figure 18. Term enrichment results for the selected clusters. Only the top 5 terms are shown for each gene set tested.

4 Discussion

The rapid advancement of next-generation sequencing (NGS) has produced vast data volumes, highlighting the need for efficient computational methods to extract novel and meaningful biological insights. Unsupervised learning for omics data analysis has benefited from Deep Learning (DL), with Autoencoders (AEs) being particularly effective (Kang *et al.*, 2022; Li *et al.*, 2023). Among them, Variational Autoencoders (VAEs) have been used successfully to integrate multiomics data in both bulk and single-cell analyses (See section 1.5). Once the AE model is trained, the encoder component serves as a feature extractor, mapping the input in a non-linear and lower dimensional space (latent space). Additionally, the use of multiple hidden layers in DL enhances this process, allowing for more effective capture of complex relationships within the data. Studies indicate that clustering algorithms perform better when using these DL-based feature mappings as input instead of raw data (Karim *et al.*, 2020; Viaud *et al.*, 2022).

Research has extensively mapped the epigenome along stem cell differentiation into cardiomyocytes (Wamstad *et al.*, 2012; Paige *et al.*, 2012) and normal heart development

(Nord *et al.*, 2013; Gilsbach *et al.*, 2018) reveal a highly dynamic epigenome. The precise timing of gene activation and repression is critical and disruptions can lead to congenital heart disease (CHD), underscoring the importance of transcriptional and epigenetic regulation (Akerberg and Pu, 2020).

Here, we tested VAEs, AEs and other dimensionality reduction techniques for their ability to compress and reconstruct experimental features collected along cardiac differentiation (FPKM values, ChIP levels, and RNA log-fold changes), and then we clustered genes based on their VAE-based representation. To visualize gene distribution in the VAE latent space, 6D latent vectors were projected into 2D using UMAP. The UMAP plot colored by various features revealed that genes with similar expression dynamics or chromatin states tended to cluster together, indicating that the latent space effectively captures meaningful biological relationships.

By selecting some clusters, we showed how VAE latent code successfully grouped together genes based on common features dynamics and biological functions. For instance, cluster 76 includes genes involved in regulation of heart development, such as *Gata4*, *Gata6*, *Hand1*, and *Hand2*, which exhibited a bivalent state in stem cells and get switched on in the mesoderm. A similar expression pattern is observed also in cluster 48, with a peak in MES, however, genes in this cluster shows a distinct epigenetic dynamic, with H3K27me3 and H3K4me3 present during all the differentiation. Interestingly, term enrichment analysis revealed an enrichment of genes associated to neuronal differentiation processes. Cluster 70 was an example of genes silenced in early stages and activated in the latter phases, important in heart function. These genes appeared to be not repressed by Polycomb, since they lacked the repression mark in early stages. Lastly, we detected a cluster (51) containing genes following a peculiar epigenetic trend, passing from an active state in ESC to a bivalent in mid-stages and being reactivated in CM, mirrored by expression peaks at differentiation extremes and low or absent expression in MES or CP. In conclusion, the analysis of some clusters revealed novel and complex gene regulation patterns during cardiac differentiation, including transitions from active to bivalent states and reactivation in later stages.

When comparing VAE, AE, Principal Component Analysis (PCA), and Uniform Manifold Approximation and Projection (UMAP) for data compression and reconstruction, the VAE consistently outperformed the other methods, both when trained and evaluated on the whole dataset and when trained on a subset and tested on unseen data. Notably, PCA achieved scores comparable to AE, despite its simplicity and lack of hyperparameter optimization. However, PCA showed lower average, and broader distributions in both reconstruction scores, compared to VAE. Lastly, UMAP consistently delivered the poorest reconstruction

performance. The inverse transformation (reconstruction) in UMAP is possible and easy to implement, however, its runtime scales exponentially with the number of components, requiring approximately 30 minutes to compute a reconstructing function from the embedding in 6 components. This long runtime made hyperparameters search and optimization infeasible. Additionally, unlike AEs and VAEs, UMAP is not designed with reconstruction as a primary objective, which limits its ability to accurately map data back from the embedding space to the original feature space. However, without labeled data, we could only rely on reconstruction performance as an indirect measure (proxy) of embedding quality and used it to evaluate and compare methods. If ground truth labels were available, clustering accuracy metrics such as Adjusted Rand Index (ARI) or Normalized Mutual Information (NMI) would offer a more direct evaluation of the models' abilities to preserve meaningful groupings and biological patterns in the latent space (or embedding).

Interestingly, incorporating fold-changes as features alongside FPKMs improved reconstruction performance. One explanation could be that while FPKMs captures absolute gene expression levels, fold-changes complements it by highlighting relative expression changes across cell types, even though they lack information on absolute quantities.

While analyzing clusters, we observed some “out-of-trend” genes, such as *Vrtn* in Cluster 48 and *Pura* in Cluster 51. The anomalous inclusion of these genes suggests possible heterogeneity within clusters, potentially due to inaccuracies in the VAE encoding or in the GMM clustering separation. A potential solution could be implementing joint training of the VAE and clustering module by incorporating clustering loss alongside VAE loss, optimizing the latent space geometry for cluster separation (Karim *et al.*, 2020). Hence, more sophisticated and more challenging to implement, deep clustering methods could be explored, such as, Deep Embedded Clustering (DEC) which employ a pretraining of the AE and fine-tuning adding the clustering module and loss (Xie *et al.*, 2016).

To build on the findings of this study, several potential improvements could enhance the analysis pipeline and broaden its applicability. Extending the feature set to include HM levels at enhancer regions would provide a more detailed view of gene regulatory dynamics, as enhancers are critical in modulating gene expression during differentiation. In such a framework, genomic bins could be the input data points of the VAE instead of genes. Testing the framework with more diverse biological datasets could further exploit the feature compression capabilities of the model. For instance, applying this method to datasets with multiple experimental conditions (e.g., healthy vs. cancerous tissue, wild-type vs. knockout models), both spatial and temporal dimensions or incorporating additional layer of genome-wide information, such as chromatin accessibility, DNA methylation, more histone marks, or

transcription factor binding data (Mora *et al.*, 2022). Also, more sophisticated approaches to feature extraction in ChIP-seq data could improve the modelling of HM levels. For example, developing metrics that capture peak shape or chromatin context (Hentges *et al.*, 2022; Oh *et al.*, 2020) could provide a more nuanced representation of HM presence. Implementing an automated software package could facilitate this multi-step analysis, from data preprocessing and feature extraction to model training and clustering, providing an end-to-end pipeline. Such a package would not only reduce manual intervention but also make this framework more accessible for other researchers, facilitating reproducibility and adaptation in various studies. To sum up, this study presents a proof of concept for a VAE-based clustering pipeline exploiting the VAE to integrate gene expression and epigenetic dynamics in a unified latent space, on which applying clustering. Selected clusters demonstrated the framework ability to detect unexpected group of genes with common dynamic features and biological functions, indicating its potential for applications in other biological scenarios.

5 Bibliography

Abadi,M. *et al.* (2016) TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

Agrawal,A. *et al.* (2024) WikiPathways 2024: next generation pathway database. *Nucleic Acids Research*, **52**, D679–D689.

Akerberg,B.N. and Pu,W.T. (2020) Genetic and Epigenetic Control of Heart Development. *Cold Spring Harbor Perspectives in Biology*, **12**, a036756.

Amemiya,H.M. *et al.* (2019) The ENCODE Blacklist: Identification of Problematic Regions of the Genome. *Sci Rep*, **9**, 9354.

Ang,S.-L. (2009) Foxa1 and Foxa2 Transcription Factors Regulate Differentiation of Midbrain Dopaminergic Neurons. In, Pasterkamp,R.J. *et al.* (eds), *Development and Engineering of Dopamine Neurons*. Springer, New York, NY, pp. 58–65.

Aranda,S. *et al.* (2015) Regulation of gene transcription by Polycomb proteins. *Science Advances*, **1**, e1500737.

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. *Nat Genet*, **25**, 25–29.

Ballard,D.H. (1987) Modular learning in neural networks. In, *Proceedings of the sixth National conference on Artificial intelligence - Volume 1*, AAAI'87. AAAI Press, Seattle, Washington, pp. 279–284.

Bernstein,B.E. *et al.* (2006) A Bivalent Chromatin Structure Marks Key Developmental Genes in Embryonic Stem Cells. *Cell*, **125**, 315–326.

Blanco,E. *et al.* (2021) Productive visualization of high-throughput sequencing data using the SeqCode open portable platform. *Sci Rep*, **11**, 19545.

Blanco,E. *et al.* (2020) The Bivalent Genome: Characterization, Structure, and Regulation. *Trends in Genetics*, **36**, 118–131.

Bruneau,B.G. (2013) Signaling and Transcriptional Networks in Heart Development and Regeneration. *Cold Spring Harb Perspect Biol*, **5**, a008292.

Chakraborty,S. *et al.* (2010) Twist1 promotes heart valve cell proliferation and extracellular matrix gene expression during development in vivo and is expressed in human diseased aortic valves. *Developmental biology*, **347**, 167.

Chollet,F. (2015) Keras: Deep Learning for humans.

Church,D.M. *et al.* (2011) Modernizing Reference Genome Assemblies. *PLoS Biology*, **9**, e1001091.

Clough,E. *et al.* (2024) NCBI GEO: archive for gene expression and epigenomics data sets: 23-year update. *Nucleic Acids Research*, **52**, D138–D144.

Danecek,P. *et al.* (2021) Twelve years of SAMtools and BCFtools. *GigaScience*, **10**, giab008.

Delgado-Olguín,P. *et al.* (2012) Epigenetic repression of cardiac progenitor gene expression by Ezh2 is required for postnatal cardiac homeostasis. *Nat Genet*, **44**, 343–347.

Ebeid,D.E. *et al.* (2020) PIM1 Promotes Survival of Cardiomyocytes by Upregulating c-Kit Protein Expression. *Cells*, **9**, 2001.

Eltager,M. *et al.* (2023) Benchmarking variational AutoEncoders on cancer transcriptomics data. *PLOS ONE*, **18**, e0292126.

Emrich,S.J. *et al.* (2007) Gene discovery and annotation using LCM-454 transcriptome sequencing. *Genome Research*, **17**, 69.

Eraslan,G. *et al.* (2019) Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun*, **10**, 390.

Furey,T.S. (2012) ChIP–seq and beyond: new and improved methodologies to detect and characterize protein–DNA interactions. *Nat Rev Genet*, **13**, 840–852.

Fyodorov,D.V. *et al.* (2018) Emerging roles of linker histones in regulating chromatin structure and function. *Nat Rev Mol Cell Biol*, **19**, 192–206.

George,R.M. and Firulli,A.B. (2019) Hand Factors in Cardiac Development. *The Anatomical Record*, **302**, 101–107.

Gilsbach,R. *et al.* (2018) Distinct epigenetic programs regulate cardiac myocyte development and disease in the human heart in vivo. *Nat Commun*, **9**, 391.

González-Ramírez,M. *et al.* (2021) Differential contribution to gene expression prediction of histone modifications at enhancers or promoters. *PLOS Computational Biology*, **17**, e1009368.

Grønbech,C.H. *et al.* (2020) scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics*, **36**, 4415–4422.

Han,J. *et al.* (2007) Concerted action of *Msx1* and *Msx2* in regulating cranial neural crest cell differentiation during frontal bone development. *Mechanisms of Development*, **124**, 729–745.

He,A. *et al.* (2012) PRC2 directly methylates GATA4 and represses its transcriptional activity. *Genes Dev.*, **26**, 37–42.

Hentges,L.D. *et al.* (2022) LanceOtron: a deep learning peak caller for genome sequencing experiments. *Bioinformatics*, **38**, 4255–4263.

Higgins,I. *et al.* (2017) beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework.

Hotelling,H. (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**, 417–441.

Hu,R. *et al.* (2021) Decoding regulatory structures and features from epigenomics profiles: A Roadmap-ENCODE Variational Auto-Encoder (RE-VAE) model. *Methods*, **189**, 44–53.

Hunter,J.D. (2007) Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, **9**, 90–95.

Joyce,W. *et al.* (2023) A Revised Perspective on the Evolution of Troponin I and Troponin T Gene Families in Vertebrates. *Genome Biology and Evolution*, **15**, evac173.

Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, **28**, 27–30.

Kang,M. *et al.* (2022) A roadmap for multi-omics data integration using deep learning. *Briefings in Bioinformatics*, **23**, bbab454.

Karim,M.R. *et al.* (2020) Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, **22**, 393.

Kent,W.J. *et al.* (2002) The Human Genome Browser at UCSC. *Genome Res.*, **12**, 996–1006.

Kim,D. *et al.* (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology*, **14**, R36.

Kingma,D.P. and Ba,J. (2017) Adam: A Method for Stochastic Optimization.

Kingma,D.P. and Welling,M. (2013) Auto-Encoding Variational Bayes.

Kwon,C. *et al.* (2007) Canonical Wnt signaling is a positive regulator of mammalian cardiac progenitors. *Proceedings of the National Academy of Sciences*, **104**, 10894–10899.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat Methods*, **9**, 357–359.

Leinonen,R. *et al.* (2010) The European Nucleotide Archive. *Nucleic Acids Research*, **39**, D28.

Leinonen,R. *et al.* (2011) The Sequence Read Archive. *Nucleic Acids Res*, **39**, D19–D21.

Li,G. and Reinberg,D. (2011) Chromatin higher-order structures and gene regulation. *Current Opinion in Genetics & Development*, **21**, 175–186.

Li,Z. *et al.* (2023) Applications of deep learning in understanding gene regulation. *Cell Reports Methods*, **3**, 100384.

Lister,R. *et al.* (2008) Highly Integrated Single-Base Resolution Maps of the Epigenome in Arabidopsis. *Cell*, **133**, 523.

Lopez,R. *et al.* (2018) Deep generative modeling for single-cell transcriptomics. *Nat Methods*, **15**, 1053–1058.

Lotfollahi,M. *et al.* (2019) scGen predicts single-cell perturbation responses. *Nat Methods*, **16**, 715–721.

Love,M.I. *et al.* (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, **15**, 550.

Lu,P. *et al.* (2022) Cardiac Myosin Heavy Chain Reporter Mice to Study Heart Development and Disease. *Circulation Research*, **131**, 364–366.

Luger,K. *et al.* (1997) Crystal structure of the nucleosome core particle at 2.8 Å resolution. *Nature*, **389**, 251–260.

Maaten,L. van der and Hinton,G. (2008) Visualizing Data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605.

Maitra,M. *et al.* (2009) Interaction of *Gata4* and *Gata6* with *Tbx5* is critical for normal cardiac development. *Developmental Biology*, **326**, 368–377.

Martire,S. and Banaszynski,L.A. (2020) The roles of histone variants in fine-tuning chromatin organization and function. *Nat Rev Mol Cell Biol*, **21**, 522–541.

Mas,G. *et al.* (2018) Promoter bivalency favors an open chromatin architecture in embryonic stem cells. *Nat Genet*, **50**, 1452–1462.

McInnes,L. *et al.* (2018) UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, **3**, 861.

Millán-Zambrano,G. *et al.* (2022) Histone post-translational modifications — cause and consequence of genome function. *Nat Rev Genet*, **23**, 563–580.

Miller,S.A. *et al.* (2010) Jmjd3 and UTX Play a Demethylase-Independent Role in Chromatin Remodeling to Regulate T-Box Family Member-Dependent Gene Expression. *Molecular Cell*, **40**, 594–605.

Moore,J.E. *et al.* (2020) Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature*, **583**, 699–710.

Mora,A. *et al.* (2022) Variational autoencoding of gene landscapes during mouse CNS development uncovers layered roles of Polycomb Repressor Complex 2. *Nucleic Acids Research*, **50**, 1280–1296.

Nord,A.S. *et al.* (2013) Rapid and Pervasive Changes in Genome-wide Enhancer Usage during Mammalian Development. *Cell*, **155**, 1521–1531.

Oh,D. *et al.* (2020) CNN-Peaks: ChIP-Seq peak detection pipeline using convolutional neural networks that imitate human visual inspection. *Sci Rep*, **10**, 7933.

Paige,S.L. *et al.* (2012) A Temporal Chromatin Signature in Human Embryonic Stem Cells Identifies Regulators of Cardiac Development. *Cell*, **151**, 221.

Park,P.J. (2009) ChIP–seq: advantages and challenges of a maturing technology. *Nat Rev Genet*, **10**, 669–680.

Pasini,D. *et al.* (2007) The Polycomb Group Protein Suz12 Is Required for Embryonic Stem Cell Differentiation. *Molecular and Cellular Biology*, **27**, 3769.

Pearson,K. (1901) LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2**, 559–572.

Pedregosa,F. *et al.* (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Piwecka,M. *et al.* (2023) Single-cell and spatial transcriptomics: deciphering brain complexity in health and disease. *Nat Rev Neurol*, **19**, 346–362.

Pruitt,K.D. *et al.* (2014) RefSeq: an update on mammalian reference sequences. *Nucleic Acids Research*, **42**, D756–D763.

Reynolds,D. (2009) Gaussian Mixture Models. In, Li,S.Z. and Jain,A. (eds), *Encyclopedia of Biometrics*. Springer US, Boston, MA, pp. 659–663.

Schuettengruber,B. *et al.* (2017) Genome Regulation by Polycomb and Trithorax: 70 Years and Counting. *Cell*, **171**, 34–57.

Shen,H. and Laird,P.W. (2013) Interplay between the Cancer Genome and Epigenome. *Cell*, **153**, 38–55.

Sitbon,Y.H. *et al.* (2020) Insights into myosin regulatory and essential light chains: a focus on their roles in cardiac and skeletal muscle function, development and disease. *J Muscle Res Cell Motil*, **41**, 313–327.

Ståhl,P.L. *et al.* (2016) Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, **353**, 78–82.

Stark,R. *et al.* (2019) RNA sequencing: the teenage years. *Nat Rev Genet*, **20**, 631–656.

Tang,F. *et al.* (2009) mRNA-Seq whole-transcriptome analysis of a single cell. *Nat Methods*, **6**, 377–382.

The Gene Ontology Consortium *et al.* (2023) The Gene Ontology knowledgebase in 2023. *Genetics*, **224**, iyad031.

Tümpel,S. *et al.* (2009) Chapter 8 *Hox* Genes and Segmentation of the Vertebrate Hindbrain. In, *Current Topics in Developmental Biology*, Genes. Academic Press, pp. 103–137.

Viaud,G. *et al.* (2022) Representation Learning for the Clustering of Multi-Omics Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **19**, 135–145.

Virtanen,P. *et al.* (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*, **17**, 261–272.

Wamstad,J.A. *et al.* (2012) Dynamic and Coordinated Epigenetic Regulation of Developmental Transitions in the Cardiac Lineage. *Cell*, **151**, 206–220.

Wang,Z. and Wang,Y. (2019) Extracting a biologically latent space of lung cancer epigenetics with variational autoencoders. *BMC Bioinformatics*, **20**, 568.

Waskom,M.L. (2021) seaborn: statistical data visualization. *Journal of Open Source Software*, **6**, 3021.

Way,G.P. and Greene,C.S. (2018) Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, **23**, 80.

Xie,J. *et al.* (2016) Unsupervised Deep Embedding for Clustering Analysis. In, *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, pp. 478–487.

6 Supplementary figures

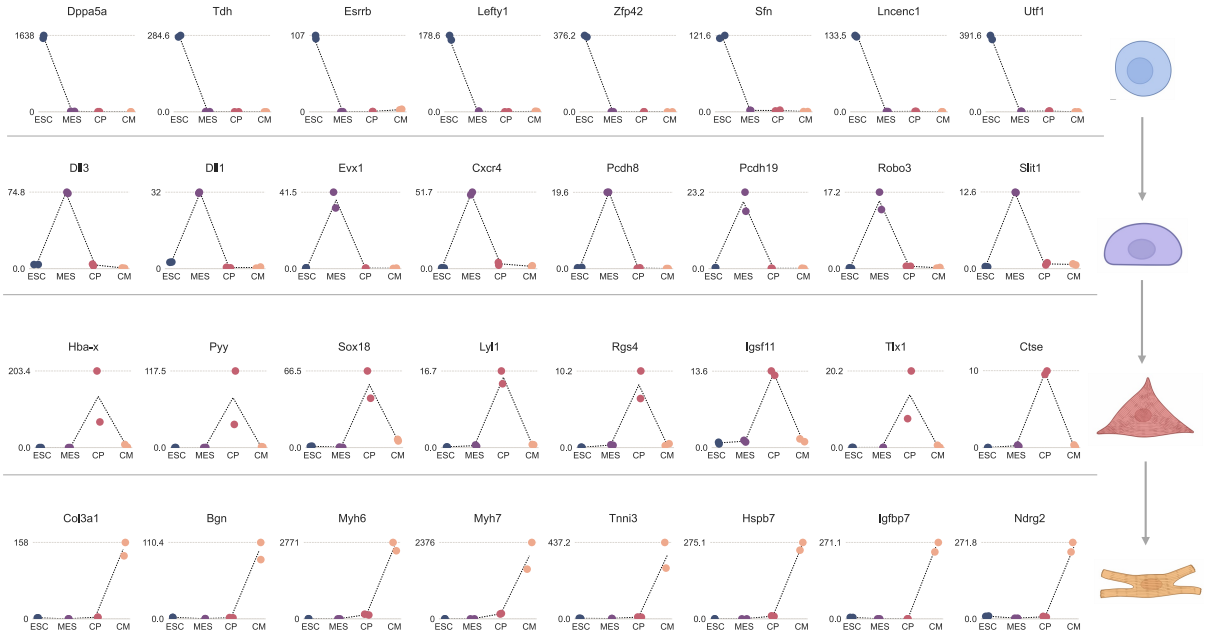

Figure S1. Gene expression profile of additional marker genes selected across cardiac differentiation stages (CT). Eight marker genes for each CT (per line) selected based on logFCs.

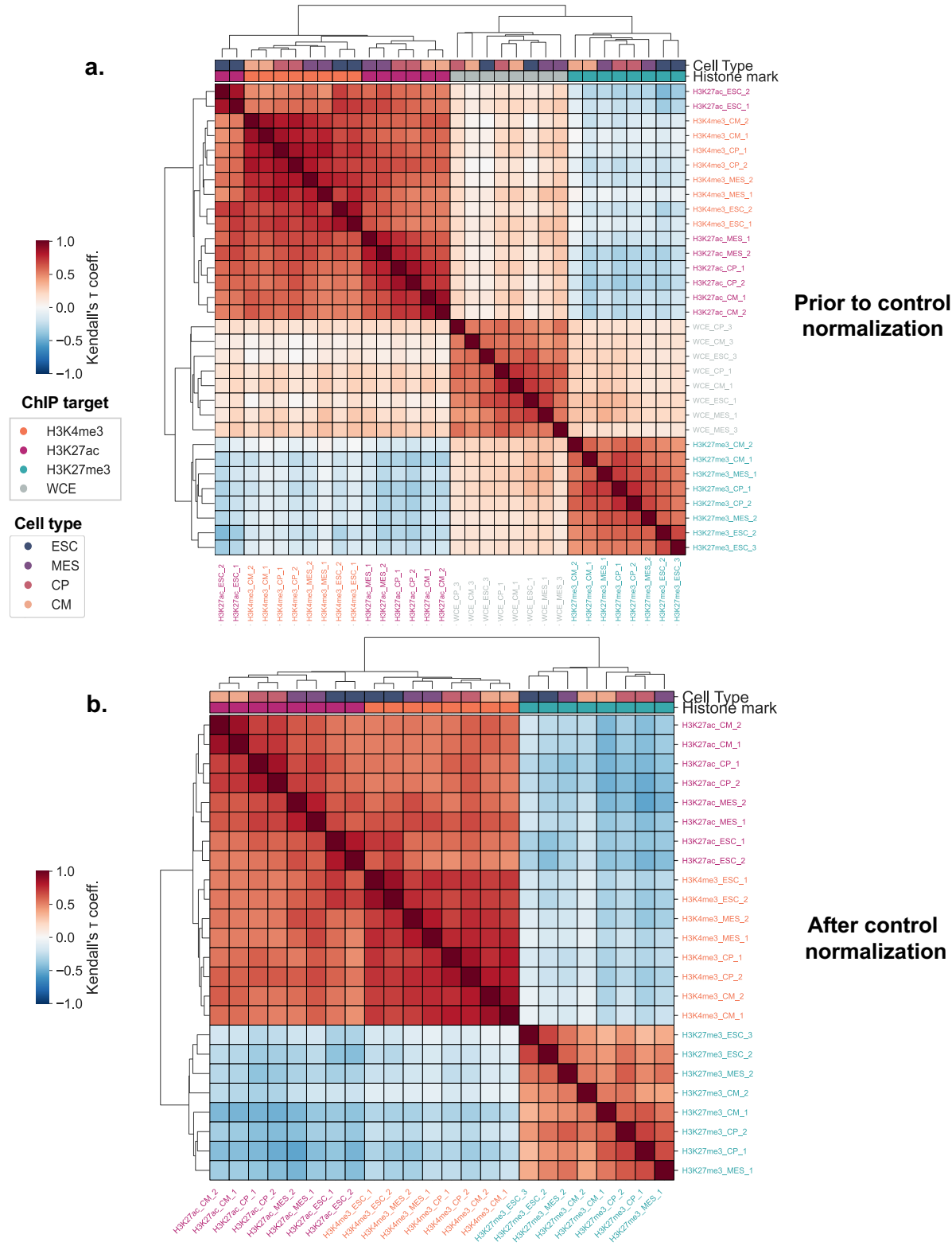

Figure S2. Heatmaps and dendrograms display Kendall's rank correlation analysis and unsupervised hierarchical clustering applied to H3K4me3, H3K27ac, and H3K27me3 datasets, before (a) and after (b) log ratio normalization to control (WCE).

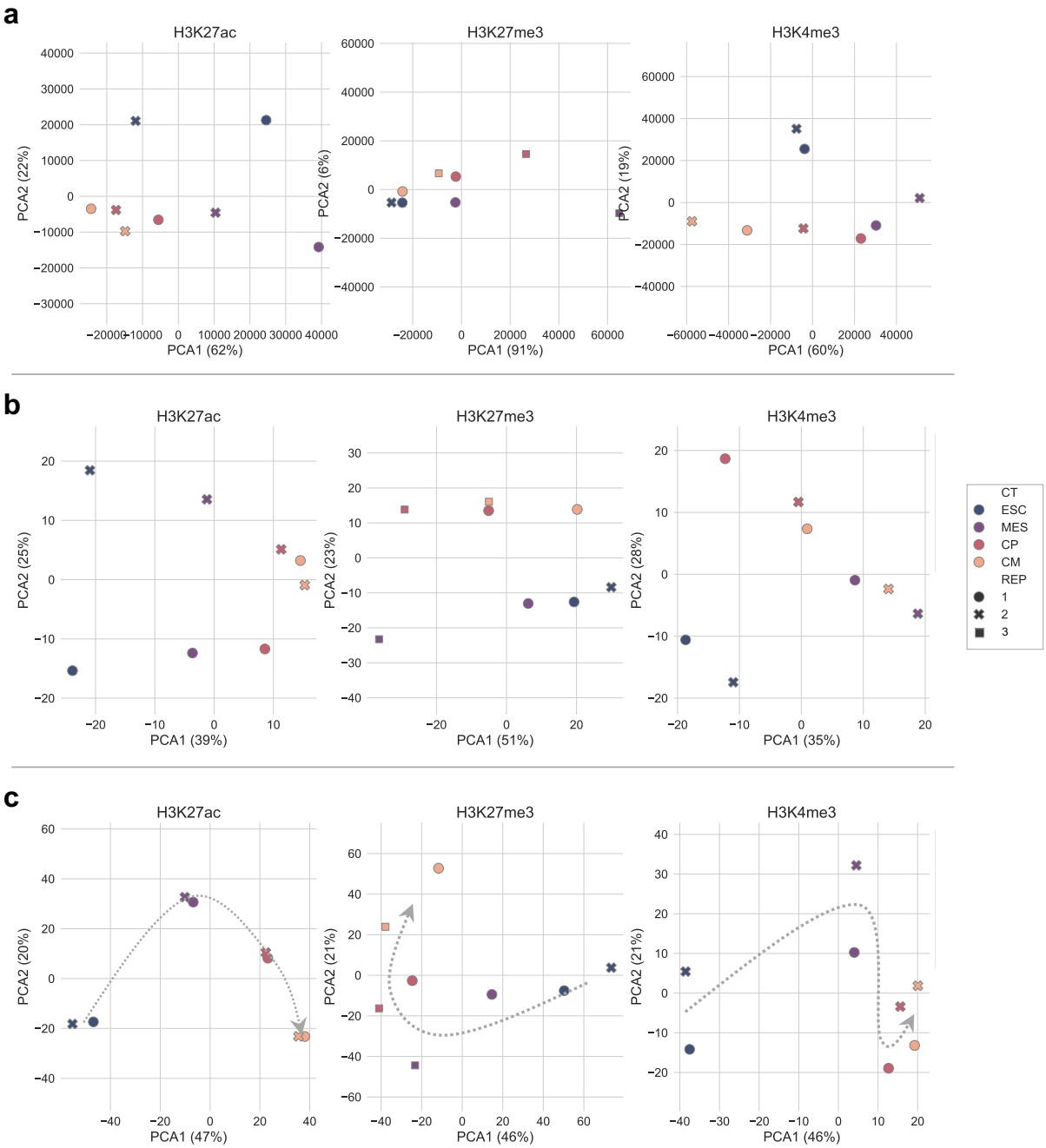

Figure S3. PCAs independently performed by grouping ChIP-seq sample based on the target HM (H3K4me3, H3K27me3 and H3K27ac) before normalization (a), after normalizing on control (b) and after Z-score normalization (c). Notice the differentiation trajectories recovered after normalizations steps (c). Replicates labeled as number 2 in H3K27me3 exhibited lower quality than number 3 in MES, CP and CM, thus the replicates 3 were adopted.

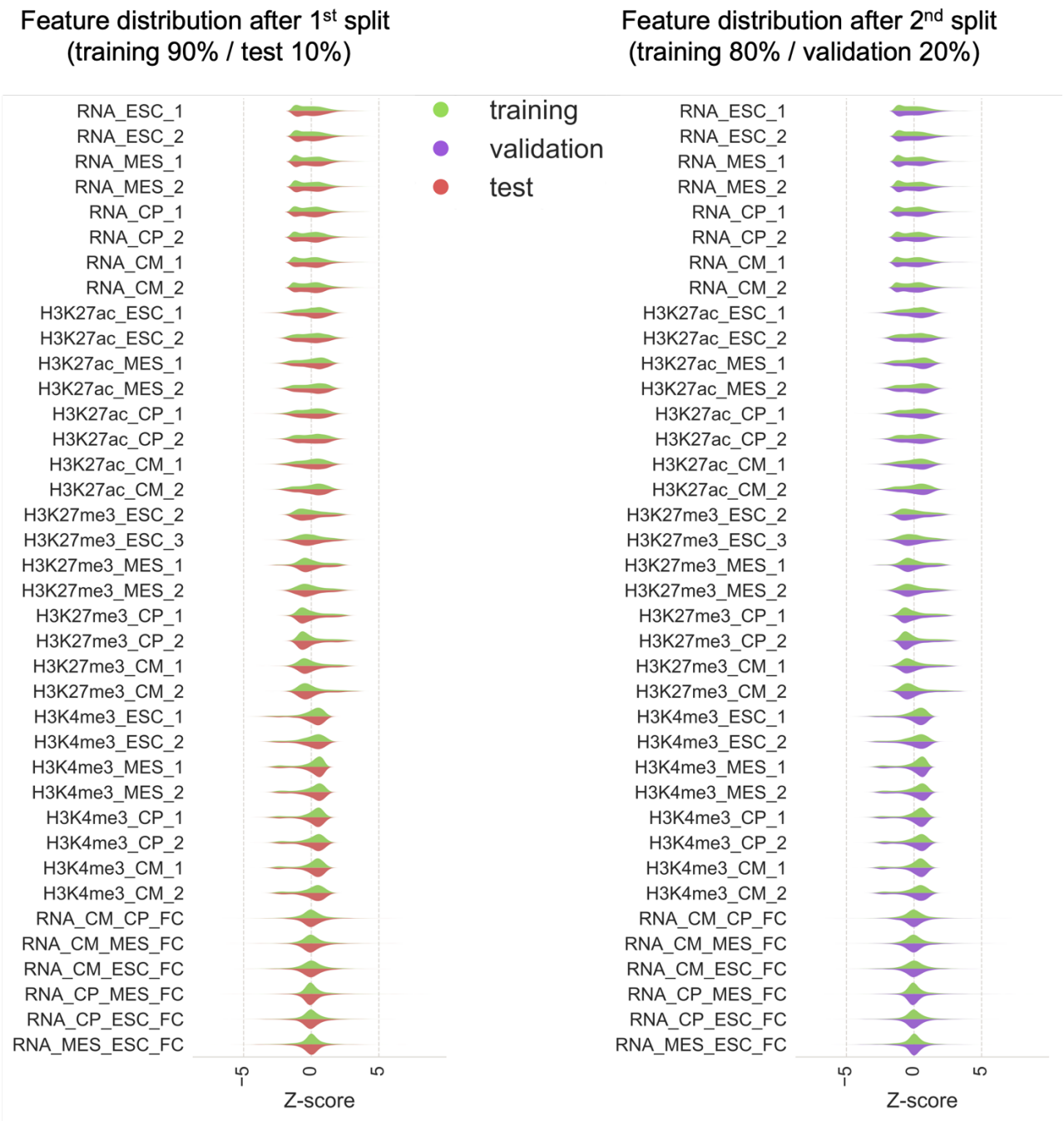

Figure S4. Feature distribution comparison between data subsets obtained after the first and second split.

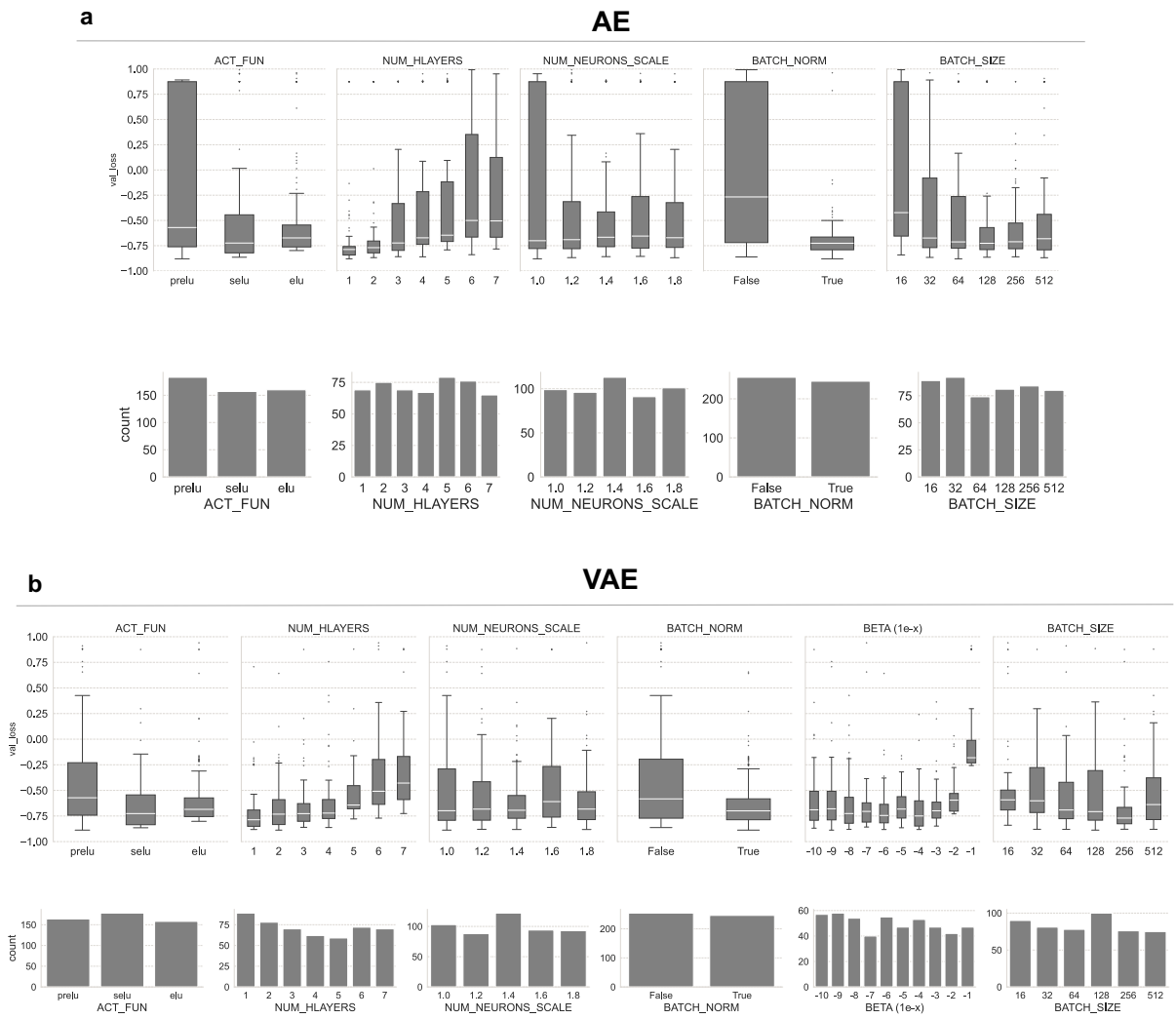

Figure S5. AE (a, top) and VAE (b, top) loss computed on validation set plotted for each HP and stratified by HP values. Frequency of each HP value tested across 500 trials, revealing nearly a uniform distribution for all hyperparameters in both AE (a, bottom) and VAE (b, bottom) models, with no single value appearing significantly more often than others.

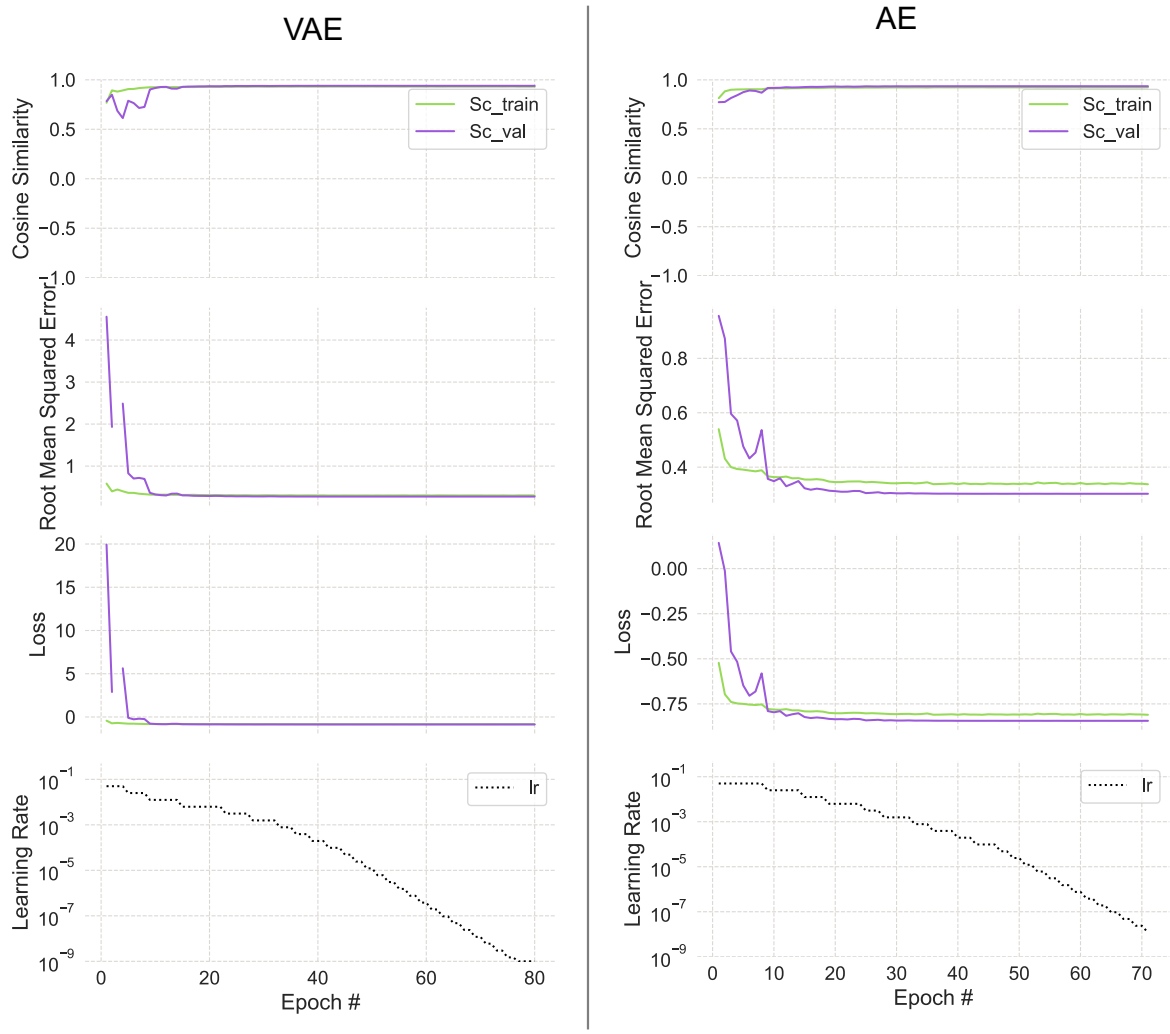

Figure S6. VAE and AE training history recorded at each epoch when training the model on the training set, capturing the loss, dynamic learning rate, and reconstruction scores computed on both training and validation sets.

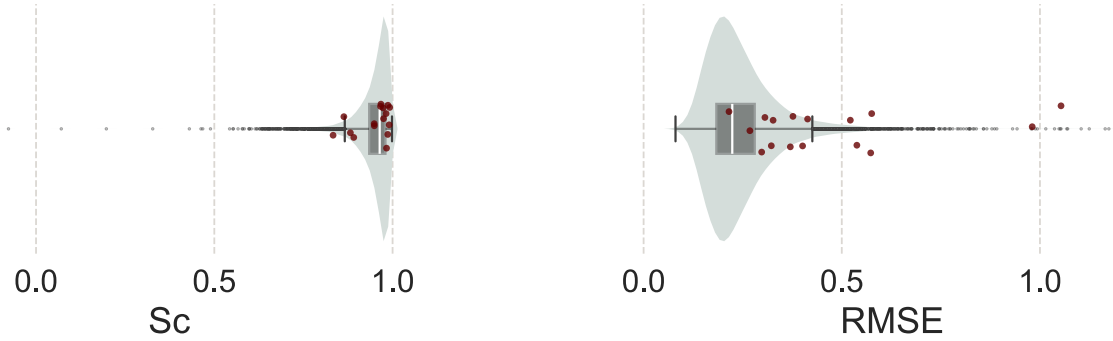

Figure S7. Distributions of reconstruction quality metrics for a subset of selected genes (CT markers) compared to all genes.

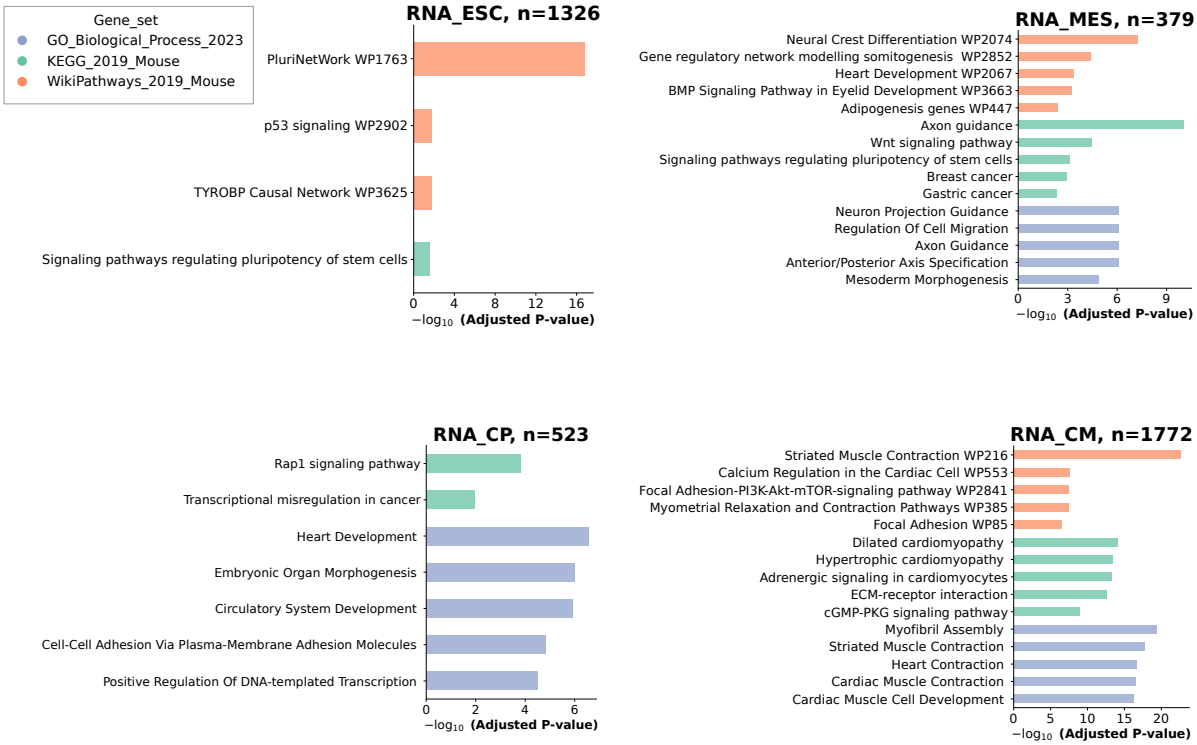

Figure S8. Term enrichment analysis performed on each of the four CT_{max} groups within the variable genes revealing CT specific terms associated with each group.

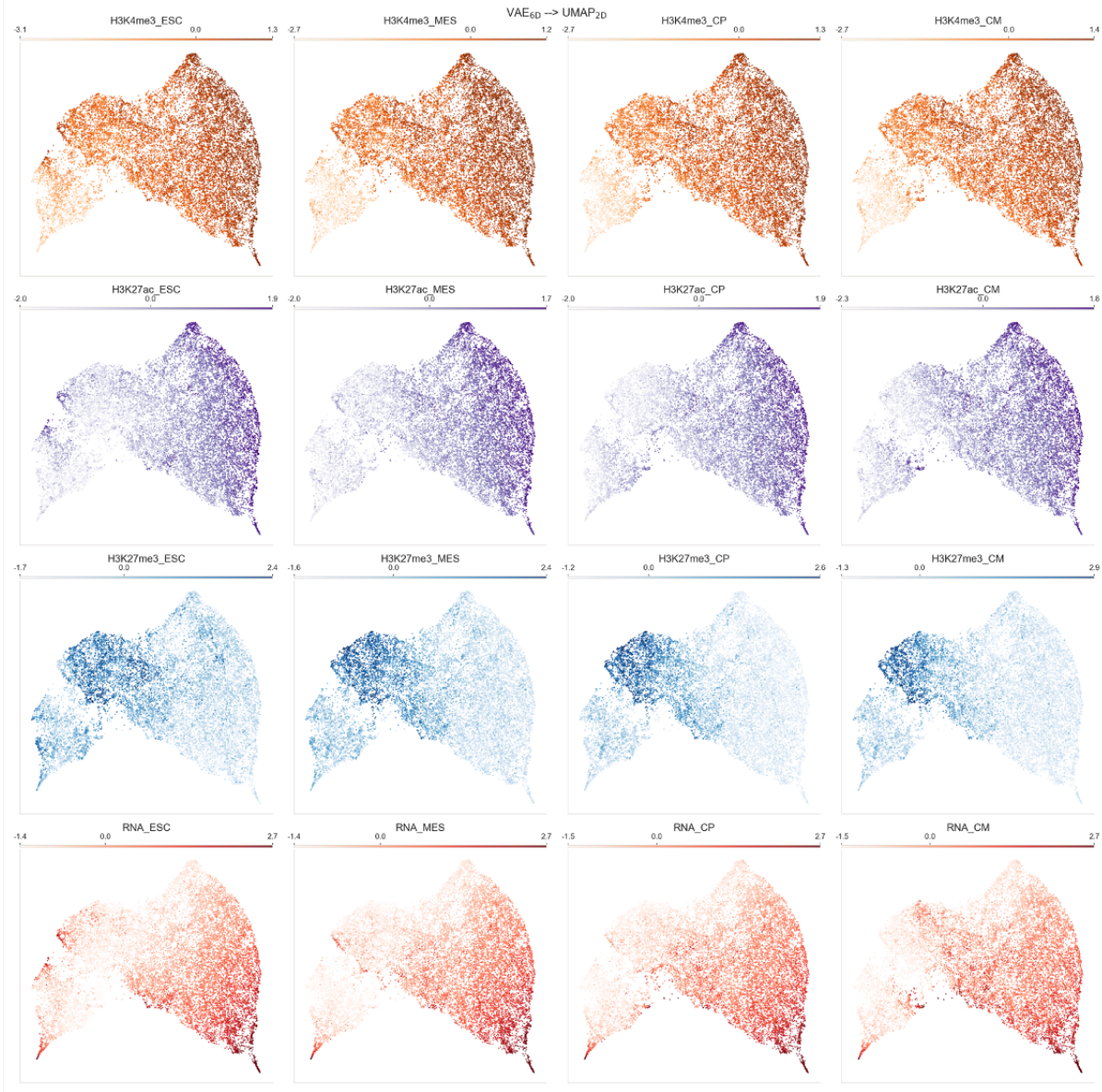

Figure S9. UMAP projection of the genes mapped in the 6-dimensional latent space of the VAE colored by input feature (replicates average) (color mapped from the 1st to the 99th percentile).

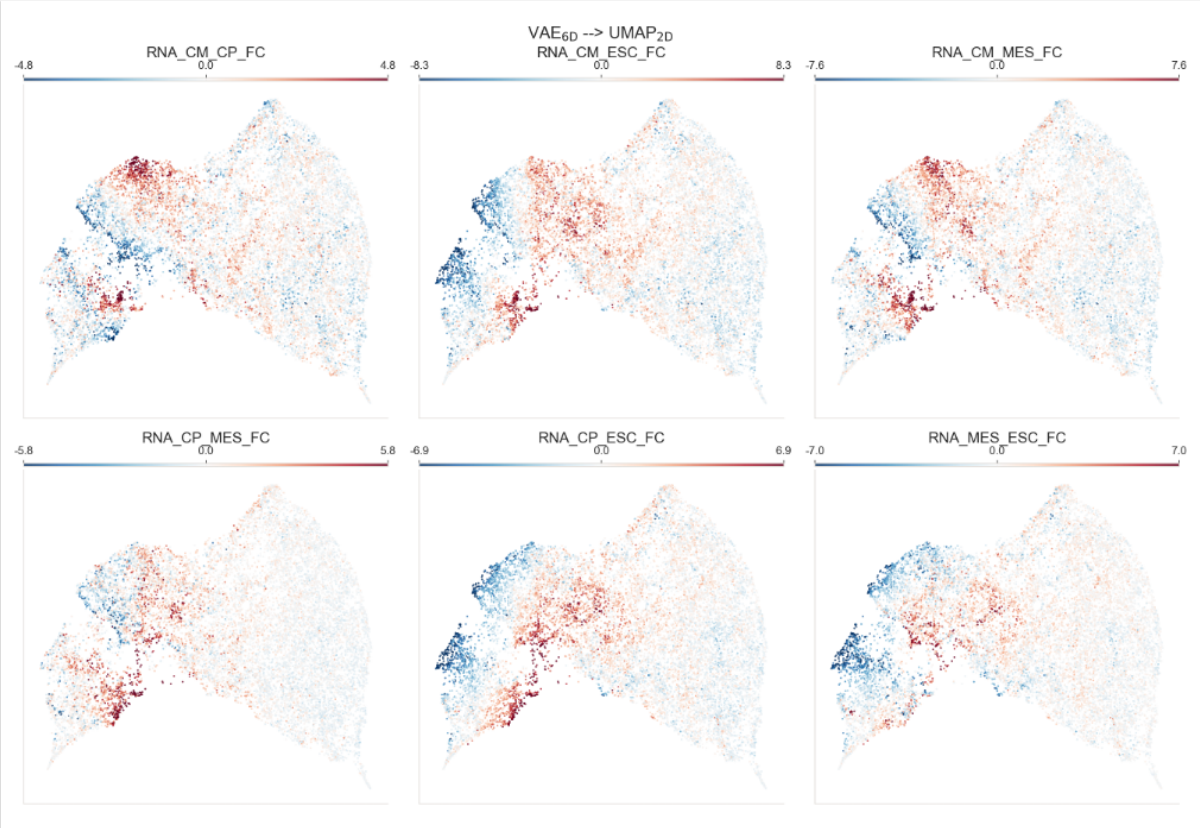

Figure S10. UMAP projection of the genes mapped in the 6-dimensional latent space of the VAE colored by logFCs (color mapped from the 1st to the 99th percentile).

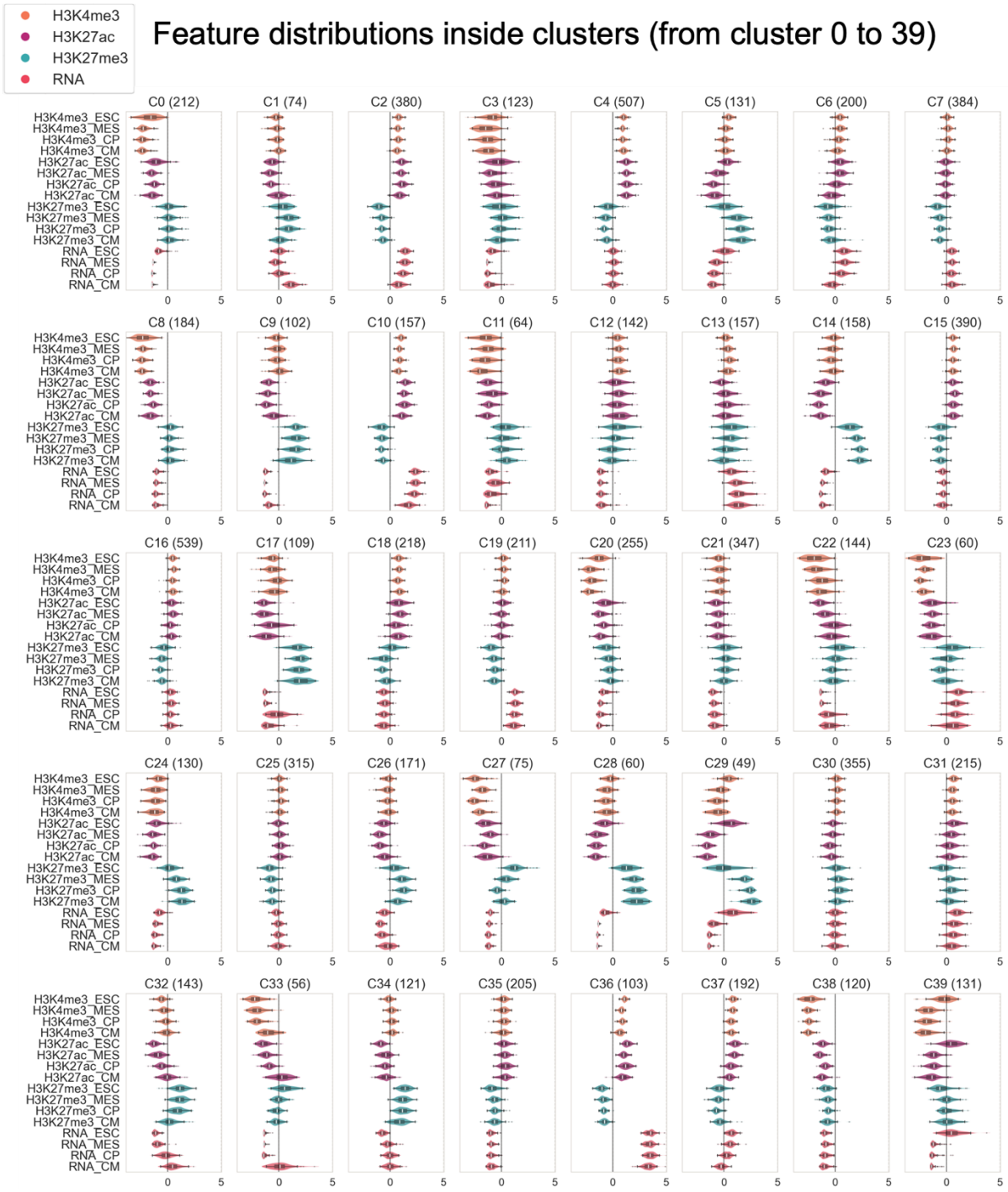

Figure S11. Feature distributions computed as average between replicates, from cluster 0 to 39. The x-axis represents Z-scores, and each subplot title indicates the cluster's gene count in brackets.

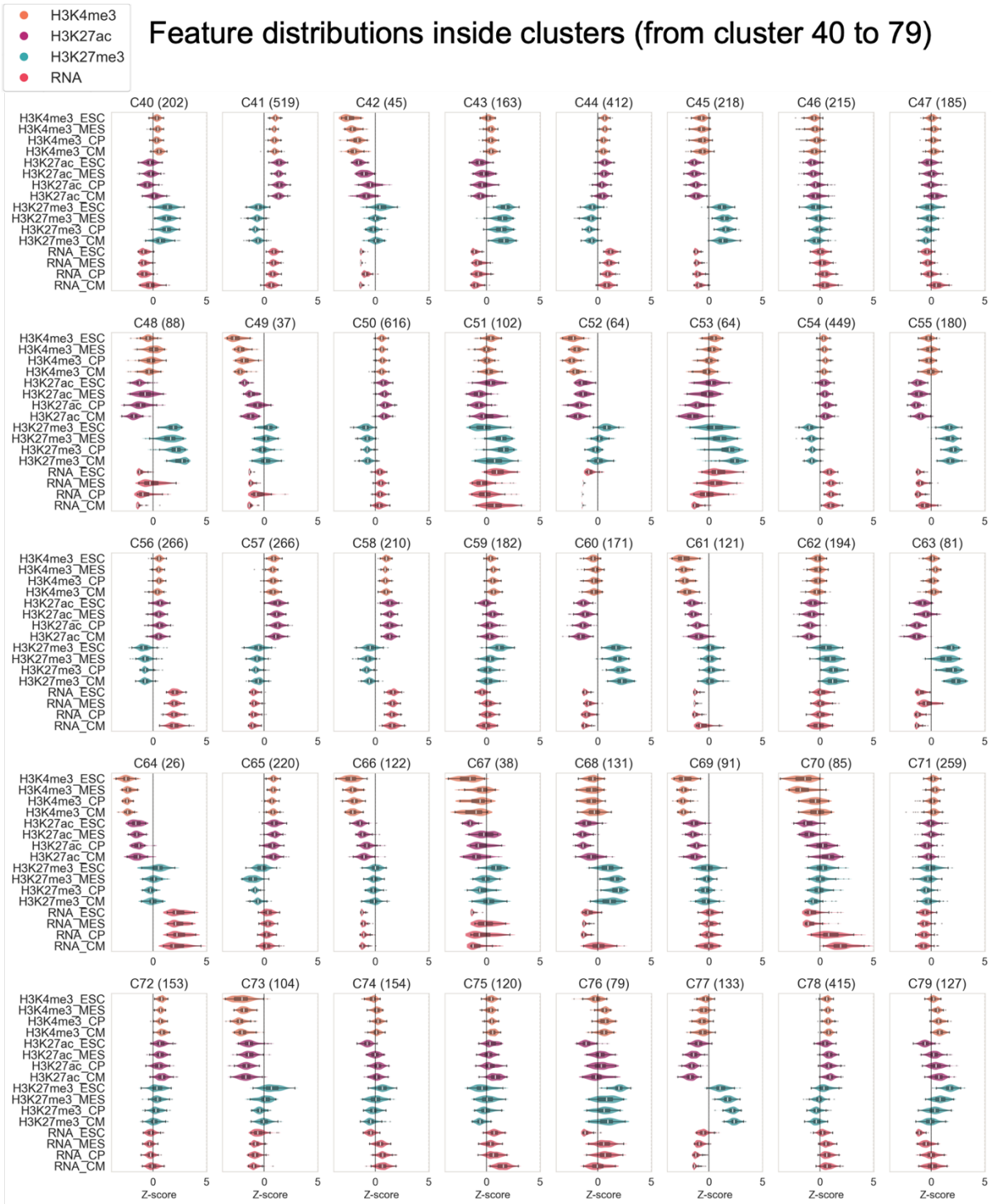

Figure S12. Feature distributions computed as average between replicates, from cluster 40 to 79. The x-axis represents Z-scores, and each subplot title indicates the cluster's gene count in brackets.

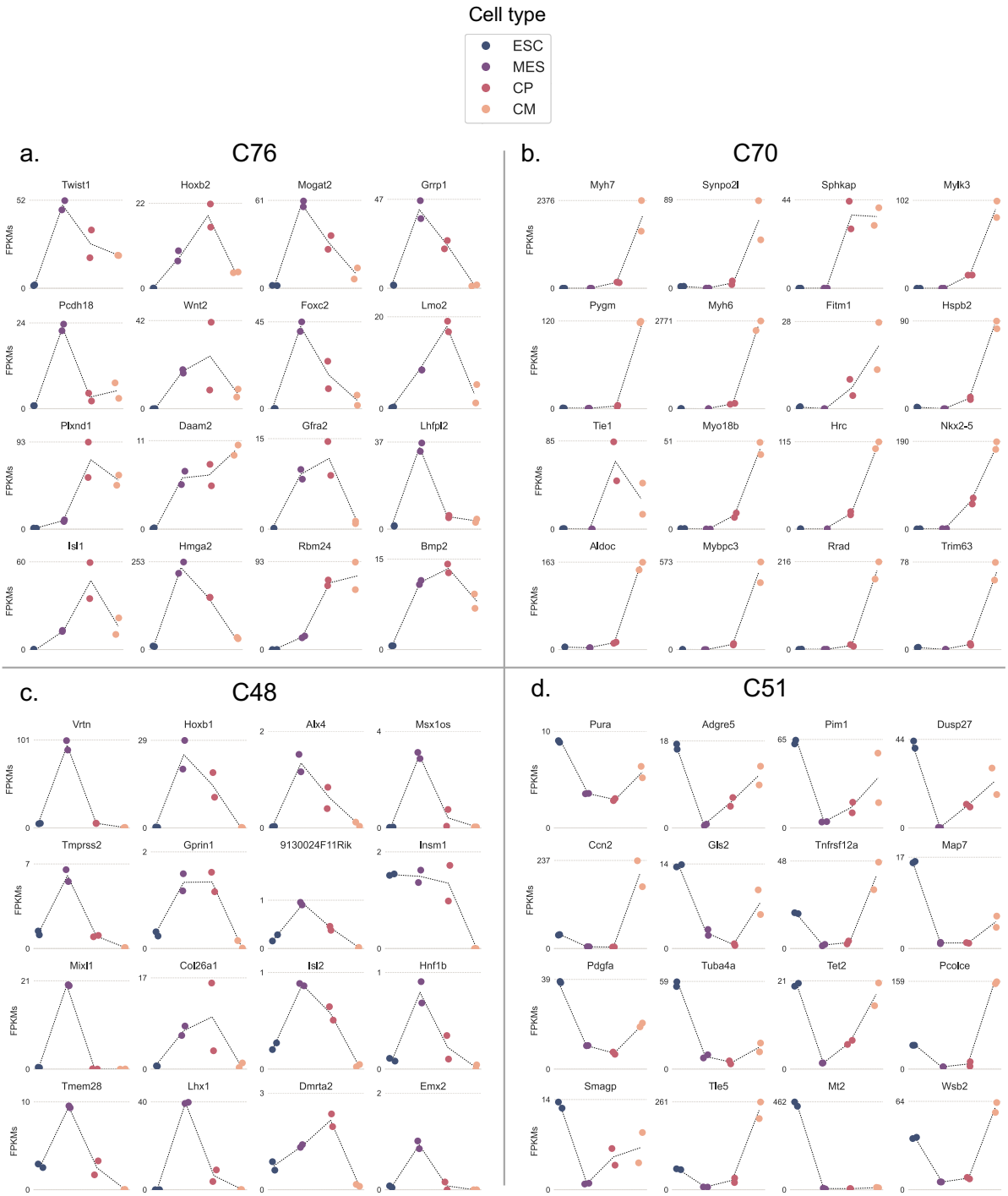

Figure S13. Expression patterns of 16 randomly selected genes per cluster. Each dot represents a replicate.

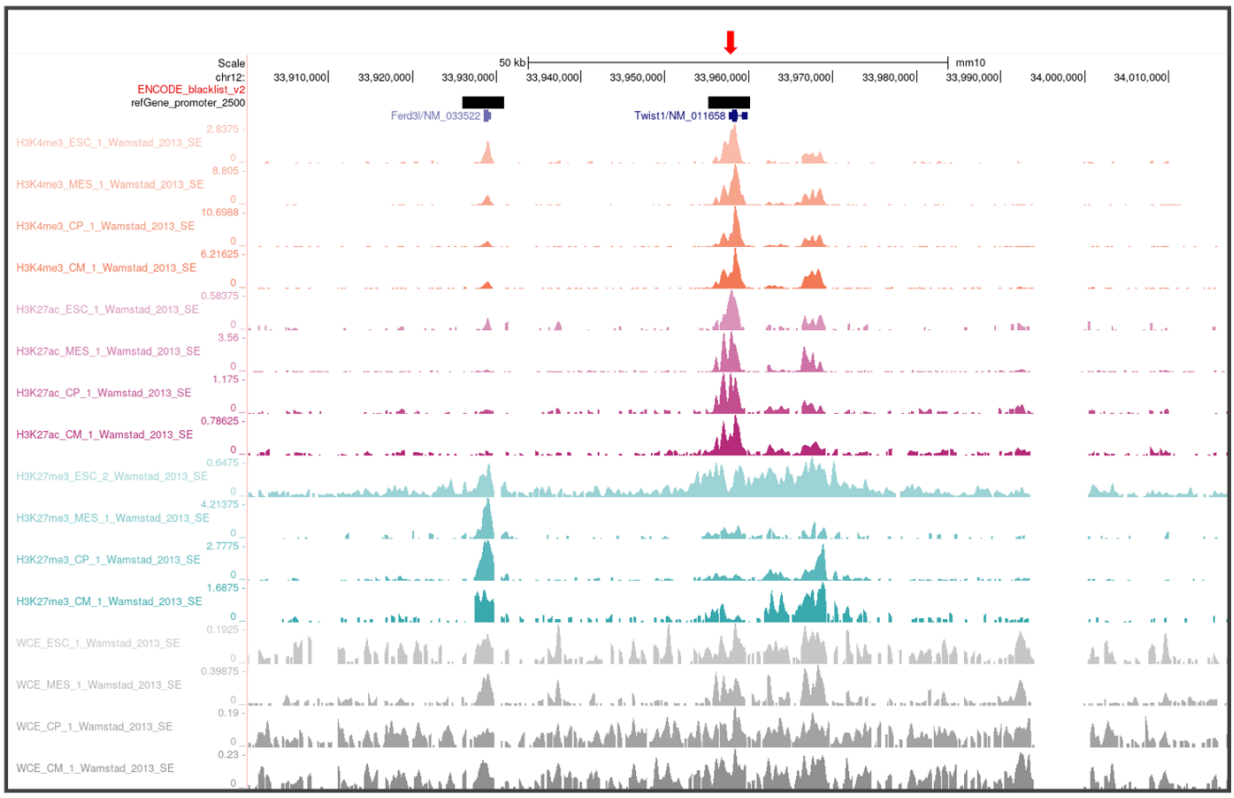

Figure S14. Twist1 region (C76: Developmental genes bivalent in ESC) shown in the genome browser with ChIP-seq experiments (Only one replicate is shown). The red arrow marks the transcription start site (TSS) of the gene of interest. Track heights are auto scaled to the maximum value within each track, and gene annotations are from UCSC RefSeq (in black).

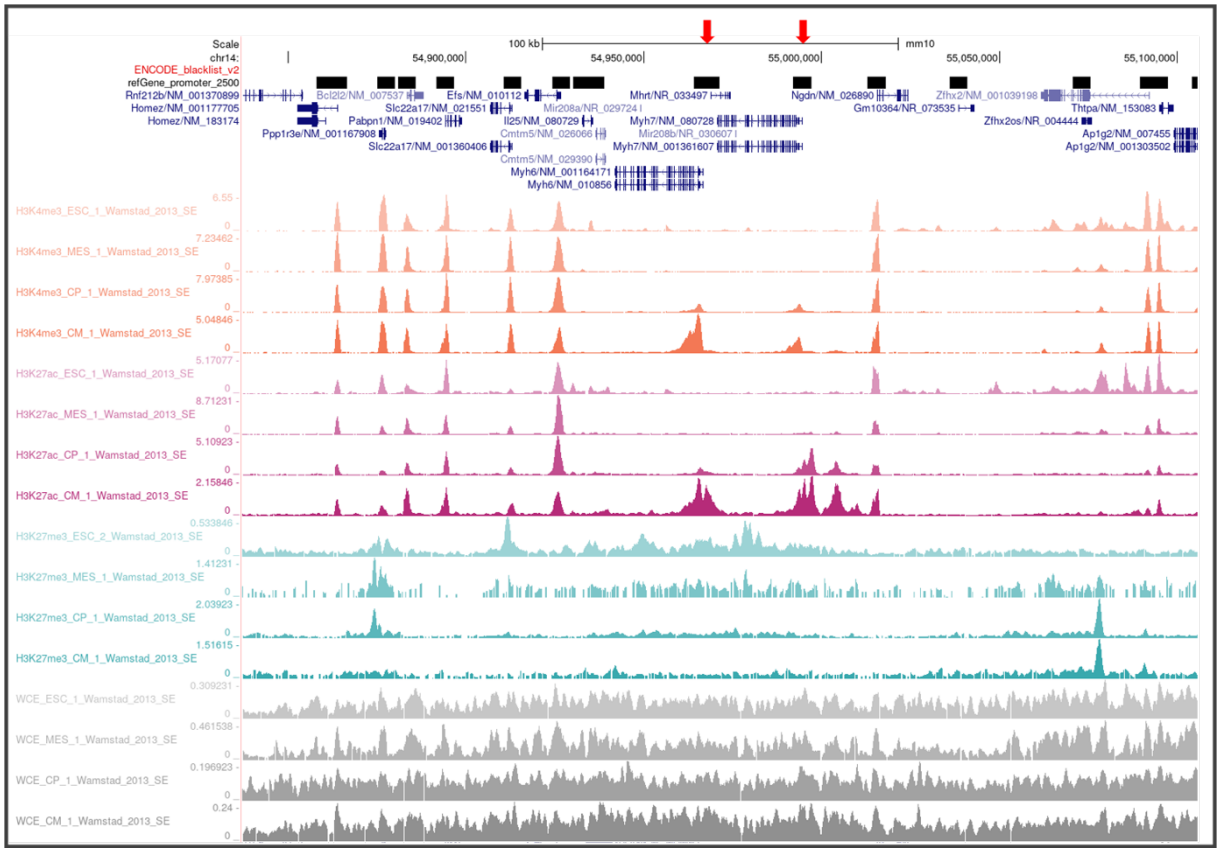

Figure S15. Myh6 and Myh7 region (C70: Silenced genes activated in CM that are non-targets of Polycomb). General plot information is consistent with those described in Figure S14.

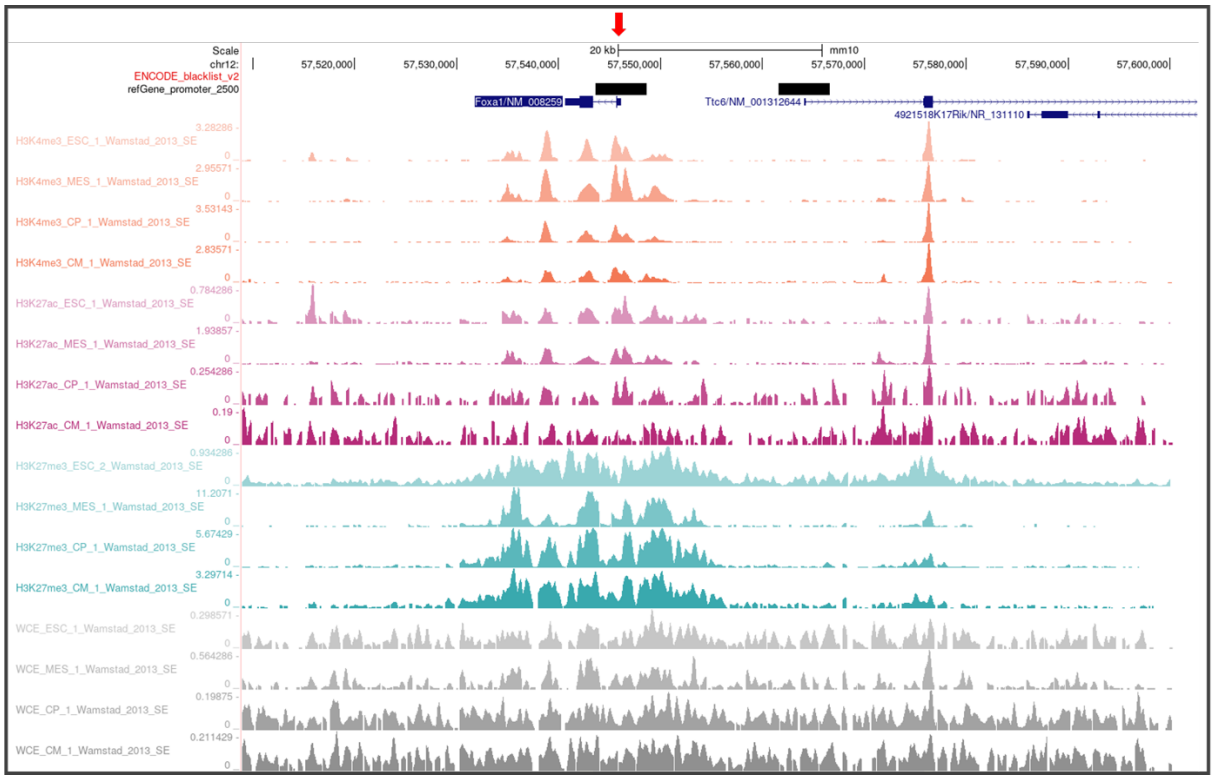

Figure S16. Foxa1 region (C48: Bivalent throughout the differentiation and expressed in mid-stages). General plot information is consistent with those described in Figure S14.

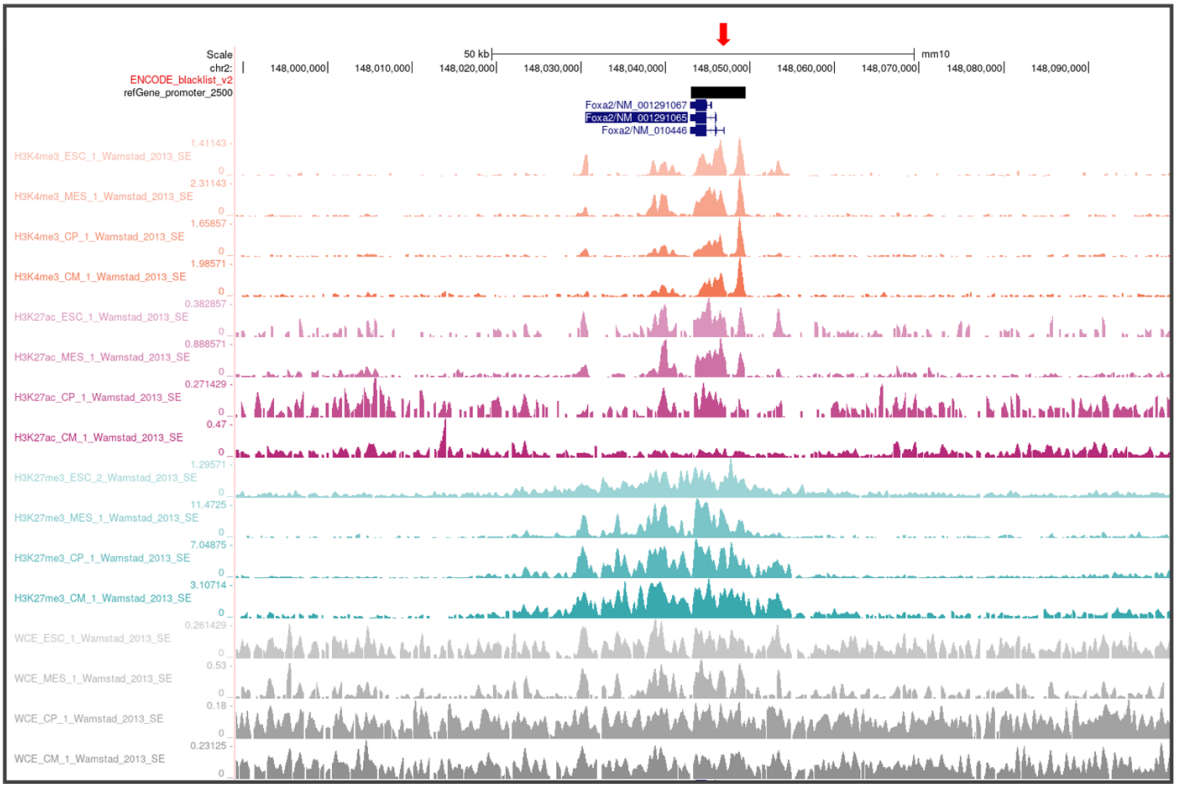

Figure S17. Foxa2 (C48: Bivalent throughout the differentiation and expressed in mid-stages). General plot information is consistent with those described in Figure S14.

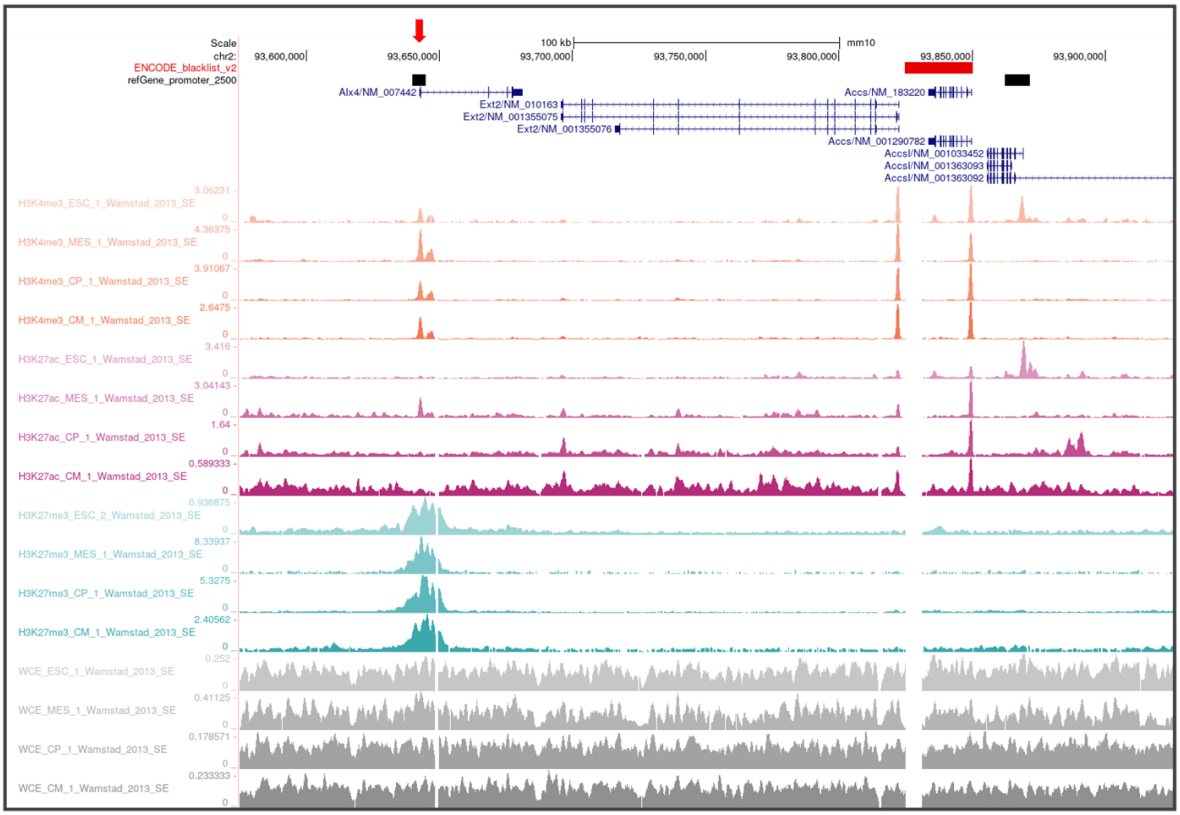

Figure S18. Alx4 region (C48: Bivalent throughout the differentiation and expressed in mid-stages). General plot information is consistent with those described in Figure S14.

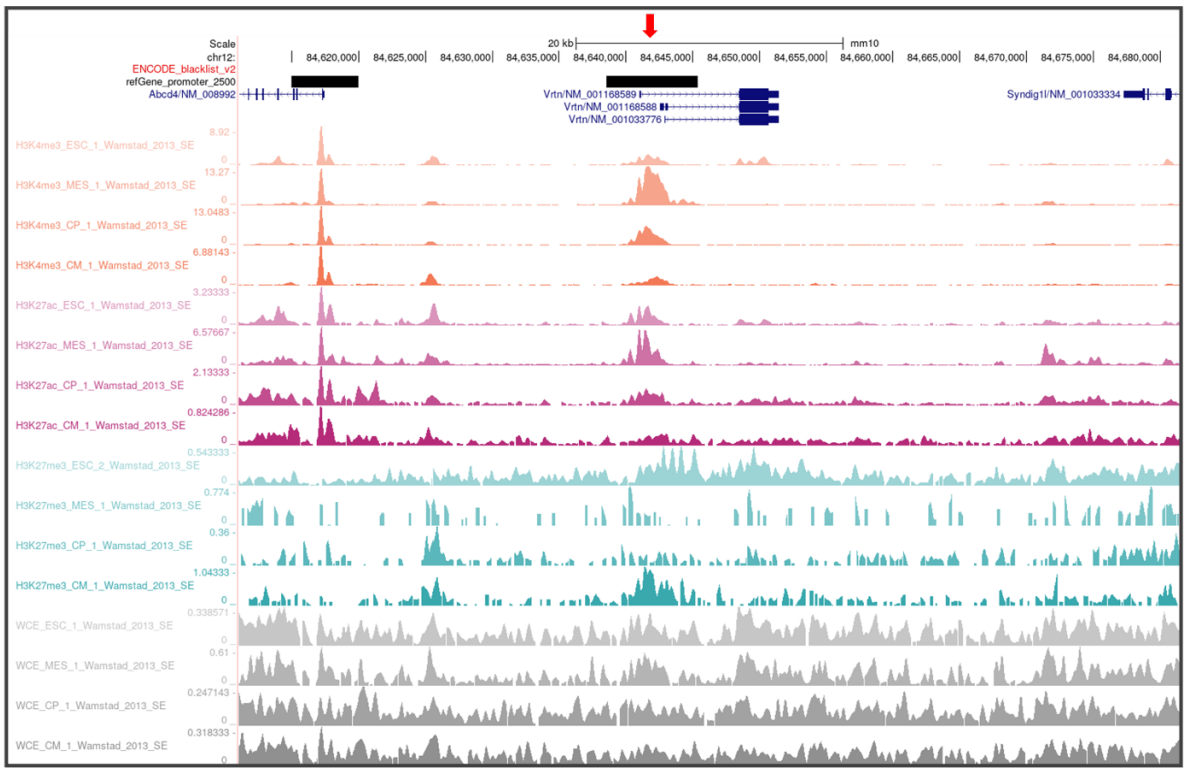

Figure S19. Vrtn region (C48: Bivalent throughout the differentiation and expressed in mid-stages). General plot information is consistent with those described in Figure S14.

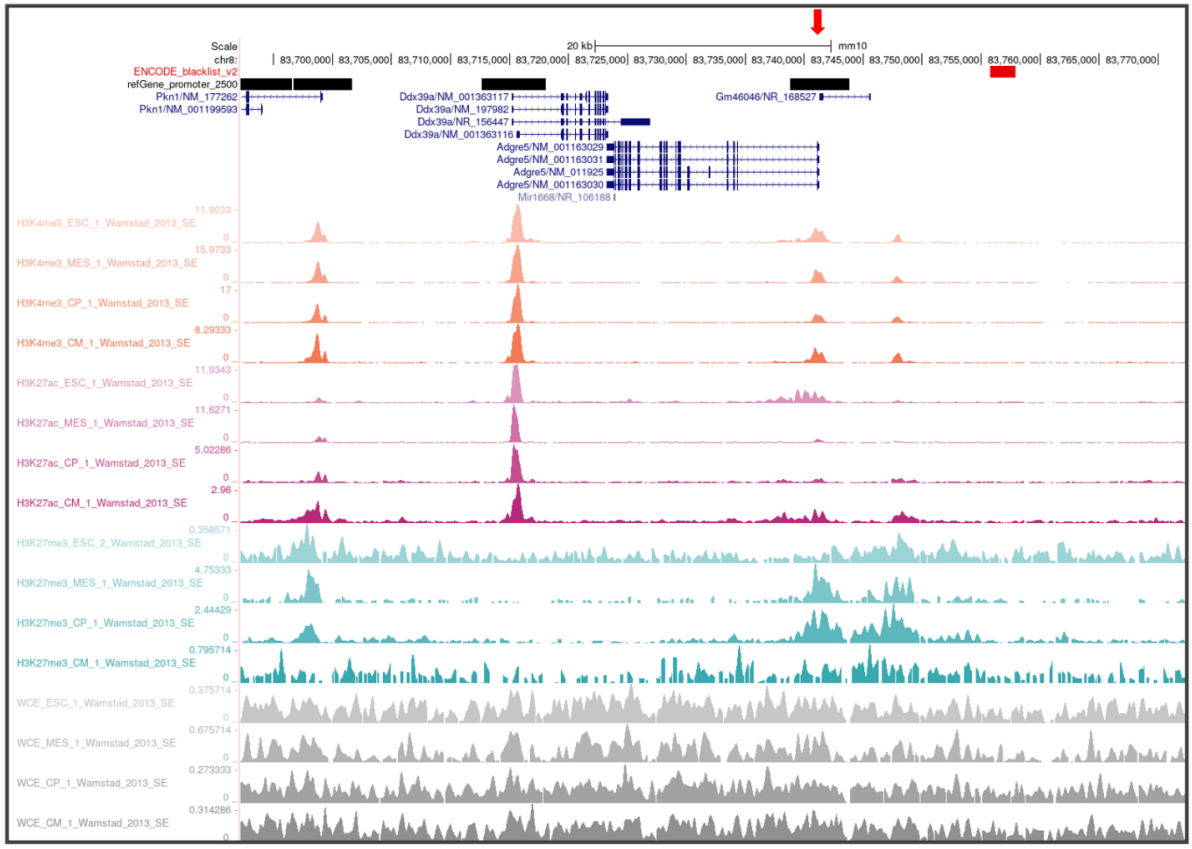

Figure S20. Adgre5 region (C51: Valley-like expression pattern, bivalent in mid-stages). General plot information is consistent with those described in Figure S14.

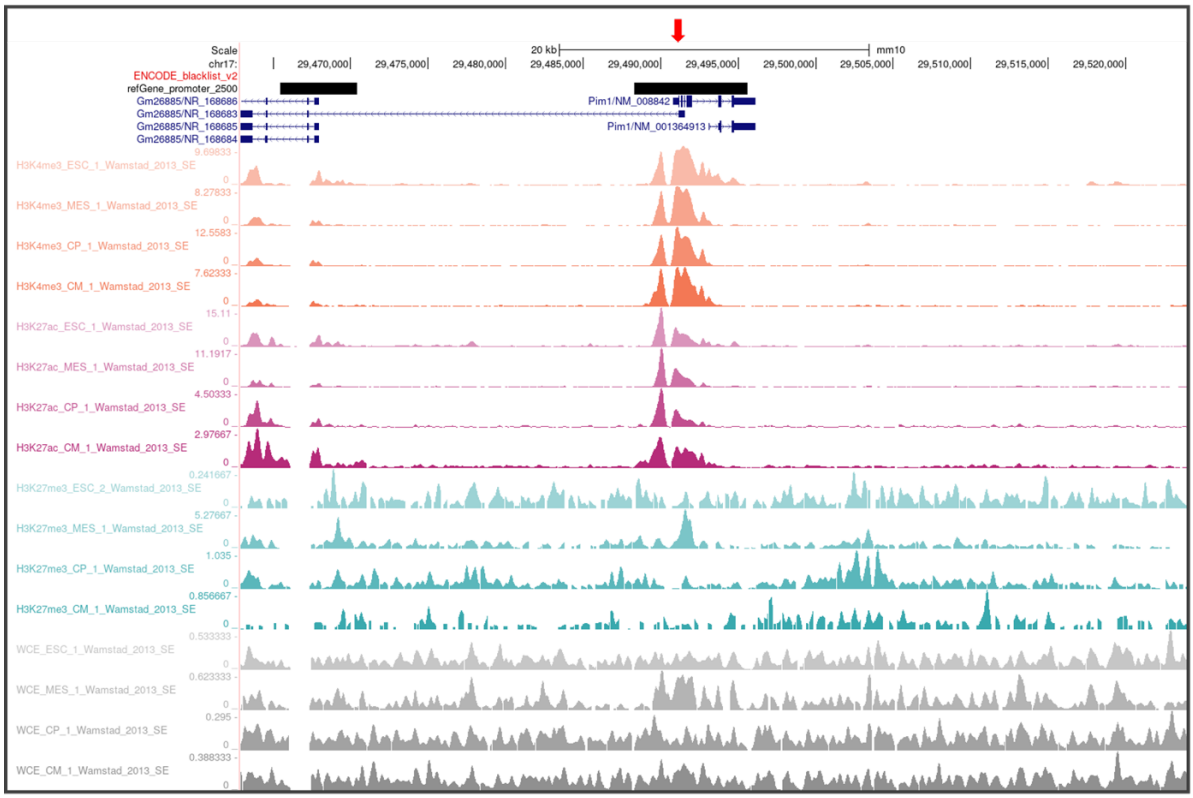

Figure S21. Pim1 region (C51: Valley-like expression pattern, bivalent in mid-stages). General plot information is consistent with those described in Figure S14.

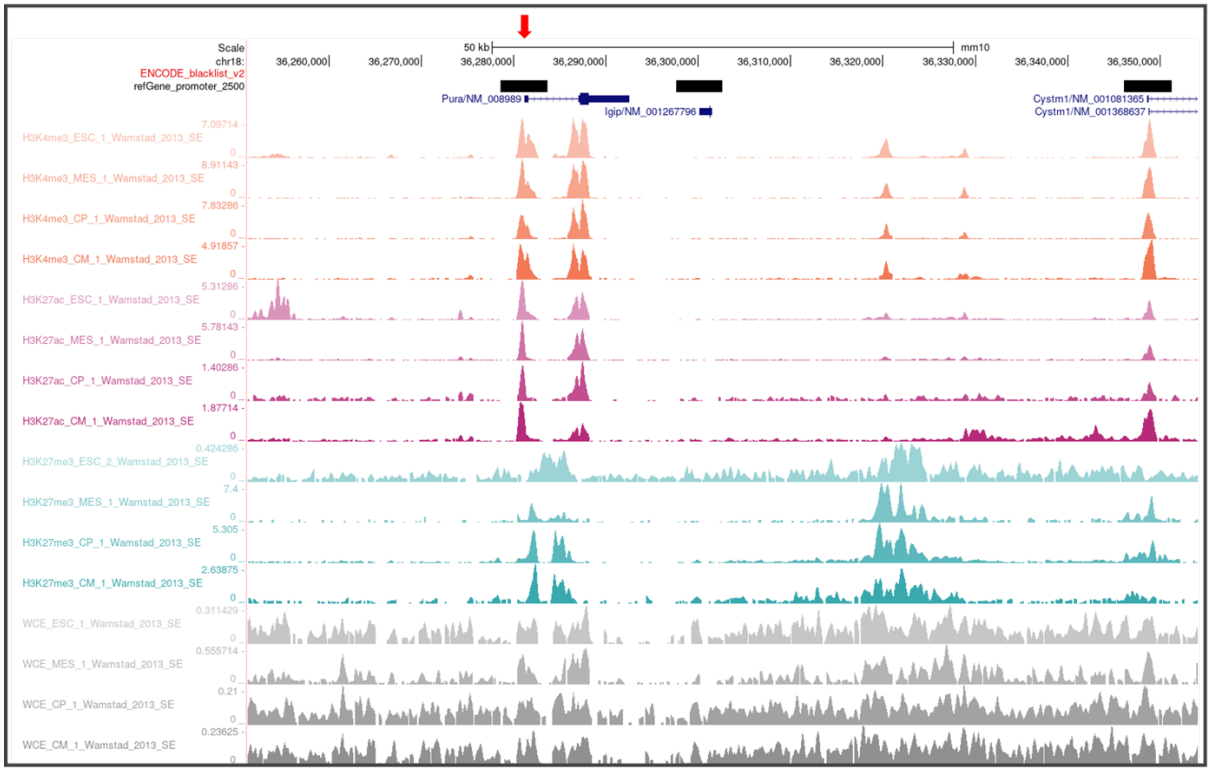

Figure S22. Pura region (C51: Valley-like expression pattern, bivalent in mid-stages). General plot information is consistent with those described in Figure S14.