Revisiting Deep Archetypal Analysis for Phenotype Discovery in High Content Imaging

Mario Wieser, Daniel Siegismund, Stephan Steigele
Genedata AG
{mario.wieser, daniel.siegismund, stephan.steigele}@genedata.com

Abstract

The discovery of unique treatment candidates for complex diseases is a challenging task for current drug discovery programs. Biopharma research has developed automated and scalable screening assays of cell culture models to screen thousands of drug candidates in parallel, e.g., by considering bio-image based assays. However, the large amount of data hinders a systematic review by human experts to distinguish between different disease and healthy phenotypes. A prevalent approach to uncover phenotypic endpoints in a dataset is based on the concept of archetypal analysis which seeks for extremal points in a dataset. State-of-the-art non-linear archetypal methods based on variational autoencoders require $k-1$ latent dimensions to encode k archetypes. However, in high content imaging we frequently require a significantly larger number of latent dimensions than archetypes to encode HCIs which results in weak latent representations and ambiguous archetypes. To overcome this limitation, we propose to relax the simplex constraint in the latent space to a unit hypersphere and learn the respective archetypes based on online dictionary learning. Extensive experiments on two industry-relevant assays and a synthetic MNIST example demonstrate that our method outperforms state-of-the-art deep archetypal analysis approaches.

1. Introduction

In recent years, early discovery research in biopharma drug discovery has been tremendously accelerated by the introduction of High-Content Imaging (HCI) to reveal novel drug candidates for sophisticated treatment strategies such as cancer immunotherapies [19]. More specifically, HCI employs a standardized experimental protocol to systematically acquire multi-spectral images which measure complex phenotypes that haven been introduced by administering certain drug candidates. In production scenarios, screening laboratories make use of automated microscopes on microtiter plates which allow for large-scale drug candidate testing and an automatic analysis procedure to assess the mechanics of many drug candidates, in the order of 1000-10.000's, for a certain disease. This process of screening many drug candidates in parallel is called high-content-screening (HCS).

In order to analyze such complex multi-channel HCI assays, in industry, screening scientists have established time-consuming but still error-prone workflows in cross-functional teams to distill information using handcrafted features from manually set up analysis pipelines [6]. In recent years, such pipelines are gradually replaced by deep learning based approaches [12, 31] which enable the screening scientist to analyze HCI assays without the usage of cross-functional teams and manual image analysis pipelines. Despite the superior performance of such models in comparison to conventional segmentation based analysis [6], the scientist still requires a manually labeling of phenotypes that may result in a biased analysis as introduced by the curating scientist. Hence, [29] introduced a self-supervised approach which distills pseudo phenotypes with linear archetypal analysis based on feature maps of a pretrained feature extractor network. Archetypal analysis (AA) [7] describes an unsupervised approach to uncover biological phenotypes in a dataset [28]. Given these pseudo phenotypes, [29] learn a representation using triplet networks to perform downstream analysis. While enabling the unsupervised analysis of HCI screening assays, this approach suffers from the drawback of relying on limiting assumption of linear archetypal analysis and requiring multiple workflow steps. As AA can be interpreted as a linear latent variable model [27], the simplifying linearity assumption can be relaxed by employing a non-linear latent variable model to uncover the archetypes in a latent space [11, 16, 17]. However, both methods rely by the simplex design choice on the limiting assumption that the method requires $k-1$ latent dimensions for encoding k archetypes. This is a limiting factor, especially in HCI, where we require significantly more latent dimensions to encode an image but require only a small number of phenotypes.

To overcome the aforementioned limitations, we revisit the non-linear extension of archetypal analysis (AA) based on neural networks and present an approach to mitigate the

dimensionality problem by leveraging findings from [24] which learns a convex hull inside a unit sphere. Therefore, we want the same simplex relaxation to a unit sphere as described in [24]. However, traditional methods [11, 16, 17] learn an Euclidean latent representation why we take advantage of the Hyperspherical VAE [9] to learn a unit sphere based on the von Mises-Fisher distribution. Subsequently, we introduce a new reparameterization of the latent space in order to estimate the mixing proportion and the archetypes while simultaneous learning the latent representation using an alternating optimization scheme. This allows not only to uncover non-linear archetypes in an end-to-end fashion but also mitigates the dimensionality problem while being able to decode the underlying biology. Please note, in the remainder of this paper, we use the term archetype and (biological) phenotype interchangeably.

Our contributions are summarized as follows:

- We propose a deep information bottleneck approach to map the input images onto a continuous low-dimensional hypersphere in order to relax the archetypal simplex assumption.
- We introduce an alternating training mechanism to learn archetypes end-to-end in an online fashion based on this representation.
- Experiments on a synthetic MNIST dataset as well as on industry relevant high content screening assays (HCS) demonstrate that the proposed model learns reasonable archetypes while outperforming state-of-the-art archetypal analysis methods.

2. Related Work

Archetypal Analysis Uncovering an arbitray number of extremal points in high-dimensional settings is a frequent task in various applications such as drug discovery. Archetypal Analysis (AA) [7] is a linear method that aims to solve this task. In AA, setting the right number of archetypes is based on prior knowledge. To account for this shortcoming, [26] introduced a model selection concept based on a group-lasso penalty in combination with the Bayesian Information Criterion. In addition, AA has been developed for static data. To extend the application spectrum, AA has been augmented to temporal [8] as well as spatio-temporal application settings [32]. In its original form, archetypal analysis does not give a probabilistic interpretation of the data. Thus, [27] provided a probabilistic interpretation of archetypal analysis which is based on the exponential family and enables the consideration of various data types such as continuous or multinominal observations. However, this formulation makes strong assumptions on the underlying data distribution. Hence, [15] introduced an approach based on the Gaussian copula construction which allows a higher model flexibility by allowing arbitrary margins with a Gaussian dependency structure.

AA is a linear model which makes it highly interpretable but makes it suffer from limited model flexibility. Traditionally, this has been improved by applying kernel methods [4, 25] to the original data. More recently, neural network based approaches have been introduced to learn latent representations from arbitrary data and archetypes end-to-end [16, 17]. However, these approaches require $k-1$ latent dimensions to encode k archetypes. This leads to problems when we are looking for a small number of archetypes while requiring a larger number of latent dimensions to learn proper representations. To account for this, we learn a hypersphere and estimate the archetypes by a modified online dictionary learning approach in an end-to-end fashion. Mairal et al. [23] applied this approach to linear archetypal analysis by feature normalization and solved the problem by a block-coordinate descent algorithm.

Information Bottleneck The information bottleneck [34] is a linear data compression technique that is derived from information theoretic quantities. Essentially, it compresses the input data into a latent representation which retains only the information necessary to reconstruct the output.

Due to the limited modeling flexibility introduced by its linear nature, [1, 3] introduced a non-linear version of the information bottleneck by utilizing neural networks. As [1, 3] did not account for all mutual information invariance properties, Wieczorek et al. [36] introduced an extension of the deep information bottleneck which restored the missing invariance properties with a copula construction and additionally learned sparse latent representations. This approach as been employed to learn a non-linear archetypes [16, 17] by a simplex construction in the latent space.

3. Foundations

3.1. Deep Information Bottleneck

The Deep Variational Information Bottleneck (DVIB) [3] is a compression technique based on mutual information. The main goal is to compress a random variable X into a random variable Z while retaining side information about a third random variable Y. I represents the mutual information between two random variables. Achieving the optimal compression requires solving the following parametric problem:

$$\min_{\phi,\theta} I_{\phi}(Z;X) - \lambda I_{\phi,\theta}(Z;Y), \tag{1}$$

where we assume a parametric form of the conditionals $p_{\phi}(z|x)$ and $p_{\theta}(y|z)$. ϕ, θ represent the neural network parameters and λ controls the degree of compression.

The mutual information terms can be expressed as:

$$I(Z;X) = \mathbb{E}_{p(x)} D_{KL}(p_\phi(z|x)\|p(z)) \quad (2)$$
$$= \mathbb{E}_{p(x)} D_{KL}(\mathcal{N}(\mu,\sigma)\|\mathcal{N}(0,1)) \quad (3)$$
$$I(Z;Y) \geq \mathbb{E}_{p(x,y)}\mathbb{E}_{p_\phi(z|x)} \log p_\theta(y|z) + h(Y) \quad (4)$$

where D_{KL} denotes the Kullback-Leibler divergence, $\mathbb{E}$ the expectation and $\mathcal{N}$ the Gaussian distribution with mean μ and variance σ.

3.2. Archetypal Analysis

Archetypal analysis (AA) introduces an unsupervised approach to uncover extremal datapoints in a dataset by representing each datapoint as the convex combination of its respective archetypes [7]. In formal terms, we can describe AA as a non-negative matrix factorization approach with simplex constraints on the weight matrices. More precisely, AA aims to factorize the data matrix $X \in \mathbb{R}^{n\times d}$ with n denoting the number of data samples and d the number of features such that $X \in \mathbb{R}^{n\times d}$ is approximated as $X \approx ABX = AZ$ with $Z \in \mathbb{R}^{k\times d}$ being the archetype matrix. Here, $A \in \mathbb{R}^{n\times k}$ and $B \in \mathbb{R}^{k\times n}$ denote the weight matrices with k representing the number of archetypes under the assumption that each row sums to one.

$$a_{ij} \geq 0 \wedge \sum_{j=1}^{k} a_{ij} = 1 \quad b_{ji} \geq 0 \wedge \sum_{i=1}^{n} b_{ji} = 1 \quad (5)$$

In general, we assume that each dataset can be approximated by the number of archetypes $k < \min\{n, d\}$. In order to solve this non-linear optimization problem, we aim to minimize the following residual sum of squares:

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{ABX}\|^2 \quad (6)$$

More recently, AA has been interpreted probabilistically by reformulating AA as a simplex latent variable model [27]. More specifically, the probabilistic model (PAA) is formalized as follows:

$$x_i \sim \mathcal{N}(a_i Z, \sigma^2, I) \quad \text{with} \quad a_i \sim \text{Dir}_k(\alpha) \quad (7)$$

where the observations x_i are drawn from the normal distribution with the mean parameterized as $\mu = a_i Z$ and variance σ^2. Z denotes the learned archetype matrix and a the mixing proportions sampled from a Dirichlet distribution with uniform concentration parameters.

However, both AA and PAA make the simplifying model assumption of linearity and thus fail to model non-linear relationships. As a consequence, [16, 17] introduced a non-linear extension based on the deep information bottleneck principle (see section 3.1) to account for this limitation. To do so, [16, 17] propose a reformulation of the mean μ of the conditional distribution $p_\phi(z|x)$ from Eq. 3 in combination with a novel archetype loss function.

$$z_i \sim \mathcal{N}(\mu_i(x_i) = a_i(x_i) Z^{\text{fixed}}, \sigma^2_{x_i} I) \quad (8)$$

where the mean μ_i of a datapoint x_i is projected into a predefined simplex Z^{fixed} with mixing proportions a_i that are determined by the weight matrix $A \in \mathbb{R}^{n\times k}$ with simplex constraint. In addition, we define an additional weight matrix $B \in \mathbb{R}^{k\times n}$ (see Eq. 6). After having defined $A \in \mathbb{R}^{n\times k}$, $B \in \mathbb{R}^{k\times n}$ and the fixed simplex Z^{fixed}, we obtain the predicted archetype positions Z^{pred} by calculating ABZ^{fixed}. In order to cover the whole simplex, [16, 17] introduce an additional loss term

$$l_{at} = ||Z^{\text{fixed}} - BAZ^{\text{fixed}}||_2^2 = ||Z^{\text{fixed}} - Z^{\text{pred}}||_2^2 \quad (9)$$

which pushes the predicted archetypes close to the simplex corners. In summary, the deep archetypal analysis method optimizes the following loss function.

$$\min_{\phi,\theta} I_\phi(Z;X) - \lambda I_{\phi,\theta}(Z;X) + l_{at} \quad (10)$$

3.3. Sparse Dictionary Learning

Sparse dictionary learning denotes a special approach of non-negative matrix factorization with the goal to represent a dataset $X \in \mathbb{R}^{n\times d}$ with n samples and d dimensions by a low-dimensional learned dictionary $D \in \mathbb{R}^{d\times m}$ with m entries and a sparse coefficient vector $\alpha \in \mathbb{R}^m$ which approximates a data point x as a linear combination of D and α. More formally, we aim to optimize the following loss function:

$$\min_{D,\alpha} ||X - D\alpha||_2^2 + \lambda||\alpha||_1 \quad (11)$$

where λ controls the amount of sparsity of the coefficient α. This optimization problem is commonly solved by an alternating optimization procedure which keeps D fixed while optimizing α and vice versa [21].

However, these methods require the full dataset X to be available which may be intractable for large scale datasets. In order to overcome this limitation, a variety of online learning methods have been developed which require only batches of the dataset at a time by introducing algorithms based on block-coordinate descent or stochastic gradient descent, amongst others [2, 23]. In the context of archetypal analysis, [24] leveraged the block-coordinate descent approach to learn the archetypes in an online fashion by replacing the Lasso constraint $\lambda||\alpha||_1$ in Eq. 11 with a simplex constraint.

4. Revisiting Deep Archetypal Analysis

As previously described in Section 1, we aim to identify a number of phenotypes k in a d-dimensional latent

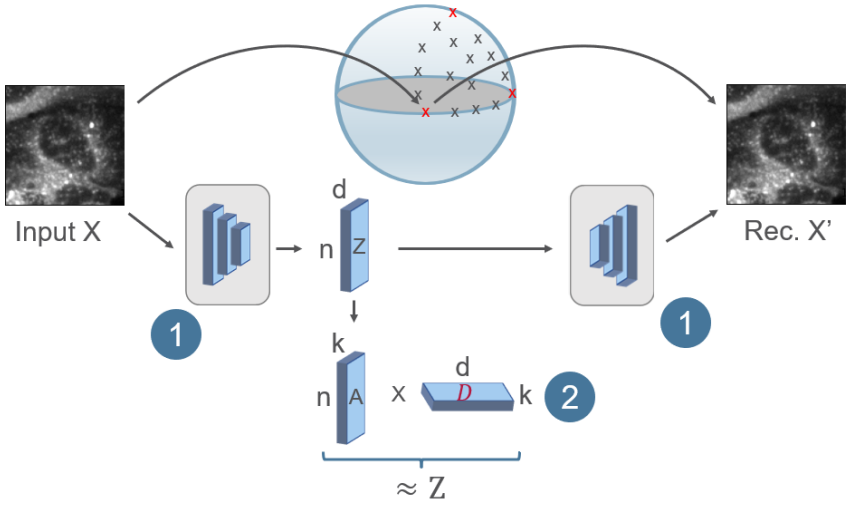

Figure 1. High level overview of the presented model. The upper part of the figure gives an intuition of the method. More specifically, we map the input X onto a latent unit sphere Z, identify the archetypes (red x) and reconstruct the input. The lower part of the image illustrates the corresponding model to achieve this. In a first step (1), we use an encoder to learn a latent representation $Z \in \mathbb{R}^{n\times d}$ of the input $X \in \mathbb{R}^{n\times w\times h}$ and subsequently try to reconstruct $X' \in \mathbb{R}^{n\times w\times h}$ with our decoder. Given this latent representation Z, we learn in step (2) the mixing coefficients $A \in \mathbb{R}^{n\times k}$ and multiply them with the archetypes $D \in \mathbb{R}^{k\times d}$ to approximate Z as close as possible.

space $Z \in \mathbb{R}^{n\times d}$ where $k << d$ based on observations $X \in \mathbb{R}^{n\times w\times h}$ (see Figure 1). In this situation, X may be large-scale, complex objects such as high content images, which are challenging to analyze at the same time. As a consequence, we learn a continuous low-dimensional representation Z of our input data X. To do so, we employ the equivalent deep information bottleneck formulation as described in [16,17] (Eq. (10)) without the additional archetype loss function (Eq. (9)). Our base model is thus defined as follows:

$$\min_{\phi,\theta} I_\phi(Z;X) - \lambda I_{\phi,\theta}(Z;X), \tag{12}$$

where ϕ, θ describe neural network parameters and λ the compression factor similar to [16, 17]. However, the traditional deep AA model suffers from the architectural shortcoming that it requires always $k-1$ latent dimensions to encode k archetypes. In order to overcome this limitation, we propose a novel deep archetypal method which is able to handle the $k << d$ situation which consists of two distinct steps, (1) learning hyperspherical representations and second (2) the corresponding archetypes. More specifically, we build on the work of [24] that proposed to relax the simplex assumption by a hypersphere assumption using l_1 normalized features and learn an approximate archetype representation via a block-coordinate descent approach.

4.1. Learning Hyperspherical Representations

In the first step (1), we aim to learn a hyperspherical representation to be able to uncover the archetypes under the relaxed constraint. However, the deep information bottleneck learns an Euclidian latent representation by parameterizing a latent point z by the mean μ and the variance σ of a Gaussian distribution to apply the reparameterization trick (see Eqs. 3 and 4) [18].

To overcome this challenge, we leverage the work of [9] to learn a hyperspherical latent representation $Z \in \mathbb{R}^{n\times d}$. To do so, we replace the Gaussian distribution in Eq. 3 with the von Mises-Fisher distribution which is essentially a Gaussian distribution on a hypersphere where μ describes the mean and κ the concentration around μ. More specifically, we rewrite the probability density function $p_\phi(z|\mu,\sigma)$ in Eq. 3 as follows:

$$p_\phi(z|\mu,\kappa) = \frac{\kappa^{(d/2)-1}}{(2\pi)^{d/2} I_{(d/2)-1}(\kappa)} \exp(\kappa\mu \cdot z), \tag{13}$$

where $I_{(d/2)-1}(\kappa))$ denotes the modified Bessel function of the first kind at order $((d/2)-1)$. Based on this result we rewrite the KL-divergence term of Eq. 3 as follows:

$$\begin{aligned} I(X;Z) &= \mathbb{E}_{p(x)} D_{KL}(p_\phi(z|x)\|p(z)) && (14)\\ &= \mathbb{E}_{p(x)} D_{KL}(\text{vMF}(\mu,\kappa)\|U(S^{m-1})) && (15)\\ &= \mathbb{E}_{p(x)} \kappa \frac{I_{(d/2)(k)}}{I_{(d/2)-1}(k)} && (16)\\ &+ \log\left(\frac{\kappa^{(d/2)-1}}{(2\pi)^{d/2} I_{(d/2)-1}(\kappa)}\right) - \log\left(\frac{2(\pi^{d/2})}{\Pi(d/2)}\right)^{-1} && (17) \end{aligned}$$

For further details regarding the sampling and optimization process, we refer the reader to [9].

Table 1. Overview of the datasets used in our experiments.

Dataset	# images	# annotated phenotypes	Phenotypes
MNIST	21000	3	0,1,3
COOS	17144	3	Endoplasmic Reticulum, Inner Mitochondrial Membrane, Cytosol
Neurite	1536	3	Healthy, Damaged, Dead

4.2. Learning Approximate Archetypes

In order to learn the archetypes based on the hyperspherical representation Z we employ the second step (2) from our model (see Fig. 1). First, we predict the mixing coefficient matrix $A \in \mathbb{R}^{n \times k}$ with k being the number of archetypes from the latent representation Z. In addition, we define the dictionary $D \in \mathbb{R}^{k \times d}$ as a free parameter. In order to project the learned archetypes in D onto the hypersphere, we add an additional normalization constraint such that $||d_i||_2 = 1$ with d_i being the i^{th} row of the archetype matrix D. To optimize the aforementioned task, we formulate the following objective function:

$$\begin{aligned} &\min_{D,A} ||Z_i - DA_i||_2^2 \\ &\text{s.t.} \quad A_i \in \Delta, ||d_j||_2 = 1, \forall i, j \end{aligned} \tag{18}$$

where i denotes the i^{th} datapoint in X and j the j^{th} archetype in D. As a consequence, we aim to solve the following optimization problem to uncover our archetypes in an end-to-end fashion:

$$\begin{aligned} &\min_{\phi,\theta,D,A} I_\phi(Z;X) - \lambda(I_{\phi,\theta}(Z;X) + ||Z - DA||_2^2) \\ &\text{s.t.} \quad A_i \in \Delta, ||d_j||_2 = 1, \forall i, j \end{aligned} \tag{19}$$

4.3. Optimizing the Objective Function

Based on the model specified in Eq. (19), we provide an alternating optimization procedure to learn the latent representation Z while uncovering the archetypes D and the mixing coefficients A given this representation Z, simultaneously. We also tried to optimize the loss function (Eq. 19) without alternating optimization which, however, resulted in worse training results (data not shown). In more detail, our approach is formulated as follows:

In the first step, our model learns a low-dimensional representation Z of X by minimizing the mutual information between $I_{\phi,\tau}(X;Z)$ and maximizing the mutual information between $I_{\phi,\theta}(Z;X)$. At the same time, we keep both the mixing coefficients A and the archetype matrix D fixed (brown part of Eq. 20).

$$\mathcal{L}_1 = \min_{\phi,\theta} I_\phi(X;Z) - \lambda\Big(I_{\phi,\theta}(Z;X) + ||Z - DA||_2^2\Big) \tag{20}$$

In the second step, we learn the mixing coefficients A and the archetypes D (black part of Eq. 21) given the current representation of Z. To do so, we fix all model parameters except of A and D and update the parameters accordingly.

$$\mathcal{L}_2 = \min_{A,B} I_\phi(X;Z) - \lambda\Big(I_{\phi,\theta}(Z;X) + ||Z - DA||_2^2\Big) \tag{21}$$

The resulting loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ are alternately optimized until convergence.

5. Experiments

To validate the effectiveness of our approach to discover phenotypes in HCI, we conduct experiments on an artificial MNIST example and on two representative, industry-relevant HCI screening experiments. An overview of the datasets is shown in Table 1. Further details regarding the datasets are given in the following sections.

Metrics. For the MNIST experiment, we assess the performance of our model by quantitatively assessing the approximation error of the convex hull and qualitatively by comparing the found archetypes of all three methods. For both HCI screening experiments, we assess the model performance by quantitatively evaluating the mean squared error of the reconstructed image as a measurement of how well the latent representations are learned and qualitatively by comparing the found archetypes of all three methods. To do so, we take the closest image in the corresponding dataset to the respective archetype.

5.1. MNIST

Dataset. As biological experiments are often difficult to interpret, we aim to provide a proof of concept in an easy and interpretable setting to demonstrate the capability of

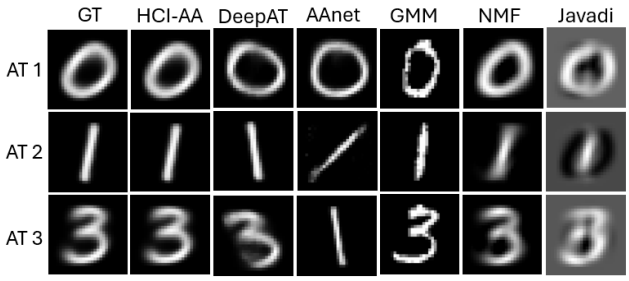

Figure 2. Visual representation of the found archetypes from the MNIST experiment, see Section 5.1. The rows describe the found archetypes whereas the columns depict the different methods.

Method	Mean Squared Error (MSE)
GMM	0.19
NMF	0.015
Javadi	0.27
DeepAT	0.171
AAnet	0.550
HCI-AA	0.002

Table 2. Quantitative summary of the results from the MNIST experiment. Here, we consider Archetypal Analysis, Coreset Archetypal Analysis and our HCI-AA model. For all methods, we measure the distance between groud truth and found archetypes by method in terms of mean squared error (MSE). Lower is better.

HCI-AA in depicting the correct archetypes. MNIST [20] consists of 70000 samples with images $x \in \mathbb{R}^{28x28x1}$ and labels y that represent numbers from 0 to 9. As an equivalent to biological phenotypes, we selected arbitrarily digests zero, one and three to represent the phenotypes in our dataset.

Baseline Methods. In this experiment, we employ five baseline methods in order to validate that HCI-AA is able to discover the correct phenotypes. To do so, we set the found archetypes on the latent representation Z of the full dataset using archetypal analysis [7] as ground truth. As baselines we compare to Gaussian Mixture Model (GMM), Factor Analysis (NMF) and Javadi [14] using the input features as well as DeepAT [16] and AAnet [35]. To ensure a fair comparision, we map all found archetypes into Z and calculate the mean squared error (MSE) between the ground truth and the respective baseline results.

5.2. Cells-out-of-Sample Dataset

Dataset. We examine the Cells-out-of-Sample (CooS) dataset [22], which includes 132,209 mouse cell microscopy images with $x \in \mathbb{R}^{64x64x2}$. Each image has two channels: the first stained with one of seven fluorescent proteins that targets a specific cell compartment, and the second consistently staining the cell nucleus. These proteins are also represented as labels y per image. Our experiment focuses on the first channel only and uses a subset of three different cell phenotypes: the Endoplasmic Reticulum, Inner Mitochondrial Membrane, and Cytosol.

Baseline Methods. Here, we use the two competitor nonlinear archetype methods [16] (DeepAT) and [11] (AAnet) and HCI-AA to compare whether and which of the methods are able to find a limited number of phenotypes from higher-dimensions and are able to learn meaningful latent representations. To perform a fair comparison, we make use of the same architecture and hyperparameters for all three methods. More details on the architecture and hyperparameter settings may be found in the appendix. However, we set the latent dimension to two as this is by design required to encode three archetypes in [16, 35]. In addition, we compare HCI-AA to GMM and NMF where the input features are the learned latent representation Z of HCI-AA. The visualized archetypes are images from the CooS dataset which are closest to the learned archetypes.

Method	CooS MSE	Neurite MSE
DeepAT	0.0284	0.0014
AAnet	0.0265	0.0014
HCI-AA	0.0072	0.0001

Table 3. Quantitative summary of the results from the CooS and NTR1 experiments. Here, we consider DeepAT, AAnet and our HCI-AA model. For all baselines, we measure the image reconstruction error in terms of mean squared error (MSE) as a proxy how well the representations had been learned.

5.3. Neurite Assay

Dataset. With the Neurite assay, the neuronal development and toxicity of neurons is studied where neurons that are derived from Induced pluripotent stem cells (IPSCs) are cultivated until they form networks. These cells are then further exposed to various compounds, and their toxicity is assessed through automated high-content imaging. From the captured images, which are represented as $x \in \mathbb{R}^{2048x2048x2}$, only the first fluorescence channel (Fluorescein isothiocyanate) is used for analysis. Using a non-overlapping sliding-window approach, 1536 image patches of 256x256 pixels each are extracted. These images are then manually categorized into three phenotype classes: healthy, damaged, and dead cells.

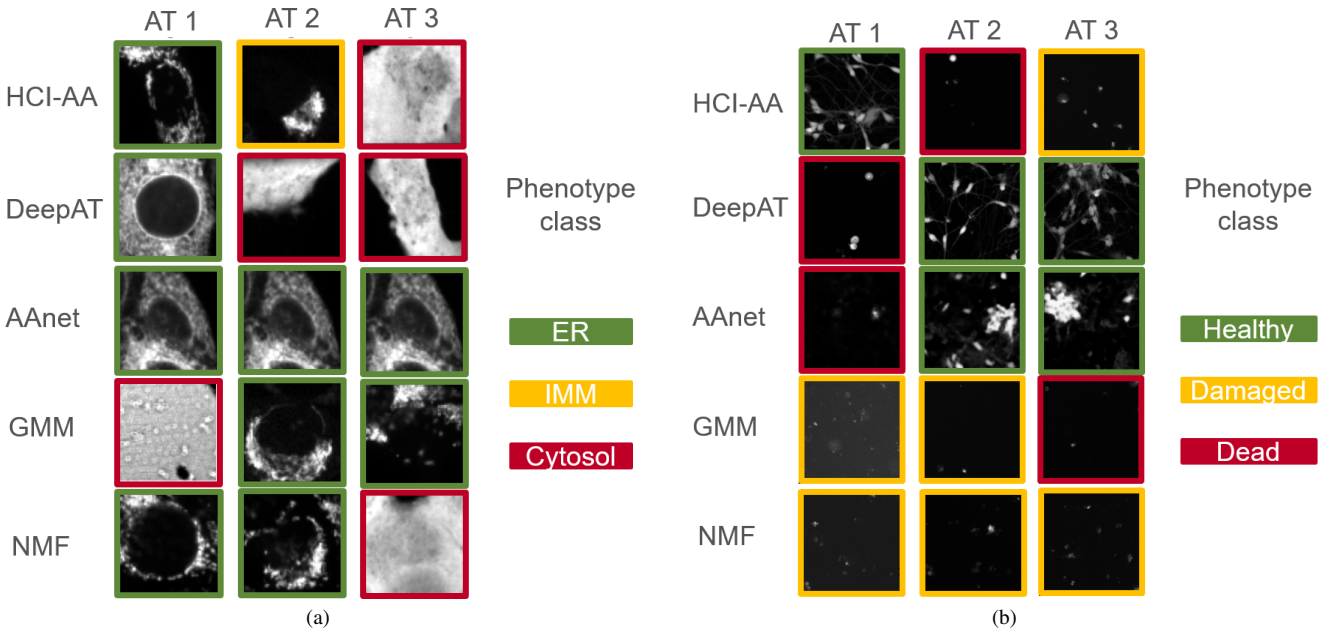

Figure 3. The found archetypes for the CooS (a) and Neurite (b) experiments. The first 3 columns denote the archetypes and the rows the corresponding baseline methods. The last column shows the color coding of the three different annotated phenotypes that are present in each assay. As shown in Table 1, both experiments exhibit three distinct phenotypes. In the case of CooS, these are the Endoplasmic Reticulum (ER), Inner Mitochondrial Membrane (IRR), and Cytosol. For Neurite, the phenotypes are Healthy, Damaged, and Dead neurons. It's important to note that only HCI-AA can unsupervised identify all phenotypes present in both assays as archetypes. All other baseline methods fail to map out the phenotypic landscape detecting one or two phenotypes only (see color coding). The illustrated images are the closest images from the corresponding dataset to the archetypes.

Baseline Methods. Equivalent to the CooS experiment in Section 5.2, we use the same two competitor methods [16,35] and HCI-AA to compare whether and which of the methods are able to find a limited number of phenotypes from higher-dimensions and are able to learn meaningful latent representations. In addition, we employ the same architecture and hyperparameters setting for all three methods to ensure a fair comparison. Similarly to the CooS experiment, we again set the latent dimension to two for both baseline methods [16, 35]. Similar to the CooS experiment, we compare HCI-AA to GMM and NMF baselines.

6. Discussion

HCI-AA reveals the correct archetypes in the MNIST dataset. We demonstrate that HCI-AA is in line with the traditional Archetypal Analysis and is able to uncover the ground truth archetypes in a dataset. In Figure 2, we show qualitatively that HCI-AA found archetypes that are close to the ground truth while all baseline methods perform worse in uncovering the ground truth phenotypes. This finding is further verified by our quantitative results in Table 2. We confirm that HCI-AA is with an MSE error of 0.002 close to the ground truth. In contrast, clustering approaches such as GMM obtain an MSE 0.19 which is up to two magnitudes higher. In addition, linear archetypal analysis on the input space [14] results in a MSE of 0.27. Moreover, DeepAT and AAnet as baseline competitors achieve an error of 0.171 and 0.550, respectively, being unable to uncover the true archetypes.

HCI-AA outperforms state-of-the-art approaches in representations learning while uncovering the correct phenotype. With two HCI screening experiments, we validated that HCI-AA learn meaningful representations which lead to the identification of reasonable phenotypes. More precisely, we compared the reconstruction errors (MSE) of HCI-AA as a proxy to two AA baselines, DeepAT and AAnet (Table 3). For the COOS experiment, HCI-AA obtained a MSE of 0.0072 whereas DeepAT and AAnet received a MSE of 0.0284 and 0.0265, respectively which is over one magnitude lower. These results are confirmed by the Neurites experiment. Here, HCI-AA has a MSE of 0.0001 compared to DeepAT and AAnet with a MSE of 0.0014. This large reconstruction error for both DeepAT and AAnet serves as a reliable proxy that a two-dimensional latent representation is not sufficient to capture the variation in the data and, thus, is unable to learn a meaningful representation which is required to uncover the correct archetypes. In addition, we

qualitatively assessed the found archetypes in Figure 3. To do so, we mapped all images from the dataset into the latent face. Subsequently, we selected the three images that have been closest to the corresponding archetypes. In contrast to DeepAT and AAnet, HCI-AA was able to uncover all three biologically expected archetype classes that are present in the datasets. These quantitative and qualitative results demonstrate that solely HCI-AA is able to learn meaningful representations while being able to uncover a low number of archetypes in higher dimensions.

HCI-AA uncovers meaningful biological phenotypes from high content images. In high content imaging (HCI) extremal cell phenotypes are of utmost interest due to the nature of the experiments where control phenotypes are frequently described as extremal phenotypes [29]. In addition, in HCI screening protocols phenotype outlier discovery and their corresponding analysis are also part of common HCI analysis pipelines [30].
However, as illustrated in Figure 3a and 3b, neither DeepAT and AAnet are able to reliably uncover the relevant phenotypes in both HCI screening experiments. In contrast, HCI-AA discovers all three archetypes in a higher dimensional space as it relaxes the simplex constraint of [16, 35] (Table 1 for an overview of the present phenotypes). In addition, both clustering (GMM) and factor analysis approaches (NMF) also fail to detect all three phenotypes in the HCI experiment as all baseline methods uncover maximal one or two of the phenotypes present. Thus, these method can not be used in routine analysis of HCI data missing potentially important phenotypes.
These findings are further supported by the neurite assay screening experiment. Here, the DeepAT and AAnet fail to identify the 'damaged' phenotype, which is a key endpoint for toxicity testing. More specifically, such assays are used in drug safety testing and overlooking a toxic phenotype during the analysis can have serious downstream implications, e.g., for the health of individuals participating in subsequent clinical drug trials [10]. At the same time, both GMM and NMF fail to identify the 'healthy' phenotype in screening experiment.
In addition to the phenotype discovery performance of HCI-AA, the method enables production of synthetic image data from outlier phenotypes, which are usually rare events and hard to collect [13, 30]. The synthetic data can therefore be used to train deep neural networks to identify such rare event outliers of interest in subsequent data acquisition runs.

7. Conclusion

While automated HCI screening protocols have helped to accelerate the discovery of novel treatment strategies, it is still challenging to perform an fully-automated analysis of the experimental results. In this work, we have introduced an approach that unifies online dictionary and representation learning in an end-to-end fashion in order to uncover phenotypes in high-dimensional HCI images. In contrast to previous approaches, our method does not require on $k-1$ latent dimensions to uncover k archetypes and thus, enabling us to detect a low dimensional number of phenotypes in a higher dimensional latent space. In our experiments, we validated that our methods uncovers the biologically relevant phenotypes on an artifical experiment and on two industry-relevant HCI screening data sets coming from drug finding campaigns. Despite being designed for HCI screening applications, our approach may be applied to other application areas where number of required latent dimensions exceed the number of archetypes.

Limitations Despite the various advances that our method offers, it also contains some limitations that we would like to address in future. First, our method builds on the work of [24] and is hence only able to find approximate archetypes in contrast to methods that consider the entire dataset [7]. Second, our method requires prior knowledge about the number of archetypes. Despite, we emphasize that this is a minor issue in HCI, as the number of phenotypes (archetypes) is typically known beforehand, especially in phenotypic and toxicity safety screening scenarios [33]. For future work, we would like to investigate solutions to select the number of archetypes automatically in a data-driven fashion similar to [26]. Finally, in certain cases it may be difficult to interpret the latent archetypal space as the inputs can be mapped to arbitrarily latent representation due to the high flexibility of neural networks. To overcome these shortcoming, we aim to incorporate additional invariances to better steer the latent mapping in future work [37]. Last, we plan to investigate HCI-AA on more sophisticated representation learning techniques such as [5].

References

[1] A. Achille and S. Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[2] Michal Aharon and Michael Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, 2008. 3

[3] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*. 2017. 2

[4] C. Bauckhage and K. Manshaei. Kernel archetypal analysis for clustering web search frequency time series. In *2014 22nd International Conference on Pattern Recognition*, pages 1544–1549, Aug 2014. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020. 8

[6] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, and others. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 2006. 1

[7] Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994. 1, 2, 3, 6, 8

[8] Adele Cutler and Emily Stone. Moving archetypes. *Physica D: Nonlinear Phenomena*, 107(1):1–16, August 1997. 2

[9] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *Uncertainty in Artificial Intelligence*, 2018. 2, 4

[10] Johannes Delp, Simon Gutbier, Stefanie Klima, Lisa Hoelting, Kevin Pinto-Gil, Jui-Hua Hsieh, Michael Aichem, Karsten Klein, Falk Schreiber, Raymond R Tice, et al. A high-throughput approach to identify specific neurotoxicants/developmental toxicants in human neuronal cell function assays. *Altex*, 35(2):235, 2018. 8

[11] David van Dijk, Daniel B. Burkhardt, Matthew Amodio, Alexander Tong, Guy Wolf, and Smita Krishnaswamy. Finding archetypal spaces using neural networks. In *IEEE International Conference on Big Data*, 2019. 1, 2, 6

[12] William J. Godinez, Imtiaz Hossain, Stanley E. Lazic, John W. Davies, and Xian Zhang. A multi-scale convolutional neural network for phenotyping high-content cellular images. *Bioinformatics (Oxford, England)*, 2017. 1

[13] Peter Goldsborough, Nick Pawlowski, Juan C Caicedo, Shantanu Singh, and Anne E Carpenter. Cytogan: generative modeling of cell images. *BioRxiv*, page 227645, 2017. 8

[14] Hamid Javadi and Andrea Montanari. Non-negative matrix factorization via archetypal analysis, 2017. 6, 7

[15] Dinu Kaufmann, Sebastian Keller, and Volker Roth. Copula archetypal analysis. In Juergen Gall, Peter Gehler, and Bastian Leibe, editors, *Pattern Recognition*, pages 117–128. Springer International Publishing, 2015. 2

[16] Sebastian Mathias Keller, Maxim Samarin, Fabricio Arend Torres, Mario Wieser, and Volker Roth. Learning extremal representations with deep archetypal analysis. In *International Journal of Computer Vision*. 2021. 1, 2, 3, 4, 6, 7, 8

[17] Sebastian Mathias Keller, Maxim Samarin, Mario Wieser, and Volker Roth. Deep archetypal analysis. In *German Conference on Pattern Recognition*. 2019. 1, 2, 3, 4

[18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 4

[19] Stephan Kruger, Matthias Ilmer, Sebastian Kobold, Bruno Loureiro Cadilha, Stefan Endres, Steffen Ormanns, Gesa Schuebbe, Bernhard Renz, Jan D'Haese, Hans Schlößer, Volker Heinemann, Marion Subklewe, Stefan Boeck, Jens Werner, and Michael von Bergwelt. Advances in cancer immunotherapy 2019 - latest trends. *Journal of Experimental and Clinical Cancer Research*, 38, 12 2019. 1

[20] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998. 6

[21] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, 2006. 3

[22] Alex X. Lu, Amy X. Lu, Wiebke Schormann, David W. Andrews, and Alan M. Moses. The cells out of sample (COOS) dataset and benchmarks for measuring out-of-sample generalization of image classifiers. *Advances in Neural Information Processing Systems*, 2019. 6

[23] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, 2009. 2, 3

[24] Jieru Mei, Chunyu Wang, and Wenjun Zeng. Online dictionary learning for approximate archetypal analysis. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 4, 8

[25] Morten Mørup and Lars Kai Hansen. Archetypal analysis for machine learning and data mining. *Neurocomputing*, 80:54–63, 2012. 2

[26] Sandhya Prabhakaran, Sudhir Raman, Julia E. Vogt, and Volker Roth. Automatic model selection in archetype analysis. In Axel Pinz, Thomas Pock, Horst Bischof, and Franz Leberl, editors, *Pattern Recognition*, pages 458–467. Springer Berlin Heidelberg, 2012. 2, 8

[27] Sohan Seth and Manuel J. A. Eugster. Probabilistic archetypal analysis. *Machine Learning*, 102(1):85–113, Jan 2016. 1, 2, 3

[28] O. Shoval, H. Sheftel, G. Shinar, Y. Hart, O. Ramote, A. Mayo, E. Dekel, K. Kavanagh, and U. Alon. Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space. *Science*, 336(6085):1157–1160, 2012. 1

[29] Daniel Siegismund, Mario Wieser, Stephan Heyse, and Stephan Steigele. Self-supervised representation learning for high-content screening. In *International Conference on Medical Imaging with Deep Learning*, 2022. 1, 8

[30] Christoph Sommer, Rudolf Hoefler, Matthias Samwer, and Daniel W Gerlich. A deep learning and novelty detection framework for rapid phenotyping in high-content screening. *Molecular biology of the cell*, 28(23):3428–3436, 2017. 8

[31] Stephan Steigele, Daniel Siegismund, Matthias Fassler, Marusa Kustec, Bernd Kappler, Tom Hasaka, Ada Yee, Annette Brodte, and Stephan Heyse. Deep Learning-Based HCS Image Analysis for the Enterprise. *SLAS DISCOVERY: Advancing the Science of Drug Discovery*, 2020. 1

[32] Emily Stone and Adele Cutler. Introduction to archetypal analysis of spatio-temporal dynamics. *Physica D: Nonlinear Phenomena*, 96(1-4):110–131, September 1996. 2

[33] Fabio Stossi, Pankaj K Singh, Kazem Safari, Michela Marini, Demetrio Labate, and Michael A Mancini. High throughput microscopy and single cell phenotypic image-based analysis in toxicology and drug discovery. *Biochemical Pharmacology*, 216:115770, 2023. 8

[34] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 2

[35] David van Dijk, Daniel B. Burkhardt, Matthew Amodio, Alexander Tong, Guy Wolf, and Smita Krishnaswamy. Finding archetypal spaces for data using neural networks. *IEEE Big Data*, 2019. 6, 7, 8

[36] Aleksander Wieczorek, Mario Wieser, Damian Murezzan, and Volker Roth. Learning Sparse Latent Representations with the Deep Copula Information Bottleneck. In *International Conference on Learning Representations*. 2018. 2

[37] Mario Wieser, Sonali Parbhoo, Aleksander Wieczorek, and Volker Roth. Inverse learning of symmetry transformations. *arXiv e-prints*, page arXiv:2002.02782, 2020. 8