HOT SINGLE CELLS IN YOUR GENE EXPRESSION SPACE

An introduction to archetypal analysis, guided by biology

Amulya Garimella, Nadine Han

December 13, 2021

Cover Letter

Over the course of this project, we received really helpful advice from our TF mentor, Caleb Miller, along with our peer reviewers, Addie Kelsey, Kamryn Ohly, Michael Hu, and Eion Chua. A lot of the advice was to further demonstrate the math behind archetypal analysis and add proofs to explain results like the RSS minimization equation. We wanted this to be a biology-oriented math paper instead of a math-oriented biology paper, and the advice from our mentor and peer reviewers was extremely helpful in achieving this goal.

We were also advised to further explain concepts, particularly PCA and its differences to archetypal analysis, as well as RSS minimization. We more clearly articulated the difference between archetypal analysis and PCA. We also fully expanded and explained the section relating to RSS minimization with diagrams and an in-depth technical walkthrough, and further expanded by including an additional proof that explains how to select archetypes in order to minimize RSS.

We had several meetings with Caleb, our mentor, to discuss the paper and help us understand rather densely-written proofs that presumed a lot of background knowledge we didn't have (particularly in one of our core papers, Cutler & Breiman) (thank you so much Caleb!!!). Part of our value added comes from breaking down these complicated concepts and unfamiliar geometry, and walking our readers through the math with care. In particular, we noticed as we read other papers in the field that there appeared to be a lack of real or accessible explanation of the archetypal analysis concepts introduced by Cutler and Breiman - even in their own paper - despite their paper being considered foundational to archetype analysis.

Of course, a lot of our value added also comes from showing numerous examples of real-world applications of the math and explaining the underlying assumptions of archetypal analysis that become important in a real-world context. We use the biological example of functional analysis of single-cell gene expression data throughout the paper to provide a concrete example that readers can keep coming back to, and ground the mathematical discussion. The biological thread culminates in the final two sections. The first of these sections briefly discusses a paper where researchers tested archetypal analysis on single-cell data to see if the data is well-described by the method. In the final section, we actually create an original example with R and apply archetypal analysis.

Thank you so much for reading our paper. We really appreciate your time, effort, and care — and most importantly, hope you enjoy!!!

1 Introduction

1.1 General Background

All the cells in our body share the same genome. So how come, say, an eye cell is different than a skin cell? Why aren't humans just homogeneous blobs?

The answer lies in gene expression: how different cells come to produce different proteins. Various protein-producing segments of the genome (genes) are regulated to produce different types of proteins in different amounts depending on their function. Through experimental biology methods, we answer the question: out of around 20,000 genes on the human genome, how much of each gene is being produced in a given tissue? (To describe "how much of each gene is being produced," we'll use the term "**gene expression**.")

Until just a few years ago, researchers relied on "bulk-tissue techniques", in which the aforementioned gene expression data was collected from homogenized samples of tissue. Homogenized tissue samples are mixed and blended such that each part of the sample is equally representative of the whole.

This hopefully seems pretty straightforward, but unfortunately, pesky biological reality gets in the way. Biological tissue is usually heterogeneous, containing many different types of cells that have different functions in the body. This complicates matters: when we analyze homogenized tissue we're ignoring the possibility that one cell type might be regulated a whole lot differently than another (because cells of different types can do wildly different things!).

Fancy new single-cell sequencing techniques allow us to separate, extract, data from, and analyze *individual* cells. We can then use that information to sort the millions of cells in a given tissue into certain cell-type categories, and look for broad trends within those cell-type categories.

But there's still a pretty significant piece of the puzzle missing...

1.2 The Problem

Let's say we're given some biological data. Suppose this data takes the form of a vector consisting of expression values for genes from a certain cell.

With no other information, **how do we figure out that cell's type?** Single-cell sequencers can't, unfortunately, automatically tell us WHAT a given cell is — only what's inside it. It's up to us, and our computers, to find out!

2 Archetypal Analysis

We can model the problem given in the introduction using the method of archetypal analysis, which is a helpful way to model single cells for reasons that will soon become clear!

First, let's discuss what we mean by an "archetype." The concept of archetypes might have made an appearance in your English class. We can ascribe to an archetype a standard set of traits that correspond wholly to a certain character type, and then consider some arbitrary character's traits in relation to that archetype. For example, Walter White (the protagonist of Breaking Bad) has some of the characteristics of a classic Hero™, but lacks many other important characteristics; thus, we call him an antihero.

What does this have to do with single-cell gene expression? Let's say we want to define an archetype for a certain biological function. Just like in English class, our cell-type archetype might consist of a bunch of traits that purely represent that one specific biological function or cell type — perhaps a gene expression profile of a cell that would be optimal for carrying out a certain task. (We can apply this concept to all kinds of data, but let's run with the single-cell example.)

How do we come up with these archetypes? They are actually derived from the data: as Cutler & Breiman [3] say,"the archetypes themselves are not wholly mythological, since each is constrained to be a mixture of points in the data." This is important to keep in mind: defining archetypes this way differentiates archetypal analysis from other methods for finding patterns in data, like Principal Component Analysis (PCA). Without going into too much detail, PCA is a technique that, in short, can distill a big dataset down to a set of principal components (PCs), by defining PCs as linear combinations of the data and using those PCs as basis vectors. This is similar to archetypal analysis except, instead of PCs, archetypal analysis uses specially constrained linear combinations (aka archetypes). The main difference between PCs and archetypes lies in interpretability, as Chan et al. [2] note: "eigenvectors determined by principal components analysis and its relatives generally do not resemble any member of the data and may be difficult to interpret," while archetypes in archetypal analysis do resemble real data points. This is due to the unique geometric properties of archetypes. Consequently, archetypes can be more helpful to researchers who are interested in characterizing cells according to their functions, as the rest of the data is plotted relative to points that clearly resemble real data.

This trait of archetypal analysis is one of the reasons why this method works so well to model single cells. We can find cell archetypes and then interpret a given archetype's gene expression profile (e.g. an archetype for a stem cell might have a lot of highly expressed division-related genes).

OK, but what does it actually mean to "represent the cell vectors" with archetypes, or to "derive" archetypes with a "mixture of points in the data"? As you may have intuited, the answer is linear combinations — but spiced up a little bit. Meet the **convex combination**, which is just a special, constrained type of linear combination:

Definition 1. *Consider a set of vectors $x_1, ...x_n$. A **convex combination** is a linear combination of these vectors $\sum_{i=1}^{n} \alpha_i x_i$ where all $\alpha_i \geq 0$ and where $\sum_{i=1}^{n} \alpha_i = 1$.*

In summary (and in a non-biology context), Eugster [4] outlines that, given a multivariate set of data represented as an $m \times n$ matrix X and some number p, archetypal analysis finds a matrix Z of p n-dimensional archetypes such that:

1. Each data point is a convex combination of the archetypes.
2. Each archetype is a convex combination of the data points.

2.1 Convex Polytopes

As we established earlier, each data point is a convex combination of the archetypes. Since every possible convex combination of two points is on the line segment connecting those two points, we can actually represent archetypes as the vertices of an n-dimensional shape with p vertices, where n is the dimension of the datapoints, and p is the number of archetypes (note that the dimensionality of the shape is not necessarily the same as the number of archetypes; for example, a triangle in 2 dimensional space has 3 vertices). We'll call the shape generated by our archetypes a "convex polytope."

Definition 2. *A **polytope** is an n-dimensional generalization of a polyhedron (which is, in turn, a 3D polygon, e.g. cube). We can call a polygon a 2-polytope, a polyhedron a 3-polytope, etc.*

Note that a low-dimensional object can easily exist in a high-dimensional space. For example, consider a plane in $\mathbb{R}^3$.

Our data points should fall within the polytope and on the lines that connect the polytope's vertices (this makes sense; recall that archetypes act as basis vectors, or axes, with which to represent data points, since each data point is a constrained linear combination of the archetypes). It then becomes helpful for us to identify a specific type of convex polytope:

Definition 3. *A **convex hull** is the smallest polytope that encloses all points in a given set (i.e. its boundary can't be arbitrarily defined to be larger than the set itself). It has a **boundary** in which each vertex is defined as an archetype, and which contains the most extremal points of the dataset. The boundary and the interior region it contains comprise the convex hull.*

But what does this really mean? In the context of archetypal analysis, and in the larger picture of cell characterization, a convex hull is essentially a model designed to represent a set of data by indicating how similar each datapoint is relative to a set of special datapoints that represent primary functions or characteristics (AKA archetypes). Here, we shall introduce an important note to keep in mind throughout the rest of the paper: **Intuitively, we hope to create the most accurate model, or convex hull that is the *most* representative of the dataset.**

Knowing this information, the next logical step would be to understand *how* to determine the archetypes that make up a convex hull. In order to do so, let's say that we have an arbitrary set of data points that looks like this:

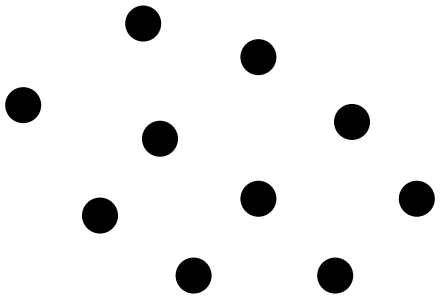

Figure 1: Simple dataset

With this data, let's construct a convex hull that contains all of the data points:

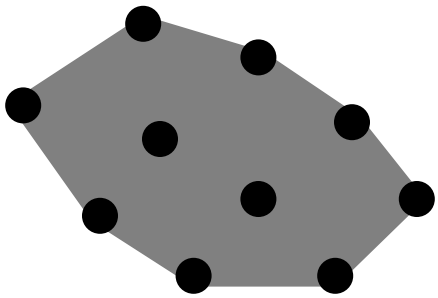

Figure 2: Convex hull with p archetypes where $p = 8 = N$

As displayed in this example, one way to define archetypes is to set all pre-existing vertices on the boundary of the convex hull as archetypes. Hence, if there are N points on the boundary of the convex hull, and we choose to use all of these points as archetypes, all of the other data points in the set can be represented by these N archetypes.

Sometimes, however, this strategy isn't the most practical - in massive datasets where N is a gigantic number, there are probably scenarios in which not *all* of the vertices are needed in order to represent the data. We can think of this as "estimating" a simpler version of the original convex hull. To introduce another way of finding archetypes, let's define a given number of archetypes less than N as p. We construct a new convex hull that may not contain all the points in the original dataset.

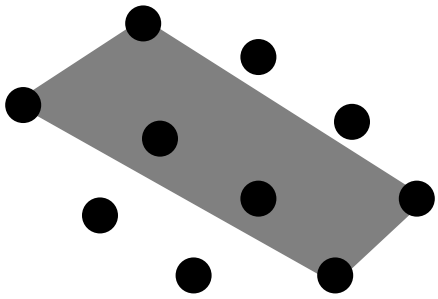

Figure 3: Convex hull with p archetypes where $p = 4 < N$

Recalling that our goal is to make the convex hull as accurate as possible while using p archetypes, how can we mathematically determine the error for the points outside of the new hull? Is there some value we can optimize to create the best archetypal representation of data with p archetypes?

The answer lies in statistics! In order for archetypal analysis to generate the

most representative archetypes of a set of data, we can look at the **Residual Sum of Squares** as a metric.

Definition 4. *Given a regression model and a set of data, the* ***Residual Sum of Squares (RSS)*** *is defined as the sum of squared residuals. A* ***residual*** *is defined as the difference between a measured value x_i in a set of data (with m values) and the corresponding predicted value from a regression model $\hat{x_i}$. Hence, the general formula for RSS:*

$$RSS = \sum_{i=1}^{m} (x_i - \hat{x_i})^2$$

In layman's terms, a RSS is an overall value that measures how off a regression model is from accurately representing the data points it is generated from. It's used to quantify the fit of a line to data, and since convex hulls produce line segments upon which we can project datapoints, we can adapt it to work well for this problem.

In this case, the "regression model" is the line segments making up the convex hull generated by the archetypal analysis algorithm, while the data points are the cells containing gene expressions. As we will show, to get "predicted values," we project datapoints outside of the hull onto the hull.

Now that we know what metric we must use, we can determine how to calculate it mathematically by deriving the RSS formula for archetypal analysis. To do so, let us visualize the same set of data points with a 4-archetype convex hull:

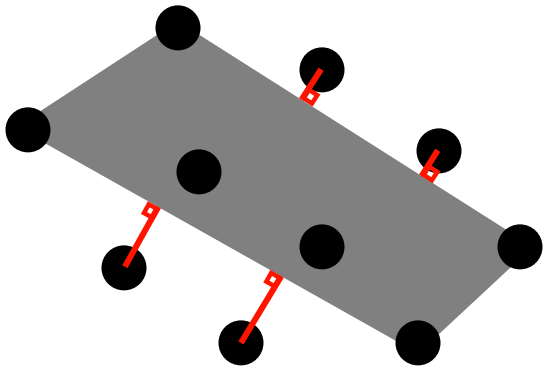

Figure 4: Convex hull with p archetypes where $p = 4 < N$, with datapoints' distance from the hull marked in red

Let's denote all the datapoints in this figure as $x_1, \ldots, x_m$ with the points outside of the convex hull being $x_1, \ldots, x_4$ and their respective distances to the

convex hull $d_1, \ldots, d_4$. Let's also denote our archetypes (the points at the vertices of the polygon) $z_1, \ldots, z_p$.

Remembering that a convex hull is defined by its vertices or archetypes, and that each datapoint is represented as a mixture of the archetypes, we *should* be able to represent each datapoint as a convex combination of the archetypes:

$$x_i = \sum_{k=1}^{p} \alpha_{ik} z_k$$

Where α is just a weight/coefficient. **Note that this equation doesn't hold for $x_1, \ldots, x_4$, since they're outside the convex hull.** Therefore, in order to represent these points in terms of the convex hull, we estimate the points by taking their orthogonal projection onto the hull. The results are shown here:

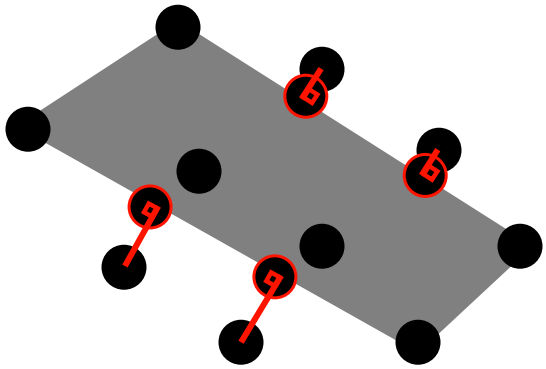

Figure 5: Convex hull with p archetypes where $p = 4 < N$, with datapoints' distance from the hull marked in red and new estimated datapoints outlined in red

These new datapoints *would* be convex combinations of the archetypes, since they're on the convex hull. Therefore, the expression $\sum_{k=1}^{p} \alpha_{ik} z_k$ can refer to either (1) an existing datapoint within the convex hull or (2) an estimation of an excluded point, which is derived by projecting the datapoint onto the convex hull.

With this information, we can now adapt the RSS formula we introduced earlier. We'll start by replacing $x - \hat{x}$ with a value d that represents the distance between the original and projected (estimated) datapoints.

$$RSS = \sum_{i=1}^{n} d_i^2$$

And we can represent d to be the length (or norm) of the difference between

the original and projected vectors:

$$\text{For datapoint } x_1,\ d_i = ||x_i - \sum_{k=1}^{p} \alpha_{ik} z_k||$$

We combine the statements to get the RSS formula for archetypal analysis!

Definition 5. *Given a set of n-dimensional data points $x_1, \ldots, x_m$ and a set of archetypes for these data points $z_1, \ldots, z_p$, the* ***Residual Sum of Squares (RSS) formula for archetypal analysis*** *can be written as:*

$$\sum_{i=1}^{m} ||x_i - \sum_{k=1}^{p} \alpha_{ik} z_k||^2$$

Where i is some number in the set 1, ... , m and k is the corresponding number in the set 1, ... , p.

Now that we understand the derivation of RSS for archetypal analysis, let's go back to our example where the number of archetypes we use p is less than N, where N is the number of vertices on the boundary of the original dataset. We can calculate the RSS using all of the points that lie outside the polytope (because, remember, the points inside the polytope can be perfectly represented as a convex combination of the archetypes!). Essentially, the RSS represents the squared distances of outside points from the hull combined. The smaller the RSS, the smaller the distance and thus the more accurate the archetypes are at representing the dataset. Remember that we hope to generate the most accurate archetypes possible, which thus presents us with an optimization problem: we need to find some vertices bounding a polytope that minimize RSS.

2.2 Minimizing RSS

Using our new knowledge of RSS, we can revisit the scenarios given earlier, which detail the three possibilities for how many archetypes we can choose, as well as how we can minimize RSS given each option:

Proposition 1. *Consider a dataset consisting of datapoints $x_1, \ldots, x_n$. Say that C represents the convex hull of that entire dataset; S is the set of datapoints on the boundary of that convex hull; N is the cardinality of S (i.e., the number of datapoints on the boundary of the convex hull encapsulating the entire original dataset).*

1. *If $p = 1$, then choosing an archetype that equals the mean of the samples minimizes RSS.*
2. *If $1 < p < N$, then there exists some set of p archetypes on the boundary of the convex hull that minimizes RSS.*
3. *If $p = N$, then RSS $= 0$.*

Let's take a stab at proving this statement!

Proof. **Part (1)** is relatively intuitive. In comparison to simply choosing an arbitrary point, or using other statistical measures such as a median, the mean considers every data point with an equal amount of importance, as per its formula. Hence, the mean minimizes the distance from each individual point to the archetype point.

To prove part (2), we want to prove that we can (and must) pick a set of archetypes specifically on the boundary of the dataset's convex hull in order to minimize RSS. Let's say for the sake of contradiction that we have a set of archetypes $z_1, \ldots, z_p$ where z_1 is strictly in the interior of the convex hull and $z_2, \ldots, z_p$ are on the boundary of the convex hull. And let's assume (wrongly) that this set of archetypes does, in fact, minimize RSS.

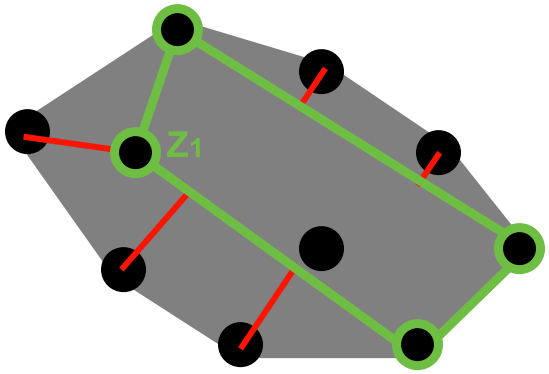

Figure 6: Our archetypes, outlined in green, along with the convex hull they create. Excluded points' distances from the hull in red. The original convex hull, in which we set $p = N$, is colored in gray in order to illustrate the boundary of the dataset.

Now, let's consider a ray going to z_1 from one of the other archetypes. We define:

$$r(t) = z_j + t(z_1 - z_j)$$

Where $j \neq 1$ and $t > 1$. We know that there must be a another point on this ray besides the archetype that lies on the boundary of the dataset — so let's define t such that $r(t)$ lies on the boundary of the dataset.

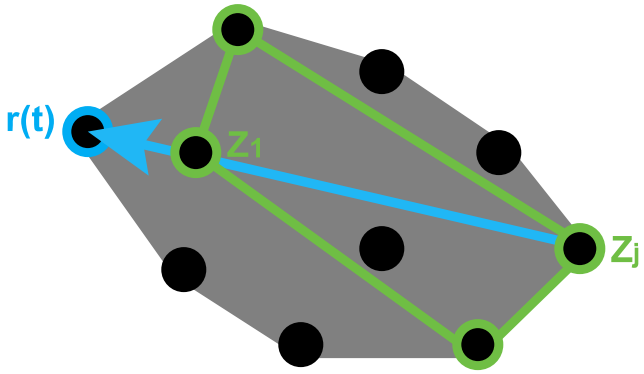

Figure 7: Our archetypes, outlined in green, along with the convex hull they create. A ray from z_1 to z_j is shown, along with the point on the ray that lies on the boundary of the dataset, $r(t)$.

So let's consider again the set of archetypes $z_1, \ldots, z_p$, and now a new set of archetypes $r(t), \ldots, z_n$. Since z_1 is strictly in the interior of the convex hull, that must mean that there is at least one datapoint in the entire set that CAN'T be captured by $z_1, \ldots, z_p$. Since $r(t)$ is on the boundary of the hull, $r(t), \ldots, z_p$ would capture that point.

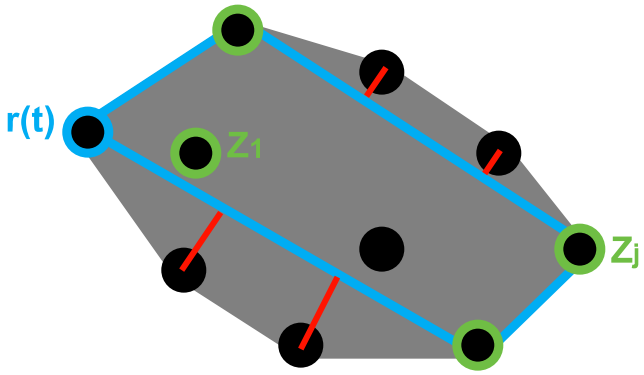

Figure 8: The new archetypes with $r(t)$ replacing z_1, along with the convex hull they create. Excluded points' distances from the new hull in red.

Finally, **part (3) is trivial**. We set our archetypes $z_1, \ldots, z_N$ such that each archetype is an existing datapoint on the boundary of the convex hull. Then, all of the points in the dataset will be within the hull produced by $z_1, \ldots, z_N$, leading to an RSS of 0! □

2.3 Selecting p

You might be wondering why we can't just select all of the extremal points of a given dataset as our archetypes. And...we hypothetically can, but consider how we'd actually apply archetypal analysis in real life. Single-cell sequencing of any given tissue yields *millions* of datapoints (which, remember, correspond to millions of cells)! Plotting these millions of cells in gene expression space space, we may have hundreds of extremal points — and using all of these points as archetypes would be counterproductive. We might expect the tissue we sequenced cells from to only have a few cell types, and so simply using all the points that happen to be on the boundaries of the convex hull would likely yield redundant results that'd be hard to interpret.

That's why we use archetype analysis to *estimate* what the convex hull looks like. In real life, we'd try to find best-fit polytopes with different numbers of vertices (we could hazard a few guesses for vertex number by starting from our biological knowledge of how many cell types might be represented in our dataset), and then see which type of polytope minimizes RSS. Check out sections 3 and 4 for a real-life example application of this idea.

2.4 The Final Step

We're almost there at a specific application! Before walking you, the reader, through archetypal analysis of real world data, we must tie up one remaining loose end: the solution to the RSS optimization problem in the case of scenario (2) outlined in Section 2.2, in which $1 < p < N$. Right now, minimization appears as a black box in which we plug a set of data points into an archetypal algorithm, and get out the archetypes.

To understand a bit more about the optimization process, we can revisit Cutler and Breiman [3] from before. In order to minimize RSS, they utilize an alternating constrained least squares algorithm, which switches between finding the optimal αs for a set of archetypes and finding the optimal archetypes for a set of αs to create the most representative convex hull with p archetypes. Let us walk through this process mathematically in order to further our understanding:

First, suppose there is a given set of p archetypes, where p is dictated by the user of the algorithm. We represent them as $z_1, dots, z_p$, and perhaps derive them by taking an arbitrary number of points on the outer bound of the dataset. To find the best αs for this set of archetypes, we then input it into the RSS formula for each datapoint:

$$\sum_{i=1}^{n} ||x_i - \sum_{k=1}^{p} \alpha_{ik} z_k||^2$$

Where x_i is the observed datapoint, and $\sum_{k=1}^{p} \alpha_{ik} z_k$ is the estimated datapoint. Then, after finding these ideal αs, we can fit the ideal archetypes to these αs. We fit one archetype at a time to the αs, and assume that the rest are held constant. We then repeat this process for each entry in the set of p archetypes, optimizing each one to get our finished set of archetypes!

2.5 Summary

When we use archetypal analysis, we're assuming that our data fits into a convex polytope; that is, we can plot all the data points and construct a "hull" around it (aka a polytope) containing all the extremal points such that all points are within that hull — either on the boundary edges and faces, or within the polytope itself. Archetypal analysis lets us *estimate* that polytope when the real boundary may be too complicated.

3 Into the Wild!

3.1 Why Use Archetypal Analysis?

Now, the goal is to place archetypal analysis in a real world context. Before we jump into a direct application, we must first discuss any underlying assumptions that archetypal analysis requires. This can be broken down into two questions: (1) *why archetypal analysis is structured the way it is*, and (2) *why it's applicable to biology.*

3.1.1 Trade-offs and Pareto Optimality

First, let us discuss one way of conceptualizing archetypal analysis' structure. Every decision we make in life has an opportunity cost, or the lost opportunity to complete another task. This can be due to a variety of reasons, but the primary two often relate to time and energy; for example, for every hour I dedicate to my Expos essay, I could have spent the same amount of time writing this math paper (oof).

This concept also applies to biology, but instead of time being the limiting factor, it's specificity. Many biological systems involve trade-offs in favor of specificity; for example, in an evolutionary context, certain aspects of animals can fall upon a spectrum of physical traits. A bird cannot have both a perfectly curved and a perfectly straight beak, but its beak can be characterized on a spectrum between the two, with some birds having more curved beaks and others having straighter ones.

For the sake of using a more intuitive example to qualify the nature of trade-offs in archetypal analysis, we can further look to a paper by Shoval et al. [6] on evolutionary trade-offs. In their research, Shoval et al. introduce the concept of **Pareto curves** as a geometric representation of archetypes.

Definition 6. *A **Pareto front**, also referred to as a **Pareto curve**, is a polytope with archetypes as vertices that represents the best trade-offs between primary functions or archetypes. This curve is created by removing all data points that have superior alternatives, or that are outperformed in all archetypes by another existing data point.*

As a visualization, we illustrate a two-archetype and a three-archetype Pareto front whose archetypes are the standard basis vectors of $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively:

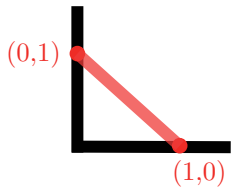

Figure 9: A two-archetype Pareto front whose archetypes are the basis vectors of $\mathbb{R}^2$

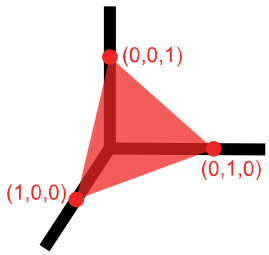

Figure 10: A three-archetype Pareto front whose archetypes are the basis vectors of $\mathbb{R}^3$

To bring linear algebra more firmly into the picture, Shoval et al. utilized a Pareto front with three archetypes, hence creating the shape of a triangle. They conclude that, because a triangle defines a plane, higher dimension data will eventually collapse onto two dimensions.

But why does this matter in the context of biology? Because of the way the convex hull is constructed, we can imagine each line segment making up the hull as a Pareto front between two archetypes. Any cell within the convex hull is, depending on its position, trading off similarity to one archetype for similarity to another. Archetypal analysis, by nature, presumes that the points within a convex hull cannot be simultaneously optimized for two tasks at once. This is the motivation behind the math of archetypal analysis.

Throughout this paper, we've assumed that it's possible to represent cells as combinations of archetypes. Now we want to address this assumption and show that yes, real cells do have gene expression patterns that fall within these geometrical constructions.

3.2 The Real World

Now that we have a good understanding of archetypal analysis, let's take it for a spin! We'll dive into a real-world example of archetypal analysis for categorizing single cells, walking you through the 2015 paper "Geometry of the Gene Expression Space of Individual Cells" by Korem et al. [5] We'll be taking you through the methods and concepts the researchers used to implement archetypal analysis on a variety of different single-cell datasets.

The researchers tested multiple different biological datasets to see if a low-dimensional, simple polytope (i.e. an n-dimensional polytope with a low number of vertices where n is also a low number) can describe the dataset well. After generating a best-fit polytope, they examine the gene expression profiles of the archetypes (at the vertices of the polytopes) and find that they correspond to certain cell types.

The assumption underlying their analysis is that archetypes should represent cells with the "ideal" expression profile for the performance of a certain task, and that each real cell in the dataset lies relative to the archetypes according to the task it performs in the body. Thus, we assume, we can categorize, sort, and better understand the cells in the dataset by examining them relative to the archetypes we generate.

3.3 The Gene Expression Space

In order to test the theory of characterizing single cells with low dimension polytopes, Korem et al. use multiple methods and types of cells, from different tissues — spleen, bone marrow, and intestine cells. We can hone in on one specific example: applying archetypal analysis algorithms to human colon crypt cells.

Applying this algorithm to a dataset of colon cells, Korem et al. tested polytopes where the dimension k ranged from 2 to 11, finding that the optimal polytope was a 3-polytope with 4 vertices, or a tetrahedron. **Note that this tetrahedron was found in the original high-dimensional gene expression space, and that each vertex of the archetype is a datapoint/vector associated with information from all genes. Archetypes, then, are akin to the most ideal/specific example of a given cell type. We'll call these "archetype-cells"**.

They then graphed the data points, or individual cells, within this tetrahedron, which contained vertices/archetypes that represented specific cell types based on the genes highly expressed by the "archetype-cells". Note that they've

used PCA to lower the dimensions of the dataset and make visualization easier, but archetypal analysis was purely used for the actual analysis and for the discovery of the tetrahedron.

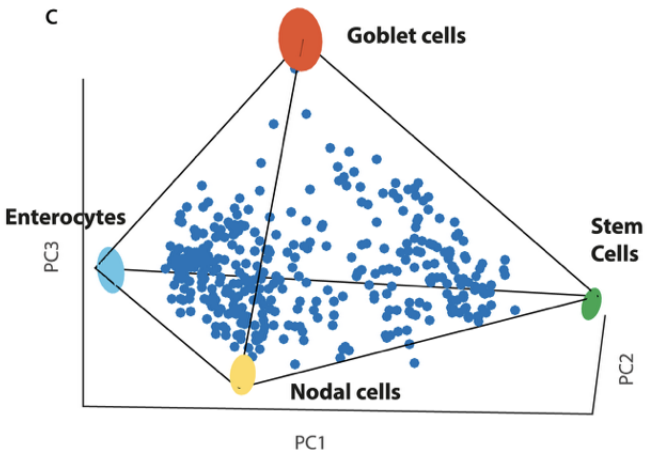

Figure 11: The human colon crypt cell tetrahedron. Each vertex represents a "pure" cell type. Adapted from Korem et al. [5]

So, as these researchers found, archetypal analysis is a good way to characterize single cells! Wahoo!

4 Trying This at Home

Finally, let's apply archetypal analysis to a small example of our own. We'll use the **archetypes** package in R [4] along with a simple toy dataset to model an archetypal analysis problem.

Let's start by generating a toy dataset. For ease of plotting and visualization, we create a dataset with 2 genes. **Note that we can perform archetypal analysis in any space, with any number of genes.** As we described in Section 3, archetype polytopes have a low number of vertices but are found in the original high-dimensional space. We do this to preserve a good idea of the expression patterns in our archetypes, so that we can categorize them later on. We wrote the following R code to model a toy problem and perform archetypal analysis:

```r
library("archetypes")

cells <- data.frame(replicate(2,sample(0:100,20,rep=TRUE)))
plot(cells)

suppressWarnings(RNGversion("3.5.0"))
set.seed(1986)

arch_plot <- function (p) {
  a <- archetypes(data = cells, k = p, verbose = TRUE)
  xyplot(a, cells, chull = chull(cells))
  xyplot(a, cells, adata.show = TRUE)
}
```

Figure 12: Our R code to perform archetypal analysis!

In lines 4 and 5, we create a 20×2 matrix and populate it with random numbers from 1 to 100. Each row represents a cell and each column represents a gene (let's call our genes "X1" and "X2"). Each value in the matrix represents an expression of a given gene in a given cell. Let's plot each cell along the axes of the two genes:

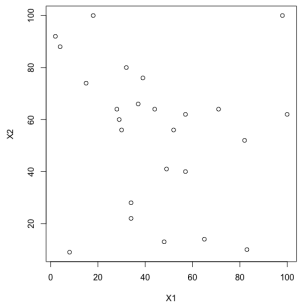

Figure 13: Dummy cells with genes "X1" and "X2"

Looking at this data, let's say we think it has three main archetypes, and find the best-fit archetype triangle. It fits OK, but there's a significant amount of data excluded, as we can see when we look at distances of each point from the estimated convex hull (Fig 14 and 15).

We calculate the RSS associated with this estimated convex hull to be 0.0674635. Pretty good, but we can do better by adding another archetype. When we fit a convex hull with 4 archetypes to the data, we get a new best-fit polygon (Fig 16 and 17). This seems to exclude far fewer points by far less distance. We calculate the RSS associated with this new estimated convex hull to be 0.01254811. Much better!

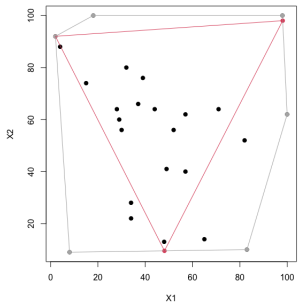

Figure 14: Convex hull of the best-fit triangle in red. Archetypes are red points. The datapoints are black, and their original convex hull is grey.

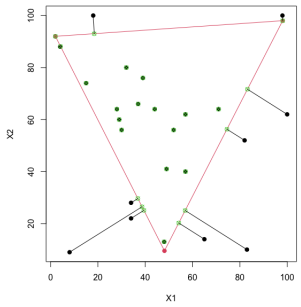

Figure 15: Distances from datapoints to the convex hull are shown as black lines with green marks corresponding to their projection onto the hull.

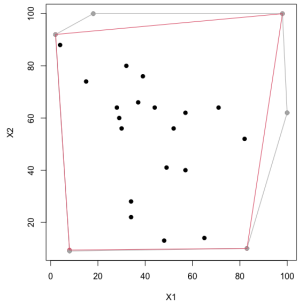

Figure 16: Convex hull of the best-fit 2-polytope, now with four vertices, in red. Archetypes are the red points. The datapoints are in black, and their original convex hull is shown in grey.

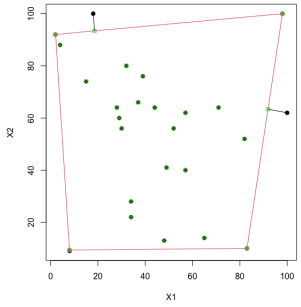

Figure 17: Distances from datapoints to the convex hull are shown as black lines with green marks corresponding to their projection onto the hull.

5 Concluding Thoughts

In this paper, we have introduced to you archetypal analysis and applied it to real-world examples in biology, specifically single-cell analysis! In detailing archetypal analysis, we discussed what archetypes are, how to represent them geometrically (including reasoning behind the structure), as well as how to find and optimize them using Residual Sum of Squares. Afterwards, we contextualized archetypes' applications in an existing study on single cell identification, before walking you through our own example using R.

For the reader's further enrichment, we have also added a bibliography below with all of the sources that we have discussed. Many of these papers go into further detail on other statistical strategies on single cell identification, as well as archetypal analysis itself.

References

[1] Thaddeus Abiy, Beakal Tiliksew, and Josh Silverman. Convex hull. *Brilliant.org*, 2021.

[2] B. H. P. Chan, D. A. Mitchell, and L. E. Cram. Archetypal analysis of galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 338(3):790–795, 01 2003.

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

[4] Manuel Eugster and Friedrich Leisch. From spider-man to hero - archetypal analysis in r. *Journal of Statistical Software*, 30, 04 2009.

[5] Yael Korem, Pablo Szekely, Yuval Hart, Hila Sheftel, Jean Hausser, Avi Mayo, Michael E. Rothenberg, Tomer Kalisky, and Uri Alon. Geometry of the gene expression space of individual cells. *PLOS Computational Biology*, 11(7):1–27, 07 2015.

[6] 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.