UNSUPERVISED RANDOM FORESTS AS APPLIED TO THE ANALYSIS OF SINGLE CELL RNA SEQUENCING DATA

by
Boris Markovich Brenerman

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
October 2021

Abstract

Single-cell RNA sequencing data contain patterns of correlation that are poorly captured by techniques that rely on linear estimation or assumptions of Gaussian behavior. We apply random forest regression to scRNAseq data from mouse brains, which identifies the co-regulation of genes within specific cellular contexts. By analyzing the estimators of the random forest, we identify several novel candidate gene regulatory relationships and compare these networks in aged and young mice. We demonstrate that cell populations have cell-type specific phenotypes of aging that are not detected by other methods, including the collapse of differentiating oligodendrocytes but not precursors or mature oligodendrocytes.

Acknowledgements

I thank Alexis Battle for the hard work of making sure this and many other projects see the light of day, and also for helping me turn half baked ideas into good ones.

I thank James Taylor for being a romantic, scientifically and otherwise

I thank the Johnston lab for their patience

I thank Benj Shapiro for an oceanic quantity of sorely needed editing, he is near-single-handedly responsible for the fact that this at least vaguely resembles English, Python, OR math.

I thank John Kim for being an advocate

I thank the committee their thoughtful scientific feedback

Table of Contents

List of Tables

List of Figures

Introduction

Single-cell RNA sequencing (scRNAseq) is a technology that has been researched extensively in the last ten years to allow analysis of transcriptomes of individual cells. Single cell RNA sequencing operates by tagging the RNA present in an individual cell with a barcode unique to that cell, followed by the sequencing of bulk RNA. By identifying the unique barcodes and assigning each sequenced read to a particular cell, researchers are able to reconstruct the transcriptomes of thousands of indvidual cells simultaneously. There are four lines of inquiry that scRNAseq is commonly used to investigate: 1.) discovery and annotation of cell types or states, 2.) discovery and annotation of gene regulatory relationships, 3.) changes in cell type populations or gene expression between samples, and 4.) the relationships of cell types to each other across differentiation trajectories and smooth progression through differentiation sometimes referred to as pseudotime. Of special interest is the usefulness of scRNAseq in helping to identify gene regulatory networks that are impossible to identify in bulk sequencing due to having their effects hidden by the averaging of several disparate cell types.

Most genes operate under the control of one or more transcription factors in response to environmental cues such as developmental signals, metabolic needs, pathogens, etc, and most transcription factors have the ability to activate or suppress the transcription of several genes. When a transcription factor or a set of transcription factors activate several genes at once to accomplish a complex effect such as stress response or cell division, the regulatory relationships between these genes are collectively known as a gene network. Elucidating gene networks clearly is of great interest in aiding the interpretation of transcriptional data, eg if we know that certain genes undergo transcriptional change, how can we understand what the overall effect on the cell is? Finding the targets that a transcription factor regulates is a problem that can be approached biochemically using immune co-precipitation sequencing, DNA footprint analysis, or other techniques, or it can be approached computationally through analysis of gene expression data.

Conventional Gene Network Discovery

Previous approaches to gene network discovery have focused on examining changes in transcription relative to some stimulus. For example, the first studies on gene regulatory blocks focused on examining the transcriptomes of hundreds of genes in hundreds of bulk RNA samples using microarray RNA quantification. Weighted gene coexpression network analysis (WGCNA) refers to the use of a correlation matrix describing pairwise correlations between all genes being investigated to construct a network. (Horvath 2011) To test whether two genes are related, a Pearson correlation is found between the expression values of each in a set of samples. Genes that are above a certain weighted threshold of correlation are designated as being connected in the regulatory network, and genes that fall below this threshold are presumed to be unconnected.

However, WGCNA suffers from a few limitations even on its face. For there to be a difference in expression for samples, usually samples with some underlying difference are examined (eg stimulated chemically, from different tissues, different developmental stages). Therefore when you consider the expression of two genes in several samples, in reality you are comparing the behavior of these samples in the presence of more variables. After all, transcription factors often affect the transcriptional levels of other transcription factors, and two transcription factors may respond pleiotropically to an external stimulus without necessarily being transcriptionally linked.

Methods developed for tackling this problem include the use of matrix inversion-based partial correlation analysis, regularized regression methods such as LASSO, and mutual information methods such as ARACNE.(Margolin et al. 2006; Ma, Song, and Huang 2007; de la Fuente et al. 2004) All of these methods attempt in different ways to eliminate the pleiotropy that plagues WCGNA by reconstructing a network where the knockon effects of one relationship inform the other relationships. These methods generally work well on constrained problems where only a small subset of genes are being evaluated, but the quality of data is often insufficient to solve the general problem of discovering transcriptome-wide gene regulatory networks in practice. For this reason much gene network research has gone into elucidating not the topology of a particular gene network, but rather finding gene modules (Akalin et al. 2009), eg sets of genes that move together and describe the presence or absence of a particular biochemical process broadly.

Use of Factor Analysis

The existence of gene regulatory blocks in some sense presents a problem, since the presence of dozens or hundreds of highly collinear variables in a dataset makes significance testing of all kinds more difficult. In order to summarize the behavior of multiple genes across thousands of cells the current best practice relies on principal component analysis (PCA), generalized linear models (GLM), and related techniques which are special cases of Projection Pursuit Regression (PPR), while the general approach is called factor analysis.

Factor analysis works by producing a small number of dummy variables that are mathematically derived from many ordinary variables, distilling the behavior of many genes into a single score. For example, if the scores of the first principal component (PC1) that are obtained by transforming raw gene expression data are strongly correlated to whether or not a cell was exposed to stimulus, an experimenter might infer that genes that are weighted highly in

PC1 also respond to the stimulus. (Jolliffe and Cadima 2016). Because PCs summarize the behavior of many collinear genes at once, it may be tempting to think that they describe the behavior of particular gene regulatory blocks, but in fact this is not the case.

PPR-family techniques find factors that explain data by employing a technique known as Gram-Schmidt orthogonalization to ensure that each successive factor that is discovered captures a maximal amount of some optimization criterion, and no further factor can capture the same information. In the case of PCA, each factor is produced by drawing a line through the dataset that minimizes ordinary least squares from the line to all data. All variance explained by that factor is then subtracted from the dataset, and the next factor repeats the process. The first problem with this process is that it is greedy: each successive PC is guaranteed to capture the maximum amount of variance present in the dataset, not necessarily to accurately predict a single biological effect. When a PC is discovered

However, this use of orthogonalization is dangerous because it makes the assumption that the relationships between the variables (genes) in the data are consistent across all samples observed. Essentially, the problem we see here is the same one as we saw early on when we talked about DNA repair: each PC assumes simple arithmetic relationships between factor values, so the first PC would observe that non-homologous end joining and homologous recombination are correlated when one looks at all cells. The first PC would then capture this relationship and regress it out.

Motivating Problem

For example, consider DNA repair of double-stranded breaks (DSB). DSB repair can proceed along two pathways, non-homologous end-joining (NHEJ) and homologous recombination (HR). When examining bulk tissue sequencing expression data, we find that there is a correlation between the expression of NHEJ and HR genes in response to the administration of DSB-inducing mutagen. (Helton and Chen 2007) In reality, however, cells have a preference for non-homologous end joining when they are quiescent and for homologous recombination when they are undergoing DNA replication, which means that the expression of genes related to either is actually anti-correlated in cells that are repairing DNA damage. (Iyama and Wilson 2013) It is possible to control for the effect of the cell cycle by synchronizing the cell cycle of dividing cells or investigating quiescent cells only, however if we did not know a-priori that such controls were necessary, we would have no reason to suspect that we reached a misleading conclusion. The situation in which relationships between certain variables change when you consider only a subset of the data is called non-linear dependence or higher-order dependence, and the presence of such dependencies violates the basic assumptions of PCA (Hyvärinen 1999).

Kernel PCA and Projection Pursuit Regression (PPR) can both be used to address this problem (Townes et al. 2020), but kernel PCA does not provide any meaningfully interpretable feature weights, and PPR can be quite slow and is sensitive to the selection of a good objective. In practice, the field is in need of a method that will allow us to find simple and intuitive factor decompositions of single-cell RNAseq data that both allow for improved clustering in a lower-dimensional space and simultaneously provide intuitive interpretations for the redundant information spread across many genes. In addition, the method should enable the interpretation of how various regulatory relationships interact with each other in a complex landscape and how they vary across the dataset. Ideally, we would want to find a method that identifies broad divisions present among cell types (e.g. stem, differentiated, cycling, quiescent, etc.), as well as the regulatory relationships present within those divisions.

The usefulness of scRNAseq lies in the fact that we can control for almost any kind of condition that is reflected in transcriptional changes post-hoc, including controls for cell type, cell cycle, stress response, etc. Unfortunately, controlling for such variables still requires a judgement call on the part of the researcher to manually identify the necessary confounders. This is a viable task when researchers have to consider a relatively constrained problem, such as identifying the regulatory relationships between up to a few dozen genes, however controlling for all possible relationships between thousands of genes becomes intractable for humans.

Non-Linear Dependence

To overcome the problems posed by higher-order dependencies, we consider the use of random forest regression (RFR). Random forests (RF) are a class of statistical models that are ensemble estimators based on bootstrapped decision trees. They are non-linear, make few assumptions about the distribution of the underlying data, are robust in a variety of contexts, and are accurate and efficient. (Breiman 2001) Unfortunately, RFs are not straightforward to interpret if a user seeks to understand the behavior of correlated features or obtain a meaningful high-level representation of a dataset, such as that provided by principal components or cluster assignments. The availability of measures such as feature importance offer insight into individual features but not the coordinated behavior of several features in the data.

In order to more effectively understand the structure of datasets such as single-cell gene expression measurements, we propose a novel method for the analysis of the latent structure of an RF. To understand the decisions that RFs make to predict gene behavior, we propose examining the commonalities across the individual nodes making up an RF directly. Given RF nodes that have many properties in common, we can cluster them and infer what we call *random forest factors* (RFF). Random forest factors trained on scRNA-seq data partition cells into groups, capture the gene covariances present within each group, and covariances present across several groups. RFFs have simple and intuitive criteria for group membership that can be extrapolated to other datasets (**Figure 1**). This approach provides a way to naturally capture higher-order interactions in single cell gene expression data, provide a partitioning scheme that is suited to irregularly-shaped clusters, produce interpretable summaries of many features, and enable comparisons between distinct datasets. When applied to mouse brain scRNAseq data, we are able to pinpoint significant changes in gene expression for differentiating oligodendrocytes as a function of aging, aiding in the interpretation of otherwise opaque data.

Chapter 1: Performance of Unsupervised Random Forests

Construction of Unsupervised Random Forests

Unsupervised training of a regressor is usually understood to be a way of constructing a compact representation of a larger (eg multivariate) dataset out of fewer latent variables, premised on the idea that the relationships of variables in the input also make information contained in those variables redundant. When the task at hand is unsupervised clustering, the only latent variable learned is categorical, while an unsupervised regression task could produce a set of latent variables that are continuous. Understanding in what way the latent variables represent the data they summarize is key to interpreting them meaningfully. For me, the goal is to generate latent variables that summarize the existence of particular relationships between gene expression, notably relationships that may potentially be different depending on the expression of particular genes. Thus the objective is to create a set of latent variables that allow for the reconstruction of gene networks and the relationships in gene expression predicted by those gene networks. To do this in an unsupervised manner, we will first need to construct a random forest that predicts the behavior of some genes based on others, establishing relationships between genes given various samples in the dataset.

The training of random forests in an unsupervised manner isn't an entirely new idea, with notable work by Shi, Afanador, and recently Mantero. (Mantero and Ishwaran 2021; Shi and Horvath 2006; Afanador et al. 2016), however, these previous methods have taken a quite different approach to unsupervised RF construction than what we propose. Shi and Afanador both take the approach of constructing a Random Forest using synthetic data intermixed with input data, creating a newly realized classification task, eg to classify whether or not the data is synthetic or input. Shi uses the newly injected data to allow the forest to make decisions regarding the grouping of correlated vs uncorrelated samples (eg the injected data is pure noise, an any set of samples that separates from it will therefore separate according the the relatedness of the signal) and produces a dissimilarity measure suitable for further use by other algorithms.

Injecting synthetic data is helpful in understanding the underlying distribution of the dataset but not necessarily in interpreting its structure. Mantero instead focuses on using an RF to predict the behavior of a large set of interaction variables based on a set of inputs. This approach takes a refreshingly direct route to estimating the relationships between various genes, but it suffers from facing a relatively unconstrained problem: which interactions ought an RF predict? Combinatorics makes for a daunting set of potential outputs to an unsupervised random forest.

I have chosen to approach the problem by training an RF to predict the behavior of some genes based on the behavior of others at random in each tree and each split. By establishing a large number of simple binary relationships between sets of genes and checking whether or not we see certain relationships appear again and again, we can investigate the interactions of large numbers of genes, and more importantly we can do so in a manner that controls for the expression of other genes. Assuming there is a reasonably finite number of gene networks that are possible to enumerate and quantify, each time a decision tree in an RF bootstraps a set of genes and then produces a split that explains the behavior of some of them, the behavior of a network that those genes belong to should be captured. Because we are creating a latent variable that represents the network, we need only capture a part of the relationship, not necessarily the entire thing. This prevents us from having to select which particular interactions to investigate and under what conditions.

Briefly, to construct an unsupervised random forest (URF), I propose that at each bootstrapping step where an ordinary RF would select a set of input features to perform splits on we instead take all available features and split them into two equal groups: the input subset and the output subset. The split is then performed as usual for a multivariate random forest (Xiao and Segal 2009) In detail, for each URF, a root node was constructed by partitioning the set of all samples. Then, for each tree among the number of trees specified by the parameters, an identical root node was taken as the first node. For each node, if maximum depth or minimum sample quantity was not reached, a split was performed. To perform any split, S samples were bootstrapped with replacement from the set of all samples satisfying the criteria of the node being split and all ancestor nodes forming the subset S'.To perform an unsupervised split, the set of all features M was split randomly into input features and output features I and O. From inputs, F inputs were bootstrapped without replacement forming the subset F'. From outputs G outputs were bootstrapped with replacement forming the subset G'. The S' x F' input matrix X' was constructed by selecting the samples S' and features F' from inputs X. The S' x G' output matrix Y' was constructed by selecting the samples S' and the features G' from outputs Y. If specified, X' and Y' were dimensionally reduced to the top K principal components by NIPALS (Wold 1975), producing a revised input and output pair X'' and Y'' with dimension S' x K and S' x K. Given the input X' or X'' and Y' or Y'', for each input feature j (eg each gene) and sample i, potential child node subsets were constructed consisting of subset 1 containing samples where $x_{i,j} < x_{s,j}$ and subset 2 containing samples where $x_{i,j} > x_{s,j}$.

The quality of each split was calculated by finding the sum of the squared deviations from the median for each feature in the newly formed nodes:

$$Q(s) = \sum^{k} \left(\sum_{i \in A} \left(\widehat{y_{A\,k}} - y_{i,k} \right)^2 + \sum_{i \in B} \left(\widehat{y_{B\,k}} - y_{i,k} \right)^2 \right)$$

Where A and B are potential children where i is in A or B if $x_{i,j}$ is greater than or less than s_j respectively, y-hat$_{Ak}$ is the median of output k in A, y-hat$_{Bk}$ is the median of k in B. After evaluating all possible splits for all selected inputs, the split producing the smallest sum of squared deviations from the median was selected and the procedure was fully repeated.

I have also investigated the use of means as the central tendency and the use of median absolute deviations from the median, variance, coefficient of determination, and several other error metrics as splitting criteria and found that the results are generally robust across a variety of parametrizations, though with some distinctions discussed later.

Quantifying Recovered Information

In order to use an unsupervised random forest (URF) constructed by random input-output splitting for further analysis we must first validate that an RF constructed this way captures meaningful information about the underlying dataset. To validate the ability of a URF to predict the behavior of genes generally we will evaluate its ability to predict the gene expression values of cells held out during training and evaluate several kinds of error criteria. To validate the ability of the RF to find distinct cell types and cell identities we will check its ability to cluster blood cells into clusters corresponding to known labels.

In order to evaluate the amount of information that a regressor captures about a dataset, it is standard procedure to train the regressor on a set of data and then evaluate its performance when faced with data that regressor has not seen before. At face value this may imply that when acting as a regressor the forest allows features to predict their own behavior, however this does not reflect a dangerous practice because features are prevented from self-predicting during forest training.

I trained a URF on a single cell gene expression dataset to conduct validation. The dataset was obtained from a publication by Ximerakis et al and contained 8 control mice whose brains were disaggregated into single-cell solution and tagged with X10 UMI chemistry to produce 16028 individually tagged cells after quality filtering, with ~2000 per mouse. The median depth of sequencing for each cell was ~20000 UMIs and mice were reared in an identical manner. 2000 genes with the highest variance across the dataset were selected for further analysis to eliminate those genes that were either too rarely expressed to be suitable for analysis or not variable enough to capture information about cell state or identity. I then trained a URF on the 16028x2000 matrix with subsampling rates of 700 features per bootstrap for input features, 700 features per bootstrap for output features, and 300 samples per bootstrap, a maximum splitting depth of 9, a minimum leaf size of 50, with each output feature standardized to node dispersion measure, and with and with an error metric of the sum of squared deviations from the median with an L1 norm across all standardized features (These parameters will constitute the default parameters for future analysis of this dataset unless otherwise specified). When a URF trained on a subset of 4 randomly selected mice is used to predict the behavior of the selected genes, the overall fraction of variance left unexplained (FVU) is 54%.

To evaluate whether or not a linear regressor captures a similar amount of information I evaluated the FVU of PCA-transforming the data for different numbers of PCs, and recovering the values that are represented in the PCs. If a matrix is decomposed into a number of PCs equal to its rank, the information in that matrix will generally be perfectly recoverable, however for biological data there is generally some number of principal components that contain the majority of recoverable information, and then a large number of PCs which recover a disproportionately small amount of total recoverable information. The information recovered by PCA with up to n components can be plotted into an “elbow” graph, where some number of PCs represent reducible information and the rest is presumed to be noise. When a 4 mouse, 2000 gene subset of the Ximerakis data is analyzed in this manner I observed this result. The elbow in the Ximerakis data rests at ~10 PCs and ~55 FVU. The elbow of the PCA matches the FVU of the URF trained on the young mice, implying that the URF and the PCA are recovering the same amount of “explainable” information from the dataset.

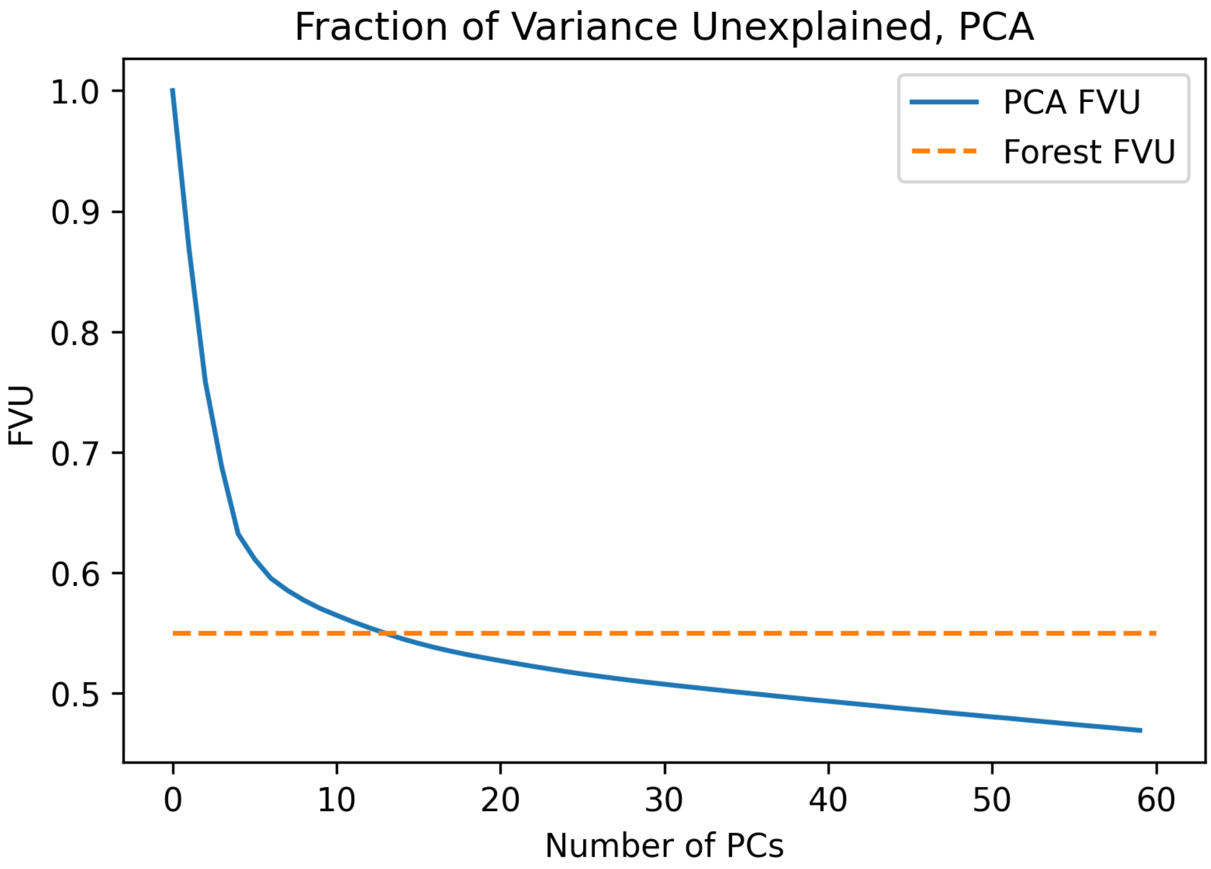


Figure 1: total recovered variance for PCA with different numbers of PCs retained, from 0 to 60. Note the “elbow” shape, where PCs above ~10th have dramatically less explanatory power compared to PCs 1-10. URF captures most of the variance that rests “under the elbow”,

eg is explainable by embedding into lower-dimensional space. This result implies that PCA and the RF have a similar power to capture the information present, though not necessarily to interpret it.

Error Domains

The fact that the magnitude of variance captured by either technique is comparable does not necessarily imply that the sources of variance captured are identical. I plotted the residuals of the PCA and URF against each other and observed that they are highly correlated, implying that the information being captured is largely the same (Fig 2)

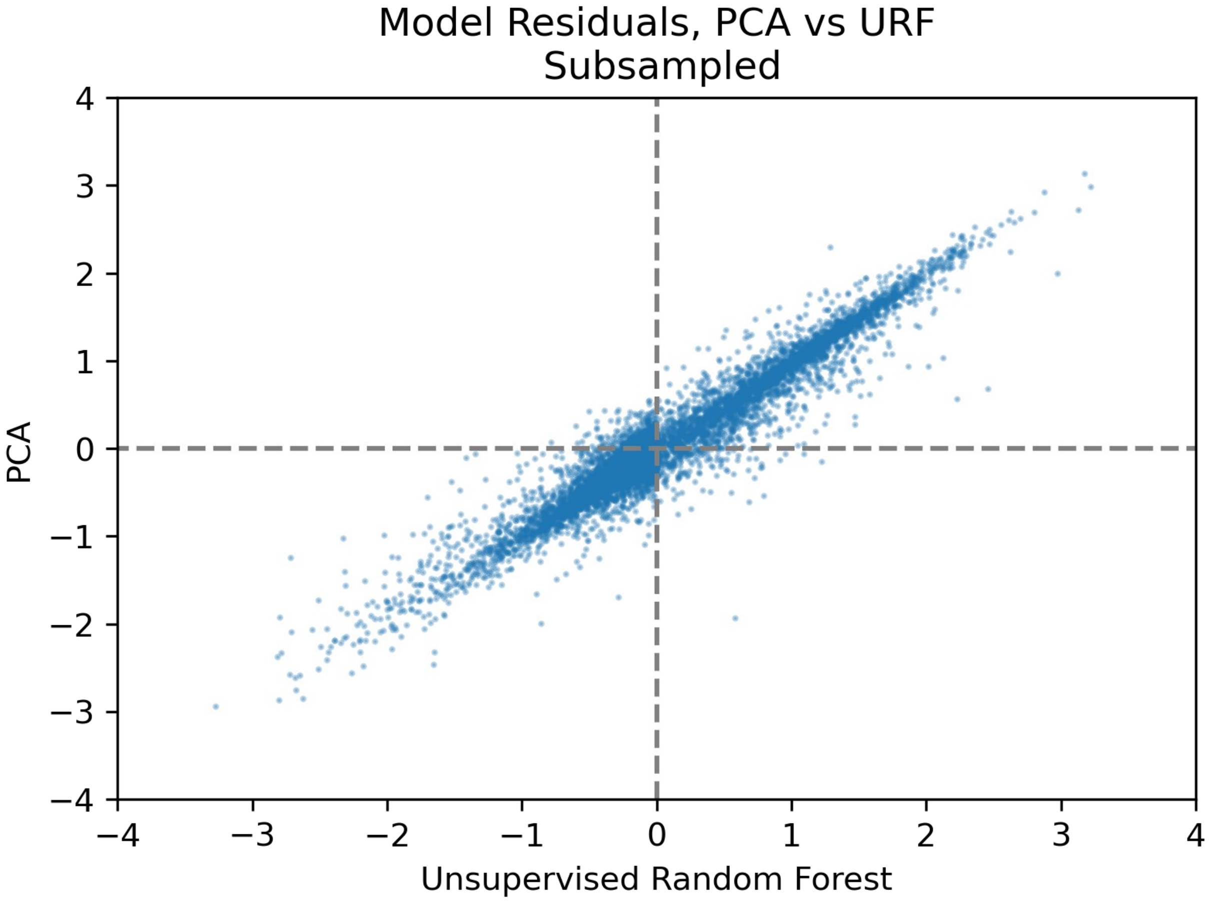


Figure 2. Residuals across all features and all samples for PCA and URF trained on 8 mouse brain scRNAseq dataset. (only 32000 data points are shown, but data not substantively different for all 320000)

Features

I plotted the explanatory power of PCA and URF for each feature within the dataset and found that PCA is superior in explaining the variance of features with very high mean expression, while the Random Forest is generally superior in explaining the expression of features with low expression.

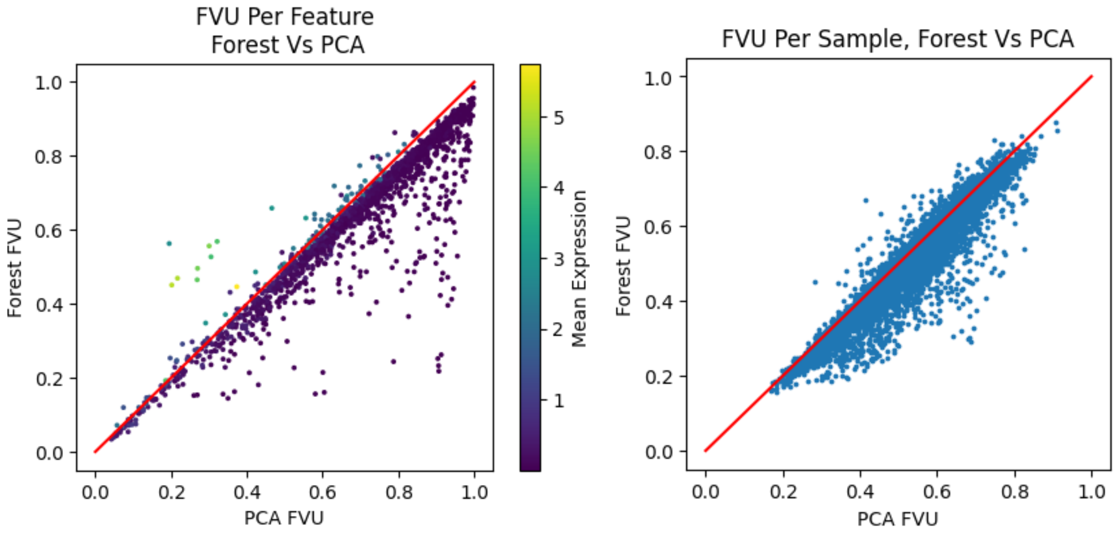


Figure 3: **A)** Fraction of total variance left unexplained for the expression of each individual gene. The mean of the log-TPM expression of each gene is ovelaid as a color, showing that highly-expressed features (eg TPM > 10000) are better predicted by PCA. **B)** Fraction of total variance left unexplained for each individual cell.

Samples

When examining prediction quality on a per-sample basis, URFs are also superior in some (though not all) samles, while PCA is superior in very few. The comparison between quality of prediction of different samples and sample library size did not yield and obvious association, however overlaying the difference in the FVU of the URF and the FVU of PCA on an embedding produces a more interesting implication:

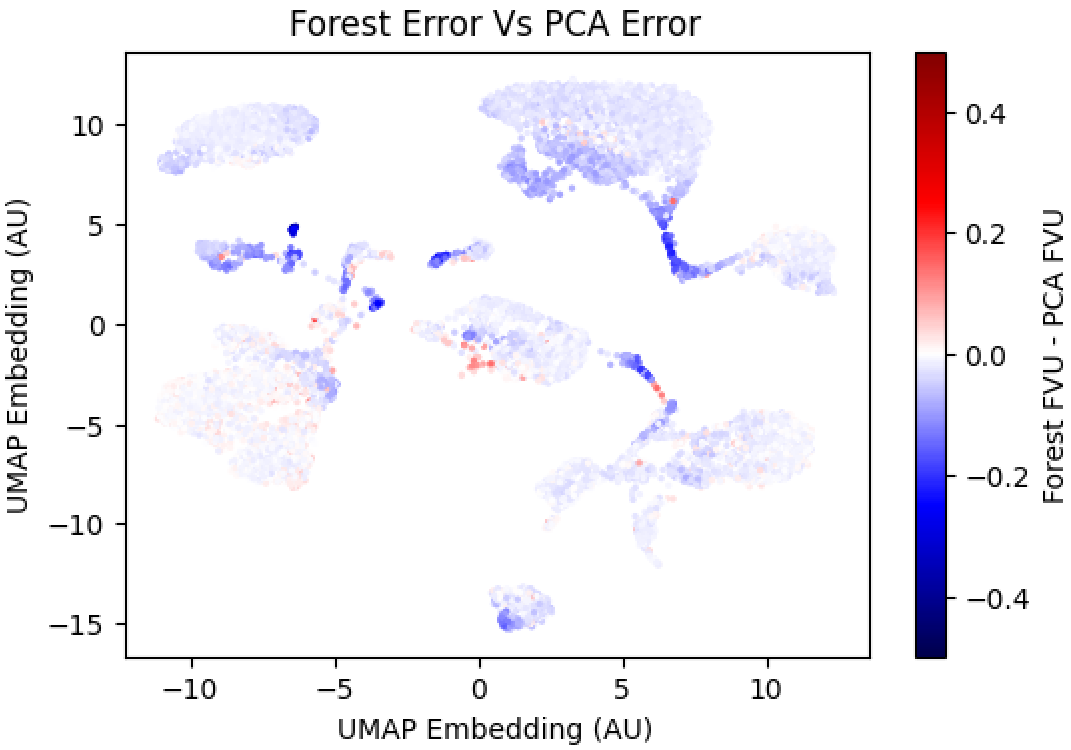


Figure 4, UMAP embedding of individual cells based on gene expression, colored by the difference in FVU for that cell between URF and PCA. A blue color indicates a forest is superior at predicting the behavior of the cell, while red indicates PCA is superior.

Here we see that PCA does a worse job capturing transitional states and the expression profiles of cells that exist in small clusters (see chapter 4 for details on cluster identities). Because predicting the behavior of transitional states is of great interest in examining stem cell biology, URFs potentially present a great boon to researchers in that context.

Inter-Individual Validity

I also wished to demonstrate that a URF trained on a set of individuals can be reasonably extrapolated to describe the behavior of ostensibly similar individuals. When a URF is trained on 4 randomly selected young mice and is used to predict the 2000 most variable genes, the final coefficient of determination (eg sum of squared residuals across all genes relative to the sum of squared residuals before partitioning) for the cells used as a training set is 54%. When the same forest is used to partition the brain cells of 4 held out mice, we instead observe an FVU of 55%, indicating that within the top 2000 genes, prediction quality shows almost no decrease between different animals, and almost all unexplained variability is presumably attributable to either measurement noise or to lack of suitability of the chosen regressor to the task at hand. Given that a linear regressor produces comparable or inferior results on a per-gene and per-cell basis, we must defer to the assumption that measurement noise makes up the majority of the remaining variance.

Effect of Robust Metrics

Unsupervised random forests have an additional unique property that I have investigated, which is that they can be easily used with optimization objectives that rely on robust statistical measures of dispersion and central tendency, which is a tall order for any other factor decomposition. Most commonly, the performance of a regressor is estimated by its ability to reduce the mean of the squared difference between the mean and individual observations (eg variance or mean squared residual as in ordinary least squares regression). The mean squared residual is attractive as a regression minimization target because it has a host of useful arithmetic properties, allowing for efficient optimization in many contexts. Notably, to estimate the mean only a sum of values and the size of the sample population need be known, and estimating the mean squared deviation from the mean is similarly easy. Comparatively, the mean squared deviation from the median or median absolute deviation from the median is not such an easily known value.

A potential advantage of the median as a measure of central tendency, however, is that it is relatively resistant to the presence of extreme outlying values in the dataset. Single-cell RNAseq data is notorious for being heteroskedastic and possessing gene expression distributions with large numbers of greatly outlying values. For this reason, I was interested in investigating the behavior of URFs that used robust measures of central tendency and dispersion in their optimization objectives.

I investigated URFs that used the following non-standard measures to evaluate the quality of a split: sum of squared deviations from the median, sum of absolute deviations from the median, and the sum of median absolute deviations from the median normalized by leaf size. These measures were compared to a URF trained using a conventional optimization objective, eg weighted variance (or sum of squared deviations from the mean). Four forests were trained, each with standard bootstrapping on features and samples, a maximum depth of 7 (to conserve compute time), dimensional reduction to 3 factors on the input, L1 normalization, and standardization of feature dispersion. When evaluating squared or even absolute deviation from the mean, the URFs that were optimizing with the median as the central tendency performed similarly to conventional URFs (within 1% of total variance explained), except for the forest based on median absolute deviations, which was highly sensitive to depth and dimensional reduction, but regained comparable performance at a depth of 9 and with dimensional reduction on outputs . However, when URFs used the median as the central tendency, they produced a superior sum of absolute deviations as a whole (49% of absolute deviations remaining for SSME, SME, and MAD at an increased depth).

Chapter 2: Latent Structure in Unsupervised Forests

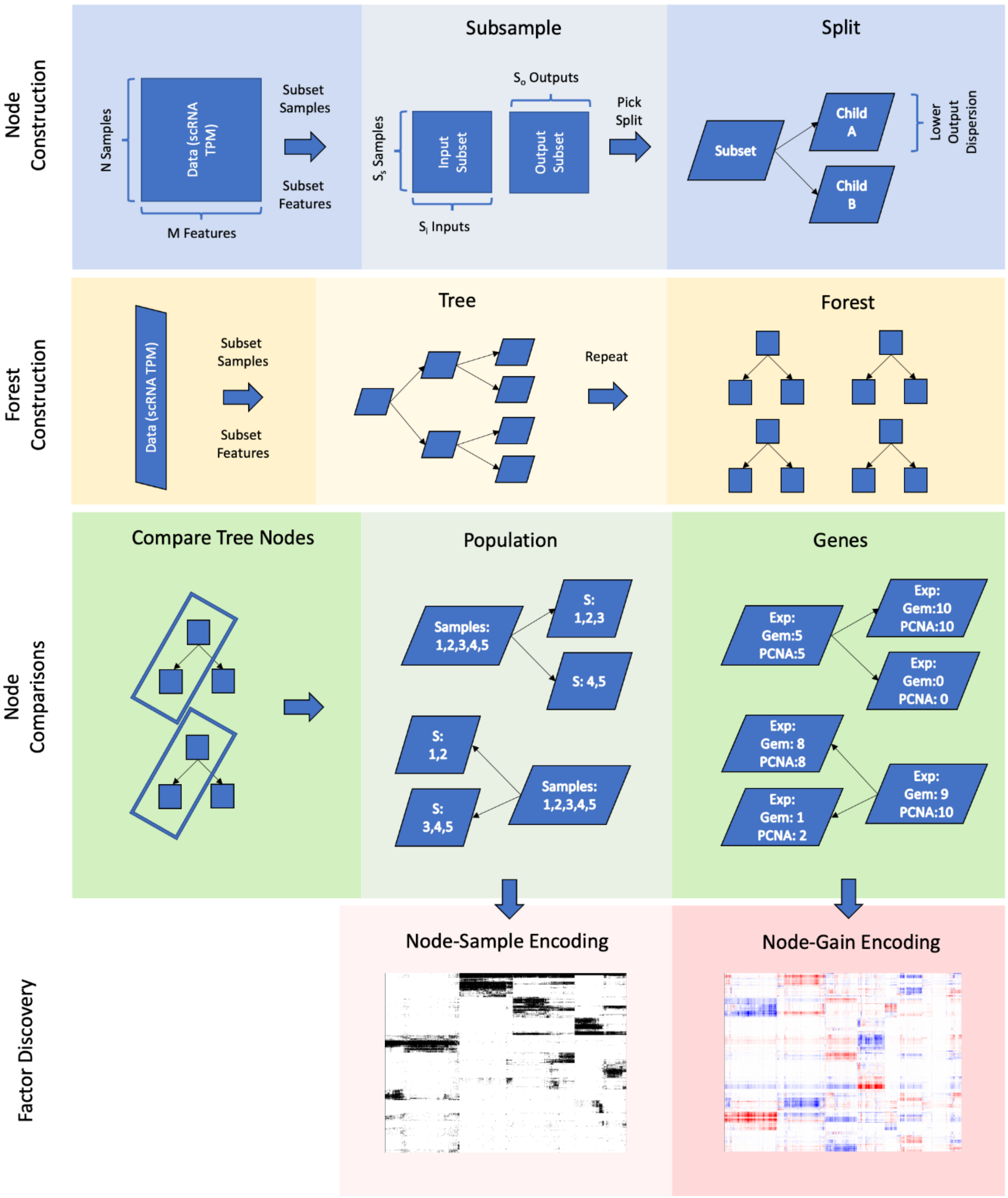


Figure 5: Overview of Random Forest Factors

Methodology

To understand scRNAseq data better we must have a way of identifying cell populations with common traits and what the traits that distinguish a cell population are, how the traits interact with each other, and which genes are co-regulated to determine the traits of a cell. In a context such as scRNAseq where there are higher-order interactions, we also ask whether or not genes and traits behave in the same way across all cell populations, or whether or not their behavior changes in some cell populations.

An RFR is an ensemble estimator based on a collection of individual decision tree estimators, which are in turn collections of individual nodes. In order to generate a tree, each individual node partitions a set of samples into two subsets based on gene expression or another criterion and generates two child nodes corresponding to the subsets. For example, a root node might coarsely partition all cells in an scRNAseq experiment into cycling and non-cycling cells based on the expression level of Geminin, a DNA replication inhibitor (McGarry and Kirschner 1998) One subset would then contain only cells that express high levels of Geminin, and the other would contain only cells expressing low levels of Geminin; cycling and non-cycling cell subsets will then have lower gene expression variance than the original set of all cells. After a node is created and a criterion is selected that determines whether or not a sample should be partitioned by that node, that node becomes an *estimator* and a *partition*. Henceforth, in order to estimate the behavior of a new sample, we find if it satisfies the criteria of a node (e.g. its Geminin expression is high) and that node then predicts that the sample will be similar to the samples in its subset. A node that uses Geminin as a criterion might predict that samples partitioned into it will also have high expression of the polymerases, both for DNA replication and repair.

To analyze the behavior of a complex dataset such as scRNAseq expression data we will use a random forest regressor with multiple outputs constructed in an unsupervised manner. We construct the RFR with no supervision by iteratively and randomly splitting all genes into inputs and outputs, and training trees to predict the expression of a set of output genes from the expression values of a set of input genes. This approach allows us to construct a random forest that relates the expression of each gene to the expression of other genes using bootstrapping, since we have no need to predict any known ground truth about the dataset.

RF nodes have an essential property that distinguishes them from other estimators: they are recursively generated. An RF node generates child nodes by finding the optimal split of the cells in its subset, without taking into account any other splits and based on entirely new criteria. In this way, each node makes a split on a population of cells where the variance explained by its parent node has been eliminated conditional on some criterion, which makes it a *conditional estimator,* otherwise known as a marginal estimator. Because each conditional estimator contains samples that meet criteria such as geminin expression, the effects discovered by conditional estimators could be dependent on the levels of expression of prerequisite genes. For example, repair of double-stranded breaks can proceed either by non-homologous end-joining (NHEJ) or homologous recombination. Homologous recombination is more prevalent in cells that are cycling while NHEJ is more common in cells that are quiescent. Within an RF, an individual node might contain only cycling cells, or only quiescent cells, as determined by Geminin levels. Such a node can detect that expression of HR pathway genes is dependent on high Rad51 and Geminin expression simultaneously. However if Geminin expression is low, Rad51 expression would have a small but positive correlation with NHEJ (Iyama and Wilson

2013). The fact that Rad51 predicts HR in cycling cells but NHEJ in quiescent cells is an example of a higher-order interaction. In this way conditional estimators allow for the possibility of discovering higher-order interactions such as conditional gene expression, which distinguishes them from purely linear estimators such as PCA.

Methodology Overview

To compare nodes across different trees in a random forest, we will describe them in terms of the samples they contain (creating a node-sample encoding or NSE) or the improvement in the conditional prediction they make (creating a node-gain encoding or NGE). By comparing the conditional estimates a node makes and samples in that node, we will be able to judge the similarities of two nodes to each other, allowing us to determine if nodes can be arranged into clusters. Then, by examining nodes that cluster together, we will be able to determine other properties of those nodes such as conditional covariance of genes or conditional predictive power.

In order to analyze whether or not similar *conditional estimates* are frequently discovered in an RFR, we construct a summary of the predictions that a node makes relative to its parent. A ***node-gain encoding (NGE)*** is a matrix of size N x L where N is the number of nodes and L is the number of regression targets (gene expression values), with each element representing a regression target; an element is valued as the difference in the predicted value for a target in a given node compared to its parent, hereafter known as ***conditional gain.***

$$CG(X_i|i \in N) = E[X_i|i \in N] - E[X_i|i \in N']$$

Where CG is conditional gain, X_i is regression target X for sample i, N is a node of interest and N' is the parent of the node of interest. In the case of the root node, which has no parent, the expectation of X given the parent is 0, so the conditional gain is the mean across the dataset. Conditional gain is gain controlled for all previously explained variation.

A **partial gain encoding** (PGE) is similar to a node-gain encoding but controls the change in the mean value for a particular node for the effects predicted by both its parent nodes and child nodes using the law of total variance. For a detailed rationale and derivation of some properties please see Appendix A.

$$PG(X_i|i \in N) = \frac{CG(X|i \in N)\sqrt{CG(X_i|i \in N)^2}}{CG(X_i|i \in N)^2 + E[CG(X_i|i \in N'')^2]}$$

Where PG is partial gain, CG is conditional gain, X_i is regression target X for sample i, N is a node of interest and N'' any descendant of the node of interest or its sisters. In practice, we get a value between -1 and 1 whose magnitude is the percentage of total variance explained for a particular gene by node N. This measure is useful because the behavior of a given sample which is partially explained by the conditional gain of a given node N may in fact be explained by a node dependent on N (ie one of its children). Thus when examining the effects that a given node predicts most accurately, we want to eliminate predictions that another node may make more accurately still. By using the law of total variance, we scale the explanatory power of every node to the same scale while retaining the sign of the estimate

The NGE and PGE allow us to summarize the conditional gain that node is predicting, so by comparing the conditional effects predicted by different nodes we can check whether certain predictions happen repeatedly. We are interested in examining these encodings because they allows us to see if nodes predict conditional changes in gene expression for the same genes repeatedly, implying that those genes are co-regulated. If we find that many nodes predict that both geminin and polymerase epsilon are upregulated simultaneously, our conclusion would be that there is a regulatory relationship between these two genes. However, the advantage compared to conventional correlation analysis like WGCNA is that many of the correlations we are examining are greatly controlled for correlations addressed by other estimators.

In order to compare the samples partitioned into nodes, we can construct a **node-sample encoding (NSE)**. A node-sample encoding is a matrix of size N x S, where N is the number of all nodes in an RFR and S is the number of samples; element i,j is 1 if node i contains element j and 0 otherwise. We constructed node sample encodings in order to examine whether certain groups of nodes often contained similar groups of samples. A node-sample score is the mean of the NSE for a given set of nodes, representing the proportion of nodes in a cluster of nodes that observe a certain sample.

A ***node-sister encoding*** is similar to a node-sample encoding, but can be valued as -1,0, or 1. An element in a node-sister encoding is valued 1 if a sample is partitioned into a node, -1 if it is partitioned into the sister of the node, and 0 if it is partitioned into neither. A node-sister encoding contains more information than a node-sample encoding, and allows us to visualize whether we we repeatedly see a single subset partitioned away from different sets, or if we see the same greater set partitioned into the same two subsets repeatedly across different trees. The node sister encoding allows us to compute the sample sister score for a given cluster, which is the number of times a particular sample was encountered in the nodes of a node cluster - the number of times that sample was encountered in their sister nodes. The advantage of the sister score is that it is close to 0 when there is no information about a sample or if the information is ambiguous (eg a sample is equally frequently discovered in a particular node or its sisters), but the score value is high when a sample is frequently discovered only in the node or only in the sisters.

Single Cell Analysis

In order to apply our factoring procedure to single cell data we trained an RFR on scRNAseq data derived from disaggregated brains of 8 young mice by Ximerakis et al. (Ximerakis et al. 2019) We recalculated the cell by gene expression matrix with an identical preprocessing pipeline as Ximerakis and obtained 16027 single cell expression profiles of disaggregated brain tissue belonging to 8 young mice. An RFR was trained in an unsupervised manner on the cells originating from young mice, trimmed, and node-sample and node-gain encodings were derived from all nodes remaining in the forest. (Figure 2)

We agglomeratively linked both nodes and genes within the NGE using cosine similarity and observed a checker pattern indicating that nodes in the RFR repeatedly predict one of several categories of conditional gains (Figure 2B). The node subpopulations didn't appear consistent in population or intra-cluster distance metrics so to obtain hard partitions rather than a dendrogram we computed a k-nearest-neighbors representation of the nodes and partitioned them via the Louvain algorithm into 39 clusters. 42% of all variance in the NGE was explained by the clustering procedure, reflecting the difficult nature of the underlying clustering task. We created several permutations of the NGE by shuffling each column independently to decorrelate the features. When the resulting permuted data was clustered by the same procedure, the resulting partitions explained 7.5 +/- 2e-7 % of the variance, indicating that the clusters found in the NGE correspond to real structure.

Importantly, when nodes are partitioned into clusters on the basis of the similarity of their NGE, the partitions explain 25% of the variance in the NSE as well, while partitioning an NSE that was shuffled as previously described explains only 7.3% of variance.

Ultimately the usefulness of these clusters of nodes is in aggregating the effects of individual nodes into the mean effect across the cluster. This is necessary for interpreting the behavior of a random forest because bootstrapping prevents us from making certain conclusions about the behavior of any individual node. The behavior of a single node could be attributed only to chance inherent in the bootstrapping procedure, but by taking a collection of nodes behaving in a similar manner, we can assert the presence of a particular effect.

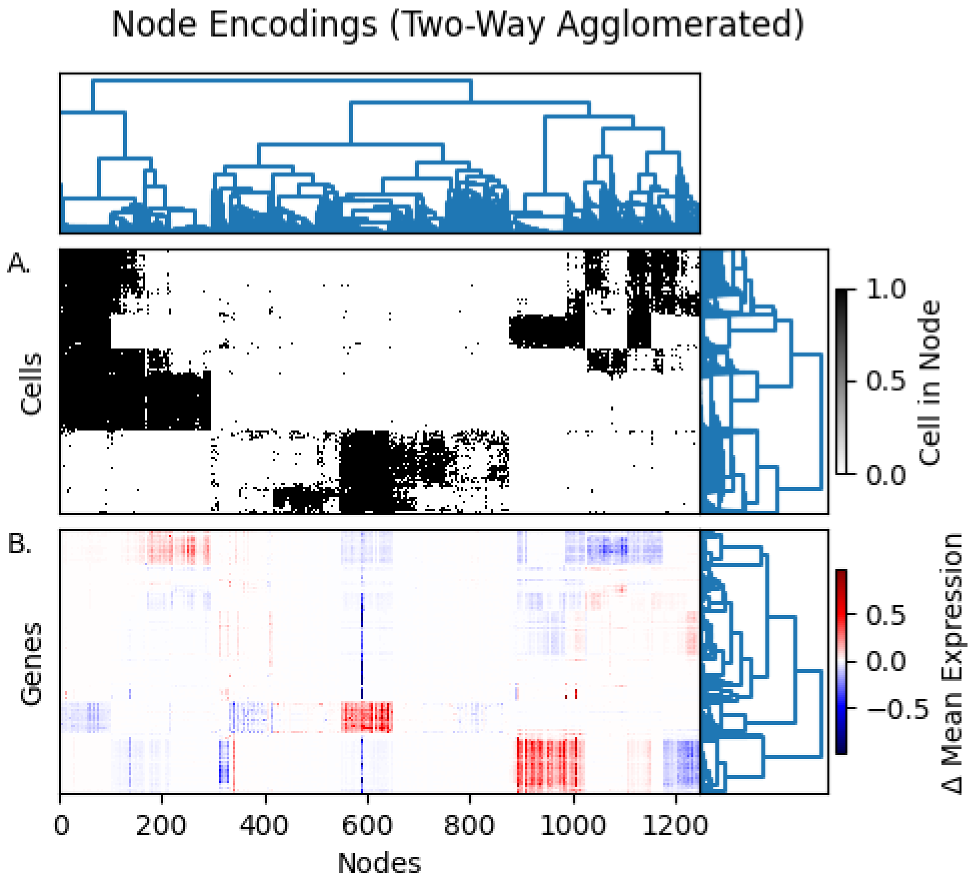


Figure 6: Node-Sample Encoding and Partial-Gain Encoding derived from an RFR trained on mouse single-cell gene expression data. Nodes below depth 4 are omitted for clarity, see Supplemental Figure 1 for complete figure. A) Node-sample encoding, element i,j is black if cell i is in node j, white otherwise. B) Partial-gain encoding, element i,j is red if expression of gene i is predicted to be higher by node j controlling for all other nodes in the tree, white if the expression

is on average the same, and blue if the expression is predicted to be lower. Dendrograms are constructed by cosine similarity on a dimensionally reduced representation of data.

Chapter 3: Local Behavior

Sister Scores Are Narrowly Distributed

The conditional dependencies demonstrated by the nodes in clusters can be different than the dependencies that are apparent across the dataset as a whole. In the brain dataset RFR the behavior of immune cells is described by nodes falling into cluster 5 and its descendants, as evident from the gene expression of cells in those clusters. There is a clear trimodal distribution of cluster 5 sister scores (**Fig. 3A**), therefore we can compare cells that occur unambiguously in cluster 5 (>.2 sister score) against all other cells. Samples that unambiguously occur in node cluster 5 significantly overexpress CD45 (.37 vs .016 TPM, p=1e-106, T Test) compared to all other cells, a ubiquitous immune cell surface marker (Trowbridge, Ralph, and Bevan 1975), and the C1Q family of genes (4.15 vs .04 TPM, p<machine epsilon, T Test), which are components of the complement system (Reid, Lowe, and Porter 1972), so cluster 5 broadly specifies immune cells. Nodes in Cluster 5 also predict the overexpression of C1Qa compared to parent nodes. The difference in mean expression of C1Qa in nodes of NC5 compared to their parents is 1.0. This is the *conditional gain* of C1qa in NC5. [1]

[1] Forests trained including CD45 also predict a conditional gain of .08 for this gene, but CD45 is excluded from the forest being presented due to its sparsity.

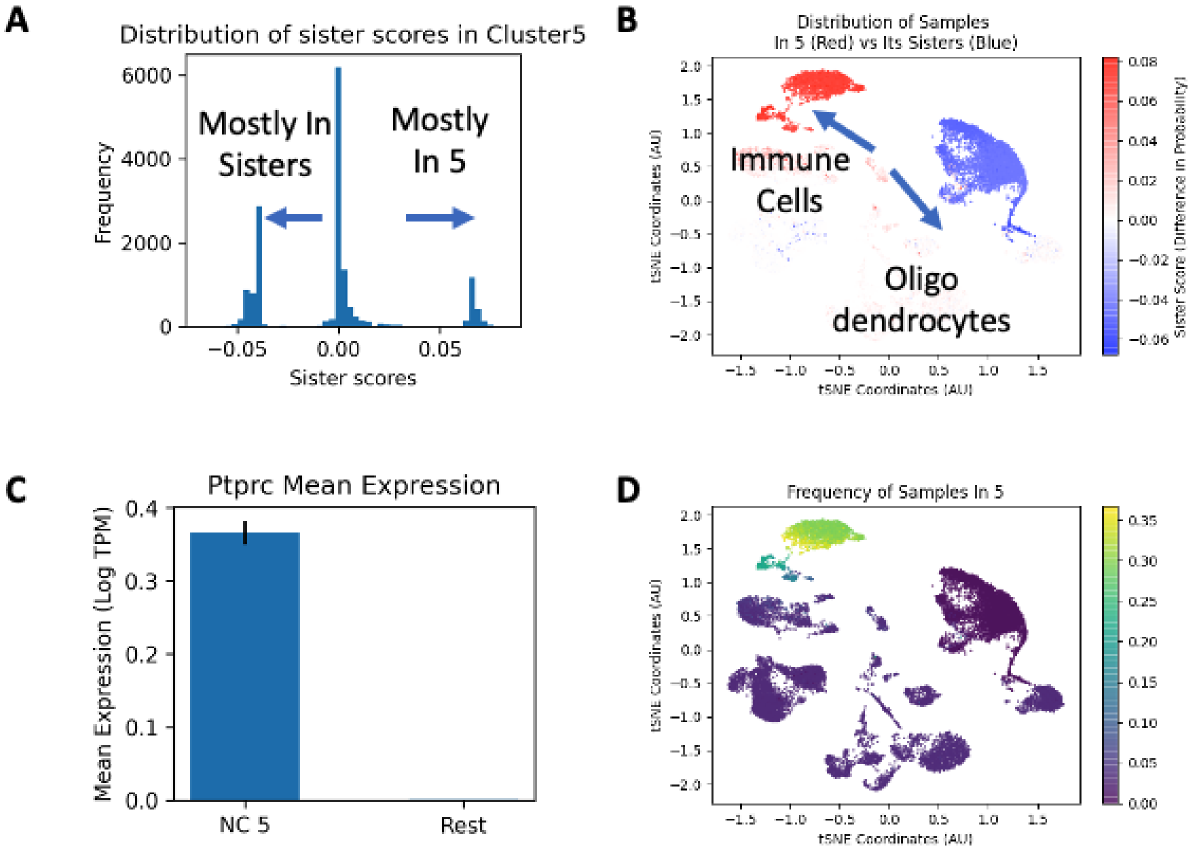


Figure 7: A. RFR nodes trained on mouse scRNAseq data and clustered by conditional gain have narrow distributions of sister scores. Samples occur mostly in nodes of cluster 5, mostly in its sisters, or mostly in neither. B. Samples that occur mostly in node cluster 5 are immune cells, as established by CD45 expression. C.Mean expression of Ptprc (CD45) in samples mostly occurring in NC5 and all other samples, whiskers are SEM. D. Frequency of samples in NC5 overall. Some samples occur frequently in both NC5 and its sisters and have a low sister score despite being seen frequently in NC5.

The latent variables that govern gene expression in mouse brain cells are not fully known, so we examined whether forest factors can discover local behavior by comparing the correlations discovered by forest factors to correlations captured by PCs and conventional correlations.

Local Behavior Reflects Nested Cell Types

The mouse brain samples we are examining contain both microglia and immune cells normally traveling in blood, likely as a consequence of blood vessels being present in the brain samples collected.[2] Microglia precursors share a lineage with other immune cells, however they migrate into the brain during early stages of mouse embryonic development and thus the gene expression profiles of general immune cells and microglia can be grouped naturally. There is a very significant distinction between microglia and other immune cells as well, which may be of interest to us; other hand, neuronal-lineage cells have a drastically different gene expression profile from any immune-lineage cells, and the gene-regulatory programs that govern immune cells and neuronal cells have little overlap. Among immune cells which are described by node cluster 5 (NC5), the RFR detects the differential regulation of several key genes that distinguish microglia and macrophages, such as CD74, Tmem119, Lyz2, and H2-Ab1.

Nodes from NC5 make up 57% of parents of NC34 and 32% of the parents of NC27. Nodes in NC34 predict the over-expression of Tmem119, which is a marker of microglia (conditional gain .91). Conversely, Cluster 27 predicts the over-expression of CD74 (conditional gain .40) and Mrc1 (conditional gain .88), which are strong markers of perivascular macrophage identity in combination. (Kim et al. 1992) Of the top 5 genes most over-expressed in NC34 compared to its sisters, all are strongly anti-correlated with the top 5 most over-expressed genes in NC27 relative to its sisters when the correlation is weighted by the frequency with which each cell is present in either of the two clusters (**Fig 4H**). However, globally the genes in

[2] Blood cells can be present in brain samples even when brain samples are perfused during processing because immune cells are frequently adherent to the walls of blood vessels and cannot be cleared by perfusion. (Anderson et al. 2012)

these two subsets are uncorrelated or even positively correlated (**Fig 4G**). For example the expression of Tmem119 and Cd74 is uncorrelated globally, however, when examining only the behavior of cells frequently present in Cluster 34 but not its sisters, we observe a strong positive correlation, which is an example of *local behavior*. Part of the reason that this correlation may be difficult to observe in other ways lies in the cells belonging to node cluster 18, where Tmem119 and Cd74 are positively correlated (**Fig 4F**). NC18 represents arachnoid barrier cells, as established by expression of Slc47a1 relative to other vascular cells.

Many genes have local behavior similar to CD74 and Tmem119, such as Pf4, Lyz2, Ifitm3, Mrc1, Ms4a7 and separately CD81, Tmem119, P2ry12, Selplg, and Hexb. Both these sets of genes are strongly anti-correlated with genes of the other set, but only among the cells unique to NC 34. (**Fig 4G, 4H**), and we propose that they occupy a gene regulatory block distinctive to microglia. Our ability to identify these sets of genes as being co-regulated in specific ways in different contexts is desirable since it allows us to identify a gene regulatory block that distinguishes two populations of a related lineage from each other, representing only the conditional difference between microglia and blood immune cells while controlling for irrelevant variation present in the rest of the dataset.

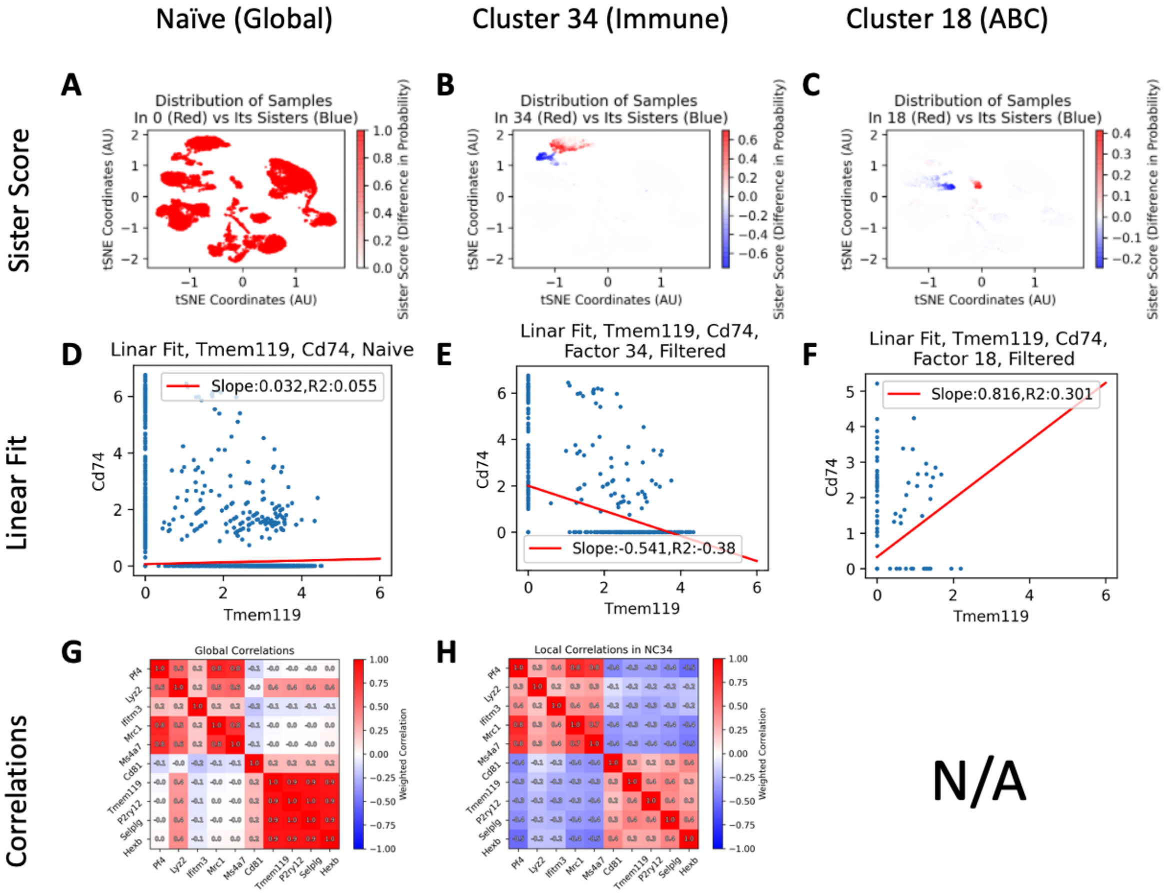
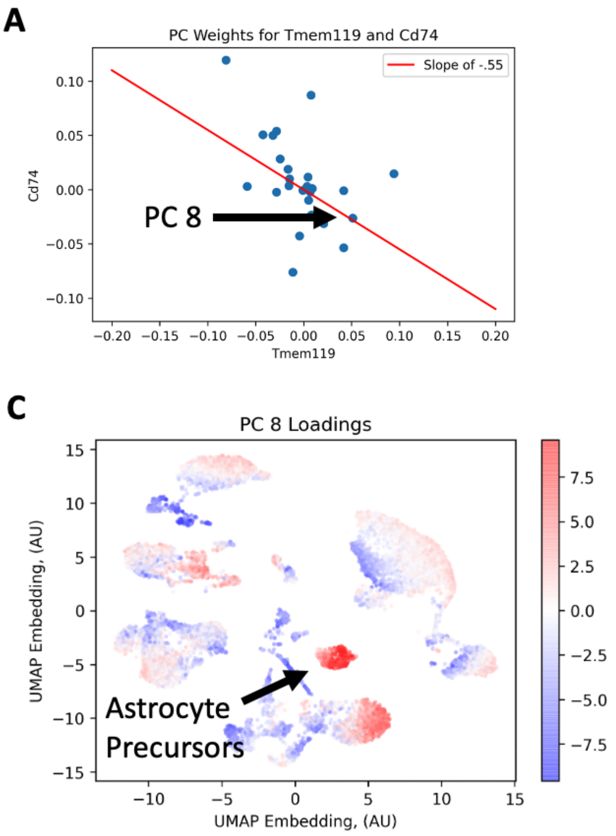


Figure 8: Certain genes behave differently when considering only cells frequently seen in NC34 than they do when considering all cells. A: UMAP projection of cell gene expression data. B: UMAP projection of cell gene expression data colored by random forest factor score for RFF 34. All cells express CD45 and are therefore immune, red cells express Tmem119 and are therefore microglia. C: UMAP projection of cell gene expression data colored by RFF 18. Red cells express Slc47a and are therefore arachnoid barrier cells .

Comparison To PCA

Finding this particular regulatory block using PCA would be challenging, because each PC could potentially be representing variation from several sources. When the log-normalized and scaled expression matrix is analyzed by PCA, the weights of individual genes in each component correspond to the regulatory relationships that PC is modeling. In order for us to find the regulatory program that distinguishes microglia from other immune cells we would need to find a PC which weighs Tmem119 and CD74 with opposite signs. We could then examine the weights of other genes in that PC and surmise that ones with weights of the same sign as Tmem119 and CD74 are upregulated in microglia and blood immune cells respectively.



B

Rank	Gene	Weight
1	Itm2b	0.23
2	Apod	0.17
3	Npy	0.16
4	Fabp7	0.16
5	Ckb	0.15
6	Ubc	0.15
7	Ubb	0.14
8	Vtn	0.14
9	Slc25a4	0.13
	…	
66	Cd74	-0.053
	…	
118	Tmem119	0.042

Supplemental: Plot of weights of Cd74 and Tmem119 among all PCs trained on mouse brain data. Red line has a slope of -.55, which is the same as the correlation coefficient of CD74 and Tmem119 among immune cells. For a PC to weigh Tmem119 and CD74 identically, it must lie on the red line. B: Genes ranked by the absolute value of their weighing in PC8 of the mouse brain dataset. C. Loadings of PC8 trained on mouse brain cells overlaid on a UMAP projection of all cells, red indicating a large positive loading, blue indicating a large negative loading. Endothelial cell identity established by

PC8 has weights of Tmem119: 0.042 and CD74:-0.053, which is the strongest negative association of loadings observed among the PCs and accurately captures the anti-correlation that is also captured by Cluster 34. This means that the weights in PC8 should come the closest to modeling the distinction between microglia and macrophages. However, PC8 has loadings across a range of cell types, and high weights for genes that are unrelated to the behavior of the immune system. The gene with the second-highest weighing in PC8 is Apod, which is not known to be expressed in any immune cells (Yoshida et al. 1996), and thus unlikely to be co-regulated with Tmem119 and Cd74. Thus, if we were to use this PC to model the distinction between microglia and macrophages, we would find a number of genes in our model whose behavior is totally irrelevant.

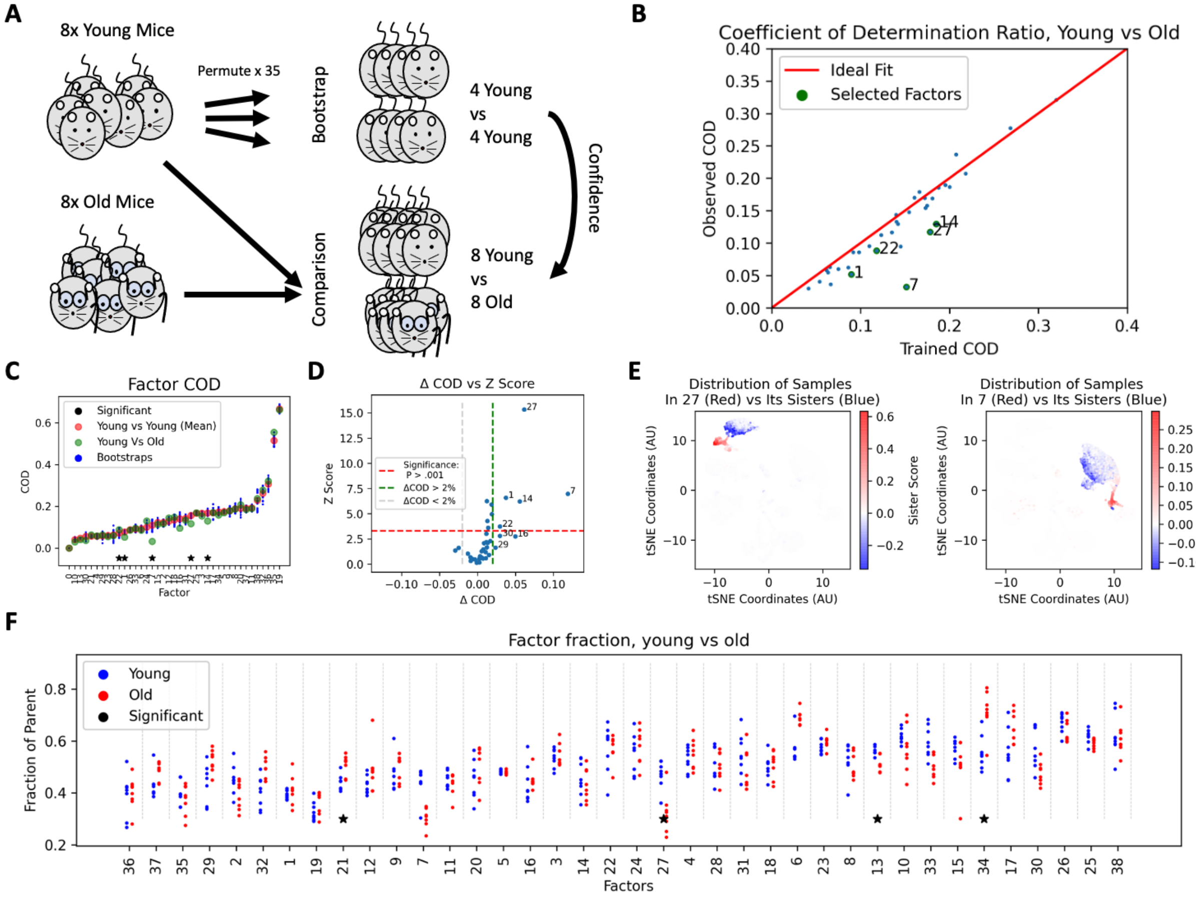


Figure 9: A: Cross-validation scheme. An RFR was trained on all young mice, 35 permutations were established consisting of 4 young mice vs 4 other young mice. The RFR node predictions were calculated using the first 4 young mice and COD was established when predicting the gene expression of the other 4 young mice. B: Mean COD when forest predicts gene expression in young vs old mice for each RFF. RFFs where COD was significantly different in old mice and young mice are labeled. C: COD of each individual factor when trained on 4 young mice and predicting 4 other young mice (blue), mean factor COD across all permutations (red), and COD of each individual factor when trained on 8 young mice and predicting 7 old mice (green). Factors with significant differences in COD are marked with black stars. D: Plot of Z score

compared with difference in mean COD in young mice and COD in old mice. Factors where differences are both large and significant are labeled. E: UMAP projection colored by the sister scores for factors 7 and 27, where greatest differences in predictive power lie. F: Mean fraction of the samples captured by nodes in each factor from the parent node is plotted for each mouse, young mice in blue, old mice in red. Factors with significant differences in percent of samples captured between young and old are labeled with black stars.

Chapter 4: Dataset Comparison

We've shown that Random Forest Factors can summarize the behavior of several features that are co-regulated in particular contexts such as specific cell types. RFFs can also be used to demonstrate that particular conditional effects change between different datasets. There are three ways for the expression of genes in a single-cell RNAseq sample to shift across two different sets of conditions: due to changes in the proportions of particular cell types in the sample, due to changes in the actual gene expression profile of cells within a cell type, or due to technical differences between the two datasets.

Random Forests don't present any unique advantage in compensating for a change in measurement, so proper normalization is required before they are applied, and we will assume it from now on. If an RFR is used to partition a dataset in which identically behaving cells are present in different proportions, the accuracy of predictions for any particular node will not change; there will only be a change in the proportion of cells present in a particular set of nodes. Conversely, if gene expression within some cell types does change, then the model we have built to describe them is no longer valid. In this case, the quality of the predictions made by individual estimators in an RFR will change (presumably for the worse), since cells will be assigned to nodes that don't fully describe their behavior. We can take advantage of this property in order to examine changes in populations and gene expressions between aged and young mouse brain cells. We used an RFR trained on young mouse brain cells to predict the behavior of aged mouse brain cells, which assigned aged brain cells to particular nodes in the RFR.

Prediction of Gene Expression

NC7 is a set of nodes describing the behavior of immature but committed oligodendrocytes, established by the differential expression of Cldn11 (oligodendrocyte transmembrane protein, 3.56 tpm in 7 vs 1.21 in all other[3],p=4e-101) and Bmp4 (early differentiation factor, 0.424 tpm in 7 vs 0.015 in all other, p=1e-18).(Gow et al. 1999) To establish how well the nodes in NC7 can predict the behavior of immature oligodendrocytes we used bootstrapping. We partitioned the 8 mice from which brain samples were obtained into 35 possible partitions of 4 training mice and 4 test mice. (**Fig 5A**) When the cells from 4 training mice are used to establish the conditional effects of nodes in NC7 and predict the behavior of cells belonging to 4 test mice, we observe a mean COD of 11% and a range of CODs from 7.7% to 13%. (**Fig 5C**), indicating that 11% of the variance present among all oligodendrocytes can be explained by the unique gene expression profile of immature oligodendrocytes.

Conversely, the COD of the NC7 when applied to cells from old mice is only 3.3%, indicating that immature oligodendrocytes in older mice have a significantly different gene expression profile. The expression for test statistics of individual CODs is not known in this context because we are uncertain of the distribution they are drawn from, however assuming them to be approximately t-distributed, 5 factors have significant shifts in prediction quality: NC7, NC27, NC14, NC1, and NC22. (**Fig 5D**)

[3] NC7 is not the only factor coding for Cldn11-positive oligodendrocytes, so there is appreciable expression in “all other cells”, the factor coding for all Cldn11-positive oligodendrocytes is NC19, for which the statistic is 3.9 TMP vs 0.094 TPM, p<machine epsilon

To establish the significance of this discrepancy in predictive power we can examine the individual genes that have better and worse prediction quality within this cluster. We list the genes that were well predicted by NC7 (COD > 15%) in **Table 2.** Certain key gene expression patterns remain accurately predicted, namely expression of Plp1, Mobp, and Gria2, all key identity markers for immature oligodendrocytes involved in maintaining the physical structure of oligodendrocytes and key physiological functions. Conversely, the predictive quality collapses for almost half of the genes that were previously significantly predicted in this sub-population, namely Fth1, Car2, and Scrg1, which are metabolic proteins responsible for ion metabolism, and Rtn, Tnr, and Bcan which are key transcriptional factors involved in oligodendrocyte differentiation.

Factor	Identity	Marker Genes	COD Young	COD Old
7	Immature oligo-dendrocytes	Cldn11, Bmp4	11%	3.3%
1	Klk6 oligo-dendrocytes	Cldn11, Klk6	7.7%	5.2%
27	Blood immune cells	CD45, Tmem119(-)	16%	12%
14	Monocytes	CD45, Plac8	17%	13%

22[4]	Perivascular Macrophages	CD45, Pf4	7.3%	8.9%

Table 1. Factor characteristics

Gene	Young COD	Δ	Old COD
Fth1:	0.48	-0.62	-0.14
Car2:	0.47	-0.59	-0.12
Scrg1:	0.29	-0.57	-0.28
Rtn1:	0.29	-0.44	-0.15
Tnr:	0.32	-0.43	-0.12
Bcan:	0.30	-0.43	-0.13
Apod:	0.31	-0.42	-0.11
Gpr17:	0.26	-0.41	-0.15
Mal:	0.46	-0.40	0.06
Stmn4:	0.41	-0.35	0.07
Lsamp:	0.23	-0.34	-0.12
Il33:	0.19	-0.32	-0.13
Ermn:	0.37	-0.31	0.06

[4] Note the increase in COD in older cells. It is unclear why COD would increase in older cells, though potentially a rare subpopulation that was co-clustering with cells in 14 is no longer present

Arpc1b:	0.17	-0.29	-0.12
Trf:	0.40	-0.29	0.11
Gpr37l1:	0.29	-0.27	0.02
Opcml:	0.25	-0.26	-0.02
Ppp1r14a:	0.36	-0.26	0.10
Opalin:	0.28	-0.21	0.06
Tubb2b:	0.17	-0.21	-0.04
Hapln2:	0.17	-0.18	-0.02
Neu4:	0.16	-0.18	-0.02
Qdpr:	0.30	-0.16	0.14
Ptgds:	0.18	-0.13	0.05
Mog:	0.22	-0.12	0.10
Pdlim2:	0.16	-0.11	0.04
Sept4:	0.28	-0.08	0.20
Mobp:	0.31	-0.08	0.24
Tsc22d1:	0.15	-0.07	0.08
Cldn11:	0.18	-0.06	0.12
Cryab:	0.23	-0.05	0.18
Gria2:	0.16	-0.05	0.11
Pex5l:	0.20	-0.04	0.16
Plp1:	0.22	0.06	0.28

Table 2: Differentially predicted factors in old vs young, factor 7.

Prediction of Population Levels

We've established that we can observe gross changes in expression profiles for certain factors, however we can also use RFFs to track changes in cell populations using our factors. In order to establish the variability of cell populations between different mice, we examined the mean proportion of cells captured from the parent node by nodes in each factor and each mouse (**Fig 5F**). We established the range of percentages of the samples in the parent that are captured by each factor across all 8 young mice and all 8 old mice. To determine whether there was a significant shift in the percentage of the parent samples captured by the nodes in each factor, we performed a Mann-Whitney U Test for the 8 fractions captured in young mice compared to 8 fractions captured in old mice. NC13 exceeds the corrected significance threshold of .00125 (NC13 MWU p=.00097), and several factors reached a significance p>.005 and are marked as well. NC13 is a child to NC3 which contains cells overexpressing Cldn5 (4.3 vs 0.19, t-test p<machine epsilon), which means NC3 contains primarily vascular cells making up the brain blood vessels. NC13 the plurality of sisters for NC13 are in NC21, and Cldn5 is highly expressed in both, so NC13 encodes for a distinction between two subtypes of vascular cells. NC21 predicts over-expression of Vtn, a marker for pericytes, and NC13 predicts over-expression of Cldn5 even relative to NC21, marking them as Endothelial Cells. (Vanlandewijck et al. 2018) The prevalence of NC21 is greater in old cells, indicating that endothelial cells are more prevalent relative to pericytes according to the model postulated by the random forest. Interestingly, the predictive power of NC21 and NC13 only has a small though significant discrepancy (4.7% vs 4.1% COD, T Test, p>.002). In absolute terms the observed changes in populations are small, so attributing significance to them is difficult without a hierarchical model of the type we demonstrate here, which aids in the detection of important effects such as the collapse of immature populations.

Chapter 5: Other Work

I have also investigated other approaches to the quantification of gene regulatory networks, especially in the context of retinal biology. Global approaches to investigating gene network behavior fail in situations where particular genes have properties that make them unsuited to analysis by pipelines designed with the average gene in mind. One example of such behavior is quantifying the expression of the opsin family of genes in human organoids. The opsin family of genes encodes for a set of proteins that are expressed in retinal sensory neurons and allow for neuron depolarization in response to specific wavelengths of light. There are three opsins found in human cone cells OPN1LW, OPN1SW, and OPN1MW 1-3, which respond respectively to long, short, and medium wavelengths of light, eg. red, green, and blue light. Each individual cone cell in a human retina commits to the expression of a different opsin over time, however the pattern in which the expression of each opsin is arranged on the retina is regulated yet random. The challenge in investigating the regulatory network that controls opsin commitment is partly due to the difficulty quantifying the expression of the opsins owing to the fact that aligning opsin cDNA generated by next-generation RNAseq techniques produces a large degree of multiple alignment between the highly homologous genes. There are only 34 loci differentiating OPN1LW from OPN1MW 1-3 in humans and only 20 in macaques, so only a small fraction of the reads imparts meaningful information about the difference in the expression of the two genes.

To investigate the gene regulatory network of the OPN family of genes Sarah Hadinyak et. al. generated a time series of bulk RNAseq libraries from human stem cells differentiating and organizing into eye organoids over the course of . In order to quantify the expression of OPN1MW and LW more effectively I generated a restricted genome index consisting of only the opsin genes in question and performed alignment without filtration for multiple alignment. The multiply aligned transcripts were then quantified through the use of the kallisto package (Bray et al. 2016) which allows for the quantification of gene expression by a bayesian approach using a system of linear equations to quantify the Maximum Likelihood solution to the quantity of transcripts present based on the number of ambiguously aligned transcripts as well as the number of uniquely aligned transcripts.

In order to validate the behavior of the kallisto package I also generated pileups of reads aligning to each unique site, allowing for manual inspection of the transcript counts corresponding to each gene at each datapoint.

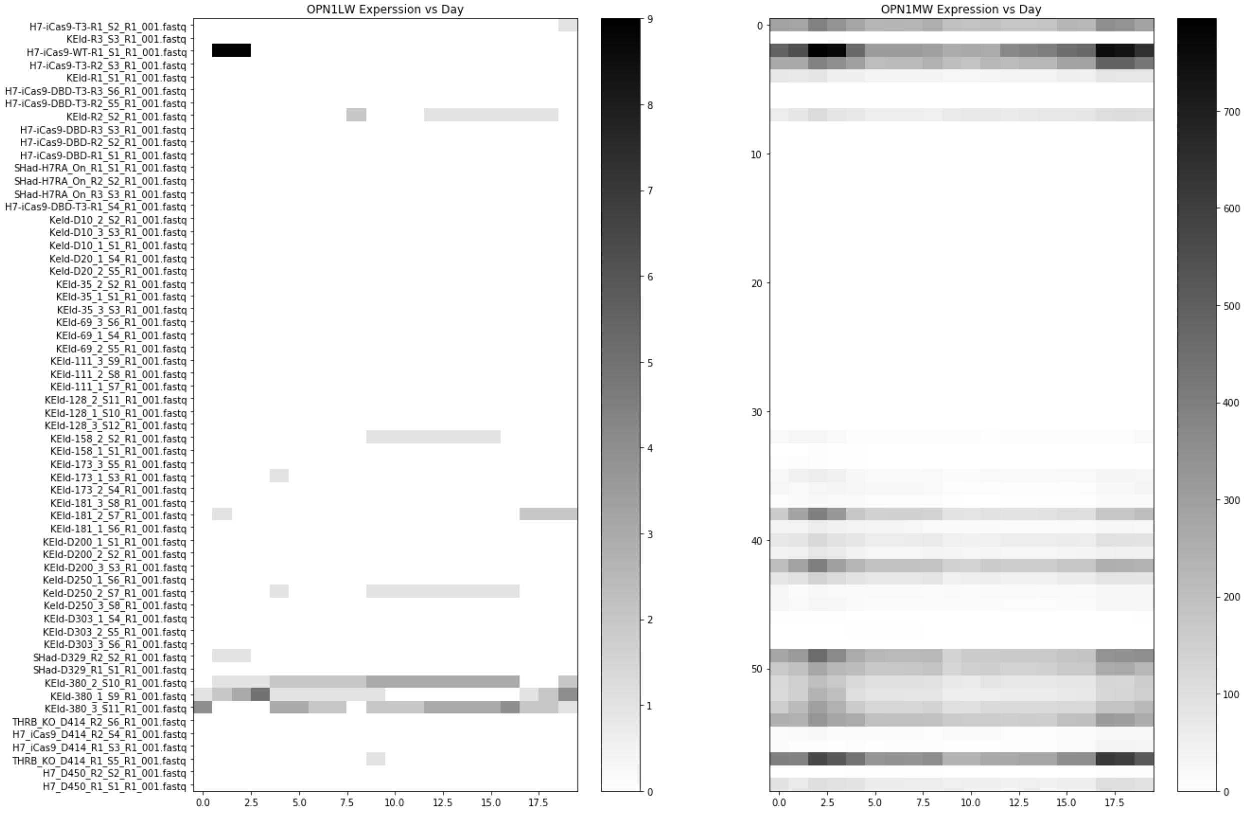


Figure 10: OPN1LW and MW expression over time in human eye organoids, transcripts per unique position per day.

Through the use of these robust measures of expression I was able to validate the gene expression levels and support the conclusion that M and L opsin commitment is regulated temporally, with an early expression of M opsin and later expression of L opsin.

Chapter 6: Conclusions

The analysis performed by Ximerakis on the mouse brain dataset is currently among the best in the field, and one of the gold-standard methods used by Ximerakis to analyze the data successfully is firstly to cluster cells coarsely, and then re-cluster the cells in coarse clusters to account for the dramatic changes in behavior that can occur in different parts of datasets. This is a popular technique that works well, as indicated by its wide usage and inclusion in the Seurat example vignettes, but it is a highly manual procedure. At an intuitive level a Random Forest Regressor takes a very similar approach by iteratively partitioning the dataset and then treating each partition as an entirely new problem, however RFRs are generally regarded as somewhat opaque tools that don't provide interpretable information on the structure of the underlying data. (Friedman et al. 2001) The method we've described here will help researchers reconsider the manual reclustering procedures that are labor intensive and require judgement calls and the opaque nature of RFRs. The intermediate partitioning steps an RFR performs can provide insight into interactions between features that would normally be hidden by Simpson's Paradox.

Conversely, RFRs don't perform as well on highly linear datasets, since each individual estimator in the ensemble ultimately outputs a step function. Additionally, RFs can be confounded by uninformative features, and if feature spartsity is uncorrelated to what? and exceeds 50%-70% RFs will generally have trouble drawing boundaries effectively, since they rely on ranked data. For sparse data we recommend using RF implementations that employ linear combinations of features (either random as originally postulated by Bierman or dimensionally-reduced as postulated by (da Silva, n.d.)). Finally, if data is too complex, the number of nodes that need to be clustered can become prohibitively large, potentially eliminating the advantages of an RF over network methods or online algorithms like DBSCAN.

References

Afanador, Nelson Lee, Agnieszka Smolinska, Thanh N. Tran, and Lionel Blanchet. 2016. “Unsupervised Random Forest: A Tutorial with Case Studies.” *Journal of Chemometrics* 30 (5): 232–41.

Akalin, Altuna, David Fredman, Erik Arner, Xianjun Dong, Jan Christian Bryne, Harukazu Suzuki, Carsten O. Daub, Yoshihide Hayashizaki, and Boris Lenhard. 2009. “Transcriptional Features of Genomic Regulatory Blocks.” *Genome Biology* 10 (4): R38.

Anderson, Kristin G., Heungsup Sung, Cara N. Skon, Leo Lefrancois, Angela Deisinger, Vaiva Vezys, and David Masopust. 2012. “Cutting Edge: Intravascular Staining Redefines Lung CD8 T Cell Responses.” *Journal of Immunology* 189 (6): 2702–6.

Bray, Nicolas L., Harold Pimentel, Páll Melsted, and Lior Pachter. 2016. “Erratum: Near-Optimal Probabilistic RNA-Seq Quantification.” *Nature Biotechnology* 34 (8): 888.

Breiman, Leo. 2001. “Random Forests.” *Machine Learning* 45 (1): 5–32.

Friedman, Jerome, Trevor Hastie, Robert Tibshirani, and Others. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics New York.

Fuente, Alberto de la, Nan Bing, Ina Hoeschele, and Pedro Mendes. 2004. “Discovery of Meaningful Associations in Genomic Data Using Partial Correlation Coefficients.” *Bioinformatics* 20 (18): 3565–74.

Gow, A., C. M. Southwood, J. S. Li, M. Pariali, G. P. Riordan, S. E. Brodie, J. Danias, J. M. Bronstein, B. Kachar, and R. A. Lazzarini. 1999. “CNS Myelin and Sertoli Cell Tight Junction Strands Are Absent in Osp/claudin-11 Null Mice.” *Cell* 99 (6): 649–59.

Helton, E. Scott, and Xinbin Chen. 2007. “p53 Modulation of the DNA Damage Response.” *Journal of Cellular Biochemistry* 100 (4): 883–96.

Horvath, Steve. 2011. *Weighted Network Analysis: Applications in Genomics and Systems Biology*. Springer Science & Business Media.

Hyvärinen, Aapo. 1999. “Survey on Independent Component Analysis.” http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.3488.

Iyama, Teruaki, and David M. Wilson 3rd. 2013. “DNA Repair Mechanisms in Dividing and Non-Dividing Cells.” *DNA Repair* 12 (8): 620–36.

Jolliffe, Ian T., and Jorge Cadima. 2016. “Principal Component Analysis: A Review and Recent Developments.” *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences* 374 (2065): 20150202.

Kim, S. J., N. Ruiz, K. Bezouska, and K. Drickamer. 1992. “Organization of the Gene Encoding the Human Macrophage Mannose Receptor (MRC1).” *Genomics* 14 (3): 721–27.

Mantero, Alejandro, and Hemant Ishwaran. 2021. “Unsupervised Random Forests.” *Statistical Analysis and Data Mining* 14 (2): 144–67.

Margolin, Adam A., Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. 2006. “ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context.” *BMC Bioinformatics* 7 Suppl 1 (March): S7.

Ma, Shuangge, Xiao Song, and Jian Huang. 2007. “Supervised Group Lasso with Applications to Microarray Data Analysis.” *BMC Bioinformatics* 8 (February): 60.

McGarry, T. J., and M. W. Kirschner. 1998. “Geminin, an Inhibitor of DNA Replication, Is Degraded during Mitosis.” *Cell* 93 (6): 1043–53.

Reid, K. B., D. M. Lowe, and R. R. Porter. 1972. “Isolation and Characterization of C1q, a Subcomponent of the First Component of Complement, from Human and Rabbit Sera.” *Biochemical Journal* 130 (3): 749–63.

Shi, Tao, and Steve Horvath. 2006. "Unsupervised Learning With Random Forest Predictors." *Journal of Computational and Graphical Statistics: A Joint Publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America* 15 (1): 118–38.

Silva, Natalia da. n.d. "PPforest Package." *R-Project.org*. https://www.r-project.org/conferences/useR-2015/presentations/62.pdf.

Townes, F. William, Stephanie C. Hicks, Martin J. Aryee, and Rafael A. Irizarry. 2020. "Author Correction: Feature Selection and Dimension Reduction for Single-Cell RNA-Seq Based on a Multinomial Model." *Genome Biology* 21 (1): 179.

Trowbridge, Ian S., Peter Ralph, and Michael J. Bevan. 1975. "Differences in the Surface Proteins of Mouse B and T Cells." *Proceedings of the National Academy of Sciences* 72 (1): 157–61.

Vanlandewijck, Michael, Liqun He, Maarja Andaloussi Mäe, Johanna Andrae, Koji Ando, Francesca Del Gaudio, Khayrun Nahar, et al. 2018. "A Molecular Atlas of Cell Types and Zonation in the Brain Vasculature." *Nature* 554 (7693): 475–80.

Wold, Herman. 1975. "Soft Modelling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach." *Journal of Applied Probability* 12 (S1): 117–42.

Xiao, Yuanyuan, and Mark R. Segal. 2009. "Identification of Yeast Transcriptional Regulation Networks Using Multivariate Random Forests." *PLoS Computational Biology* 5 (6): e1000414.

Ximerakis, Methodios, Scott L. Lipnick, Brendan T. Innes, Sean K. Simmons, Xian Adiconis, Danielle Dionne, Brittany A. Mayweather, et al. 2019. "Single-Cell Transcriptomic Profiling of the Aging Mouse Brain." *Nature Neuroscience* 22 (10): 1696–1708.

Yoshida, K., E. S. Cleaveland, J. W. Nagle, S. French, L. Yaswen, T. Ohshima, R. O. Brady, P. G. Pentchev, and A. B. Kulkarni. 1996. "Molecular Cloning of the Mouse Apolipoprotein D Gene and Its Upregulated Expression in Niemann-Pick Disease Type C Mouse Model." *DNA and Cell Biology* 15 (10): 873–82.