Active Acquisition Methods for Single Cell Genomics

Thesis by
Xiaoqiao Chen

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2025
Defended June 27, 2024

Xiaoqiao Chen
ORCID: 0000-0003-4685-3466

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Professor Dr. Matt Thomson, for his invaluable advice and guidance in my academic and research endeavors. Six years ago, as a student researcher in Thomson's lab, I harbored serious doubts about myself. Professor Thomson assured me of my talent and expressed firm belief in my abilities, providing the initial courage I needed for this journey. Throughout, he has been a steadfast source of unconditional support, teaching me invaluable lessons along the way. He is always patiently and passionately answering even the simplest questions from me. This thesis project would not have been possible without his support over the past six years. Matt, thank you for being such an inspiring advisor!

I would like to thank my committee, Prof. Long Cai, Prof. Yisong Yue, and Prof. Katherine Louise Bouman, for their suggestions and advice on my research during my time in Caltech. I'm grateful to all of them for having taken out copious amounts of time from their busy schedules to examine this thesis and give their suggestions and feedback. Special thanks to Prof. Long Cai, who first gave me the inspiration about active acquisition for spatial genomics, an important part of this thesis. I'm also very grateful to Prof. Yisong Yue for his mentorship, providing a valuable perspective about machine learning and research. I also would like to thank Prof. Katherine Louise Bouman for all valuable suggestions and feedback to my works.

Thanks to my friends at Caltech—Yuchun Sun, Tian Xie, Ruizhi Cao, Zhiquan Yuan, Yongzhao Guo, Yang Zhang, Tianzhe Zheng, Ding Zhong, Christina Wang, Jialong Jiang, and others—for the wonderful meals, game nights, trips, and other enjoyable activities we shared.

I acknowledge the countless individuals who have supported me at every stage of my life.

Finally, I dedicate this thesis to my family. I am profoundly thankful to my mother, Jing Liu, and my father, Qiang Chen, for your unwavering support, love, and encouragement.

ABSTRACT

We introduce two novel computational methodologies, ActiveSVM and Active Cell Inference, aimed at reducing the costs and enhancing the efficiency of single-cell mRNA sequencing and spatial transcriptomics, respectively. ActiveSVM employs an active learning approach to identify minimal yet highly informative gene sets for cell-type classification, physiological state identification, and genetic perturbation responses in single-cell datasets. By focusing on misclassified cells through an iterative process, ActiveSVM efficiently scales to analyze over a million cells, demonstrating around 90% accuracy across various datasets, including cell atlas and disease characterization studies.

Active Cell Inference complements this by utilizing ordered gene sets, developed through ActiveSVM, to streamline spatial genomics measurements. This end-to-end pipeline significantly reduces measurement time and costs by up to 100-fold in scientific and clinical settings. It optimizes the gene probing process by identifying well-classified cells early, allowing for targeted gene application based on cell classification certainty. This method's efficacy is further enhanced by a temporal scaling calibration scheme, improving calibration accuracy throughout its iterative process.

Both methodologies were rigorously tested on the expansive Human Cell Atlas dataset, using the advanced computational tool, CellxGene-Census, involving over 60 million cells. This integration facilitated the creation of precise gene sets for various human tissues, dramatically improving the efficiency and reliability of these cutting-edge genomic techniques. Together, ActiveSVM and Active Cell Inference represent significant advancements in the application of genomics to clinical diagnostics, therapeutic discovery, and genetic screens, promising substantial reductions in the operational complexities and costs associated with next-generation sequencing technologies.

PUBLISHED CONTENT AND CONTRIBUTIONS

Chen, Xiaoqiao, Sisi Chen, and Matt Thomson (2022). "Minimal gene set discovery in single-cell mRNA-seq datasets with ActiveSVM". In: *Nature Computational Science* 2.6. Xiaoqiao Chen proposed the algorithm and applied all experiments, pp. 387–398. DOI: 10.1038/s43588-022-00263-8.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background and Motivation

Single-cell genomics is a powerful technique in genomics that examines individual cells in different tissues, capturing their unique genetic information. Unlike traditional methods that blend and average data from many cells, this approach highlights the distinct roles and functions of each cell. This field has grown quickly due to breakthroughs in DNA and RNA amplification from single cells, enhanced sequencing technologies, and advanced bioinformatics. These advancements allow scientists to detect minor genetic differences and expression patterns between cells, identify rare cell types, and trace cell lineage in developmental studies. Consequently, single-cell genomics has become essential for exploring complex biological processes in health and disease. It provides detailed insights into cancer, neurological disorders, and immune system dynamics, contributing to a detailed map of human cell biology and advancing personalized medicine by targeting specific cellular issues.

Single-cell mRNA sequencing has advanced to routinely profile thousands of cells in each experiment. However, despite its potential to unravel complex biological questions, the high cost of sequencing limits its application in preliminary experiments like drug and genetic screens, as well as in budget-sensitive clinical tests. To mitigate costs, targeted mRNA sequencing has been developed, focusing on key genes to cut costs by as much as 90

In gene regulation, cells orchestrate their gene expression through programs governed by common transcription factors, leading to correlated expressions within these groups. This correlation allows for reconstructing a cell's transcriptional state by analyzing a select few critical genes. Yet selecting these genes computationally can be demanding, especially with the large data volumes produced by single-cell sequencing.

Addressing this, we developed ActiveSVM, an innovative computational approach that selects a minimal yet effective gene set for identifying cell types and transcriptional states. ActiveSVM refines its gene selection through a cyclic process that classifies cells, assesses misclassifications, and enhances the gene set iteratively. This method leverages active learning by consulting an SVM classifier's output to

focus on misclassified cells, thereby ensuring the biological relevance of the selected genes.

ActiveSVM excels in scalability, handling data sets with millions of cells by concentrating on those poorly classified in initial assessments. We have successfully applied this method to various large-scale single-cell studies, effectively identifying small and potent gene sets for tasks ranging from distinguishing cell types in human blood and mouse brain to pinpointing disease markers and region-specific genes in spatial transcriptomics. This approach not only confirms known markers but also discovers novel genes, showcasing the utility of active learning in managing large-scale genomic data efficiently.

Spatial genomics measures the expression of RNA in single cells with spatial resolution using imaging. Right now, the problem is that spatial genomics measurements image many genes in many cells and imaging time is limiting the scaling of spatial genomics to clinical diagnostics. Active acquisition methods can optimize imaging protocols so that the minimum amount of imaging is performed on a sample to accomplish a given clinical task. In many cases, the goal of spatial genomics is to identify the cell types within a given region of sample, for example identifying immune cells within a tumor. For immunotherapy, determining the abundance of T cell and within the tumor and identifying whether the tumor is well or poorly infiltrated a critical.

Here we developed active cell inference and active data acquisition protocol which uses ordered gene sets and a classifier to classify cell types within a sample while minimizing the number of genes that must be imaged per cell. We demonstrate that active inference can reduce aging time dramatically by a factor of 10 to 100 enabling cell classification with this few as 10 genes per cell on average. We demonstrate that acquisition cost varies by tissue with the kidney requiring 10 times more rounds of imaging for comprehensive type classification then the tissues. Finally, we demonstrate that acquisition cost can be modulated dependent on the certainty required in the cell identification task.

1.2 Single-cell Genomics

Single-cell genomics is a powerful and rapidly advancing field of genetic research that enables scientists to examine the genomic material at the level of individual cells. This approach provides a high-resolution view of genetic diversity and function within a mixed population of cells, which is crucial for understanding complex

biological systems and diseases.

Origins and Development

The development of single-cell genomics has been driven by advancements in microfluidic technology, sophisticated imaging techniques, and high-throughput sequencing. Traditional genomic studies, which often analyze bulk samples containing numerous cells, tend to obscure the distinct genetic identities and states of individual cells. Single-cell genomics emerged to address these limitations, offering a more granular view of genomics (Tang et al., 2009) (Stuart and Satija, 2019).

Techniques and Technologies

Key technologies in single-cell genomics include single-cell RNA sequencing (scRNA-seq), single-cell DNA sequencing, and more recently, single-cell epigenomic sequencing. These technologies allow researchers to explore not only the genetic code but also the active gene expression and regulatory mechanisms at the single-cell level.

Single-cell RNA sequencing (scRNA-seq) provides a snapshot of the active genes within a cell at the moment of sampling, offering insights into the cell's functional state.(Hebenstreit, 2012).

Single-cell DNA sequencing is used to examine the genomic variations, such as mutations and rearrangements, within individual cells, which is particularly useful in cancer research.(Navin et al., 2011).

Single-cell epigenomics focuses on detecting chemical modifications of the DNA and histones that regulate gene expression without altering the underlying DNA sequence.(Assaf Rotem et al., 2015).

Applications

The applications of single-cell genomics are vast and transformative across various fields:

In developmental Biology, it helps in mapping cellular differentiation pathways and understanding the complex dynamics of embryonic development.

In oncology, single-cell techniques are crucial for identifying tumor heterogeneity, tracking the evolution of cancer cells, and developing targeted therapies.(Trapnell et al., 2014). It helps in identifying intra-tumor heterogeneity and in the development of targeted therapies.(Tirosh et al., 2016).

In immunology, researchers can define the phenotypes of immune cells more precisely and understand their roles in health and disease.

In neuroscience, single-cell genomics enables the classification of neuron types and helps in mapping complex neural circuits.(Villani et al., 2017). It facilitates neuron type classification and neural circuit mapping.(Zeisel et al., 2015).

Challenges and Future Directions

While single-cell genomics is highly informative, it faces challenges such as technical noise and the need for complex data analysis. Future efforts will likely focus on enhancing data accuracy and developing better computational tools to handle large datasets.(Stegle, Teichmann, and Marioni, 2015).

In summary, single-cell genomics offers unparalleled insights into the molecular mechanics of individual cells, driving advances in many biomedical fields and potentially revolutionizing clinical diagnostics and personalized medicine. (Regev et al., 2017).

1.3 Spatial Genomics

Spatial genomics is an innovative field that combines the power of genomic analysis with the precise localization of genetic activity within tissues. This approach allows scientists to not only understand what genes are active in a sample but also where those genes are expressing within the tissue's architecture. This spatial context adds a crucial dimension to genomic data, enhancing our understanding of the complex interplay between cells and their microenvironments in health and disease.

The technology behind spatial genomics involves techniques that maintain the spatial integrity of a tissue sample while performing high-throughput genomic or transcriptomic analysis. By mapping the gene expression profiles to specific locations, researchers can observe how cells and their genes interact within their native environments. This is particularly valuable in complex tissues like the brain or tumors, where the spatial arrangement of cells contributes significantly to function and disease progression.

Spatial genomics applications are broad, ranging from oncology, where it can help identify the tumor microenvironment and its impact on cancer progression, to neuroscience, where it elucidates the organization of cell types in the brain. It's also proving invaluable in developmental biology, providing insights into how cells differentiate and organize during growth.

For example, the work of Ståhl et al., 2016 introduced a methodology that combines histological analysis with high-throughput RNA sequencing. This approach allows for the visualization and quantitative analysis of gene expression across tissue sections, providing a powerful tool for biomedical research.

Similarly, the technology has been further refined and utilized in studies such as those by Rodriques et al., 2019, who demonstrated the ability to capture the transcriptome of tissues with precise spatial resolution, significantly enhancing the understanding of tissue architecture and function.

These examples illustrate how spatial genomics is transforming our ability to correlate genetic activity with specific locations within tissues, enhancing the potential for breakthroughs in personalized medicine and targeted therapy development.

Chapter 2

ACTIVESVM: DISCOVERS MINIMAL GENE-SETS FOR CLASSIFYING CELL-TYPES AND DISEASE STATES WITH SINGLE-CELL MRNA-SEQ

Sequencing costs currently prohibit the application of single-cell mRNA-seq to many biological and clinical analyses. Targeted single-cell mRNA-sequencing reduces sequencing costs by profiling reduced gene sets that capture biological information with a minimal number of genes. Here, we introduce an active learning method (ActiveSVM) that identifies minimal but highly-informative gene sets that enable the identification of cell-types, physiological states, and genetic perturbations in single-cell data using a small number of genes. Our active feature selection procedure generates minimal gene sets from single-cell data through an iterative cell-type classification task where misclassified cells are examined at each round of analysis to identify maximally informative genes through an 'active' support vector machine (ActiveSVM) classifier. By focusing computational resources on misclassified cells, ActiveSVM scales to analyze data sets with over a million single cells. We demonstrate that ActiveSVM feature selection identifies gene sets that enable 90% cell-type classification accuracy across a variety of data sets including cell atlas and disease characterization data sets. The method generalizes to reveal genes that respond to genetic perturbations and to identify region specific gene expression patterns in spatial transcriptomics data. The discovery of small but highly informative gene sets should enable substantial reductions in the number of measurements necessary for application of single-cell mRNA-seq to clinical tests, therapeutic discovery, and genetic screens.

2.1 Introduction

Single-cell mRNA-seq methods have scaled to allow routine transcriptome-scale profiling of thousands of cells per experimental run. While single cell mRNA-seq approaches provide insights into many different biological and biomedical problems, high sequencing costs prohibit the broad application of single-cell mRNA-seq in many exploratory assays such as small molecule and genetic screens, and in cost-sensitive clinical assays. The sequencing bottleneck has led to the development of targeted mRNA-seq strategies that reduce sequencing costs, by up to 90%, by

focusing sequencing resources on highly informative genes for a given biological question or an analysis (Heimberg et al., 2016; H. C. Fan, G. K. Fu, and Fodor, 2015; Replogle et al., 2020; Marshall et al., 2020; Riemondy et al., 2019). Commercial gene-targeting kits, for example, reduce sequencing costs through selective amplification of specific transcripts using $\sim$ 1000 gene-targeting primers.

Cells modulate gene expression through the regulation of transcriptional programs or modules that contain multiple genes regulated by common sets of transcription factors (Heimberg et al., 2016). Genes within transcriptional modules exhibit correlated gene expression due to co-regulation. Correlations in gene expression can enable the transcriptional state of a cell to be reconstructed through the targeted mRNA profiling of a small number of highly informative genes (Heimberg et al., 2016; Replogle et al., 2020). However, such targeted sequencing approaches require computational methods to identify highly informative genes for specific biological questions, systems, or conditions. A range of computational approaches including differential gene expression analysis and principal components analysis (PCA) can be applied to identify highly informative genes (Heimberg et al., 2016). Yet current methods for defining minimal gene sets are computationally expensive to apply to large single-cell mRNA-seq data sets and often require heuristic user-defined thresholds for gene selection (Delaney et al., 2019; F. Wang et al., 2019). As an example, computational approaches based upon matrix factorization (PCA,Non-negative matrix factorization), are typically applied to complete data sets and so are computationally intensive when data sets scale into the millions of cells (Bhaduri et al., 2018). Further, gene set selection after matrix factorization requires heuristic strategies for thresholding coefficients in gene vectors extracted by PCA or NNMF, and then asking whether the selected genes retain core biological information.

Here, inspired by active learning(Felder and Brent, 2009) approaches, we develop a computational method that selects minimal gene sets capable of reliably identifying cell-types and transcriptional states through an 'active' support vector machine classification task (ActiveSVM) (Rückstieß, Osendorfer, and Smagt, 2011; Noble, 2006). The ActiveSVM algorithm constructs a minimal gene set through an iterative cell-state classification task. At each iteration, ActiveSVM applies the current gene set to classify cells into classes that are provided by unsupervised clustering of cell-states or by used-supplied experimental labels. The procedure analyzes cells that are misclassified with the current gene set, and, then, identifies maximally informative genes that are added to the growing gene set to improve classification. Traditional

active learning algorithms query an oracle for training examples that meet a criteria (Settles, 2009). The ActiveSVM procedure actively queries the output of an SVM classifier for cells that classify poorly, and then performs detailed analysis of the specific misclassified cells to select maximally informative genes. By selecting minimal gene sets through a well-defined, classification task, we ensure that the gene sets discovered by ActiveSVM retain biological information.

The central contribution of ActiveSVM is that the method can scale to large single-cell data sets with more than one million cells. ActiveSVM scales to large data sets because the procedure focuses computational resources on poorly classified cells. Since the algorithm only analyzes the full-transcriptome of cells that classify poorly with the current gene set, the method can be applied to large data sets to discover small sets of genes that can distinguish between cell-types at high accuracy. We demonstrate that ActiveSVM can analyze a mouse brain data set with 1.3 million cells and requires only hours of computational time. In addition to scaling, the ActiveSVM classification paradigm generalizes to a range of single-cell data analysis tasks including the identification of disease markers, genes that respond to Cas9 perturbation, and the identification of region specific genes in spatial transcriptomics.

In conventional SVM based feature selection, the user would first train an SVM classifier on the complete data set and then select features according to the absolute values of the individual gene weights w (Chang and Lin, 2008) requiring analysis of the complete data set as well as heuristic strategies for defining weight thresholds.

To demonstrate the performance of ActiveSVM, we apply the method to a series of single-cell genomics data sets and analysis tasks. We identify minimal gene sets for cell-state classification in human peripheral blood mononuclear cells (PBMCs) (G. X. Zheng et al., 2017), the megacell mouse brain data set (Genomics, 2017), and the Tabula Muris mouse tissue survey(Consortium et al., 2018). We identify disease markers that distinguish healthy and multiple myeloma patient PBMCs (S. Chen et al., 2020). To highlight the generality of the method, we apply ActiveSVM to identify genes impacted by Cas9 based gene-knock down in perturb-seq (Dixit et al., 2016) and demonstrate that ActiveSVM can identify gene sets that mark specific spatial locations of a tissue through analysis of spatial transcriptomics data (Eng et al., 2019). Gene sets constructed by ActiveSVM are both small and highly efficient, for example, classifying human immune cell types within PMBCs using as few as 15 genes and classifying 55 cell-states in Tabula Muris with < 150 genes.

The gene sets we discover include both classical markers and genes not previously established as canonical cell-state markers. Conceptually, ActiveSVM demonstrates that active sampling strategies can be applied to enable the scaling of algorithms to the large data sets generated single-cell genomics.

2.2 Method

In the conventional Sequential Feature Selection (SFS) (Rückstieß, Osendorfer, and Smagt, 2011), features are selected one-by-one in a greedy strategy to optimize an objective function. Here, we develop an active SVM (ActiveSVM) feature selection method, where we only analyze the subset of incorrectly classified cells at the current step and then select the new gene features based upon those cells. This active learning strategy enables the efficient computation of small gene sets across large data sets by minimizing the total number of cells and genes that are analyzed.

A common work-flow in single-cell mRNA-seq experiments defines a series of cell-states or cell-types using unsupervised clustering of cells (Wolf, Angerer, and Theis, 2018; Macosko et al., 2015). We developed a computational framework based on support vector machine (SVM) classifier to identify minimal gene sets that distinguish a set of cell-states in single-cell data. The procedure is an 'active' formulation of classical Sequential Feature Selection (SFS) (Rückstieß, Osendorfer, and Smagt, 2011). In the conventional SFS approach, features are selected one-by-one in a greedy fashion to optimize an objective function. To reduce computational burden, we propose an active feature selection framework where we only use the subset of incorrectly classified cells at current step and then select additional features based upon those cells. The active learning strategy enables efficient computation of minimal gene sets across large data sets by minimizing the total number of cells and genes that are analyzed.

The algorithm can accept the cell-state labels that are typically derived from unsupervised clustering. We, then, utilize the cell-state labels to identify a minimal set of marker genes that can retain the separation between cell-states with a minimal set of gene features. We note that our method can also accept user supplied cell-type labels as input if a user seeks to identify new genes that separate cell-states based upon biologically curated markers.

In summary, our algorithm is applied to single-cell gene expression data and takes, as input, gene expression data and cell-type labels. Alternatively, we generate the cell-type labels using unsupervised clustering. Our procedure, then, starts with

an empty gene set, an empty cell set and a list of candidate genes and cells. The algorithm iteratively selects genes and classifies cells using identified genes by training an SVM model to classify cell-types. The algorithm identifies cells in the data set that classify poorly given the current gene set, and uses those cells to select additional genes to improve classification accuracy on the entire data set.

In single cell gene expression data, we use $x_i^{(j)} \in \mathbb{R}$ to denote the measurement of the j-th gene of the i-th cell. We assume the classification labels are given and consider a data-set $\{x_i, y_i\}_{i\in\{1,\ldots,N\}}$ contains N cells with total M genes, where $x_i = [x_i^{(j)}]_{j\in\{1,\ldots,M\}}$ and $y_i \in \mathbb{Z}^N$ are labels. The labels could be binary or multi-class and can be derived from clustering. We also denote the gene expression vector of i-th cell with part of genes as $x_i^{(D)} = [x_i^{(j)}]_{j\in D}$, where $D \subset \{1, \ldots, M\}$. And we use J and I to refer to the set of selected genes and cells.

First, to seed the gene list, the algorithm selects c cells at random out of the total set of N cells and adds them to the cell set. The parameter c is determined by the user. The algorithm, then, trains an SVM on the cell set, which defines an SVM margin w that optimally separates cells into classes that are consistent with labels on this seed set. A gene selection strategy we developed, max margin rotation, evaluates all candidate genes based on the margin w and one gene with the highest score is added to the gene set. A second SVM model is, then, learned given the current gene set, and, we identify cells that classify poorly given the current gene list. Then we sample c misclassified cells to identify genes that improve classification in the next step of our procedure. The integrated algorithm is shown in Algorithm 1.

Algorithm 1: ActiveSVM

Input: $c, k \in \mathbb{N}, J = \emptyset$

Output: J

Randomly or 'balanced' select c cells $I \subset \{1, \ldots, N\}, |I| = c$

Train a 1-D SVM model on training set I for each candidate gene: $\{h_{w,b}^{(j)}\}_{j \in \{1,\ldots,M\}}$

$loss_j = \sum_{i \in I} \max\{0, 1 - y_i h_{w,b}^{(j)}(x_i^{(j)})\}$

Select one gene $j_0 \in \{1, \ldots, M\}$ with lowest $loss_j$

$J = J \cup \{j_0\}$

repeat

- Optimize (1) and get optimal solution $\{\alpha_i^*\}_{i=1}^N$
- Get the the set of misclassified cells $S \subset \{1, \ldots, N\}$ with $\alpha_i^* = C$
- **if** *min-complexity* **then**
  - Randomly or 'balanced' select c cells $I \subset S$, where $|I| = c$;
- **else**
  - **if** *min-cell* **then**
    - $c' = \min\{c, |I \cap S|\}$;
    - Randomly or 'balanced' select $c - c'$ cells $I' \subset S \setminus I$, where $|I'| = c - c'$;
    - $I = I \cup I'$
  - **end**
- **end**
- $w = \sum_{i \subset I} \alpha_i^* y_i x_i^{(J)}$
- $w_{padded} = [w, 0]$
- For each $j \in \{1, \ldots, M\} \setminus J$, optimize (4) and get optimal solution $\{\alpha_i^{*(j)}\}_{i \in I}$
- $w_j = \sum_{i \in I} \alpha_i^{*(j)} y_i x_i^{(J \cup \{j\})}$
- $\vartheta_j = \arccos \cos \vartheta_j = \arccos \frac{<w_j, w_{padded}>}{\|w_j\| \|w_{padded}\|}$
- Select one gene $j^* \in \{1, \ldots, M\} \setminus J$ with largest ϑ_j
- $J = J \cup \{j^*\}$

until $|J| \geq k$

The two novel components of the method are the strategies to evaluate and select genes and cells at each iteration. Specifically, we identify cells that classify poorly and use misclassified cells to identify highly informative genes. To select highly informative genes given the misclassified cells, a range of different strategies can be applied. In the conventional SVM, the procedure would sort features according

to the absolute values of the components of weight w.(Chang and Lin, 2008) we developed a gene selection strategy, Max Margin Rotation (MMR), that evaluates all candidate genes and selects the gene that induces maximum rotation of the margin w. The ActiveSVM algorithm continues iteration until a max gene number, k, is reached. The max gene number k can be set as any integer smaller than M and can be set to small values during exploratory analysis and to larger values for more exhaustive exploration of a data set. The integrated algorithm is shown in Algorithm 1 and visualized as Figure 2.1.

The most important feature of our ActiveSVM procedure is that the algorithm must never load an entire data set into memory. At each step, the procedure performs classification of cells using a minimal gene set, and then performs detailed (all genes) analysis of only a subset of misclassified cells. Due to the design of the procedure, ActiveSVM can analyze large data sets that do not easily fit in memory. In conventional SVM based feature selection, the user would first train an SVM classifier on the complete data set and then select features according to the absolute values of the components of weight w (Chang and Lin, 2008). We note that conventional feature selection procedures typically apply classification accuracy for feature selection. Conventional SFS often selects features based upon improvement in classification accuracy. We found empirically that MMR provides improved classification results and so selected MMR as our gene selection strategy.

Based on the above outline of ActiveSVM, we can formalize the specific, mathematical gene and cell selection strategies into two defined rules. Assume the SVM classifier notation of one observation is $h_{w,b}(x_i^{(D)}) = g(w^T x_i^{(D)} + b)$ for any $i \in \{1, 2, \ldots, N\}$ and $D \subset \{1, 2, \ldots, M\}$ with respect to observation $x \in \mathbb{R}^{|D|}$, where $w \in \mathbb{R}^{|D|}$ and $b \in \mathbb{R}$ are parameters (the margin and bias respectively). Here, $g(z) = 1$ if $z \geq 0$, and $g(z) = -1$ otherwise. And the loss function is Hinge Loss (Rosasco et al., 2004) $\text{loss}_i = \max\{0, 1 - y_i(w^T x_i^{(D)} + b)\}$, where $y_i \in \mathbb{R}$ is the ground truth label of observation x_i.

Cell selection: identification of maximally informative cells

We formalize the ActiveSVM procedure and define mathematical rules that encode our specific gene and cell selection strategies. For notation, in single-cell gene expression data, we use $x_i^{(j)} \in \mathbb{R}$ to denote the measurement of the j-th gene of the i-th cell. We assume the classification labels are given and consider a data-set $\{x_i, y_i\}_{i \in \{1,\ldots,N\}}$ contains N cells with total M genes, where $x_i = [x_i^{(j)}]_{j \in \{1,\ldots,M\}}$ and

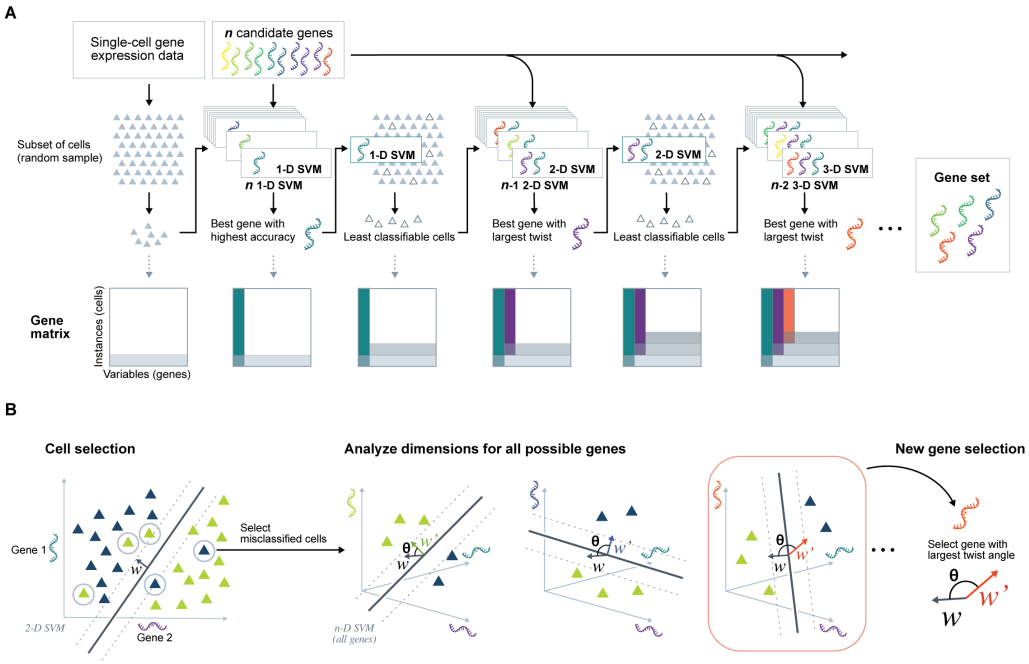


Figure 2.1: Figure A shows the gene selection workflow with single cell gene expression dataset. The purpose is to select a gene set from n candidate genes for cell classification based on SVM. First a subset of cells are randomly sampled for selecting the first gene. Then we train n single 1-D SVM models for each candidate gene, where the training set is only the set of randomly sampled cells at previous step. The gene (the green one) corresponding to the SVM with highest accuracy is selected. Then the least classifiable cells of this selected 1-D SVM should be the training set to select the second gene. Based on this, n-1 2-D SVM are trained, where one dimension is the selected green gene and the other dimension is each unselected candidate gene. The gene (the purple one) corresponding to the SVM with largest twist angle of SVM weight is the second selected gene. The procedure repeats many times and the final gene set would be built. The gene matrix at the last row shows the part of training data we use. The total size of data should be much smaller than the entire original dataset. Figure B is the details of twist angle when selecting the third gene. The mis-classified cells are sampled from the selected 2-D SVM to train 3-D SVMs. w is the weight of the 2-D SVM. For each unselected candidate gene, a 3-D SVM is trained, where two dimensions are the same genes with the selected 2-D SVM and the second dimension is the new candidate gene. The new weight of each 3-D SVM is w' and the twist angle θ is the angle between w and w'. The number of 3-D SVMs and their corresponding twist angles is totally n-2. The gene corresponding to the 3-D SVM weight with largest twist angle θ is selected.

$y_i \in \mathbb{Z}$ are labels. The labels could be binary or multi-class and can be derived from clustering. We also denote the gene expression vector of i-th cell with part of genes as $x_i^{(D)} = [x_i^{(j)}]_{j \in D}$, where $D \subset \{1, \ldots, M\}$. And we use J and I to refer to the set of selected genes and cell set.

We adopt the SVM classifier notation of one observation is $h_{w,b}(x_i^{(D)}) = g(w^T x_i^{(D)} + b)$ for any $i \in \{1, 2, \ldots, N\}$ and $D \subset \{1, 2, \ldots, M\}$ with respect to observation $x \in \mathbb{R}^{|D|}$, where $w \in \mathbb{R}^{|D|}$ and $b \in \mathbb{R}$ are parameters (the margin and bias respectively). Here, $g(z) = 1$ if $z \geq 0$, and $g(z) = -1$ otherwise. And the loss function is Hinge Loss(Rosasco et al., 2004) $\text{loss}_i = \max\{0, 1 - y_i(w^T x_i^{(D)} + b)\}$, where $y_i \in \mathbb{R}$ is the ground truth label of observation x_i.

For the cell selection strategy, we identify cells with the largest SVM classification loss. In SVM classification, samples separable in n-D are also separable in $(n+1)$-D as they are at least separated by the same boundary with zero at the $(n+1)$-th dimension. Therefore, to improve the classification accuracy with a new gene, we should only consider the misclassified cells. We identify such cells through analysis of the dual form of the classical SVM classification problem. After solving the primal optimization problem of soft margin SVM, we have the dual optimization problem with a non-negative Lagrange multiplier $\alpha_i \in \mathbb{R}$ for each inequality constraint.(Bottou and Lin, 2007).

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i_1, i_2 = 1}^{N} y_{i_1} y_{i_2} \alpha_{i_1} \alpha_{i_2} < x_{i_1}^{(J)}, x_{i_2}^{(J)} > \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{2.1}$$

Here $x_i^{(J)}$ refers to the measurement of the i-th cell with all selected genes, and $C \in \mathbb{R}$ is a hyper-parameter we set to control the trade-offs between size of margin and margin violations when samples are non-separable.

We solve the optimal solution α^* and apply the Karush-Kuhn-Tucker(KKT) dual-complementarity conditions (Gordon and Tibshirani, 2012) to obtain the following results where $w \in \mathbb{R}^{|J|}$ and the intercept term $b \in \mathbb{R}$ are optimal.

$$\begin{aligned} \alpha_i^* = 0 \quad &\Rightarrow \quad y_i(w^T x_i^{(J)} + b) > 1 \\ \alpha_i^* = C \quad &\Rightarrow \quad y_i(w^T x_i^{(J)} + b) < 1 \\ 0 < \alpha_i^* < C \quad &\Rightarrow \quad y_i(w^T x_i^{(J)} + b) = 1. \end{aligned} \tag{2.2}$$

Therefore, for each cell, the Lagrange multiplier α_i indicates whether the cell falls within the SVM margin defined by the vector w. $\alpha_i > 0$ means $y_i(w^T x_i + b) \leq 1$, i.e. cells are on or inside the SVM margin. Hence, we can directly select cells with $\alpha_i > 0$. In practice, we typically only select cells with $\alpha_i = C$, which indicates incorrectly classified cells.

Using this mathematical formulation, we develop two different versions of the ActiveSVM procedure, the min-complexity strategy and min-cell strategy, for distinct goals. The min-complexity strategy minimizes the time and memory consumption when computational resources are restricted or where a user desires to reduce runtime. In the min-complexity strategy, a fixed number of cells is sampled among all misclassified cells and used as the cell set for gene selection in each iteration. Therefore, a small number of cells can be analyzed at each round and typically only few cells might be selected repeatedly. The two strategies are discussed in more detail in the Methods section. We also developed random and balanced strategies for sampling cells across a series of cell-states with varying cell membership.

Gene selection by maximizing margin rotation

To select maximally informative genes at each round, we analyze misclassified cells and identify genes that would induce the largest rotation of the classification margin. Our procedure is inspired by the active learning method, Expected Model ChangeSettles, 2009 twist angle induced in w when we add a new dimension (gene) to the classifier. Assume J is the set of genes we have selected so far. Once we add a gene into the $|J|$-dimensional data space, the parameter w will have one more dimension. The rotation of margin measures how much w twists after adding the new dimension compared with weight in the previous iteration.

Specifically, assume J is the set of genes we have selected so far. We derive the corresponding w from the optimal solution α^*.(Bottou and Lin, 2007) After solving the dual optimization problem (1), we have:

$$w = \sum_{i \in I} \alpha_i^* y_i x_i^{(J)}. \tag{2.3}$$

Then we pad w with zero to get a $|J+1|$-dimensional weight w_{padded}, whose first $|J|$ dimensions is w and the $|J+1|$-th dimension is zero.

For each candidate gene j, we train a new $|J+1|$-dimensional SVM model and have weight w_j,where $j \in \{1,\ldots,M\} \setminus J$. That is to say, for candidate gene j, we solve the dual optimization problem (4) and find a new optimal multiplier $\alpha^{*(j)}$. Note that we only use the selected cells here, $i_1, i_2 \in I$.

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i\in I} \alpha_i^{(j)} - \frac{1}{2} \sum_{i_1,i_2\in I} y_{i_1} y_{i_2} \alpha_{i_1}^{(j)} \alpha_{i_2}^{(j)} \langle x_{i_1}^{(J\cup\{j\})}, x_{i_2}^{(J\cup\{j\})} \rangle \\ \text{s.t.} \quad & 0 \le \alpha_i^{(j)} \le C \\ & \sum_{i\in I} \alpha_i^{(j)} y_i = 0 \end{aligned} \tag{2.4}$$

Then we have w_j as shown in equation (5):

$$w_j = \sum_{i\in I} \alpha_i^{*(j)} y_i x_i^{(J\cup\{j\})}. \tag{2.5}$$

The angle θ_j between w_j and w_{padded} is the expected angle the margin rotates, corresponding to the j-th candidate gene. Then the j-th gene with largest angle θ_j will be selected. We measure the angle between two vectors using cosine similarity(P. Xia, L. Zhang, and F. Li, 2015):

$$\vartheta_j = \arccos\cos\vartheta_j = \arccos\frac{\langle w_j, w_{padded}\rangle}{\| w_j \| \| w_{padded} \|}. \tag{2.6}$$

Therefore, a new gene, which maximizes ϑ_j, is selected to maximize the expected model change.

Multi-class ActiveSVM

For multi-class classification, the SVM is handled according to a one-vs-rest scheme, where a separate classifier is fit for each class, against all other classes. Margin rotation is represented as the sum of weight components in each class dimension. Hence with Z classes, we get Z weight components corresponding to Z one-vs-the-rest classification decision boundaries. Assume the weight component for class z of the previous $|J|$-dimensional SVM model is $w^{(z)}$. Denote the $|J+1|$-dimensional weight after zero-padding of $w^{(z)}$ as $w_{padded}^{(z)}$ and the new $|J+1|$-dimensional weight component of class z with j-th gene as $w_j^{(z)}$, where $z \in 1,\ldots,Z$. Then we have:

$$\vartheta_j^{(z)} = \arccos\cos\vartheta_j^{(z)} = \arccos\frac{\langle w_j^{(z)}, w_{padded}^{(z)}\rangle}{\| w_j^{(z)} \| \| w_{padded}^{(z)} \|} \tag{2.7}$$

$$\vartheta_j = \sum_{z=1}^{Z} \vartheta_j^{(z)}. \tag{2.8}$$

Min-cell and min-complexity cell selection strategies

In the min-cell strategy, to reduce the number of unique cells required, the misclassified cells already used in previous steps are given the highest priority to select again. Therefore, the min-cell strategy attempts to re-use cells across rounds of iteration and aims to minimize the total number of unique cells we acquire during the entire procedure. The min-cell strategy can be applied to limit the number of cells required to perform the analysis in settings where cell acquisition might be limiting including in the analysis of rare cell populations or in clinical data sets.

For the min-cell strategy, assume we select c cells for each iteration and there are $a+b$ misclassified cells at the current iteration, where a cells have been used at least once in previous iterations while b cells are new cells. If $a \geq c$, we do not need to add any new cells to current cell set. If $a < c$, we sample $c - a$ cells among the b new cells. Then the algorithm uses the whole selected cell set for the next gene selection step. When using the min-cell strategy, cells tend to be re-used many times and the curve of number of unique cells we acquire converges to a fixed value along with the number of genes we select. In experiments, the number of cells selected for each step, c, is a hyper-parameter set by the user. Typically, the parameter can be set to a small number using the min-complexity strategy, as a sufficient number of new cells is considered in the procedure. Selecting a small number of cells each round reduces computational complexity. In the min-cell strategy it can be advantageous to select a larger number of total cells to guarantee diversity of training cells while still bounding the total number of cells used.

Balancing cell-sampling across cell-classes

In addition to the min-cell and min-complexity options, we also include two version of cell sampling strategies. The first one is uniform, random sampling. Another option is cell 'balanced' sampling that can be applied to balance sampling across a series of cell classes. In the 'balanced' strategy, we sample a fixed number of cells from each cell class, and for classes with insufficient cells we sample all the cells in the class. Mathematically, assume there are Z classes and S is the set of all

misclassified cells this step. We should sample c' cells from a candidate cell set, S', for the current iteration. In min-complexity strategy, $c' = c$ and the candidate cell set, S', should be S itself. For the min-cell strategy, $c' = c - \min\{c, |I \cap S|\}$, where I is the cell set before current iteration, and the candidate cell set $S' = S \setminus I$. Assume $S' = \cup_{z=1}^{Z} S'_z$, where S'_z are the set of cells in class z, and $|S'_z| \leq |S'_{(z+1)}|$ for any $z \in \{1, 2, ..., Z-1\}$. We sample cells in order from class 1 to class Z and denote P_z as the union set of all selected cells from all classes after class z. Then, for class z, if $|S'_z| \leq (|S'| - |P_{z-1}|)/(Z - z + 1)$, we select all cells in S'_z. Otherwise, if $|S'_z| > (|S'| - |P_{z-1}|)/(Z - z + 1)$, we randomly sample $(|S'| - |P_{z-1}|)/(Z - z + 1)$ cells in S'_z. The procedure repeats for all classes and then we have P_Z as the cells we select at this iteration.

Incorporation of cell labels derived from unsupervised analysis, experimental conditions, or biological knowledge

The goal of ActiveSVM is to discover minimal gene sets for extracting biological information from single-cell data sets. To define minimal gene sets, we apply a classification task in which we find genes that enable a SVM classifier to distinguish single-cells with different labels (y_i). In practice, explicit cell-type labels are often not known for a data set. An extremely common work-flow in single-cell genomics applies Louvain clustering algorithms to identify cell classes and visualizes these cell classes in UMAP or t-SNE plots (Macosko et al., 2015; Wolf, Angerer, and Theis, 2018). The cell clusters output by clustering work-flows in commonly used single-cell analysis frameworks provide a natural set of labels for down-stream analysis. In fact, ActiveSVM can, then, identify specific marker genes for interpreting the identified cell-clusters and determining their biological identify. More broadly, cell-class labels can be quite general including the identity of a genetic perturbation (Figure 2.6), the spatial location of a cell (Figure 2.7). We can imagine the application of ActiveSVM to a broad set of additional labels including membership to a differentiation trajectory or lineage tree (Street et al., 2018).

2.3 Time and Memory Complexity

Memory complexity of ActiveSVM

One of the key contribution of ActiveSVM is that it substantially saves memory usage because only a small part of data is used at each iteration. The entire dataset can be stored in disk and the algorithm only loads two small matrices into memory: a $N \times |J|$ matrix of all cells with the currently selected genes and a $|I| \times M$ matrix of

the cell set with all genes. The memory complexity is $O(M+N)$ while the memory complexity of algorithms using the entire dataset is at least $O(MN)$. The min-cell strategy minimizes the total number of unique cells required to reduce the cost of data measurement, acquisition, and storage.

Time complexity of ActiveSVM

The time complexity of the complete procedure depends primarily on the training of SVM. The standard time complexity of SVM training is usually $O(MN^2)$ (Abdiansah and Wardoyo, 2015). Assume that we plan to select $k \in \mathbb{N}$ genes in total and use the cell set I_i of poorly classified cells at i-th iteration, where $k, k^2 \ll M$ and $|I_i|, |I_i|^2 \ll N$ are constants. Then the computational complexity of ActiveSVM is:

$$O(\sum_{i=1}^{k} (i \cdot N^2 + (M-i) \cdot (i+1) \cdot |I_i|^2)) \sim O(N^2 + M).$$

The key reduction in total complexity occurs because each step is performed using N cells with of order $k, k^2 \ll M$ genes or using order M genes with $|I_i|$ cells. Therefore, the polynomial $O(MN^2)$ is reduced to two separate steps that are individually $O(N^2)$ and $O(M)$.

And in practice, we implement ActiveSVM using the linear SVM library LIBLINEAR(R.-E. Fan et al., 2008), whose time complexity is $O(MN)$. Therefore, and the corresponding time complexity of ActiveSVM with LIBLINEAR is:

$$O(\sum_{i=1}^{k} (i \cdot N + (M-i) \cdot (i+1) \cdot |I_i|)) \sim O(N + M).$$

In the gene selection part, the margin rotation angles of all candidate genes can be computed in parallel, which also accelerates the algorithm. The complexity provides a substantial improvement in marker gene selection methods especially for large-scale datasets.

Computational Infrastructure

To analyze computational requirements of ActiveSVM, we performed analysis using an r5n.24xlarge, a type of EC2 virtual server instance on AWS, with 96 virtual central processing units (vCPU) and 768 GiB memory on Linux system. The instance allowed us to track run time and memory usage. As an example, for the largest data set analysis, we applied ActiveSVM to select 50 genes on the largest

dataset, mouse brain 'megacell' dataset, which contains 1306127 cells and 27998 genes, using ActiveSVM and some other popular feature selected methods, including correlation coefficient, mutual information, feature importance by decision tree, and conventional SVM. The peak memory usage of ActiveSVM is 2111 MB while other methods all consume more than 78600 MB. The run time of the min-complexity method is about 69 minutes and of the min-cell method is about 243 minutes. Each comparison method takes more than 4 days on the same server machine. The run time and peak memory usage of ActiveSVM on all six datasets are shown in Supplementary Table 2.1. The ActiveSVM package used for the brain megacell dataset only loads the selected genes and cells into memory at each iteration while other two experiments called the package loading the entire dataset. Both packages are provided in Code Availability Section.

2.4 Datasets and Experiments

We test our ActiveSVM feature-selection method on four single-cell mRNA-seq datasets: a dataset of peripheral blood mononuclear cells (PBMCs) (G. X. Zheng et al., 2017), the megacell 1.3 million cell mouse brain data set (Genomics, 2017), the Tabula Muris mouse tissue survery dataset (Consortium et al., 2018), and a multiple myeloma human disease dataset (S. Chen et al., 2020). Later, we demonstrate generalization of the strategy to additional types of single-cell data analysis, including a perturb-seq dataset where genes impacted by Cas9 based genetic perturbation, and a spatial transcriptomics dataset by seqFish+.

For each analysis, we show the classification accuracy of the test set along with the number of genes we select. We also compare the classification performance to several widely-used feature selection methods, including conventional SVM, correlation coefficient (Taylor, 1990), mutual information (Vergara and Estévez, 2014), Chi-square (McHugh, 2013), feature importance by decision tree (Safavian and Landgrebe, 1991), and randomly sample genes, showing that ActiveSVM obtains the highest accuracy. All of the comparison methods select genes one by one and select a new gene with the largest score in terms of the corresponding evaluation functions while using the same number of cells as our method. However, all methods randomly sample cells at each iteration without an active learning approach. For perturb-seq and seqFish+ datasets, we also show the accuracy performance of comparison methods, where the entire dataset is used. Specifically, conventional SVM based feature selection also called naive SVM selects the gene with largest weight component, which is the most popular SVM feature selection method. In our

application of ActiveSVM, we tested both the min-cell strategy and min-complexity strategies as well as randomly sampling and 'balanced' sampling.

In each experiment, the data set was first pre-processed and normalized using standard single-cell genomics strategies (See Data Pre-processing). The entire dataset was, then, randomly split into training set with the size of 80% and test set with the size of 20%. For conventional and ActiveSVM, we found the approximately optimal parameter by grid-search (Syarif, Prugel-Bennett, and Wills, 2016) across lists of candidate values for some key parameters in the framework of 3-fold cross validation (Arlot and Celisse, 2010). The optimal parameters were fixed during all iterations. For the comparison methods, we use 3-fold cross validation grid-search to obtain the optimal parameters at each single iteration. We also implemented the algorithms called `min_complexity_cv` and `min_acquisition_cv` that apply grid-search and cross validation for each single SVM trained in each iteration (see Code Availability). The parameter setting details are shown in the Parameters section.

In our evaluation, besides accuracy curves with proportion confidence interval(L. D. Brown, Cai, and DasGupta, 2001), we also show the distribution of gene markers we selected and the relation with classification target. The subplots include the gene expression values on t-SNE projection, the mean of each class, histogram distribution, violin plot, the correlation coefficient heatmap, etc.

To indicate the efficiency, we also recorded the run time, peak memory usage, and the total number of unique cells we used of ActiveSVM on these datasets.

Pre-processing

For PBMC, Tabula Muris, Multiple-Myeloma datasets, they were pre-processed for a prior publication (S. Chen et al., 2020) via column normalizaition. In each experiment, we removed the columns and rows where all values are zero. Then, gene expression matrices were first columns normalized and log transformed. For a cell i, each gene $x_i^{(j)}$ (gene j in cell i) is first normalized as $\tilde{x}_i^{(j)} = \frac{x_i^{(j)}}{\sum_{i=1}^{M} x_i^{(j)}}$ where M is the number of genes in the transcriptome. And then we did l^2-normalization for each cell, which means scaling each cell vector individually to unit l^2-norm.

For Mega-cell data set, perturb-seq, spatial transcritomics data sets, we removed the columns and rows where all values are zero. Then we performed l^2-normalization along each cell.

Parameter Optimization

For conventional and ActiveSVM, we found the approximately optimal parameter by grid-search (Syarif, Prugel-Bennett, and Wills, 2016) across lists of candidate values for some key parameters in the framework of 3-fold cross validation (Arlot and Celisse, 2010). The optimal parameters were fixed during all iterations. For the comparison methods, we use 3-fold cross validation grid-search to obtain the optimal parameters at each single iteration. We also implemented the algorithms called `min_complexity_cv` and `min_acquisition_cv` that apply grid-search and cross validation for each single SVM trained in each iteration (see Code Availability).

Here we provide the algorithm parameters we used for ActiveSVM in Appendix A Table A.1-A.3. Besides the training set and test set, there are 15 user-defined hyper-parameters in ActiveSVM, five of which are about the feature selection procedure and the other ten are commonly-used parameters for linear SVM classifier. The detailed description about all parameters of ActiveSVM are detailed described in the integrated package page https://pypi.org/project/activeSVC/.

As for comparison methods, correlation coefficient, mutual information, and chi-squared methods don't have specific parameters to set. We implemented them with scikit-learn(Pedregosa et al., 2011) package 'SelectKBest'. For feature importance scores from decision tree and naive SVM, we did grid-search on key parameters based on 3-fold cross validation at each step. The parameters of decision tree are $criterion$ and $min\_samples\_leaf$ and of naive SVM are tol and C.

Time and Memory Usage

To indicate the efficiency, we also recorded the run time, peak memory usage, and the total number of unique cells we used of ActiveSVM on these datasets. We used r5n.24xlarge (Amazon, n.d.[c]), a type of EC2 (Amazon, n.d.[a]) virtual server instance on AWS (Amazon, n.d.[b]), with 96 virtual central processing units (vCPU) and 768 GiB memory on Linux (Torvalds, n.d.) system. For example, we selected 50 genes on the largest dataset, mouse brain 'megacell' dataset, which contains 1306127 cells and 27998 genes, using ActiveSVM and some other popular feature selected methods, including mutual information, feature importance by decision tree, and conventional SVM. The peak memory usage of ActiveSVM is 2111 MB while other methods all consume more than 78600 MB. The run time of the min-complexity method is about 69 minutes and of the min-cell method is about 243 minutes. Each comparison method takes more than 4 days on the same server machine. The run

time and peak memory usage of ActiveSVM on all six datasets are shown in Table 2.1. The ActiveSVM package used for the brain megacell dataset only loads the selected genes and cells into memory at each iteration while other two experiments called the package loading the entire dataset. Both packages are provided in the Code Availability Section.

Data Availability

All data used in the paper has been previously published. Source Data for main figures (except Figure 1) and extended data figures is available with this manuscript.

The PBMC Single-cell RNA-seq data have been deposited in the Short Read Archive under accession number SRP073767 by the authors of (G. X. Zheng et al., 2017). Data are also available at http://support.10xgenomics.com/single-cell/datasets.

The original Tabula Muris dataset is available at https://figshare.com/projects/Tabula_Muris_Transcriptomic_characterization_of_20_organs_and_tissues_from_Mus_musculus_at_single_cell_resolution/27733.

The original multiple myeloma PBMC data, containing 2 healthy donors and 4 multiple myeloma donors, is available at https://figshare.com/articles/dataset/PopAlign_Data/11837097/3.

The 10x genomics Megacell data set is available at http://support.10xgenomics.com/single-cell/datasets.

The perturb-seq data set (Dixit et al., 2016) is available at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2396856

The spatial transcriptomics data (Eng et al., 2019) is available https://github.com/CaiGroup/seqFISH-PLUS.

Code Availability

Our method is integrated as a install-able Python package called activeSVC. The installation instructions and user guidance are shown at https://pypi.org/project/activeSVC. The source codes of activeSVC and some demo examples are publicly available on GitHub at https://github.com/xqchen/activeSVC and Zenodo (xqchen, 2022).

In addition we created Google Colaboratory project for three examples that PBMC demo is at https://colab.research.google.com/drive/16h8hsnJ3ukTWAPnCB581dwj-nN5oopyM?usp=sharing, Tabula Muris demo is at https://colab.research.g

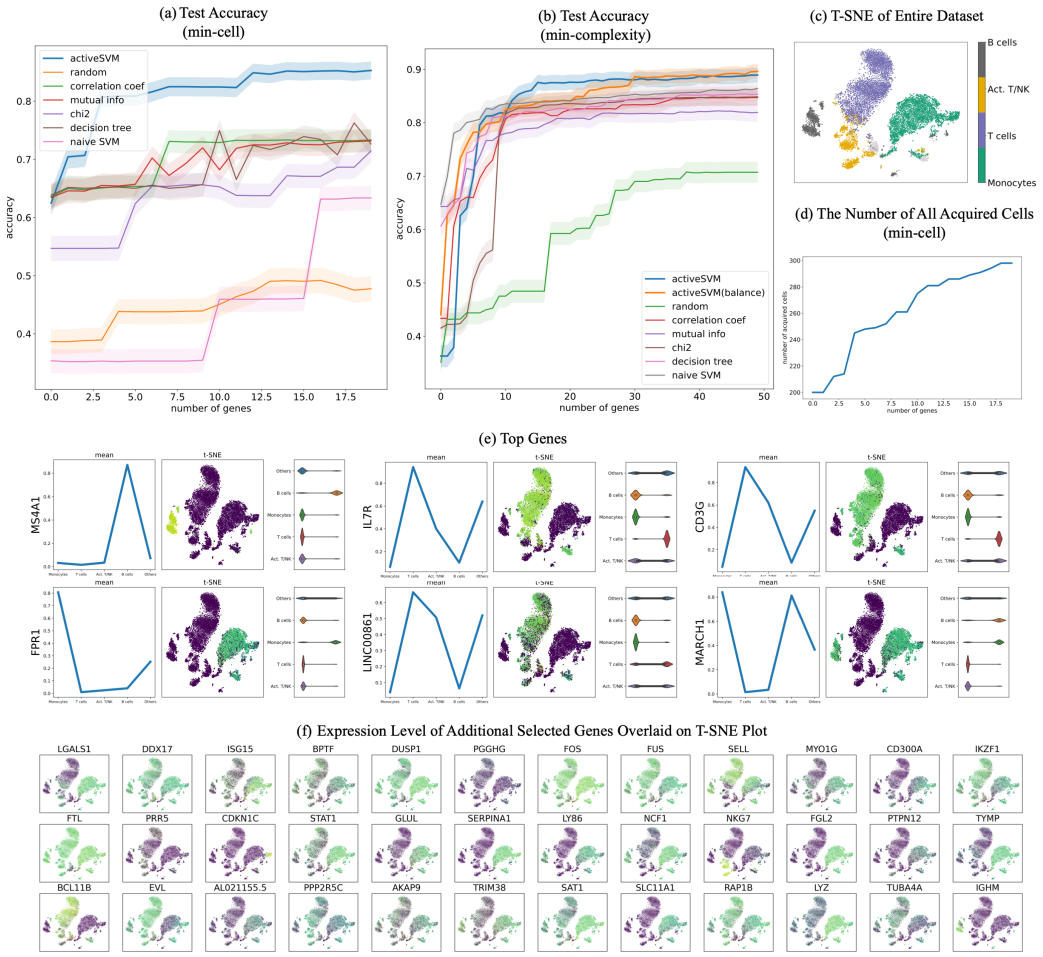
Table 2.1: Run Time and Peak Memory Usage of ActiveSVM.

	matrix size (cells, genes)	min-complexity run time (s/gene)	min-cell run time (s/gene)	memory (MB)	unique cells (min-cell)
mouse megacell	(1306127, 27998)	4142/50	14580/50	2111	712
PBMC	(10194, 6915)	121/50	176/20	1325	298
Tabula Muris	(55656, 8661)	737/150	7701/100	1093	779
MM	(35159, 32527)	127/40	449/40	1616	445
seqFish	(913, 10000)	33/30	728/30	887	428
perturb-seq	(10895, 15976)	3424/50		9493	3827

oogle.com/drive/1SLehIKIQqpjK6BzEKc9m0y3uJ_LBqRzA?usp=sharing, and PBMC cross-validation demo is at https://colab.research.google.com/drive/1fhQ8GD3NyzB3w0vof9WimXK6BLqDNuDC?usp=sharing.

The Python package provides six callable functions: (1) *min_complexity*; (2) *min_acquisition*, the min-cell strategy; (3) *min_complexity_cv*, which use cross validation (Arlot and Celisse, 2010) and grid-search (Syarif, Prugel-Bennett, and Wills, 2016) to train the best SVM estimator at each iteration; (4) *min_acquisition_cv*, the min-cell strategy with cross validation and grid-search; (5) *min_complexity_h5py*, for large h5py data files, it only loads the part of data, the rows and columns of selected genes and cells, instead of loading the entire dataset into memory; (4) *min_acquisition_h5py*, is similar with *min_complexity_h5py* but uses min-cell strategy. All include the algorithm for both randomly and 'balanced' sampling. We implement the SVM classifier with the LinearSVC package from scikit-learn (Pedregosa et al., 2011) library, which is implemented in term of LIBLINEAR (R.-E. Fan et al., 2008). And we use parfor package to parallelize for-loops to accelerate algorithm for large datasets. There are three hyper-parameters to set: balance (boolean), num_features (int), and num_samples (int), to identify the sampling strategy, the number of genes to select, and the number of cells in each iteration.

In the GitHub project, we use the PBMC dataset (G. X. Zheng et al., 2017) and Tabula Muris dataset (Consortium et al., 2018) as examples to show the procedure and its performance of *min_complexity* and *min_acquisition*. We also have the test examples of *min_complexity_cv* and *min_acquisition_cv* on PBMC dataset and the demo projects of *min_complexity_h5py* and *min_acquisition_h5py* on 1.3 millions mouse brain 'megacell' dataset(Genomics, 2017). The notebooks contain downloading dataset, preprocessing, and selecting genes with our method. Additionally, we created Google Colaboratory project for these two examples that PBMC demo is at https://colab.research.google.com/drive/16h8hsnJ3ukTWAPnCB581dwj-nN5oopyM?usp=sharing, Tabula Muris demo is at https://colab.research.google.com/drive/1SLehIKIQqpjK6BzEKc9m0y3uJ_LBqRzA?usp=sharing, and PBMC cross-validation demo is at https://colab.research.google.com/drive/1fhQ8GD3NyzB3w0vof9WimXK6BLqDNuDC?usp=sharing.



2.5 Results

Active feature selection on human PBMC data

To test the performance of ActiveSVM, we used the method to extract classifying gene subsets for human PBMCs. We analyzed a single-cell transcriptional profiling data set for 10194 cells (G. X. Zheng et al., 2017) with 6915 genes. We used Louvain clustering (Blondel et al., 2008) to identify T-cells, activated T/NK cells, B-cells, and Monocytes (Figure 2.2(c)).

The min-cell strategy classified the 5 major cell-types at greater than 85% accuracy with as few as 15 total genes (Figure 2.2(a)) and the test accuracy of min-cell, with both randomly sampling and 'balanced' sampling, also reached much higher accuracy than the comparison methods.

A key benefit of the active learning strategy is that a relatively small fraction of the data set is analyzed, so that the procedure can generate the gene sets while

only analyzing 298 cells (Figure 2.2(d)). At each iteration, a specific number of misclassified cells ($c = 100$) are selected but the total number of cells used does not increase in increments of 100, since some cells are repeatedly misclassified and are thus repeatedly used for each iteration.

In addition to enabling cell-type classification of the data set, the ActiveSVM gene sets provide a low-dimensional space in which to analyze the data. When we reduced our analysis to consider only the top 100 genes selected by the ActiveSVM algorithm, we were able to generate a low-dimensional representations of the cell population (t-SNE) that preserved critical structural features of the data, including the distinct cell-type clusters (Figure 2.2(c)).

The procedure generates gene sets that contain known and novel markers, each plotted individually in a t-SNE grid (Figure 2.2(e)(f)). For instance, MS4A1 and CD79 are well-established B-cell markers, and IL7R and CD3G are well-established T-cell markers. However, we also find genes which are not commonly used as markers, but whose expression is cell-type specific. For instance, we find highly monocyte-specific expression of FPR1, which encodes N-formylpeptide receptor, which was recently discovered to be the receptor for plague effector proteins (Osei-Owusu et al., 2019). We also find T-cell/NK-cell specific expression of a long noncoding RNA, LINC00861, whose function is unknown but has been correlated with better patient outcome in lung adenocarcinoma (Sage et al., 2020). The marker genes are generally highly specific for individual cell types, but some mark multiple cell types (i.e. MARCH1, which marks monocytes and B-cells).

Scaling of ActiveSVM feature selection to million cell dataset

To demonstrate the scaling of the ActiveSVM feature selection method to large single cell mRNA-seq data sets, we applied the method to extract compact gene sets from the 10x genomics the 'megacell' demonstration data set (Genomics, 2017). The megacell dataset was collected by 10x genomics as a scaling demonstration of their droplet scRNA-seq technology. The data set contains full transcriptome mRNA-seq data for 1.3 million cells from the developing mouse brain profiled at embryonic day 18 (E18) (Genomics, 2017). The data set is one of the largest single cell mRNA-seq data sets currently available. The size of the data set has been a challenge for data analysis, and a previous analysis paper was published that developed sub-sampling methods that extract marker genes and cell-types by extracting sub-sets of of the data set containing $\sim 100,000$ cells (Bhaduri et al., 2018).

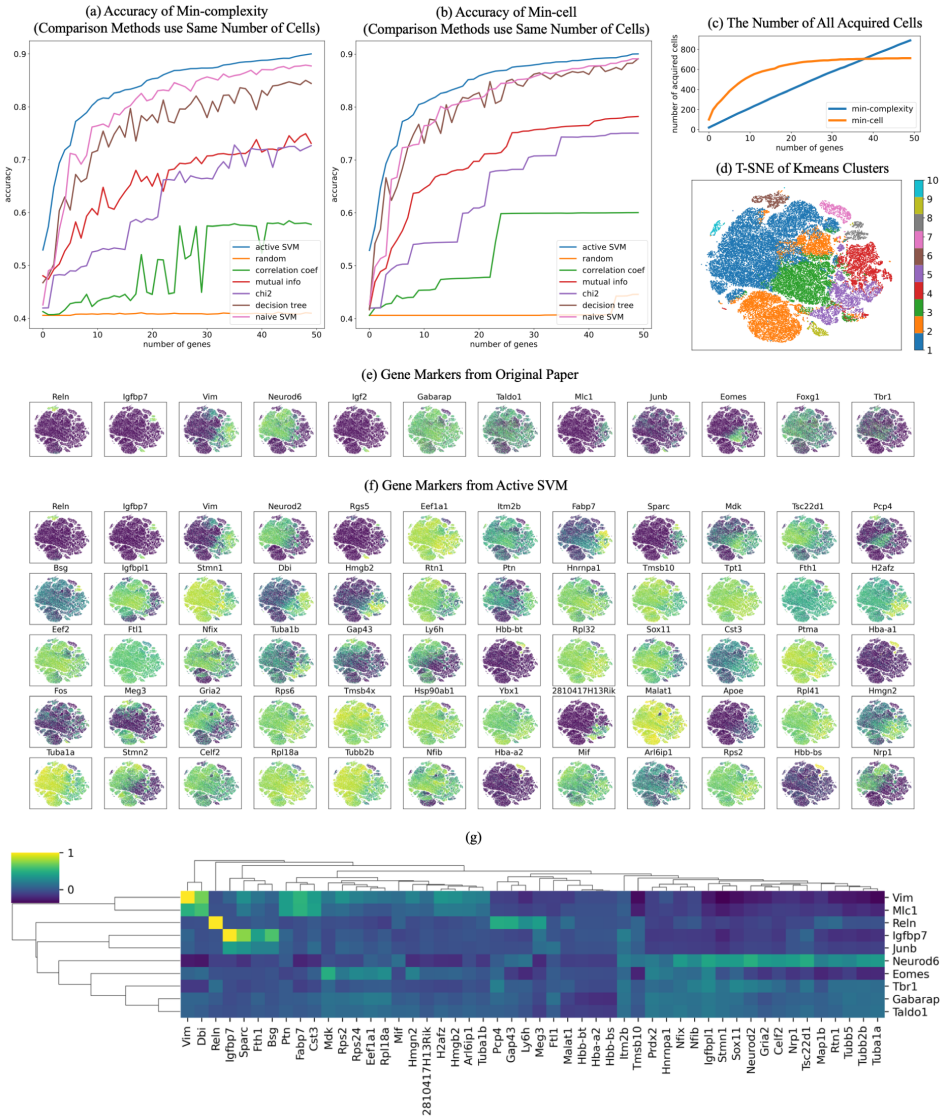


Figure 2.2: **Scaling of ActiveSVM feature selection to 1.3 million cell mouse brain data set** (a) The test accuracy of min-complexity strategy that selects 50 genes using 20 cells each iteration; (b) The test accuracy of min-cell strategy that selects 50 genes using 100 cells each iteration; (c) The total number of unique cells used vs gene set size with both the min-complexity and the min-cell strategy; (d) The t-SNE plots of the entire filtered dataset with 10 classes by k-means clustering; (e) Expression level of the gene markers from previously published analysis overlaid on t-SNE plot; (f) Expression level of the gene markers selected by ActiveSVM overlaid on t-SNE plot, where the first row are the genes that have similar distribution with gene markers from previously analysis and other genes are new markers correlated with the classification target. (g) Correlation matrix of literature markers (y-axis) from (Bhaduri et al., 2018) versus ActiveSVM selected genes (x-axis).

We applied our ActiveSVM method to extract minimal genes sets for classifying the 10 classes of cells that were extracted through k-means (Likas, Vlassis, and Verbeek, 2003) clustering in the internal analysis of the data (Figure 2.3(a)(b)). The min-complexity algorithm used 20 cells at each iteration and the min-cell algorithm selected 100 cells each loop. The min-cell algorithm acquired fewer unique cells, as cells are selected repeatedly (Figure 2.3(c)). On this dataset, both algorithms use 'balanced' sampling for both min-complexity and min-cell strategies. As the dataset is too large to produce t-SNE, we randomly sampled $30,000$ cells and find the tSNE projection, which is shown with the input cell clusters in Figure 2.3(d).

While the size of the data set has presented challenges for conventional sampling methods, the ActiveSVM algorithm must only acquire from memory a small number of genes or cells at each round of analysis, and therefore, the method avoids computing across the entire 1.3 million cells and $\sim 30,000$ gene data set. We found that it was possible to run ActiveSVM on a conventional lap-top. For decreasing compute time, we analyzed the megacell data set on an AWS instance r5n.24xlarge. On this instance, ActiveSVM ran in 69 minutes for the min-complexity strategy and 243 minutes for the min cell strategy. As a comparison, naive SVM required greater than four days of computation to run on all 1.3 million cells on the same AWS instance (Table 2.1).

To provide a bench-marking for ActiveSVM, we instead compared the accuracy of ActiveSVM to a data set where we allow ActiveSVM to run on the data set;we extract the number of analyzed cells, and then provide this same number of cells to the other methods shown in figure 3(a)(b). Applying the other methods to sub-sampled data, allowed us to extract the classification accuracy as a bench-marking for ActiveSVM.

In addition to performing the classification task, the ActiveSVM procedure discovers gene sets that achieve $\sim$ 90% classification accuracy with only 50 genes. The procedure discovered a series of cluster specific marker genes that extend prior analysis. For example, the analysis in (Bhaduri et al., 2018) identified marker genes through sub-sampling and prior biological literature. A set of genes identified previously is shown in Figure 2.3(e). The ActiveSVM analysis discovered several of the same markers as the previous work (Reln, Vim, Igfbp7) (Figure 2.3(f)).

Further, ActiveSVM extended previous analysis by identifying additional markers that correlate with the previously analysis as well as marker genes of additional cell states. The development of radial glial cells, in particular, has been of intense recent interest because radial glial cells are the stem cells of the neocortex in mouse

and human (Pollen et al., 2015). Careful molecular analysis has defined markers of radial glial cells including Vim. ActiveSVM identified a group of genes whose expression correlates with Vim across the E18 mouse brain. Our analysis identified an additional set of genes expressed in the same cell population as Vim including, Dbi (Diazepam Binding Inhibitor, Acyl-CoA Binding Protein), Hmgb2, and Ptn. A correlation matrix (Figure 2.3(g)) showing the correlation of ActiveSVM identified genes (x-axis) with literature markers (y-axis) discussed in (Bhaduri et al., 2018) reveals the existence of Vim correlated genes. The Vim genes were of interest because they include additional transcription factors Hmgb2 (Pollen et al., 2015) and also a core group of genes, Ptn and Fabp7 (also Brain Lipid Binding Protein), two components of a radial glia signaling network (Anthony et al., 2005; M. G. Andrews, L. Subramanian, and Kriegstein, 2020; Pollen et al., 2015) that has been identified as a core regulatory module supporting the proliferation and stem cell state in the radial glial cell population.

The neural progenitor transcription factor Neurod6 marked a separate cell population that we identified to contain genes including Neurod2 (a transcription factor) and Sox11 (a transcription factor) as well as glial transcription factors Nfib and Nfix and the receptor Gria2 (Glutamate Ionotropic Receptor AMPA Type Subunit 2). The marker genes observed in Neurod6 expressing cells were anti-correlated with the Vim correlated markers suggesting that ActiveSVM identified two distinct regulatory modules. Structurally, the tubulin proteins Tuba1b and Tuba1a were expressed in Vim and Neurod6 populations respectively. In addition to genes correlated or anti-correlated with existing markers, ActiveSVM identified markers of additional cell populations including Meg3, a long non-coding RNA expressed in cluster 2.

Broadly, the analysis of the 'megacell' mouse brain data set demonstrates that ActiveSVM scales to analyze a large data set with > 1 million cells. The analysis of such large data sets has been challenging with conventional approaches that attempt to store the entire set in memory for analysis. Previous analysis of the 10x megacell dataset found that sub-samples with greater than $100,000$ cells would yield an out of memory error on a server node with 64 cores, a 2.6 GHz processor, and 512 GB of RAM (Bhaduri et al., 2018).

ActiveSVM iterates through analysis of cells and genes while focusing computational resources on poorly classified cells, and so ActiveSVM does not load the entire dataset into memory but can read cells and genes from disk as needed. Further, through iterative analysis, ActiveSVM identifies known marker and regulatory

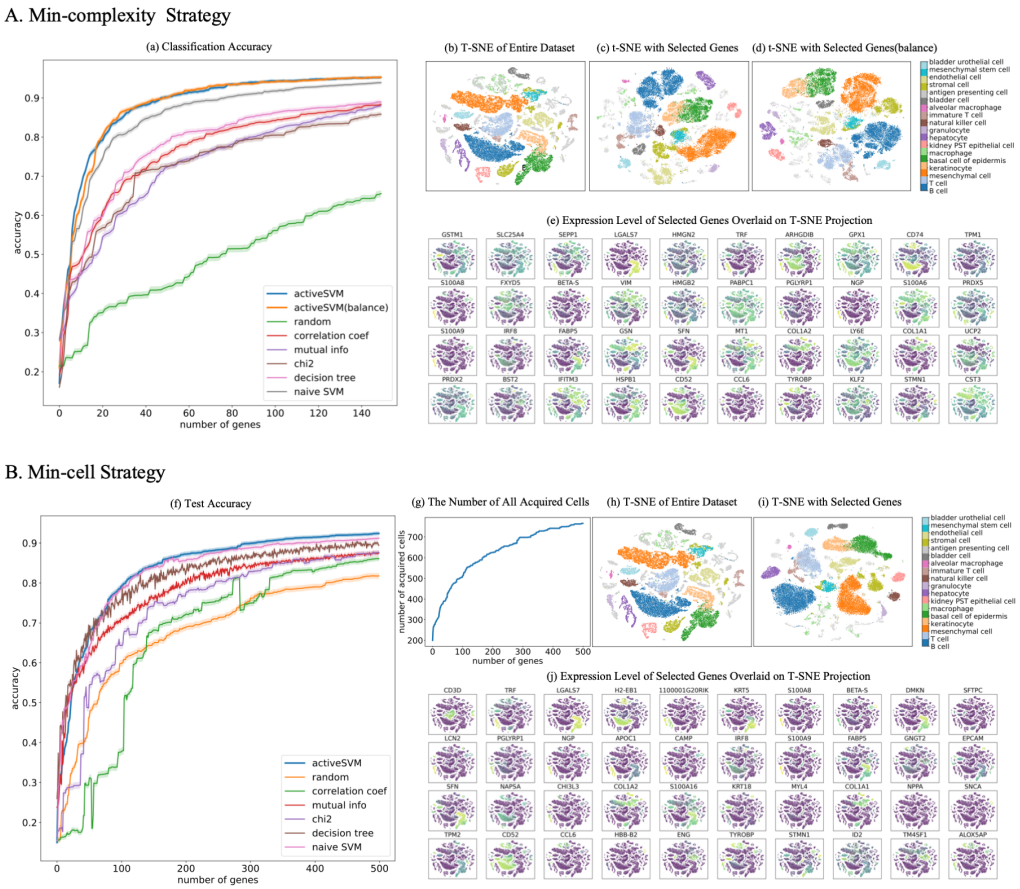


Figure 2.3: **Minimal gene sets for cell-type classification in the Tabula Muris mouse tissue survey** (A) Classification results of 150 genes selected using the min-complexity strategy with 20 cells each iteration. (B) 500 genes selected using the min-cell strategy with 200 cells per iteration. Results for standard and balanced strategy shown with comparison methods and confidence intervals. The subplots contain: classification accuracy vs gene set size using the min-complexity strategy (a) and min-cell strategy (f); the t-SNE plots of the entire filtered dataset (b)(h); the t-SNE plots of the gene set selected using min-complexity strategy with randomly sampling (c) and 'balanced' sampling (d), and gene set selected using the min-cell strategy (i); the expression level overlaid on t-SNE projection for genes selected by min-complexity (e) and by min-cell (j); and the total number of unique cells used vs gene set size with the min-cell strategy (g).

genes, genes that correlate with known markers as well as marker genes of additional cell populations that could provide a starting point for future experimental investigations.

Identifying gene sets for cell-type classification in the Tabula Muris tissue survey
In addition to analyzing a data set with a large number of total cells, we sought to benchmark performance of ActiveSVM feature selection on a data set with a large number of distinct cell types. We applied ActiveSVM to the Tabula Muris mouse tissue survey, a droplet-based scRNA-sequencing data-set, that contains 55,656 single cells across 58 annotated cell types, and 12 major tissues (Consortium et al., 2018). For each cell, 8,661 genes are measured. In our analysis, we used the supplied cell-type labels, agnostic of tissue type. Thus, cells labeled 'macrophage' from the spleen are considered to belong to the same class as cells labeled 'macrophage' from the mammary gland.

Even with a large number of cell types, ActiveSVM can construct gene sets that achieve high accuracy (> 90%), compared to other methods (Figure 2.4(a)(f)). To construct a gene set of size 500, ActiveSVM feature selection used fewer than 800 unique cells (Figure 2.4(g)) or an average of 14 cells per cell type. We were able to recreate the clustering patterns from the original data (Figure 2.4(b)(h)) when analyzing the cells within the low dimensional t-SNE space spanned by the selected 150 genes (Figure 2.4(c)(d)) or 500 genes (Figure 2.4(i)).

Our approach allowed us to construct a set of marker genes able to identify mouse cell types across disparate tissues. Even when analyzing a large number of cell types, we were able to identify highly cell-type specific genes, such as CD3D, a well-established T-cell marker, or TRF (transferrin), which is selectively secreted by hepatocytes (Guan et al., 2020), or LGALS7 (galectin-7), which is specific for basal and differentiated cells of stratified epithelium (Magnaldo, Fowlis, and Darmon, 1998). However, given the functional overlap between different cell types, the genes within our set include many that mark multiple cell types. For instance, H2-EB1 (Stables et al., 2011), a protein important in antigen presentation, is expressed in B-cells and Macrophages, both of which are professional antigen presenting cells (APCs). Our analysis also identified cell type-specific expression for a number of poorly studied genes, such as granulocyte- and hepatocyte- specific expression of 1100001G20RIK (also known as Wdnm-like adipokine), which has previously only been associated with adipocytes (Wu and Smas, 2008).

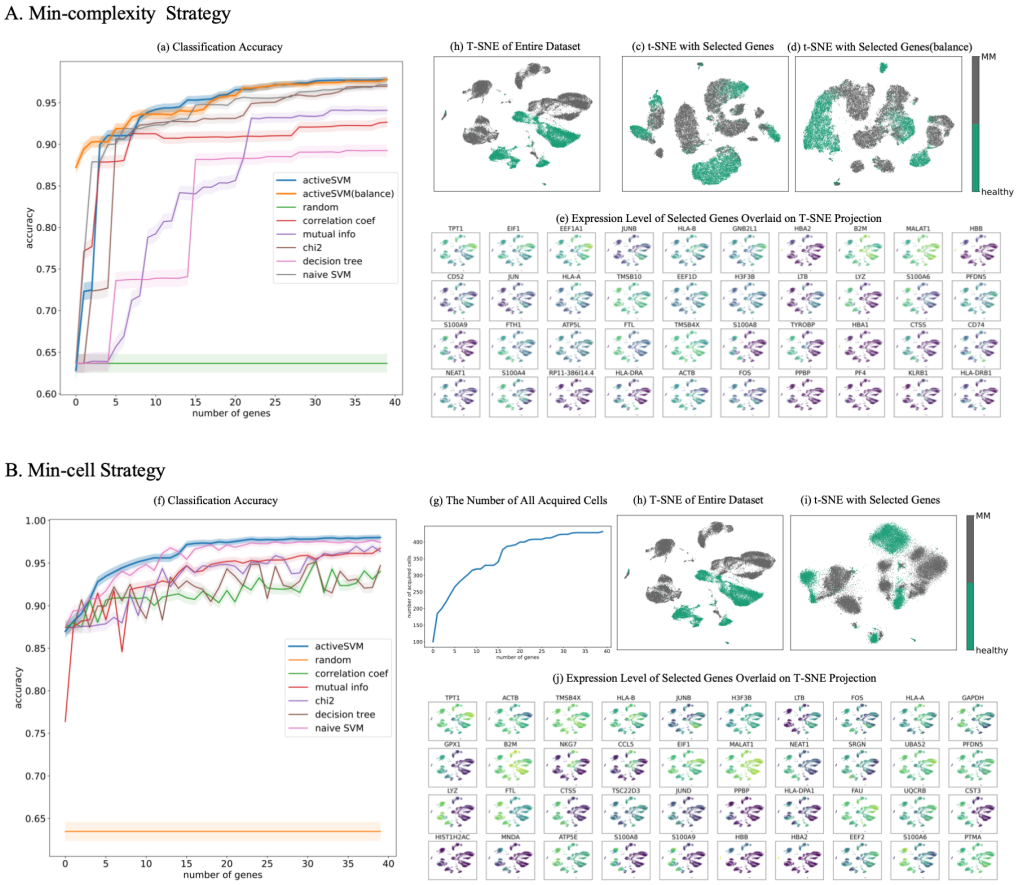


Figure 2.4: **Gene set selection for healthy vs disease classification in multiple myeloma dataset.** (A) classification results of 40 genes selected by min-complexity strategy using 20 cells each iteration. (B) 40 genes selected using Min-cell strategy with 100 cells per iteration. Results for standard and balanced strategy shown with comparison methods and confidence intervals. As in Figure 2.4, each sub-figure, sub-panels show the number of acquired cells per iteration, tSNE visualizations of using the complete data set, visualizations using only the ActiveSVM extracted data set, and marker genes identified by ActiveSVM.

Extraction of gene sets for classification of disease state in peripheral blood cells from multiple myeloma patient samples

To analyze ActiveSVM as a tool for the discovery of disease-specific markers, we used single-cell data from peripheral blood immune cells collected from two healthy donors and four patients who have been diagnosed with multiple myeloma (MM)(S. Chen et al., 2020). MM is an incurable cancer of plasma cells, known as myeloma cells, that over-proliferate in the bone marrow. Although myeloma cells are typically the target of analysis because they are the causative agent of disease, peripherally circulating immune cells also contain signatures of disease, including a depleted B-cell population (Rawstron et al., 1998; Magalhães et al., 2013), an increased myeloid-derived suppressor cell count (Malek et al., 2016), and T-cell immunosenescence (Suen et al., 2016; Magalhães et al., 2013).

We sought to further define transcriptional markers that distinguish healthy peripheral immune cells from the cells of MM patients. We performed feature selection using heterogeneous populations of cells labeled only by disease state. The data set contains 35159 with 32527 genes (Table 2.1).

We compared the classification accuracy for ActiveSVM vs the other methods (Figure 2.5(a)(f)), and found that ActiveSVM achieved high accuracy in a limited number of steps and consistently outperformed the other methods. We tested ActiveSVM with two different cell sampling strategies, randomly sampling, and 'balanced' sampling, in which equal numbers of cells from each cell type are sampled to correct for artifacts due to different cell-type proportions between samples. We noted that although the balanced approach gave higher classification accuracy at early iterations, these differences are no longer apparent after selecting 20 genes (Figure 2.5(a)).

Non-overlapping cell-type clusters were identified for healthy and MM cells in the original dataset in t-SNE projections (Figure 2.5(b)(h)). The non-overlapping clusters are replicated in t-SNEs constructed from 40 genes selected using both the min-complexity strategy (Figure 2.5(c)(d)) and the min-cell strategy (Figure 2.5(i)).

Analysis of the function of the genes identified by ActiveSVM revealed most regulate house-keeping functions, suggesting that global shifts in translation and motility are disrupted in multiple myeloma patients. Translation-associated markers include Eukaryotic Translation Initiation Factor 1 (EIF1), Eukaryotic Translation Elongation Factor 1 Alpha 1 (EEF1A1), and prefoldin subunit 5 (PFDN5). Motility associated genes include ACTB, putative anti-adhesion molecule CD52, and actin-sequestering protein TMSB4X.

We also found both known and novel markers of MM within the peripheral blood immune cells. Our analysis identified TPT1, previously associated with MM (Ge et al., 2011), and RACK1 (also known as GNB2L1), a scaffolding protein that coordinates critical functions including cell motility, survival and death, which is broadly upregulated in peripheral immune cells from MM patients. Although this gene has been previously associated with myeloma cells (Xiao et al., 2018), its regulation had not been reported in peripherally circulating immune cells. Our ability to discover MM-specific genes within peripheral immune cells suggests a broader use for discovering disease-specific genes across many different types of pathologies.

Interestingly, the procedure also identifies multiple members of the S100 Calcium Binding Protein Family (S100A8, S100A9 and S100A6, and S10084) (C. Xia et al., 2018; M. Liu et al., 2021; Dobreva et al., 2020) as members of the genes sets that separate MM vs healthy samples. The S100 protein family defines a module of genes that are associated with the induction of stress response pathways. The expression of S100 genes is prognostic for a number of diseases. Specifically, a recent study found that S100A4 expression correlates with poor patient survival in mulitple myeloma and that S100A8, and S100A9 are markers that correlate with poor response of multiple myeloma patients to treatment with proteasome inhibitors and the and histone deacetylase inhibitor panobinostat (M. Liu et al., 2021). The result demonstrates that ActiveSVM can automatically define groups of genes that have clinical association with disease progression and treatment outcome. The minimal gene sets generated by ActiveSVM could provide useful targeted sequencing panels for a variety of clinical tasks.

ActiveSVM identifies genes impacted by Cas9 based genetic perturbation

The previous analyses above have demonstrated that ActiveSVM identifies minimal gene sets for cell-state identification across a range of single-cell mRNA-seq data sets. We next demonstrate that ActiveSVM provides a more general analysis tool with potential applications to a range of single-cell genomics analysis tasks. To demonstrate generalization of ActiveSVM based gene set selection across single-cell genomics tasks, we applied the method to identify marker genes in two additional applications: perturb-seq and spatial transcriptomics.

Perturb-seq is an experimental method for performing Cas9-based genetic screens with single-cell mRNA-seq read-outs. In perturb-seq, cells are induced with libraries

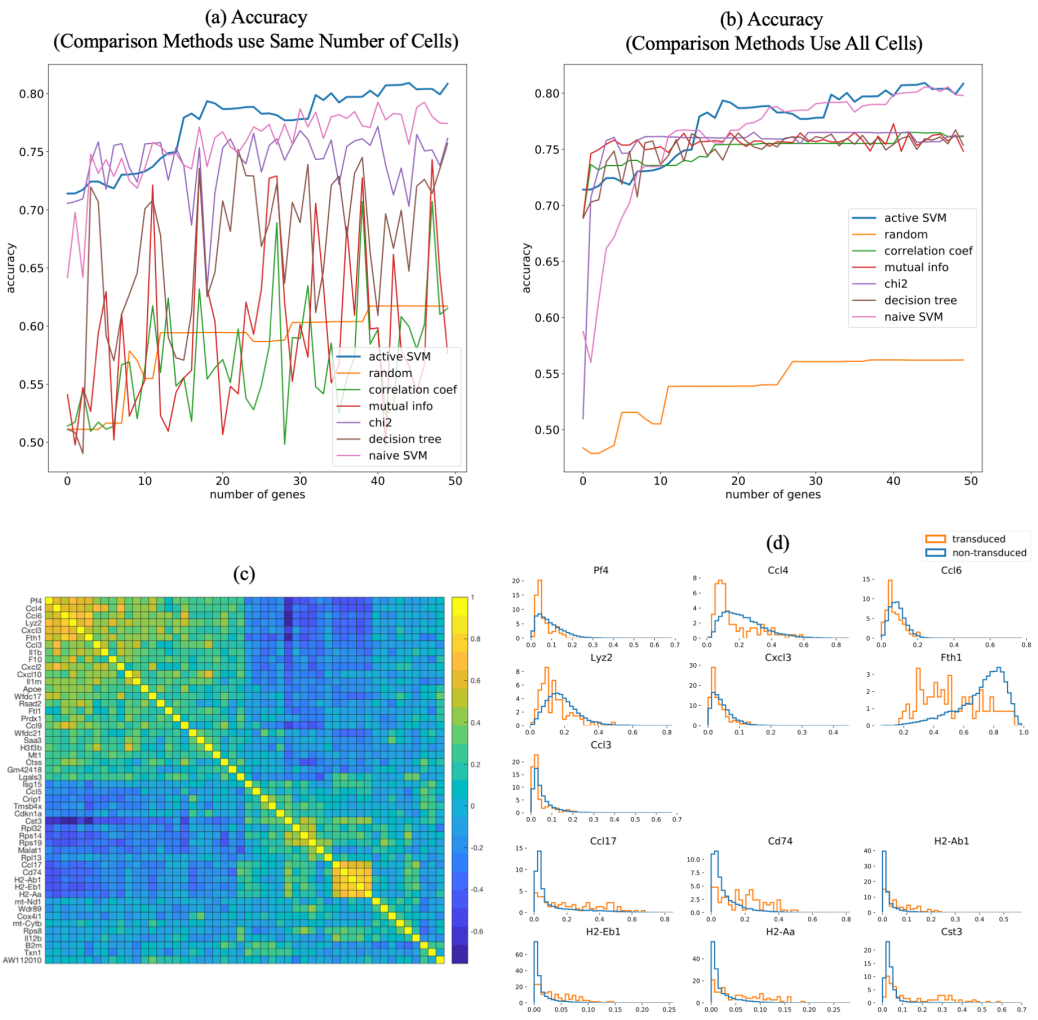


Figure 2.5: **Application of ActiveSVM to identify genes expression changes following Cebp knock-down with perturb-seq** The results of classification on perturb-seq data (Dixit et al., 2016) where cells are labeled and classified as Cebp sgRNA transduced or not-transduced with a guide RNA. (a-b) accuracy of entire dataset with min-complexity strategy, where comparison methods use the same number of cells as ActiveSVM in (a) and use the entire dataset in (b). (c) correlation matrix showing pair-wise correlation coefficients for genes in Cebp perturbed cells. Correlation matrix identifies two gene modules. (d) Distributions of gene expression in Cebp sgRNA transduced (orange) or not transduced (blue) cells. Selected genes from modules in (c) shown and organized so that genes whose expression increases with Cebp perturbation are on top and repressed genes are on the bottom of the figure.

of guide RNA's that target the Cas9 protein to cut and silence specific genes (Dixit et al., 2016; Replogle et al., 2020). Perturb-seq is performed in a pooled fashion so that a pooled set of sgRNA molecules is delivered to a cell population. Individual cells stochastically take-up specific guide RNAs, and the guide RNAs target Cas9 cuts and silences genes in the genome. Following the perturbaation experiment, single-cell mRNA-seq is applied to read both the transcriptome of each cell and the identify of the delivered sgRNA through sequencing. The advantage of the perturb-seq method is that many knock-out experiments can be performed simultaneously. However, a challenge is that noise impacts the measurement of guide RNA identify, and, further, the cutting of the genome by the Cas9 molecule is not complete. Due to measurement and experimental noise, identifying the impact of genetic perturbation on a cell population can be challenging, and various methods have been developed to boost signal (Replogle et al., 2020). We applied ActiveSVM to identify a minimal gene set as well as down-stream effects of gene knock-down in perturb-seq data.

We specifically applied ActiveSVM to analyze public data collected from mouse dendritic cells with transcription factor knock-downs (Dixit et al., 2016). The experiment analyzed cells in which transcription factors has been knocked-down using perturb-seq in mouse dendritic cells stimulated for 3 hours with LPS, a signal that mimics bacterial infection.

To apply ActiveSVM to the data, we focused our analysis on knock-down of Cebp an pioneer transcription factor. We pre-processed the data to identify cells induced with sgRNA against Cebp and non-induced cells, and used transduced and non-transduced as our cell-labels. We applied ActiveSVM to select a minimal gene set that could classify transduced versus non-transduced cells. ActiveSVM identified minimal gene sets (50 genes) that achieved 80% classification accuracy on the Cebp sgRNA cell label. As we applied the class-balanced model to obtain the classification accuracy and there are only about 20 transduced cells in test set, we show the accuracy on entire dataset instead of test set. On this noisy dataset, ActiveSVM worked better than comparison methods with the condition that ActiveSVM only used a small subset of data while comparison methods performed on the entire dataset (Figure 2.6(b)).

We found that the discovered gene set could be decomposed into two modules of correlated genes (Figure 2.6c). Figure 2.6(c) shows a clustered correlation matrix for the 50 identified genes. Gene expression distributions for cells in transduced vs non-transduced cells demonstrated that the modules represented two groups of

genes. One group (including Pf4, Ccl4, Ccl6, Lyz2) was repressed by Cebp knock-down, and the second gene group was activated by Cebp knock-down including (Ccl17, Cd74, H2-Ab1) (Figure 2.6d).

In both cases, the identified gene sets contained known targets of Cebp, the perturbed transcription factor. For example, Fth1 (ferritin, heavy polypeptide 1), Cst3, Tmsb4x, Lgals3, Ccl4, and Cd74 are all previously identified as direct binding targets of Cebp as determined by Chip-seq (Rouillard et al., 2016). Since Cebp knock-down leads to both up-regulation and down-regulation of genes, the results suggest that the factor can play both activating and repressive roles consistent with prior literature (Pei and Shih, 1990).

Our analysis of the perturb-seq data set, therefore, demonstrates that ActiveSVM can be applied as a useful tool for the identification of genes modulated by perturb-seq experiments. ActiveSVM can return minimal genes sets that contain functional information. Moreover, perturb-seq has been a main application of gene targeting approaches (Replogle et al., 2020). Therefore, ActiveSVM could provide a method for identifying minimal gene sets that can be applied to increase the scale of perturb-seq data collection.

ActiveSVM defines region specific markers in spatial transcriptomics data

Finally, to further demonstrate the generality of the ActiveSVM approach, we applied the procedure to identify minimal gene sets for classification of cells by spatial location in spatial transcriptomics data. Spatial transcriptomics is an emerging method for measuring mRNA expression within single cells while retaining spatial information and cellular proximity within a tissue. As an example, in SeqFish+, an imaging based spatial transcriptomics method, cells are imaged in their tissue environment, and mRNA transcripts are counted using single-molecule imaging of mRNA spots (Eng et al., 2019). In all spatial transcriptomics applications, a common goal is the identification of genes that mark specific spatial locations within a tissue sample. Additionally, spatial imaging methods are commonly limited by imaging time. While Seqfish+ can profile $10,000$ mRNA molecules per cell, the identification of reduced gene sets would reduce imaging time and throughput.

We applied ActiveSVM to identify genes associated with specific spatial locations in the mouse brain. We used a seqFISH+ data set in which the authors profile $10,000$ mRNA molecules in 7 fields of view (FOV) in the mouse brain (Eng et al., 2019). Fields of view correspond to spatially distinct regions of the mouse cortex as well

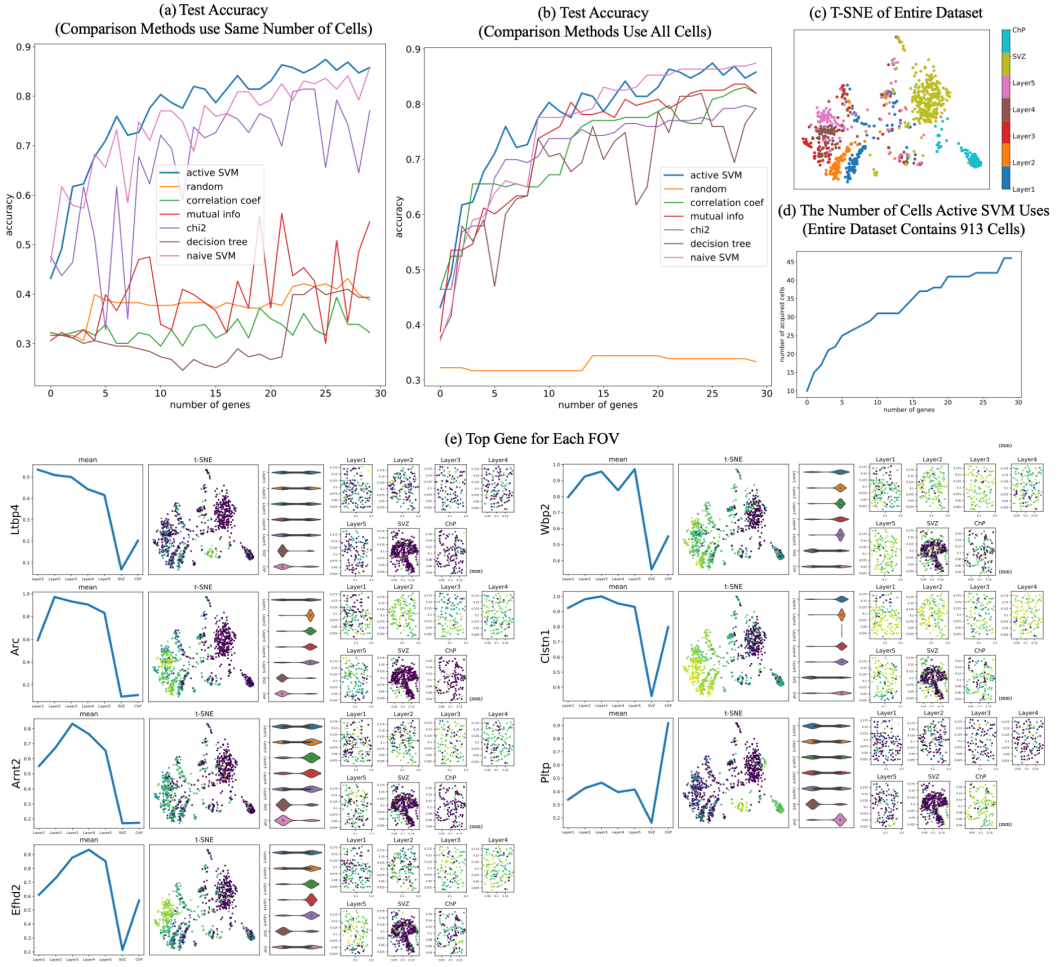


Figure 2.6: **Application of ActiveSVM to identify region specific marker genes in the mouse brain with spatial transcriptomic data** The results of classification where cells are labeled according to fields of view (FOV) in (Eng et al., 2019). (a-b) test accuracy with min-complexity strategy, where comparison methods use the same number of cells as ActiveSVM in (a) and use the entire dataset in (b). Fields of view 1-5 correspond to 5 regions of the mouse cortex, additional fields of view are labeled SVZ (sub-ventricular zone) and ChP (chordid plexus). (c) tSNE of cell transcriptomes for all cells (d) number of cells used per iteration (e) Sample of identified genes where each sub-panel shows mean expression across FOV/brain regions for selected gene, a tSNE plot colored by expression of selected gene, a violin plot of single cell gene expression values for selected gene in FOV/brain region, and spatial plots of each field of view where dots represents cells in 2D imaging slice, cells are colored by intensity of selected gene and units are in millimeters.

as the sub-ventricular zone and chordid plexus. We used the spatial location labels provided by Eng et al., 2019 to identify seven different brain locations (Fields of view 1-5 corresponding to Cortex Layers 2/3 through Layer 6; FOV 6 is sub-ventricular zone, and FOV 7 is chordid plexus). Applying the spatial location labels as class labels, we applied ActiveSVM to identify genes that could allow classification of single-cells by their location in one of the seven classes and to define marker genes that correspond to specific spatial locations.

We identified gene sets of < 30 genes that enabled location classification with greater than 85% accuracy with min-complexity strategy (Figure 2.7(a)). ActiveSVM used only 10 cell at each iteration but worked better than comparison methods who performed on the entire dataset (Figure 2.7(b)).

In the spatial application, the result means that the ~ 30 genes are sufficient to classify single-cells as belonging to one of the 7 spatial classes. In Figure 2.7, we show the mean expression of identified genes across cortical fields of view corresponding to a sweep through cortical layers 2/3 through 6 as well as SVZ and CP. Our analysis identifies markers Prex1 that are specific to the upper cortical layers of the brain. Efhd2, a calcium binding protein linked to Alzheimer's disease and dementia, was similarly expressed in lower cortical layers (Vega, 2016; Borger et al., 2014). Finally, Pltp, a Phospholipid transfer protein, was localized to the chordid plexus. In Figure 2.7(e), we show the spatial distribution of these genes including their mean expression across regions, violin plots documenting expression distribution, and renderings of the single-cells within the field of view and the relative expression of each gene.

The spatial analysis demonstrates that a broad range of different experimental variables can be applied as labels. In each case ActiveSVM discovers genes that allow classification of cells according to labels and identifies interesting genes. Regional gene marker identification is a major task in seqFish data analysis and ActiveSVM is able to identify genes enriched in different brain regions automatically. Such spatial information could provide interesting new insights into disease processes mediated by genes like Efhd2.

2.6 Discussion

In this chapter, we introduce ActiveSVM as a feature selection procedure for discovering minimal gene sets in large single-cell mRNA-seq datasets. ActiveSVM extracts minimal gene sets through an iterative cell-state classification strategy. At

each round the algorithm applies the current gene set to identify cells that classify poorly. Through analysis of misclassified cells, the algorithm identifies maximally informative genes to incorporate into the target gene set. The iterative,active strategy reduces memory and computational costs by focusing resources on a highly informative subset of cells within a larger data set. By focusing computational resources on misclassified cells, the method can run on large data sets with more than one million cells. We demonstrate that ActiveSVM is able to identify compact gene sets with tens to hundreds of genes that still enable highly accurate cell-type classification. We demonstrate that the method can be applied to a variety of different types of data set and single-cell analysis tasks including perturb-seq data analysis and spatial transcriptomic marker gene analysis.

Conceptually, we refer to our strategy 'active' because it actively selects data examples (here cells) at each iteration for detailed analysis . Our algorithm specifically selects cells that within the margin of the SVM classifier, and uses these poorly classified cells to search for maximally informative genes (features). In traditional active learning strategies, an algorithm is typically called active when it can directly query an oracle for data examples that meet a criteria (Settles, 2009; Settles, 2011). In the tradition of active learning, our ActiveSVM procedure queries the SVM classifier for cells that have been misclassified, and then expends computational resources to analyze all genes within that limited subset of cells to discover informative genes. Thus, while our algorithm cannot query the biological system directly for cells that meet a specific criteria, the algorithm queries the data set itself for informative examples, and therefore we refer to it as 'active'. Our current work focuses on a single classification method, the support vector machine, as the computational engine. Active learning methods can be applied more broadly to additional classification strategies like neural network based classification as well as to additional types of analysis like data clustering and gene regulatory network inference.

Our method also has some conceptual similarity to boosting methods (Schapire, 2003; Schapire, 1999). Boosting algorithms (e.g AdaBoost) train a series of 'weak' learners for a classification tasks, and then combine these weak classifiers to generate a strong classifier. In boosting a single weak learner may initially obtain moderate performance on a task. The performance of weak learners is improved through iterative training of additional learners and focusing their training on difficult data examples, for example, misclassified examples. The boosting algorithm constructs a final, strong classifier by combining the results of the ensemble of

weak classifiers through a weighted majority vote. Our method is distinct from conventional boosting, because we search for a minimal set of features in our data that allows a single SVM classifier to achieve high-accuracy classification. However, ActiveSVM feature selection shares conceptual ideas with boosting in that both methods focus analysis on challenging examples and combine information to achieve strong classification from initially weak classifiers.

ActiveSVM provides an iterative strategy for extracting a compact set of highly informative genes from large single cell data sets. Biologically, recent work highlights the presence of low-dimensional structure within the transcriptome (Heimberg et al., 2016). Low-dimensional structure emerges in gene expression data because cells modulate their physiological state through gene expression programs or modules that contain large groups of genes. Since genes within transcriptional modules have highly correlated expression, measurements performed on a small number of highly informative signature genes can be sufficient to infer the state of a cell (Cleary et al., 2017). Low-dimensional structure can be exploited to decrease measurement and analysis costs since a small fraction of the transcriptome must be measured to infer cellular state. We developed ActiveSVM as a scalable strategy for extracting high information content genes within a sharply defined task, cell-state classification.

In ActiveSVM we apply an active learning strategy to reduce the computational and memory requirements for analyzing single-cell data sets by focusing computational resources on 'difficult to classify' cells. In the future, active learning strategies could be applied directly at the point of measurement. In genomics measurement resources often limit the scale of data acquisition. In future work we aim to develop strategies that can improve the on-line acquisition of single-cell data. Active strategies could be implemented at the point of measurement by only sequencing or imaging the content of cells that meet a criteria. Even more broadly, it might be possible to induce a biological system to generate highly informative examples through designed experimental perturbation (Jiang, Sivak, and Thomson, 2019).

Chapter 3

ACTIVECELLINFERENCE: DECREASE ACQUISITION COST IN SPATIAL GENOMICS

Spatial transcriptomics assigns cell types and states to their locations in histological sections at single-cell and subcellular resolutions by measuring the expression of a predefined set of genes. The high temporal cost of measurements is a major limiting factor in introducing spatial genomics into clinical practice. We present Active Cell Inference, an end-to-end pipeline that uses ordered gene sets to enable fast and low cost spatial genomics measurements in scientific and clinical settings. The developed algorithm identifies well-classified cells that require no further probing, reducing the number of cells each gene marks, which in turn reduces measurement costs by 10 to 100 fold.

Our Active Cell Inference procedure starts with a set of all cells and an ordered gene set from developed ActiveSVM (Chapter II). The algorithm iteratively classifies cells with cell-types or states labels using probed genes from the ordered gene set by training a SVM model with probability calibration and identifies uncertainty of cells. Cells that are certainly well-classified and require no additional gene markers are removed from the set. Subsequently, the next genes to be probed are marked in cells that are misclassified or classified unreliably, to improve classification accuracy and certainty.

To refine the sequence of predictions made by our model, I implemented a temporal scaling calibration scheme based on Platt scaling (Platt et al., 1999) integrated with vector scaling (Guo et al., 2017). Our experimental assessments demonstrate that this calibration method significantly improves the probability calibration throughout the iterative process of the algorithm.

Furthermore, we applied this algorithm to the expansive Human Cell Atlas dataset (Regev et al., 2017) using the advanced computational tool, CellxGene-Census (Biology et al., 2023), which includes data on over 60 million cells. This integration has enabled us to establish precisely targeted gene sets for various human tissues, greatly enhancing the efficiency of the Active Cell Inference process. This strategy has consistently delivered highly reliable and accurate results across different human tissues.

3.1 Introduction

Spatial genomics is an advanced technique that measures RNA expression at the single-cell level, with the added dimension of maintaining spatial context through imaging. This methodology is pivotal for understanding the complex spatial arrangements and interactions of cells within a specific tissue or environment. The challenge currently facing the field is the extensive amount of time required to image a large number of genes across numerous cells. This bottleneck significantly impedes the broader application of spatial genomics, particularly in clinical diagnostics where speed and scalability are crucial. (Moffitt et al., 2016) (Lubeck et al., 2014).

A common objective of spatial genomics is to identify and classify different cell types within a sample. For instance, in oncology, a critical application is the identification of immune cells, such as T cells, within a tumor. This information is essential for evaluating the immune response to the tumor and for tailoring immunotherapy treatments. Immunotherapy relies heavily on understanding whether a tumor is densely infiltrated by immune cells or not, as this can influence the effectiveness of treatment strategies. Therefore, the ability to accurately and efficiently determine the presence and abundance of specific cell types within a tumor using spatial genomics is of paramount importance for advancing personalized medicine and improving patient outcomes (Ståhl et al., 2016) (Rodriques et al., 2019).

For example, in tumor analysis, spatial genomics can illuminate the distribution and types of immune cells across different regions of a tumor. This granularity helps in discerning patterns of immune evasion by cancer cells and identifying potential targets for immunotherapy. By pinpointing where immune cells are located and how densely they populate various tumor areas, researchers can better assess the immunological landscape of the cancer. This assessment is critical in deciding whether a tumor is likely to respond to immunotherapies that rely on boosting the body's immune response to cancer cells.

Active acquisition methods represent a promising solution to this challenge. These methods involve optimizing imaging protocols to ensure that only the necessary amount of imaging is performed to achieve specific clinical outcomes. By prioritizing efficiency, these techniques can significantly reduce the time and resources required for spatial genomics studies, making them more feasible for clinical applications. (K. H. Chen et al., 2015).

Enhancing the capabilities of spatial genomics further, active acquisition strategies not only streamline imaging processes but also enable the detailed examination of

cellular microenvironments in health and disease. This is especially relevant in the field of oncology, where understanding the cellular composition of tumors can provide insights into tumor behavior and response to therapies. (J. H. Lee et al., 2014) (Eng et al., 2019).

Moreover, the ability to perform this analysis efficiently—by reducing the imaging load without compromising the quality of data—can significantly accelerate the transition from experimental research to practical, clinical applications. Active acquisition methods, by focusing imaging efforts on gene markers most relevant to the clinical question at hand, ensure that the data collected is both scientifically robust and clinically relevant.

This targeted approach not only conserves valuable laboratory resources but also opens the door for real-time spatial genomic diagnostics. In the clinical setting, such diagnostics could be transformative, enabling the rapid stratification of patients to appropriate therapies based on the spatial molecular signatures of their tumors. For instance, patients whose tumors show high levels of T cell infiltration might be excellent candidates for aggressive immunotherapeutic regimens, whereas those with poor infiltration might require alternative strategies.

In essence, the integration of active acquisition methods with spatial genomics represents a significant leap towards personalized medicine, where the precise cellular and molecular characteristics of a patient's disease can guide therapy choices. This convergence promises to refine our understanding of disease pathology at an unprecedented scale, potentially ushering in a new era of targeted and effective treatment strategies across various medical disciplines.

We have devised an active cell inference and data acquisition protocol utilizing ordered gene sets and classifiers to efficiently identify cell types within a sample, significantly reducing the number of genes imaged per cell. Our approach has successfully decreased imaging time by a factor of 10 to 100, allowing for cell classification with an average of just 10 genes per cell. Our findings indicate that imaging costs differ among tissue types; for instance, kidney tissues require ten times more imaging rounds than other tissues for accurate cell type classification. Additionally, we observed that the cost of acquisition can be tailored based on the precision needed for cell identification.

I'll introduce the probability calibration in the section Probability Calibration. The proposed ActiveCellInference and temporal scaling are explained in detail in the

Methods section. Datasets and experiments are described in the next two section. And the results of experiments are detailed in two subsections: (1) the accuracy relative to the number of genes selected and the average number of genes needed per cell, demonstrating that the algorithm maintains high accuracy even with a reduced number of genes; (2) a comparison between temporal scaling and Platt scaling, highlighting differences in scalability and efficiency.

3.2 Probability Calibration

In classification tasks, accurately predicting class labels and assessing the probability associated with these labels is essential. This probability serves as a measure of confidence in the predictions. However, some models either provide inaccurate probability estimates or lack the capability to offer such predictions at all. To resolve this, calibration modules are employed to either improve the model's probability estimates or to add functionality for predicting probabilities.

A well-calibrated probabilistic classifier produces results that can be directly interpreted as confidence levels. For instance, a perfectly calibrated classifier that predicts with an 80% confidence should have about 80% of such predictions turn out to be correct.

Let's consider a classifier $\hat{y}_i = f(X_i)$ derived from dataset $\{X_i, y_i\}_{i=0}^{N}$, where $y_i \in \{1, ..., K\}$ are ground-truth labels. The goal is for the probability calibration $\hat{P}$ to mirror the true probability accurately. Perfect calibration can be described by the equation:

$$\mathbb{P}(\hat{y} = y|\hat{P} = p) = p, \forall p \in [0, 1]. \tag{3.1}$$

Achieving this level of precision in calibration is typically unattainable in practical settings due to the limited sample sizes in datasets. This limitation necessitates the use of empirical methods to approximate this ideal. In the upcoming sections, I will introduce proxy metrics that evaluate how closely empirical probability estimates match this ideal calibration, and discuss several well-established techniques for probability calibration.

Evaluating Calibration

I will discuss two commonly used calibration errors for multiclass classification settings and one visualization technique to assess the effectiveness of probability calibration.

These three evaluation methods rely on comparing the model's predicted probability with the actual sample accuracy. This comparison is conducted by dividing the predictions into M interval bins, where each bin represents a probability range of size $\frac{1}{M}$. For each bin, the average predicted probability of the samples is calculated, and the actual sample accuracy is determined by the fraction of correct predictions in that bin, specifically, the proportion of samples correctly identified as belonging to the positive class. Let B_m represent the set of samples whose predicted confidence falls within the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$. The expected sample accuracy for B_m is then calculated by:

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i)$$

while the average predicted probability in each bin is:

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{P}_i.$$

The good calibration in empirical sense means $acc(B_m) = conf(B_m)$. The calibration gap of a bin is the absolute difference between $acc(B_m)$ and $conf(B_m)$.

$$gap(B_m) = |acc(B_m) - conf(B_m)|.$$

Reliability Diagrams (DeGroot and Fienberg, 1983) (Wilks, 1990), also referred to as calibration curves, illustrate the accuracy of calibration in probabilistic predictions from a binary classifier. The y-axis indicates the expected sample accuracy, and the x-axis represents the predicted probability of the model. For a model that is perfectly calibrated, the curve should ideally follow the identity function, where the predicted probabilities align precisely with the actual outcomes. Deviations from this line, typically marked in red on the diagrams, highlight a calibration gap, signaling that the model is miscalibrated.

Expected Calibration Error (ECE) (Naeini, Cooper, and Hauskrecht, 2015) is the weighted mean of the calibration gaps of all bins. More precisely,

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} gap(B_m).$$

Maximum Calibration Error (MCE) (Naeini, Cooper, and Hauskrecht, 2015) is especially important in high-stakes settings where precise confidence measures

are crucial. This metric aims to minimize the greatest discrepancy between the predicted confidence and actual accuracy, enhancing reliability in critical situations.

$$MCE = \max_{m=\{1,...M\}} gap(B_m).$$

Calibration Methods

Probability calibration models for classification refine a classifier to provide true likelihood estimates or probabilities of classification, rather than just class labels. These models are generally categorized into parametric and nonparametric types. In this section, I'll discuss three well-known methods: Platt scaling (Platt et al., 1999), which is a parametric approach, and Isotonic Regression (Zadrozny and Elkan, 2002) and Histogram Binning (Zadrozny and Elkan, 2001), which are nonparametric methods. In addition to these, there are several other effective calibration techniques applicable in various scenarios, such as Bayesian Binning (Naeini, Cooper, and Hauskrecht, 2015), Beta Calibration (Kull, Silva Filho, and Flach, 2017), and methods that utilize Gaussian processes (Küppers, Schneider, and Haselhoff, 2022) (Song et al., 2019).

Platt Scaling Platt scaling is a well-known method for calibrating probabilities in classification tasks. It uses the outputs from the original classifier, represented as f_i, as inputs to a logistic regression model. This model is defined by parameters α and β, and it is optimized by minimizing the Negative Log Likelihood (NLL) between the predicted outputs and the actual labels y_i. The resulting calibrated probability is computed using the sigmoid function in the logistic regression model, expressed as:

$$\hat{p}_i = \sigma(\alpha f_i + \beta) = \frac{1}{1 + exp(\alpha f_i + \beta)}.$$

Here, y_i represents the true label of the i-th sample, and f_i is the score provided by the original, uncalibrated classifier for this sample. The parameters α and β are real numbers optimized through maximum likelihood estimation.

In scenarios involving multi-class classification, it's common to calibrate each class separately using a one-vs-rest strategy. Studies often indicate that replacing the sigmoid function with the softmax function can improve results for multi-class classification. However, in datasets where class distribution is imbalanced, both sigmoid and softmax approaches may introduce bias in the probability estimates, often favoring the majority class disproportionately due to the skewed distribution.

To address the challenges of imbalanced multi-class classification, vector scaling and matrix scaling are suggested as solutions by (Guo et al., 2017). These approaches are similar to multinomial logistic regression, where a weight matrix $\mathbf{W}$ and a bias vector $\mathbf{b}$ are optimized. Vector scaling simplifies the calibration process by zeroing out all elements not directly involved in calibration, effectively reducing the number of parameters to be trained. This technique can be seen as an extension of Platt scaling applied within a one-vs-rest framework, but with a focus on jointly optimizing the weights. The probability in this method is calculated as follows:

$$\hat{p}_i = \sigma(\mathbf{W} f_i + \mathbf{b}).$$

Isotonic Regression Isotonic regression is a non-parametric technique that constructs a step-wise non-decreasing function. It is designed to minimize the difference between the predicted probabilities and the actual labels, as illustrated by the equation below:

$$\begin{aligned} &\min \ \sum_{i=1}^{N} (y_i - \hat{f}_i) \\ &\text{subject to} \ \ \hat{f}_i \geq \hat{f}_j \ \text{whenever} \ \ f_i \geq f_j \end{aligned} \tag{3.2}$$

Histogram Binning Histogram binning organizes the output space of a model into M distinct bins, using either equal-width or equal-frequency techniques to ensure a uniform distribution of data across these bins. Each bin receives a calibrated probability designed to minimize the mean squared error (MSE) within that specific segment. This probability is set to reflect the proportion of positive outcomes observed in the bin. Consequently, when a new test instance lands in one of these bins, it inherits the calibrated probability assigned to that bin, which helps to standardize the accuracy of predictions throughout the model's output spectrum.

3.3 Method

The Active Cell Inference method sequentially uses genes from a pre-defined, ordered gene set as probes and trains SVM models with probability calibration to assess classification confidence of cells. Cells identified with high certainty are systematically removed from further analysis, as they do not require additional probing. The dynamic filtration of cells based on the confidence of their classification ensures that only cells requiring more gene markers remain in focus, thereby reduces the

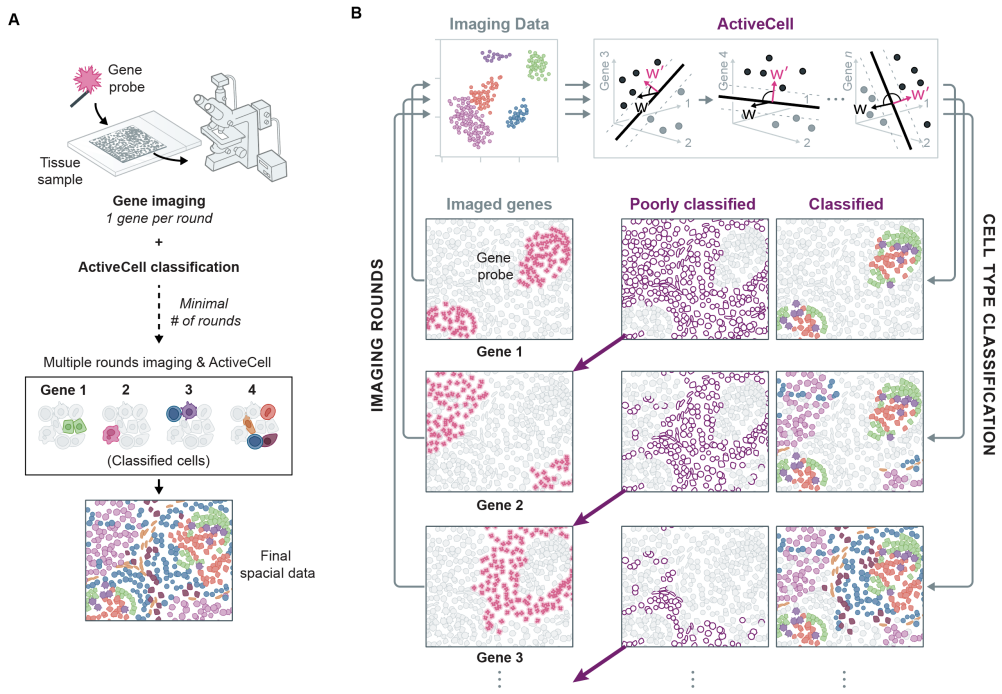


Figure 3.1: **Active Cell Inference:** Starting with N cells and an ordered gene set of M genes, this method uses SVM models and probability calibration to sequentially classify cells. Cells meeting a high certainty threshold are removed, minimizing further analysis. The process continues until all cells meet the classification criteria or a minimal number remains, guiding targeted gene probing in spatial genomics.

measurement cost of spatial genomics. Moreover, we have derived ready-to-use ordered gene sets for all human organs from the Human Cell Atlas dataset, enabling direct application of the Active Cell Inference method on human tissues.

In detail, the iterative process starts with a complete cell set with N cells and an ordered gene set with M genes. At k-th step, a SVM model is trained with the cell set and the first k genes. Probability calibration models are instrumental in evaluating and quantifying the level of certainty associated with the classification of each cell. The cells with classification certainty higher than preset threshold are removed from the cell set, requiring no further gene probing. The process continues until all cells are well-classified, or the number of remaining cells is negligible, or it meets a user-defined criterion for a specific task. The integrated algorithm is shown in Algorithm 2. This procedure guides spatial data measurement, allowing biologists to target cells with specific gene probes as instructed by our algorithm. The framework are shown in Figure 3.1.

In this framework, various probability calibration methods are applicable. Recognizing the sequence of classification outputs, I have developed a modification of Platt scaling called temporal scaling, which enhances calibration by utilizing a broader range of historical data rather than solely relying on the outputs from the current iteration. Experiments demonstrate that this approach surpasses traditional Platt scaling in terms of performance within this iterative framework.

The algorithm applies to single-cell gene expression data with annotations or clustering labels. Cell labels can be derived from unsupservised analysis, experimental meta-data, or biological knowledge of cell-type marker genes. Notably, our method can also integrate user-supplied labels for specific spatial genomics tasks.

The ordered gene set we used here is derived from ActiveSVM, a iterative gene selection algorithm based on SVM. ActiveSVM generates ordered gene sets from single-cell data through an iterative cell-type classification task where only misclassified cells are examined at each round. To refine this process for our Active Cell Inference approach, we have enhanced ActiveSVM with classification probability calibration. Rather than limiting our attention to merely misclassified cells, we now prioritize cells that exhibit the highest uncertainty in their classification. This strategic adjustment ensures the resultant gene set is optimally aligned with the requirements of Active Cell Inference, thereby improving its efficiency and accuracy in identifying pertinent genes for spatial genomics analysis.

Algorithm 2: Active Cell Inference

Input: $X \in \mathbb{R}^{N \times M}$, $X_i^j \in \mathbb{R}$: the sample of i-th cell and j-th gene, $y \in \{1, ..., K\}^N$; test set $X' \in \mathbb{R}^{N' \times M}$, $X'^j_i \in \mathbb{R}$: the sample of i-th cell and j-th gene; threshold $c \in (0, 1)$; $T \in \mathbb{N}$ is the size of sliding window; $K \in \mathbb{N}$ is the number of classes; ordered gene set J; set of all cells I; stop criterion S

$j = 1$

repeat

- Train SVM on $X_i^j, \forall i \in 1, ..., N$
- Get optimal $W^j \in \mathbb{R}^{K \times M}$, $b^j \in \mathbb{R}^K$
- $f_i^j = X_i W^T + b, \forall i \in 1, ..., N$
- Train temporal scaling model on $[f_i^{max(0,j-T)}, ..., f_i^j], \forall i \in 1, ..., N$
- Get optimal W'^j, v^j, b'^j
- $j = j + 1$

until $j > |J|$

$j = 1$

repeat

- $f'^j_i = X'_i W^T + b, \forall i \in 1, ..., N'$
- $F'^j_i = [f_i^{max(0,j-T)}, ..., f_i^j]$
- $\hat{P}_i = \sigma(\mathbf{W}' F'^j_i \mathbf{v} + \mathbf{b}'), \forall i \in 1, ..., N'$
- $\hat{p}_i = max(\hat{P}_i^k), k \in 1, ...K, , \forall i \in 1, ..., N'$
- $I' = \{i | \hat{p}_i >= c\}$
- $I = I \setminus I'$
- $j = j + 1$

until $j > |J|$ *or* $I = \emptyset$ *or* S

Temporal Scaling

Various probability calibration methods can be effectively integrated into our framework. Notably, both Platt scaling and Isotonic Regression have shown impressive performance in our experiments using the Human Cell Atlas (HCA) dataset.

Given that the classification outputs in our iterative framework are produced in a sequential manner, I have developed a method known as temporal scaling to enhance probability calibration. This method extends Platt scaling by considering a broader history of outputs, rather than just the current iteration. It incorporates a sliding window mechanism with a fixed length to capture the sequence of outputs over time. Let's assume there are N samples (cells), K classes, and the sliding window

has a length of T. The output of the classifier is denoted as $f \in \mathbb{R}^{N\times K\times T}$, and the classification output matrix for each sample is $f_i \in \mathbb{R}^{K\times T}$. The calibration is then applied using the following formula:

$$\hat{p}_i = \sigma(\mathbf{W} f_i \mathbf{v} + \mathbf{b}).$$

Here, σ represents the softmax function, suitable for multi-class classification tasks. $\mathbf{W} \in \mathbb{R}^{K\times K}$ s a diagonal weight matrix used in vector scaling, which adjusts the calibration for different classes to address biases in imbalanced datasets. It has non-zero diagonal elements to minimize the complexity of the model. The vector $\mathbf{v} \in \mathbb{R}^T$ is a temporal kernel that processes the sequence data, with the same kernel being shared across all classes to further reduce parameter count. Additionally, $\mathbf{b} \in \mathbb{R}^K$ is the bias vector. Altogether, the model has $2K + T$ parameters, making it a compact yet effective solution for probability calibration in simple models.

Temporal scaling has proven to be more effective than traditional Platt scaling in this context, demonstrating enhanced performance in our iterative setup. Detailed results of this approach are presented in the Results section.

Preset Threshold

In our iterative framework, a preset threshold for classification certainty determines which cells require further analysis with additional gene probes. Cells that meet or exceed this certainty threshold—calculated based on the calibrated probability—require no further probing and are classified using all genes selected up to that point. Cells below this threshold proceed to the next round to query more gene probes.

The calibrated probability reflects the proportion of samples that are correctly classified among all evaluated samples. For instance, if the calibrated probability is 80%, theoretically, 80% of the samples within this grouping should be correctly classified. However, due to imperfections in calibration, there might be a discrepancy between this theoretical accuracy and the actual observed accuracy. Interestingly, samples that meet an 80% certainty threshold often possess a higher true certainty, leading to experimental classification accuracies that exceed the theoretical threshold.

In practice, setting the certainty threshold slightly below the desired accuracy level is advisable. This approach has been validated across experiments involving over 60 different tissues, where the empirical accuracy typically aligns closely with the

preset threshold. Additionally, setting a lower threshold can reduce the average number of genes needed per cell, optimizing the efficiency of the classification process.

Stop Criterion

The iterative process allows for flexible termination, meaning it can be halted at any stage. This process typically continues until all cells are accurately classified, the number of unclassified cells becomes minimal, or it satisfies a specific user-defined criterion relevant to the task at hand. Such criteria might include limitations on the total number of genes utilized, a convergence point in the number of cells classified, overall classification accuracy, or other relevant metrics.

3.4 Datasets

The experiments were conducted on every single human organ tissue sample available in the Human Cell Atlas (HCA) dataset. The ordered gene sets, derived through the ActiveSVM algorithm, were tailored based on the convergence of ActiveSVM accuracy for each specific tissue. I analyzed the classification accuracy of all cells and the average number of genes required per cell, revealing that the ActiveCellInference algorithm consistently achieved high classification accuracy with a minimal number of genes.

Additionally, I tested temporal scaling against traditional Platt scaling to compare their performance concerning the Expected Calibration Error (ECE) and Maximum Calibration Error (MCE). The evaluation metrics demonstrated that temporal scaling significantly enhances probability calibration performance.

I also examined variations among different organs, identifying the number of cell types and the size of sub-datasets as the primary factors influencing classification performance and the average gene requirement per cell.

The experiments leveraged the expansive Human Cell Atlas dataset, employing both the ActiveSVM for feature selection and ActiveCellInference for cell classification across all organ tissues. For efficient data handling, I utilized the Census cloud-based dataset access tool to query sliding sub-datasets within the analysis pipelines.

Human Cell Atlas Dataset

The Human Cell Atlas (HCA) is a pivotal global research initiative aimed at creating a comprehensive map of all cell types in a healthy human body across various

life stages. This atlas is expected to significantly enhance our understanding of human biology and has the potential to drive substantial advancements in health and medicine.

The dataset comprises over 61.5 million cells collected from more than 8.7 thousand donors across 455 projects facilitated by 761 labs worldwide, involving tissues from 61 human organs such as the brain, heart, lungs, and many more. Each cell has been analyzed for 60,664 genes. This extensive and well-annotated dataset allows for detailed exploration and comparison across a vast range of cell types and tissues, as detailed in Tables 3.1 and 3.2.

The HCA not only aims to delineate the complex roles and functions of individual cells and their genetic profiles but also serves as a monumental project on the scale of the Human Genome Project. It is akin to creating a "Google Maps" for human cells, providing not just a genomic blueprint but also a functional landscape of how cells utilize these genetic instructions. This atlas is revolutionizing our understanding of the 37.2 trillion cells in the human body and their implications in health and disease.

Researchers are compiling this comprehensive cell map, focusing initially on key biological systems like the lungs and brain. The data, accessible via the HCA Data Portal, will eventually encompass 10 billion cells from all body tissues, transforming the landscape of global health research.

Census

For these experiments, I accessed human organ sub-datasets from the HCA using the computational tool Census API, which provides efficient, cloud-based access to structured HCA datasets. This tool facilitates rapid interaction with the data, allowing researchers to query and analyze single-cell RNA data seamlessly. By employing a cell-based slicing and querying approach through TileDB-SOMA, researchers can acquire data slices in formats like AnnData, Seurat, or SingleCell-Experiment, greatly enhancing the efficiency of data analysis. For this project, I used the stable Census dataset version dated "2023-05-15".

Census supports the handling of larger-than-memory data slices, which means the algorithm processes only manageable sub-datasets at each iteration rather than the entire dataset, significantly reducing memory load and computational demands.

3.5 Experiment Details

In this subsection, I'll provide a comprehensive overview of the experimental setup, including dataset preprocessing, parameter optimization, and the computational infrastructure used.

Pre-processing

The initial dataset consists of raw read counts matrix. To ensure consistency, I excluded datasets from the Smart-seq2 assay technique due to its differing data format from more recent methods. For each human organ tissue sub-dataset, I removed all columns and rows where values were entirely zero. The gene expression matrices were first column-normalized and then log-transformed. For cell i and gene j, normalization was performed as $\tilde{x}_i^{(j)} = \frac{x_i^{(j)}}{\sum_{i=1}^{M} x_i^{(j)}}$ where M is the total number of genes. Subsequently, each cell vector was scaled to unit l^2-norm through l^2-normalization, enhancing the efficiency of model training and optimization.

For both ActiveSVM and ActiveCellInference, datasets were randomly split into training and test sets in a 4:1 ratio. The training set for ActiveSVM was utilized for selecting gene markers, with their effectiveness validated on the test set. In ActiveCellInference, training involved SVM and calibration model training, ideally performed on independent datasets to avoid biased calibrators. We used 3-fold cross-validation, where two folds trained the SVM and the third trained the calibration model based on SVM output. This cycle was repeated three times, and the final calibration models were averaged from these iterations.

Normalization transformations were applied column-by-column or row-by-row using stored means and standard deviations, facilitated by the Census API, which loads only necessary data subsets into memory for each iteration.

Parameter Optimization

Parameter tuning for ActiveSVM and ActiveCellInference was conducted via a grid search (Syarif, Prugel-Bennett, and Wills, 2016) across lists of candidate values for key parameters within a 5-fold cross-validation framework (Arlot and Celisse, 2010). The optimal parameters, once determined, were consistently used throughout all iterations.

Thresholds for different organs varied based on the accuracy profile of ActiveSVM, generally set at 80%, 90%, or 95%. Organs with lower accuracy from ActiveSVM had reduced thresholds, such as 70% or 65%. Details of the preset thresholds for all

organs are provided in Tables 3.1 and 3.2.

The temporal scaling sliding window size was set to 3, which proved most effective across most tissues. The "liblinear" solver was used to optimize the probability calibration models.

Computational Infrastructure

All experiments were conducted on computational clusters accessed via the Open OnDemand service portal (Hudak et al., 2018), supported by the US National Science Foundation (Alexandria, n.d.). For ActiveSVM, each experiment utilized 4 virtual central processing units (vCPUs) with 512 GiB of memory on a Linux system. For ActiveCellInference, each experiment was allocated 1 vCPU with 256 GiB of memory. This setup provided the necessary computational resources to handle the extensive data processing and analysis requirements of our experiments.

3.6 Ready-to-use ActiveSVM Gene Sets

I have developed ready-to-use gene sets and cell inference pipelines using the Human Cell Atlas datasets. For each human organ, an ordered set of gene markers was established using the ActiveSVM feature selection algorithm. Given that the organ tissue dataset comprises a vast collection of single-cell mRNA-seq datasets, the resulting gene set is both reliable and generalizable, suitable for any new tissue samples from the same human organ. The associated probability calibration models and SVM classifiers are also designed to be generalizable across new spatial genomics tissues.

For each organ, I have documented the classification accuracy for both the training and test sets, alongside the number of genes selected by the algorithm. Comprehensive lists of genes for all organs are available in Appendix C: Full Lists of ActiveSVM Gene Sets for HCA Data. The quantity of gene markers in each set varies among different organs, with the required number of genes correlating positively with the size of the datasets and the diversity of cell types present. Additionally, the final accuracy is influenced by the dataset's noise levels and the reliability of its annotations. These factors similarly affect the outcomes of the ActiveCellInference experiments.

The gene sets for all organs were curated through ActiveSVM. Details regarding the size of the dataset, the number of cell types, and the extent of the genes in the ActiveSVM gene sets can be found in Tables 3.1 and 3.2. Further information about the gene sets are shown in Appendix A and B.

Table 3.1: Size of Dataset, ActiveSVM Gene Set, Genes used in ActiveCellInference, and preset threshold

Organ	dataset size	cell types	gene set size	genes used	threshold
abdomen	32635	9	50	20	0.8
abdominal wall	5154	8	60	29	0.97
adipose tissue	93319	28	120	60	0.6
adrenal gland	437955	43	350	70	0.95
ascitic fluid	108287	10	70	60	0.9
axilla	6484	10	120	14	0.97
bladder organ	32470	26	80	60	0.8
blood	8841851	143	300	250	0.6
bone marrow	303147	109	200	150	0.6
brain	9300186	100	350	250	0.8
breast	1555995	51	100	115	0.8
colon	508127	112	300	300	0.7
digestive system	2710	9	200	28	0.8
embryo	79012	23	80	80	0.9
endocrine gland	397653	109	300	150	0.8
esophagogastric junction	12771	29	150	9	0.9
esophagus	170900	73	300	200	0.8
exocrine gland	37162	28	100	100	0.8
eye	760892	79	200	194	0.9
fallopian tube	164336	25	80	80	0.7
gallbladder	9769	20	100	50	0.9
heart	1559696	71	600	200	0.6
immune system	50982	22	300	300	0.7
intestine	167838	44	120	120	0.8
kidney	712494	112	500	400	0.7
lamina propria	23687	30	80	80	0.8
large intestine	106196	101	200	180	0.6
liver	564748	115	300	300	0.7
lung	2882265	188	400	260	0.6
lymph node	402978	109	300	300	0.6
mucosa	26060	18	120	120	0.8
musculature	133866	75	150	150	0.8
nose	313887	75	350	350	0.7
omentum	222303	42	150	150	0.9
ovary	194433	32	100	100	0.9
pancreas	182746	47	100	100	0.8
paracolic gutter	8012	9	30	30	0.9

Table 3.2: Size of Dataset, ActiveSVM Gene Set, Genes used in ActiveCellInference, and preset threshold. (Continued)

Organ	dataset size	cell types	gene set size	genes used	threshold
parietal peritoneum	10546	9	30	29	0.95
peritoneum	86704	9	60	60	0.9
placenta	94204	42	150	35	0.8
pleura	19695	20	200	17	0.97
pleural fluid	25331	23	100	100	0.8
prostate gland	136295	57	100	100	0.6
reproductive system	389407	42	250	200	0.9
respiratory system	368376	106	300	300	0.8
saliva	14502	13	600	100	0.7
scalp	19408	8	150	100	0.95
skeletal system	12329	24	100	50	0.9
skin of body	177132	100	260	260	0.8
small intestine	829526	153	500	290	0.6
spinal cord	30106	38	150	25	0.8
spleen	359155	105	270	270	0.7
stomach	278277	73	500	300	0.9
testis	13211	24	175	60	0.9
tongue	13629	12	100	100	0.9
trunk	23646	8	150	40	0.95
ureter	2390	11	100	1	0.9
urinary bladder	6266	8	100	20	0.97
uterus	285234	37	120	100	0.8
vasculature	31218	23	70	70	0.8
yolk sac	40544	41	120	120	0.9

3.7 Results: Accuracy and Gene Utilization Per Cell

This section is divided into several subsections, each illustrating how ActiveCellInference accurately selects the appropriate cells using a minimal number of gene markers, achieving high classification accuracy.

ActiveCellInference's accuracy is defined as the cumulative accuracy of all cells classified in previous iterations. Classification occurs when the probability calibration model indicates no further genes are necessary for a cell. For instance, if 10 cells are resolved in the first round with 9 correctly classified and 1 misclassified, and in the second round 20 cells are settled with 15 correctly classified, then the accuracy at the end of the second round would be calculated as $\frac{9+15}{10+20}$.

In the final iteration, I implement an early stop of the algorithm, opting to classify all remaining unclassified cells with the genes utilized up to that point. Consequently, for most organs, there is a slight decline in accuracy in the last round, as many of these cells require more genes than those available when the algorithm is prematurely halted.

The subsections are organized as follows: (1) Results along with the increasing number of genes, including the accuracy and the fraction of all classified cells in previous rounds; (2) Comparison the accuracy of ActiveSVM and ActiveCellInference; (3) The final results after the last iteration, including the accuracy of all classified cells and the average number of genes queried per cell.

Results with Increasing Number of Genes

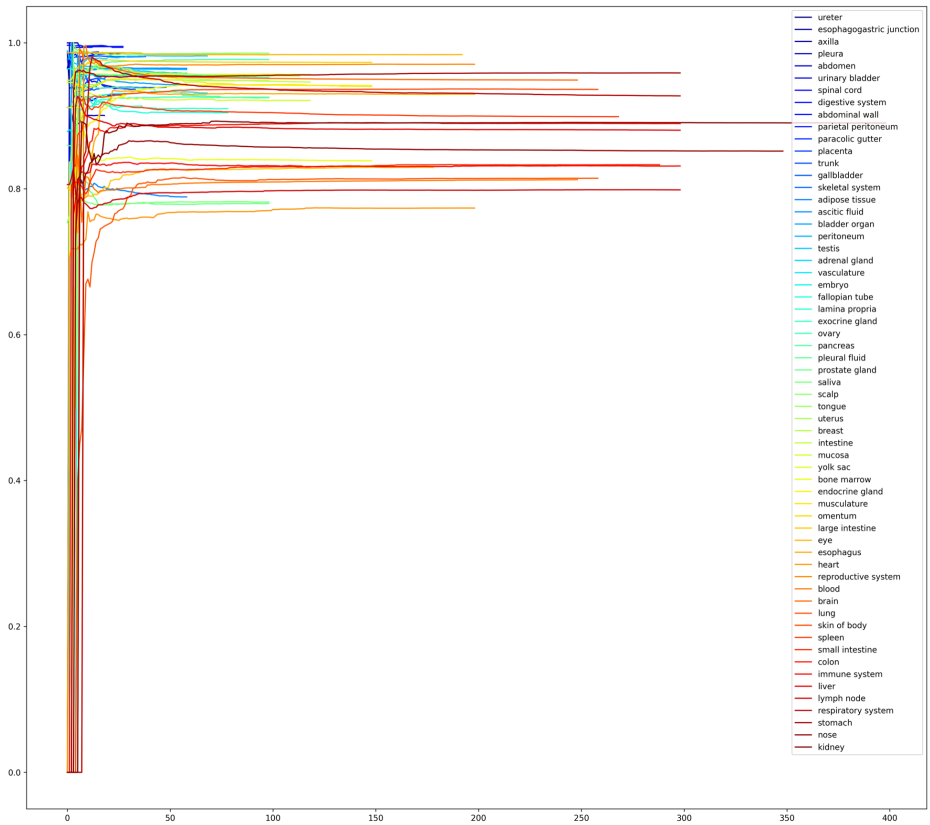
Figure 3.2 illustrates the accuracy of ActiveCellInference as it correlates with the number of genes utilized, while Figure 3.3 displays the proportion of cells classified in all previous iterations relative to the total number of cells in each respective organ.

Initially, the accuracy is approximately equal to the preset threshold and remains largely consistent throughout the successive iterations until the final round. Most organs demonstrate an accuracy exceeding 80%, maintaining this high level of accuracy consistently across the process.

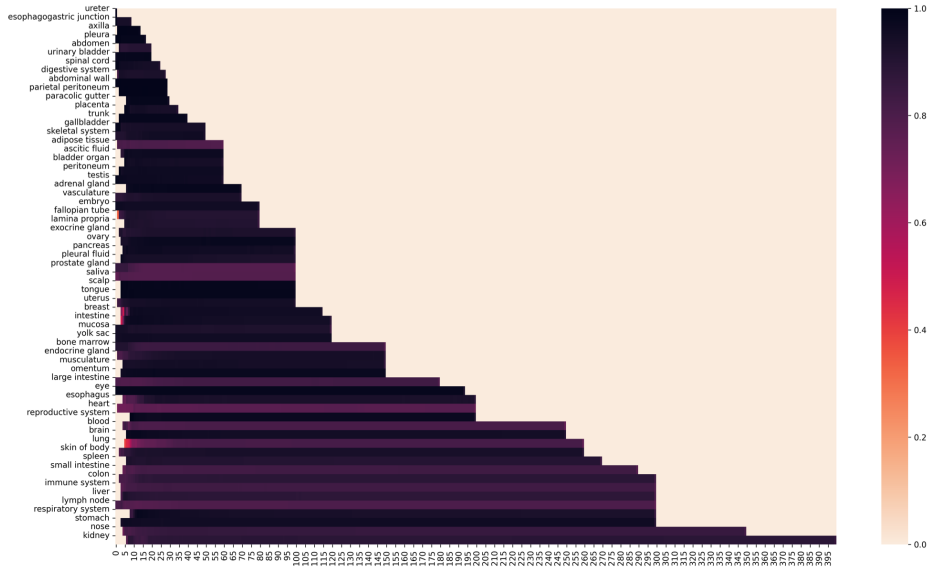
For all organs, ActiveCellInference successfully classifies over 70% of the cells, as shown in the sub-figures depicting the fraction of classified cells. Some organs achieve well over 90% classification accuracy with fewer than 50 genes used. Throughout the iterations, the fraction of classified cells gradually increases, though the rate of increase slows over time. This indicates that most cells are classified at an early stage with a relatively small number of genes.

Comparison of Accuracy Between ActiveSVM and ActiveCellInference

Figure 3.4 illustrates the comparative accuracy of ActiveSVM and ActiveCellInference, with the number of genes used plotted on the x-axis. ActiveSVM's accuracy demonstrates a gradual increase, whereas ActiveCellInference achieves high accuracy right from the start and maintains this level consistently until the final round. This performance suggests that ActiveCellInference effectively identifies and selects the appropriate cells in each iteration, ensuring that cells requiring additional genes are accurately targeted and that the gene querying process concludes precisely when the cells are well-classified.

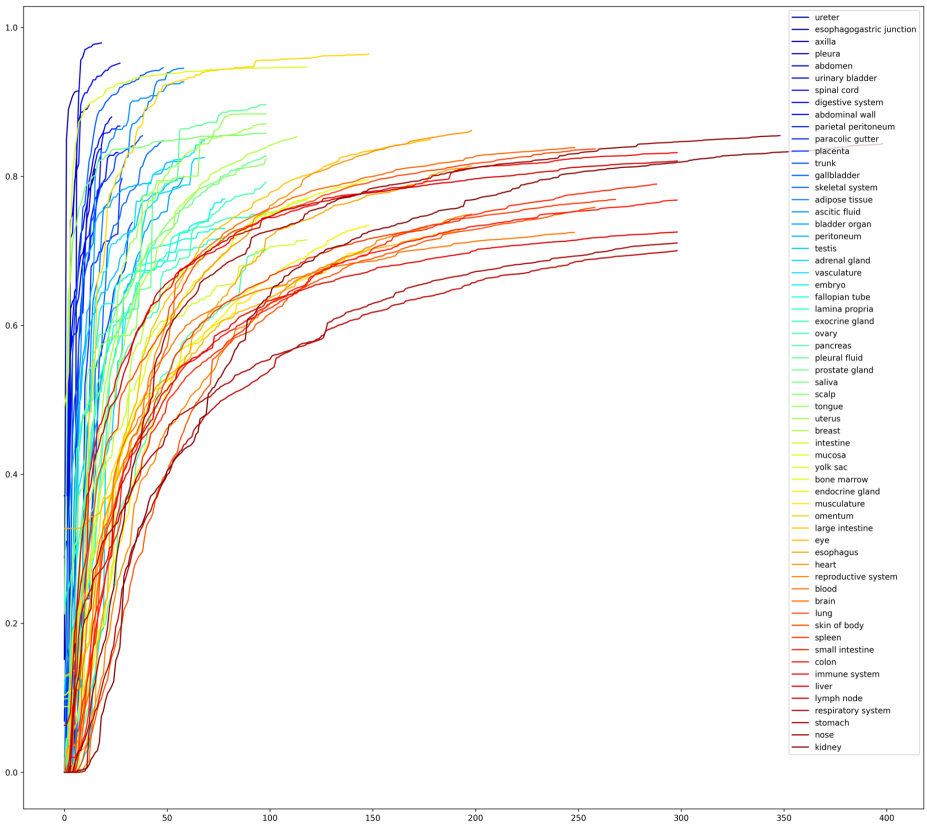


(a)

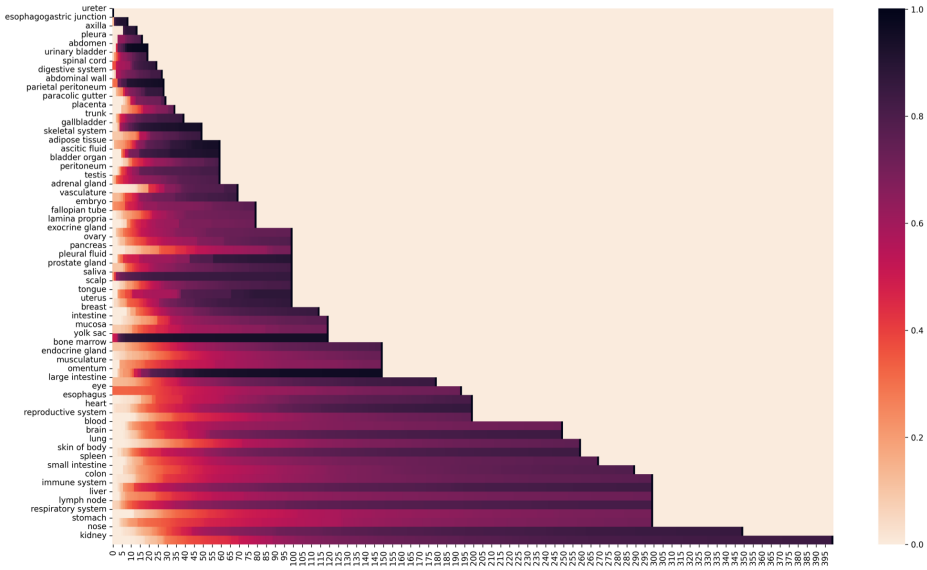


(b)

Figure 3.2: The line plot (a) and heatmap (b) of the classification accuracy of all classified cells in all previous rounds.

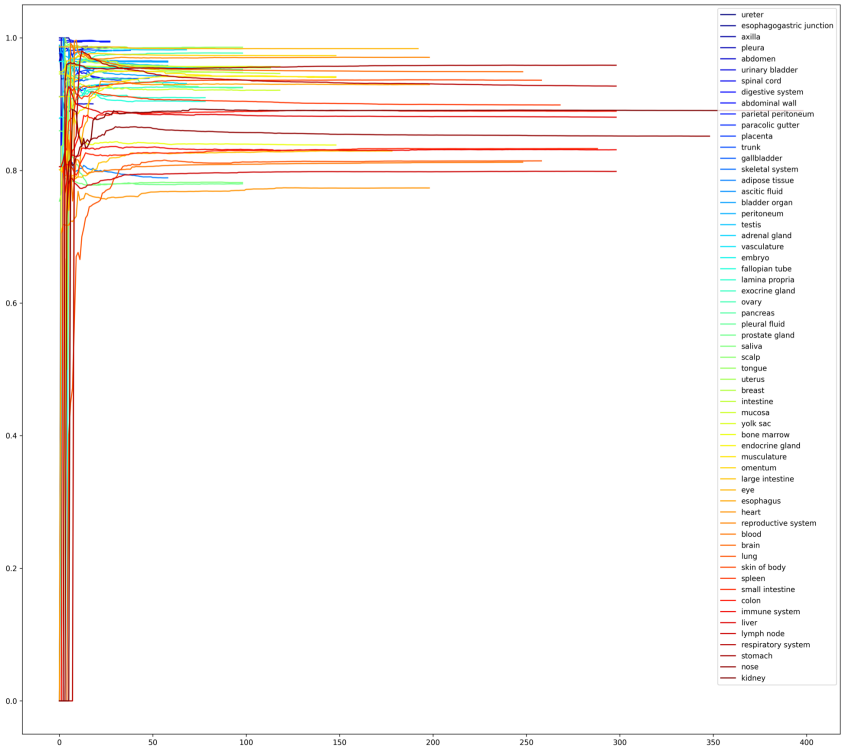


(a)

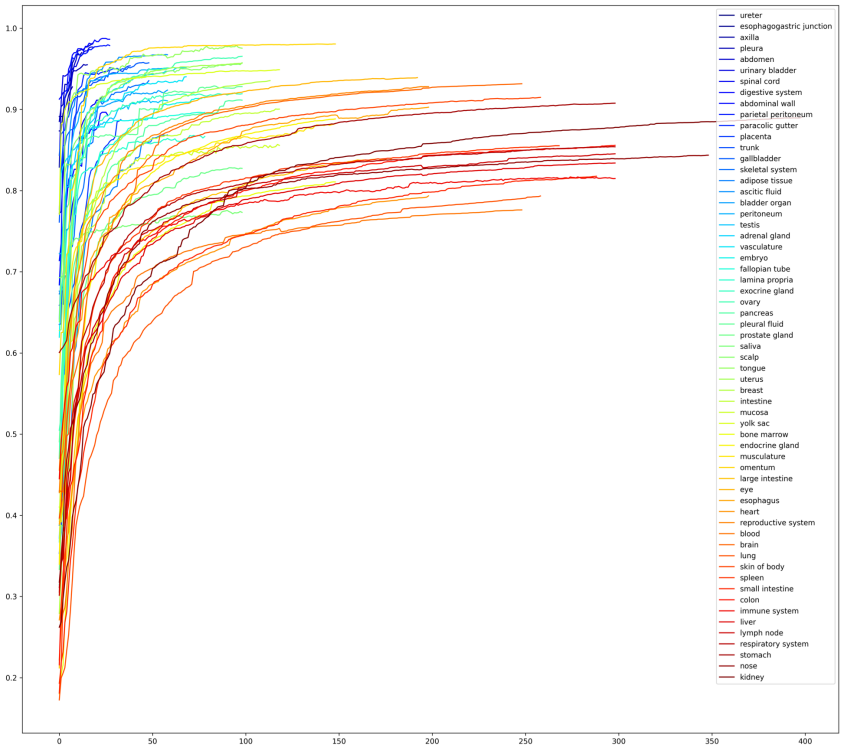


(b)

Figure 3.3: The line plot (a) and heatmap (b) of the fraction of all classified cells in all previous rounds over the total number of cells.



(a)



(b)

Figure 3.4: Compare the accuracy of ActiveCellInference (a) and ActiveSVM (b).

Further detailed comparisons for all organs are displayed in Figures 3.5 and 3.6, where the accuracy of ActiveCellInference is represented by orange lines and that of ActiveSVM by blue lines, again with the number of genes used along the x-axis. This visual comparison underscores the efficiency and effectiveness of ActiveCellInference in reaching and maintaining high accuracy levels across different organ datasets.

Final Results

Figure 3.7 displays two sets of accuracy measurements: one for cells classified by the end of the iterative process and another after classifying all remaining unclassified cells in the final round. As expected, the accuracy for the latter group is higher due to the inclusion of all cells.

The accuracy for most organs exceeds 80%, aligned with the predefined thresholds set for most organs at levels such as 80%, 90%, or 95%. Several organs even achieve accuracies surpassing 95%. Lower accuracy levels are observed in organs with either a large number of cells, a wide variety of cell types, or where the datasets and annotations are particularly noisy. For instance, the prostate gland dataset exhibits notably low classification accuracy due to its noise levels, despite utilizing all available genes. Similarly, the lung dataset's vast size and numerous cell types pose significant challenges to achieving high accuracy with a limited number of genes.

The average number of genes required per cell varies significantly across different organs, as shown in the results. This variation is based on the application of ActiveCellInference using the complete ActiveSVM gene set without implementing an early stop strategy. In practice, while the ureter dataset requires as few as one gene, the respiratory system may need upwards of 130 genes for effective classification. Generally, for most organs, the algorithm successfully classifies all cells using only tens of genes on average.

3.8 Results: Temporal Scaling vs. Platt Scaling

In my analysis, I evaluated the temporal scaling method I developed for ActiveCellInference against the traditional Platt Scaling using Expected Calibration Error (ECE). ECE is a commonly used metric for assessing probability calibration models, measuring the expected calibration gap, which is the discrepancy between the expected sample accuracy and the average predicted probability for each bin.

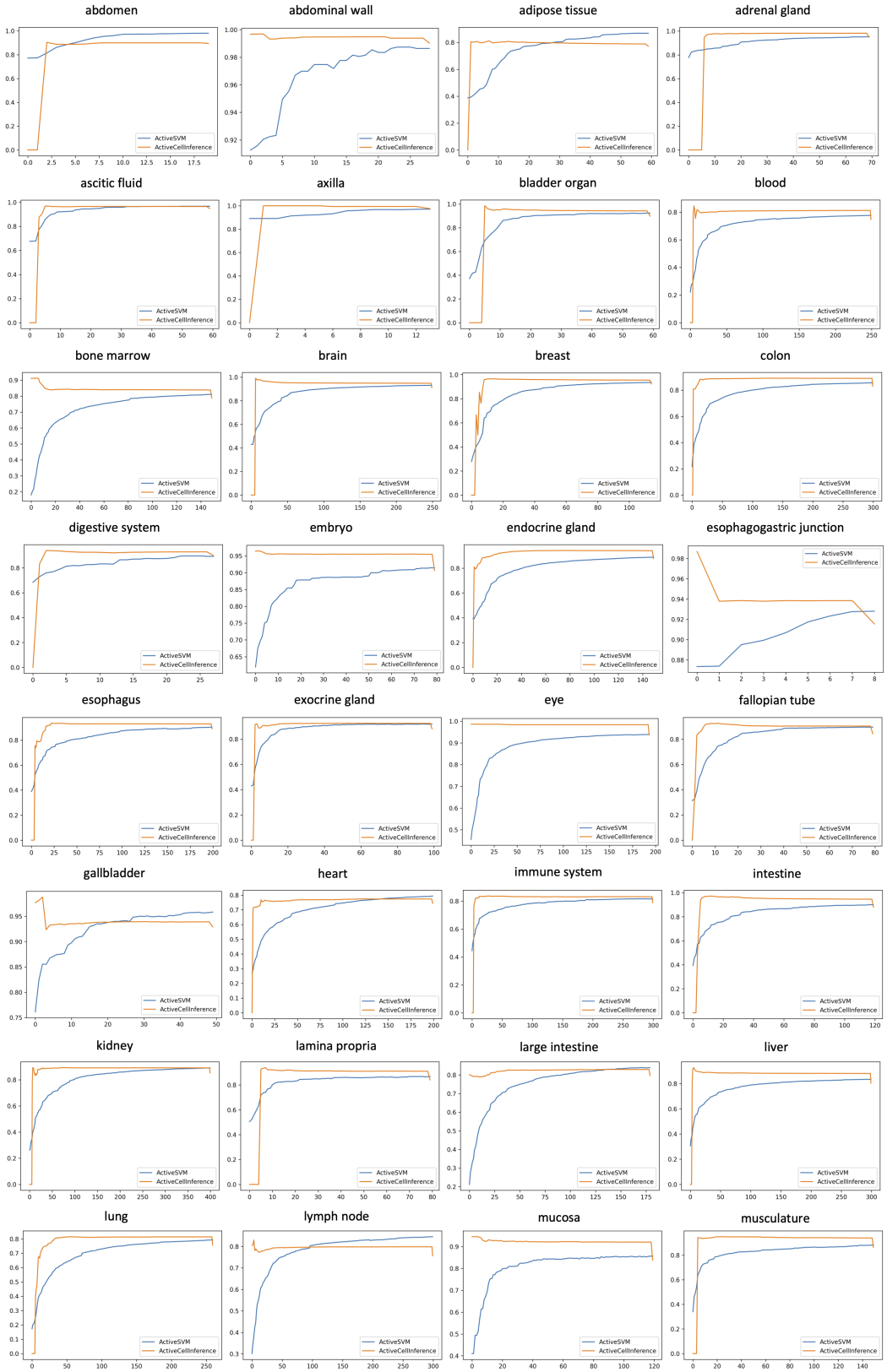


Figure 3.5: Comparison between the accuracy of ActiveCellInference and ActiveSVM of all organs.

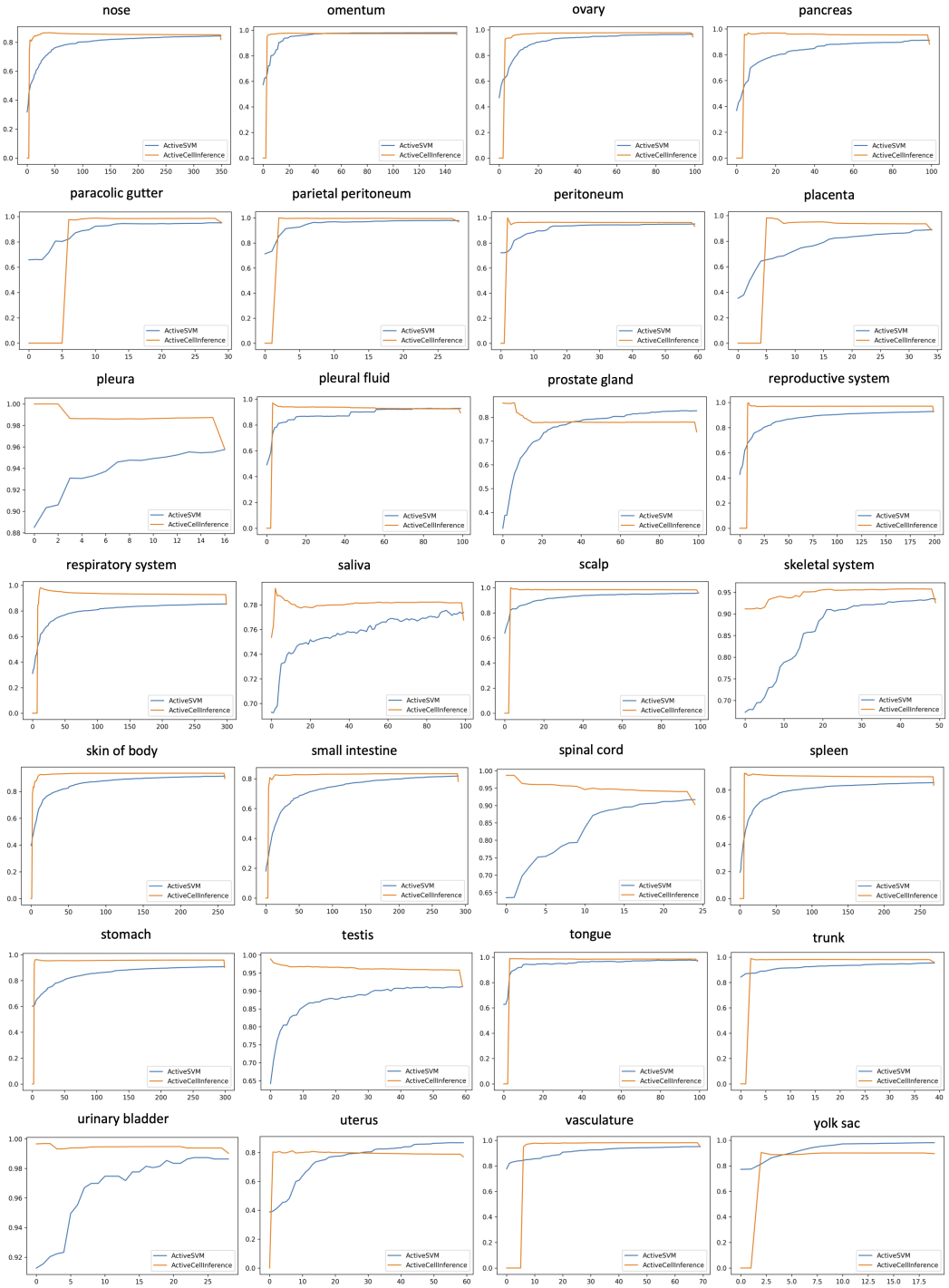


Figure 3.6: Comparison between the accuracy of ActiveCellInference and ActiveSVM of all organs.

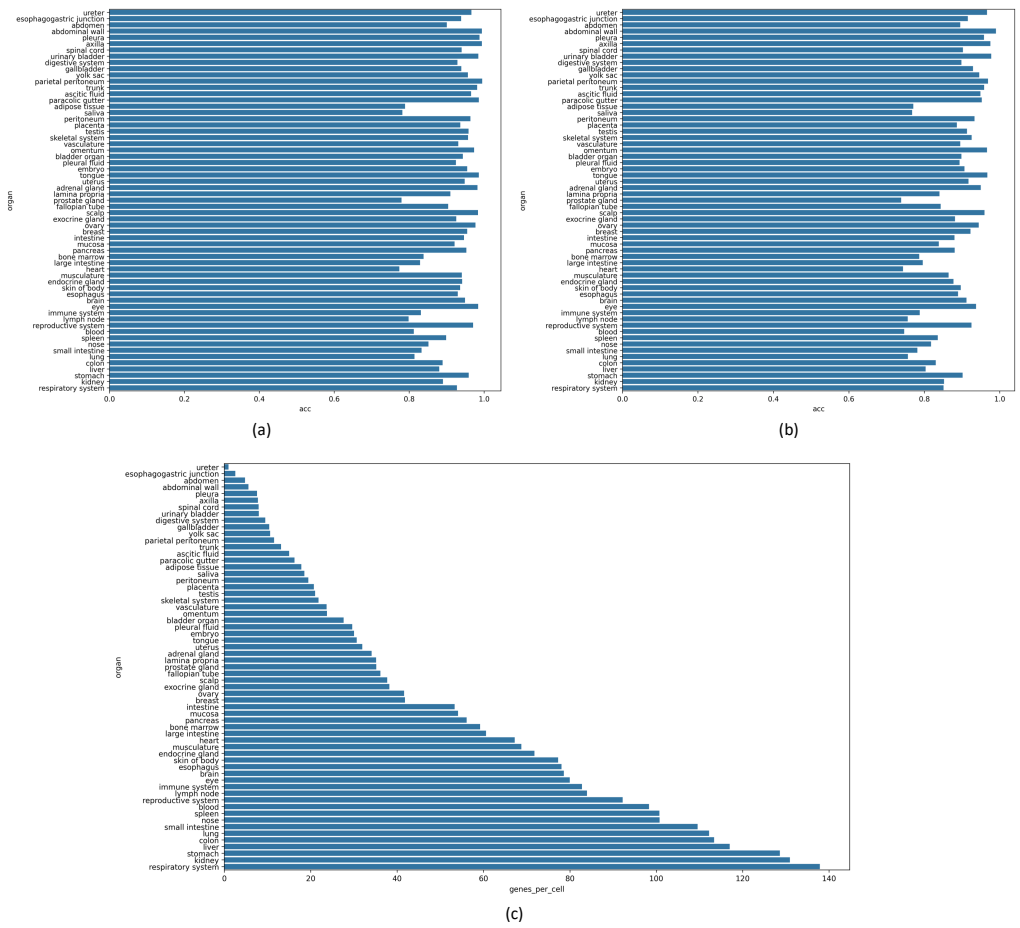


Figure 3.7: (a) The accuracy of all cells classified by ActiveCellInference; (b) The accuracy of all cells classified by ActiveCellInference with all unclassified cells classified at the last round; (c) The average number of genes queried per cell.

These comparative experiments were conducted across 31 human organs, with temporal scaling showing significant improvements in ECE over Platt Scaling. The results for some of these organs are displayed in Figure 3.8, illustrating the enhanced calibration performance achieved through temporal scaling.

3.9 Discussion

ActiveCellInference offers a versatile iterative framework designed to facilitate active cell acquisition in spatial genomics, enhancing the efficiency and cost-effectiveness of spatial sequencing processes. While the framework initially employs Support Vector Machines (SVM) for classification, it is flexible enough to incorporate any classification model as its computational engine. Similar probability calibration methods that are used with SVM can also be adapted to work with other

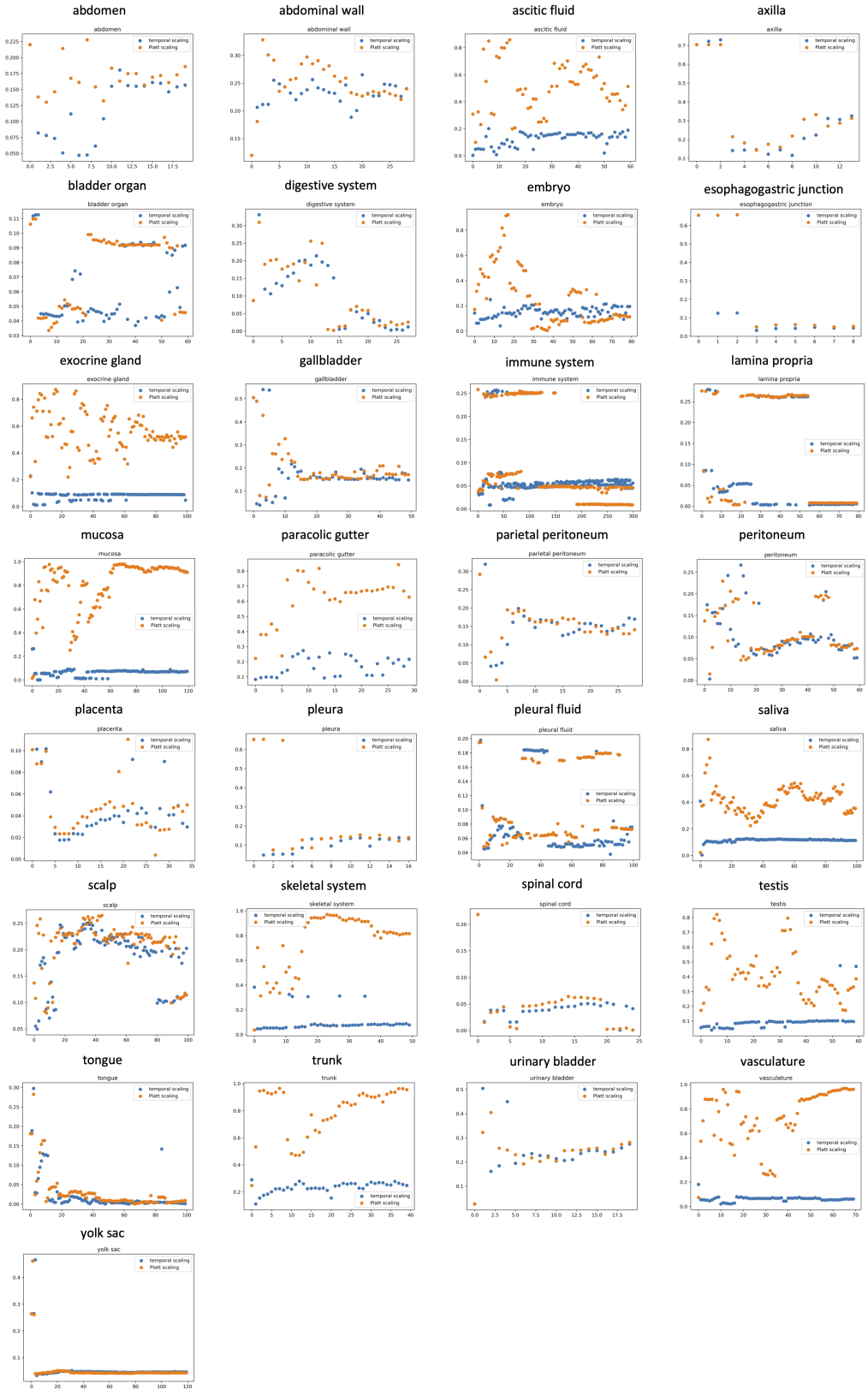


Figure 3.8: The Expected Calibration Error (ECE) values for Temporal Scaling and Platt Scaling across 31 organs are depicted here. The x-axis represents the number of genes. Orange lines indicate temporal Scaling, while blue lines represent Platt Scaling. Temporal Scaling demonstrates a lower calibration error compared to Platt Scaling for almost all the organs evaluated.

classifiers within this framework, ensuring robustness in predictions.

Additionally, the framework is not limited to classification models alone; it can integrate regression models to predict continuous outputs. However, adapting regression models requires alternative approaches to measure uncertainty, as traditional probability calibration techniques are not suitable. Techniques such as predictive variance from Gaussian processes or bootstrapping methods can be employed to estimate uncertainty in regression contexts.

Although the ActiveCellInference acquisition process is not universally applicable to all spatial sequencing techniques, it significantly contributes to the development of more cost-effective and advanced methods. It guides the strategic design of gene probes, potentially reducing the complexity and expenses associated with current spatial sequencing techniques. The ongoing development in this area suggests that more refined and economically feasible spatial sequencing technologies are likely to emerge, driven by innovations such as ActiveCellInference, which streamline genomic analyses and make high-resolution spatial genomics more accessible.

Chapter 4

DISSCUSSION AND OUTLOOK

We introduce and detail the implementation of two advanced methodologies, ActiveSVM and ActiveCellInference, both of which enhance the efficiency and precision of genomic analyses through innovative feature selection and active learning strategies.

ActiveSVM operates on large single-cell mRNA-seq datasets, utilizing an iterative cell-state classification approach to refine and minimize gene sets effectively. By selectively focusing on cells that poorly classify under current models, ActiveSVM iteratively integrates maximally informative genes, thereby optimizing the computational resources by concentrating efforts on a subset of highly informative cells. This method allows the handling of extensive datasets, showcasing its capability to identify compact gene sets that achieve high accuracy in cell-type classification across various genomic tasks, including perturb-seq data analysis and spatial transcriptomics. Conceptually, ActiveSVM embodies the principles of active learning by targeting data examples (cells) that need further analysis to refine classification accuracy, paralleling concepts in boosting methods where the focus is on challenging examples to enhance overall model performance.

Complementing this, ActiveCellInference provides a robust iterative framework for active cell acquisition in spatial genomics, designed to integrate seamlessly with various classification or regression models. This flexibility allows the framework to adapt to different analytical needs, including the prediction of continuous outputs. ActiveCellInference enhances the efficiency and cost-effectiveness of spatial sequencing by guiding the strategic design of gene probes and reducing the complexity and expenses associated with spatial sequencing techniques. Despite its specific application constraints, the potential for broader application and methodological refinement suggests a promising future for more advanced genomic technologies.

Both methodologies leverage the inherent low-dimensional structure found within transcriptomes, where significant information about cellular states can be inferred from a small subset of genes. This understanding allows for a reduction in the scale of necessary measurements, significantly lowering the costs and complexity of data acquisition. Future directions for these strategies include their application at the

measurement phase, potentially using active learning to select only those cells that meet specific criteria for sequencing or imaging, thereby optimizing resource use and operational efficiency.

By integrating ActiveSVM's feature selection capabilities with ActiveCellInference's framework for spatial genomics, these methodologies not only streamline genomic analysis but also pave the way for significant advancements in how we approach complex disease understanding and personalized medicine. The continuous development and application of these strategies are set to revolutionize genomic studies, making high-resolution analyses more accessible and impactful in both research and clinical settings.

In the future, the active acquisition framework, which focuses on both gene and cell selection, presents a promising avenue for transitioning advanced sequencing techniques into clinical applications. Currently, the high costs associated with single-cell sequencing techniques limit their practical use in real-world settings. However, by selectively targeting specific genes and cells that provide the most informative data, this approach could significantly reduce the financial burden associated with comprehensive genomic analyses.

This proactive strategy could not only streamline the sequencing process but also enhance its affordability, making these advanced technologies accessible for routine clinical diagnostics and personalized medicine. The implementation of such targeted sequencing methods promises to bring high-throughput genomic technologies to the forefront of clinical practice, where they can be used for more precise disease diagnosis, treatment monitoring, and the development of tailored therapies.

By refining the selection process to focus only on the most crucial genetic information needed for specific clinical outcomes, we can maximize the efficiency and efficacy of genomic sequencing. This approach would not only lower costs but also reduce the time and resources required for data processing and analysis, further enhancing the practicality of these techniques for everyday clinical use. The continued development and refinement of such frameworks hold the potential to revolutionize how we integrate genomic data into healthcare, bridging the gap between high-tech research methodologies and patient-centered clinical solutions.

BIBLIOGRAPHY

Abdiansah, Abdiansah and Retantyo Wardoyo (2015). "Time complexity analysis of support vector machines (SVM) in LibSVM". In: *International journal computer and application* 128.3, pp. 28–34.

Alexandria, VA (n.d.). "National Science Foundation". In: *Retrieved from* ().

Amazon (n.d.[a]). *Amazon Elastic Compute Cloud Documentation*. URL: https://docs.aws.amazon.com/ec2/.

– (n.d.[b]). *AWS Innovate*. URL: https://aws.amazon.com.

– (n.d.[c]). *Instance types*. URL: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html.

Andrews, Madeline G, Lakshmi Subramanian, and Arnold R Kriegstein (2020). "mTOR signaling regulates the morphology and migration of outer radial glia in developing human cortex". In: *Elife* 9, e58737.

Anthony, Todd E et al. (2005). "Brain lipid-binding protein is a direct target of Notch signaling in radial glial cells". In: *Genes & development* 19.9, pp. 1028–1033.

Arlot, Sylvain and Alain Celisse (2010). "A survey of cross-validation procedures for model selection". In: *Statistics surveys* 4, pp. 40–79.

Bhaduri, Aparna et al. (2018). "Identification of cell types in a mouse brain single-cell atlas using low sampling coverage". In: *BMC biology* 16.1, pp. 1–10.

Biology, CZI Single-Cell et al. (2023). "CZ CELLxGENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data". In: *bioRxiv*, pp. 2023–10.

Blondel, Vincent D et al. (2008). "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10, P10008.

Borger, Eva et al. (2014). "The calcium-binding protein EFhd2 modulates synapse formation in vitro and is linked to human dementia". In: *Journal of Neuropathology & Experimental Neurology* 73.12, pp. 1166–1182.

Bottou, Léon and Chih-Jen Lin (2007). "Support vector machine solvers". In: *Large scale kernel machines* 3.1, pp. 301–320.

Brown, Lawrence D, T Tony Cai, and Anirban DasGupta (2001). "Interval estimation for a binomial proportion". In: *Statistical science* 16.2, pp. 101–133.

Chang, Yin-Wen and Chih-Jen Lin (Mar. 2008). "Feature Ranking Using Linear SVM". In: *Proceedings of the Workshop on the Causation and Prediction Challenge at WCCI 2008*. Ed. by Isabelle Guyon et al. Vol. 3. Proceedings of Machine Learning Research. Hong Kong: PMLR, pp. 53–64. URL: http://proceedings.mlr.press/v3/chang08a.html.

Chen, Kok Hao et al. (2015). "Spatially resolved, highly multiplexed RNA profiling in single cells". In: *Science* 348.6233, aaa6090.

Chen, Sisi et al. (2020). "Dissecting heterogeneous cell populations across drug and disease conditions with PopAlign". In: *Proceedings of the National Academy of Sciences* 117.46, pp. 28784–28794.

Cleary, Brian et al. (2017). "Efficient generation of transcriptomic profiles by random composite measurements". In: *Cell* 171.6, pp. 1424–1436.

Consortium, Tabula Muris et al. (2018). "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris." In: *Nature* 562.7727, p. 367.

DeGroot, Morris H and Stephen E Fienberg (1983). "The comparison and evaluation of forecasters". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 32.1-2, pp. 12–22.

Delaney, Conor et al. (Oct. 2019). "Combinatorial prediction of marker panels from single-cell transcriptomic data". en. In: *Mol. Syst. Biol.* 15.10, e9005.

Dixit, Atray et al. (2016). "Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens". In: *cell* 167.7, pp. 1853–1866.

Dobreva, Tatyana et al. (2020). "Single cell profiling of capillary blood enables out of clinic human immunity studies". In: *Scientific reports* 10.1, pp. 1–9.

Eng, Chee-Huat Linus et al. (2019). "Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+". In: *Nature* 568.7751, pp. 235–239.

Fan, H Christina, Glenn K Fu, and Stephen PA Fodor (2015). "Combinatorial labeling of single cells for gene expression cytometry". In: *Science* 347.6222.

Fan, Rong-En et al. (2008). "LIBLINEAR: A library for large linear classification". In: *the Journal of machine Learning research* 9, pp. 1871–1874.

Felder, Richard M and Rebecca Brent (2009). "Active learning: An introduction". In: *ASQ higher education brief* 2.4, pp. 1–5.

Ge, Feng et al. (2011). "Quantitative proteomic analysis of tumor reversion in multiple myeloma cells". In: *Journal of proteome research* 10.2, pp. 845–855.

Genomics, X (2017). "1.3 million brain cells from E18 mice". In: *CC BY* 4.

Gordon, Geoff and Ryan Tibshirani (2012). "Karush-kuhn-tucker conditions". In: *Optimization* 10.725/36, p. 725.

Guan, Wenqian et al. (Apr. 2020). "The diagnostic value of serum DSA-TRF in hepatocellular carcinoma". en. In: *Glycoconj. J.* 37.2, pp. 231–240.

Guo, Chuan et al. (2017). "On calibration of modern neural networks". In: *International conference on machine learning*. PMLR, pp. 1321–1330.

Hebenstreit, Daniel (2012). “Methods, challenges and potentials of single cell RNA-seq”. In: *Biology* 1.3, pp. 658–667.

Heimberg, Graham et al. (2016). “Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing”. In: *Cell systems* 2.4, pp. 239–250.

Hudak, Dave et al. (2018). “Open OnDemand: A web-based client portal for HPC centers”. In: *Journal of Open Source Software* 3.25, p. 622.

Jiang, Jialong, David A Sivak, and Matt Thomson (2019). “Active Learning of Spin Network Models”. In: *arXiv preprint arXiv:1903.10474*.

Kull, Meelis, Telmo Silva Filho, and Peter Flach (2017). “Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers”. In: *Artificial intelligence and statistics*. PMLR, pp. 623–631.

Küppers, Fabian, Jonas Schneider, and Anselm Haselhoff (2022). “Parametric and multivariate uncertainty calibration for regression and object detection”. In: *European Conference on Computer Vision*. Springer, pp. 426–442.

Lee, Je Hyuk et al. (2014). “Highly multiplexed subcellular RNA sequencing in situ”. In: *science* 343.6177, pp. 1360–1363.

Likas, Aristidis, Nikos Vlassis, and Jakob J Verbeek (2003). “The global k-means clustering algorithm”. In: *Pattern recognition* 36.2, pp. 451–461.

Liu, Minxia et al. (2021). “S100 Calcium Binding Protein Family Members Associate With Poor Patient Outcome and Response to Proteasome Inhibition in Multiple Myeloma”. In: *Frontiers in Cell and Developmental Biology*, p. 2261.

Lubeck, Eric et al. (2014). “Single-cell in situ RNA profiling by sequential hybridization”. In: *Nature methods* 11.4, pp. 360–361.

Macosko, Evan Z et al. (2015). “Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets”. In: *Cell* 161.5, pp. 1202–1214.

Magalhães, Roberto J Pessoa de et al. (2013). “Analysis of the immune system of multiple myeloma patients achieving long-term disease control by multidimensional flow cytometry”. In: *Haematologica* 98.1, p. 79.

Magnaldo, T, D Fowlis, and M Darmon (July 1998). “Galectin-7, a marker of all types of stratified epithelia”. en. In: *Differentiation* 63.3, pp. 159–168.

Malek, Ehsan et al. (2016). “Myeloid-derived suppressor cells: The green light for myeloma immune escape”. In: *Blood reviews* 30.5, pp. 341–348.

Marshall, Jamie L et al. (2020). “HyPR-seq: Single-cell quantification of chosen RNAs via hybridization and sequencing of DNA probes”. In: *Proceedings of the National Academy of Sciences* 117.52, pp. 33404–33413.

McHugh, Mary L (2013). “The chi-square test of independence”. In: *Biochemia medica: Biochemia medica* 23.2, pp. 143–149.

Moffitt, Jeffrey R et al. (2016). “High-throughput single-cell gene-expression profiling with multiplexed error-robust fluorescence in situ hybridization”. In: *Proceedings of the National Academy of Sciences* 113.39, pp. 11046–11051.

Naeini, Mahdi Pakdaman, Gregory Cooper, and Milos Hauskrecht (2015). “Obtaining well calibrated probabilities using bayesian binning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 29. 1.

Navin, Nicholas et al. (2011). “Tumour evolution inferred by single-cell sequencing”. In: *Nature* 472.7341, pp. 90–94.

Noble, William S (2006). “What is a support vector machine?” In: *Nature biotechnology* 24.12, pp. 1565–1567.

Osei-Owusu, Patrick et al. (Oct. 2019). “FPR1 is the plague receptor on host immune cells”. en. In: *Nature* 574.7776, pp. 57–62.

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

Pei, DQ and CH Shih (1990). “Transcriptional activation and repression by cellular DNA-binding protein C/EBP”. In: *Journal of virology* 64.4, pp. 1517–1522.

Platt, John et al. (1999). “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods”. In: *Advances in large margin classifiers* 10.3, pp. 61–74.

Pollen, Alex A et al. (2015). “Molecular identity of human outer radial glia during cortical development”. In: *Cell* 163.1, pp. 55–67.

Rawstron, Andrew C et al. (1998). “B-lymphocyte suppression in multiple myeloma is a reversible phenomenon specific to normal B-cell progenitors and plasma cell precursors”. In: *British journal of haematology* 100.1, pp. 176–183.

Regev, Aviv et al. (2017). “The human cell atlas”. In: *elife* 6, e27041.

Replogle, Joseph M et al. (2020). “Combinatorial single-cell CRISPR screens by direct guide RNA capture and targeted sequencing”. In: *Nature biotechnology* 38.8, pp. 954–961.

Riemondy, Kent A et al. (2019). “Recovery and analysis of transcriptome subsets from pooled single-cell RNA-seq libraries”. In: *Nucleic acids research* 47.4, e20–e20.

Rodriques, Samuel G et al. (2019). “Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution”. In: *Science* 363.6434, pp. 1463–1467.

Rosasco, Lorenzo et al. (2004). “Are loss functions all the same?” In: *Neural computation* 16.5, pp. 1063–1076.

Rotem, Assaf et al. (2015). “Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state”. In: *Nature biotechnology* 33.11, pp. 1165–1172.

Rouillard, Andrew D et al. (2016). “The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins”. In: *Database* 2016.

Rückstieß, Thomas, Christian Osendorfer, and Patrick van der Smagt (2011). “Sequential feature selection for classification”. In: *Australasian Joint Conference on Artificial Intelligence*. Springer, pp. 132–141.

Safavian, S Rasoul and David Landgrebe (1991). “A survey of decision tree classifier methodology”. In: *IEEE transactions on systems, man, and cybernetics* 21.3, pp. 660–674.

Sage, Adam P et al. (Oct. 2020). “Assessment of long non-coding RNA expression reveals novel mediators of the lung tumour immune response”. en. In: *Sci. Rep.* 10.1, p. 16945.

Schapire, Robert E (1999). “A brief introduction to boosting”. In: *Ijcai*. Vol. 99. Citeseer, pp. 1401–1406.

– (2003). “The boosting approach to machine learning: An overview”. In: *Nonlinear estimation and classification*, pp. 149–171.

Settles, Burr (2009). “Active learning literature survey”. In.

– (2011). “From theories to queries: Active learning in practice”. In: *Active learning and experimental design workshop in conjunction with AISTATS 2010*. JMLR Workshop and Conference Proceedings, pp. 1–18.

Song, Hao et al. (2019). “Distribution calibration for regression”. In: *International Conference on Machine Learning*. PMLR, pp. 5897–5906.

Stables, Melanie J et al. (Dec. 2011). “Transcriptomic analyses of murine resolution-phase macrophages”. en. In: *Blood* 118.26, e192–208.

Ståhl, Patrik L et al. (2016). “Visualization and analysis of gene expression in tissue sections by spatial transcriptomics”. In: *Science* 353.6294, pp. 78–82.

Stegle, Oliver, Sarah A Teichmann, and John C Marioni (2015). “Computational and analytical challenges in single-cell transcriptomics”. In: *Nature Reviews Genetics* 16.3, pp. 133–145.

Street, Kelly et al. (2018). “Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics”. In: *BMC genomics* 19.1, pp. 1–16.

Stuart, Tim and Rahul Satija (2019). “Integrative single-cell analysis”. In: *Nature reviews genetics* 20.5, pp. 257–272.

Suen, Hayley et al. (2016). “Multiple myeloma causes clonal T-cell immunosenescence: identification of potential novel targets for promoting tumour immunity and implications for checkpoint blockade”. In: *Leukemia* 30.8, pp. 1716–1724.

Syarif, Iwan, Adam Prugel-Bennett, and Gary Wills (2016). “SVM parameter optimization using grid search and genetic algorithm to improve classification performance”. In: *Telkomnika* 14.4, p. 1502.

Tang, Fuchou et al. (2009). "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature methods* 6.5, pp. 377–382.

Taylor, Richard (1990). "Interpretation of the correlation coefficient: a basic review". In: *Journal of diagnostic medical sonography* 6.1, pp. 35–39.

Tirosh, Itay et al. (2016). "Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq". In: *Science* 352.6282, pp. 189–196.

Torvalds, Linus (n.d.). *Linux Kernel*. URL: https://kernel.org.

Trapnell, Cole et al. (2014). "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells". In: *Nature biotechnology* 32.4, pp. 381–386.

Vega, Irving E (2016). "EFhd2, a protein linked to Alzheimer's disease and other neurological disorders". In: *Frontiers in neuroscience* 10, p. 150.

Vergara, Jorge R and Pablo A Estévez (2014). "A review of feature selection methods based on mutual information". In: *Neural computing and applications* 24.1, pp. 175–186.

Villani, Alexandra-Chloé et al. (2017). "Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors". In: *Science* 356.6335, eaah4573.

Wang, Fang et al. (2019). "SCMarker: ab initio marker selection for single cell transcriptome profiling". In: *PLoS computational biology* 15.10, e1007445.

Wilks, Daniel S (1990). "On the combination of forecast probabilities for consecutive precipitation periods". In: *Weather and forecasting* 5.4, pp. 640–650.

Wolf, F Alexander, Philipp Angerer, and Fabian J Theis (2018). "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19.1, pp. 1–5.

Wu, Yu and Cynthia M Smas (July 2008). "Wdnm1-like, a new adipokine with a role in MMP-2 activation". en. In: *Am. J. Physiol. Endocrinol. Metab.* 295.1, E205–15.

Xia, Chang et al. (2018). "S100 proteins as an important regulator of macrophage inflammation". In: *Frontiers in immunology* 8, p. 1908.

Xia, Peipei, Li Zhang, and Fanzhang Li (2015). "Learning similarity with cosine similarity ensemble". In: *Information Sciences* 307, pp. 39–52.

Xiao, Ta et al. (2018). "RACK1 promotes tumorigenicity of colon cancer by inducing cell autophagy". In: *Cell death & disease* 9.12, pp. 1–13.

xqchen (Apr. 2022). *xqchen/activeSVC: ActiveSVM*. Version 4.0.4. DOI: 10.5281/zenodo.6481687. URL: https://doi.org/10.5281/zenodo.6481687.

Zadrozny, Bianca and Charles Elkan (2001). “Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers”. In: *Icml*. Vol. 1, pp. 609–616.

– (2002). “Transforming classifier scores into accurate multiclass probability estimates”. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699.

Zeisel, Amit et al. (2015). “Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq”. In: *Science* 347.6226, pp. 1138–1142.

Zheng, Grace XY et al. (2017). “Massively parallel digital transcriptional profiling of single cells”. In: *Nature communications* 8.1, pp. 1–12.

Appendix A

OPTIMAL PARAMETERS FOR ACTIVESVM EXPERIMENTS

In this appendix, we detail the algorithm parameters employed in our ActiveSVM experiments, which are listed in Tables A.1 to A.3. ActiveSVM incorporates 15 user-defined hyperparameters: five are specific to the feature selection process, while the remaining ten are standard parameters used in linear SVM classifiers. Comprehensive descriptions of all ActiveSVM parameters are available on the integrated package page at https://pypi.org/project/activeSVC/.

Table A.1: Parameters of ActiveSVM (PBMC and mouse megacell datasets).

	PBMC (min-complexity)	PBMC (min-cell)	mouse megacell (min-complexity)	mouse megacell (min-cell)
num_features	50	20	50	50
num_samples	20	100	20	100
init_features	1	1	1	1
init_samples	20	200	20	100
balance	True/False	True	True	True
penalty	'l2'	'l2'	'l2'	'l2'
loss	squared_hinge	squared_hinge	squared_hinge	squared_hinge
dual	True	True	True	True
tol	1e-4	1e-4	1e-4	1e-4
C	1.0	1.0	1.0	1.0
fit_intercept	True	True	True	True
intercept_scaling	1	1	1	1
class_weight	None	None	'balanced'	'balanced'
random_state	None	None	None	None
max_iter	1000	1000	1000	1000

Table A.2: Parameters of ActiveSVM (Tabula Muris and MM datasets).

	Tabula Muris (min-complexity)	Tabula Muris (min-cell)	MM (min-complexity)	MM (min-cell)
num_features	150	500	40	40
num_samples	20	200	20	100
init_features	1	1	1	1
init_samples	20	200	20	100
balance	True/False	False	True/False	False
penalty	'l2'	'l2'	'l2'	'l2'
loss	squared_hinge	squared_hinge	squared_hinge	squared_hinge
dual	True	True	True	True
tol	1e-4	1e-4	1e-4	1e-4
C	1.0	1.0	1.0	1.0
fit_intercept	True	True	True	True
intercept_scaling	1	1	1	1
class_weight	None	None	None	None
random_state	None	None	None	None
max_iter	1000	1000	1000	1000

Table A.3: Parameters of ActiveSVM (perturb-seq and seqFish datasets).

	perturb-seq (min-complexity)	seqFish (min-complexity)
$num\_features$	50	30
$num\_samples$	500	10
$init\_features$	1	1
$init\_samples$	1000	10
$balance$	True	False
$penalty$	'l2'	'l2'
$loss$	squared_hinge	squared_hinge
$dual$	True	True
tol	1e-6	1
C	1.0	10
$fit\_intercept$	True	True
$intercept\_scaling$	1	1
$class\_weight$	'balanced'	None
$random\_state$	None	None
max_iter	1,000,000	100,000

Appendix B

READY-TO-USE GENE SET FIGURES

Figures A.1 through A.9 showcase the plots of ActiveSVM gene sets for all organs. In these figures, the left sub-figures display the UMAP visualizations of cell types. The middle sub-figures illustrate the accuracy of the ActiveSVM gene sets, with the x-axis representing the number of genes used. The right sub-figures highlight the first four genes selected for each organ. Comprehensive lists detailing the ordered gene sets for all organs are available in Appendix C: Full Lists of ActiveSVM Gene Sets for HCA Data.

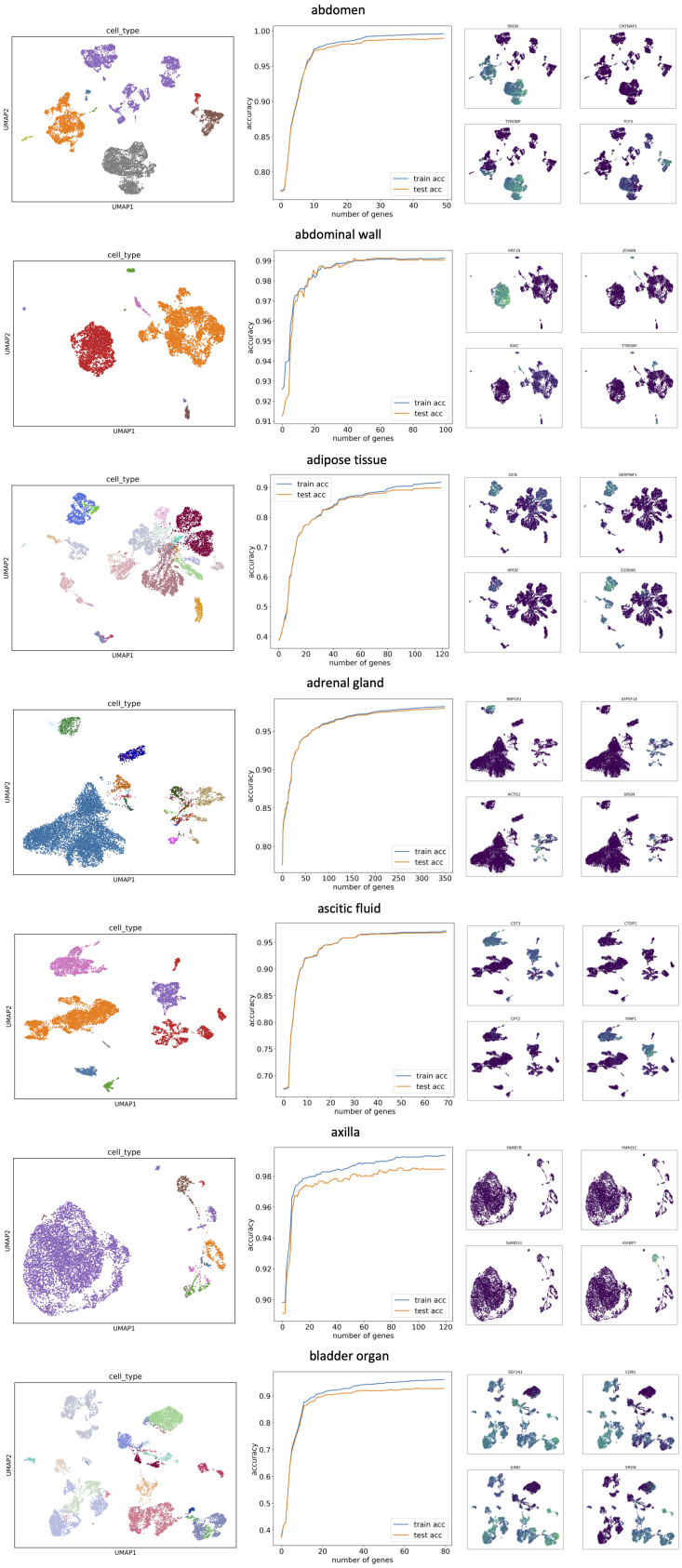


Figure B.1: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

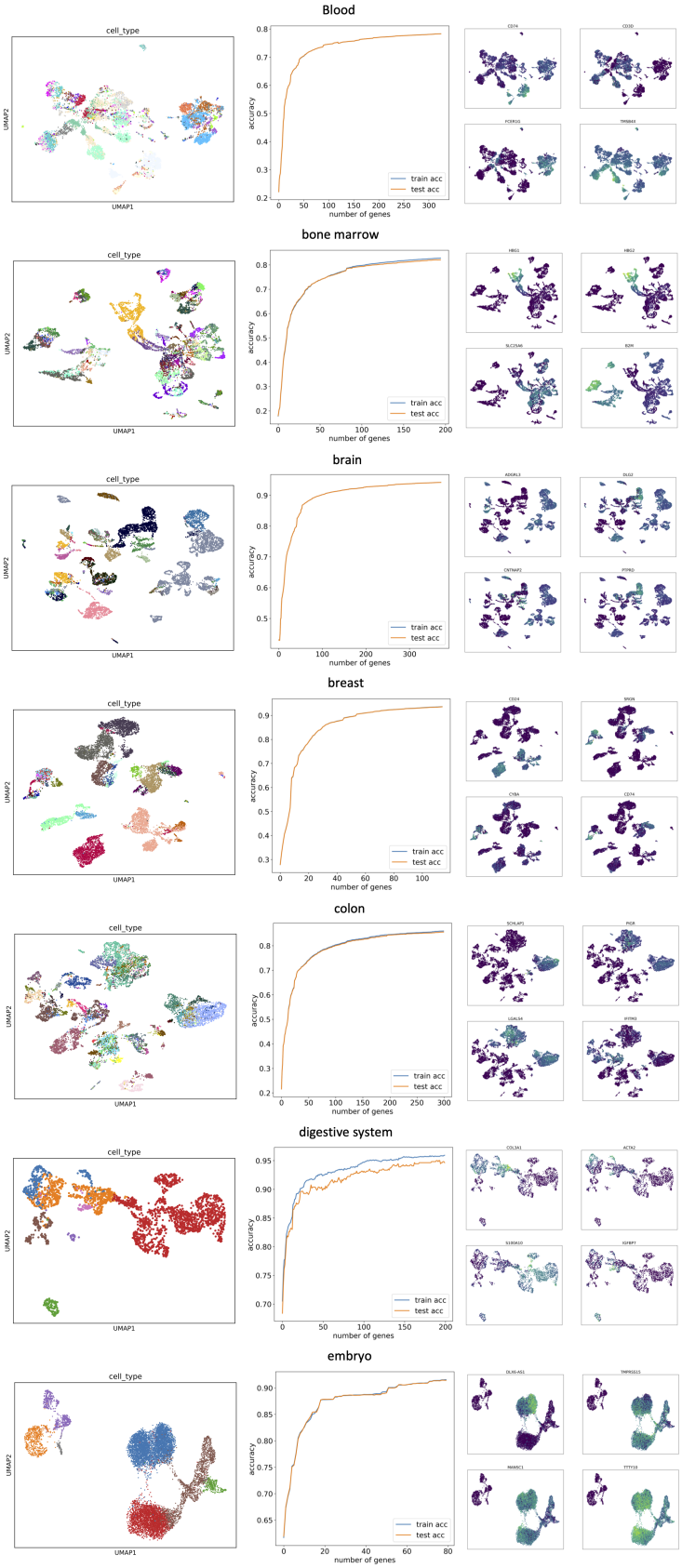


Figure B.2: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

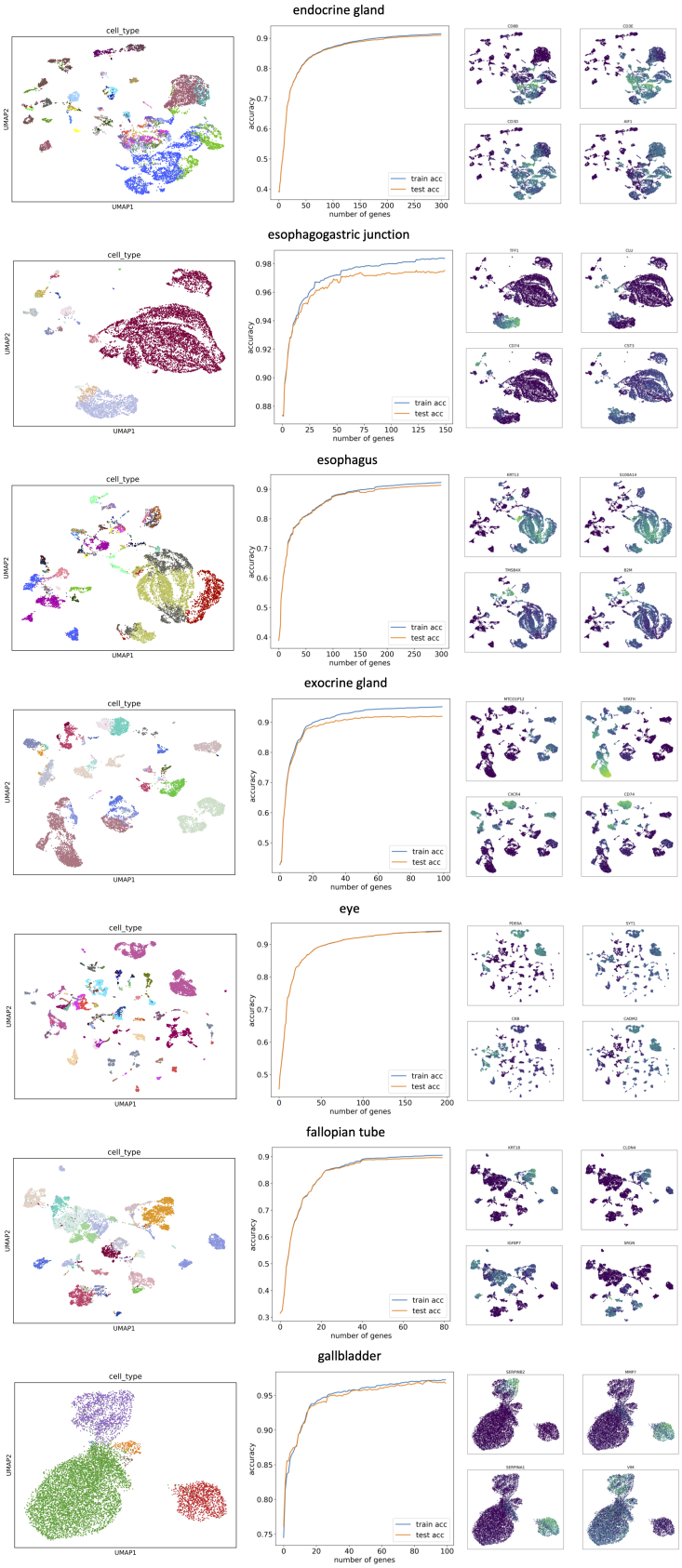


Figure B.3: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

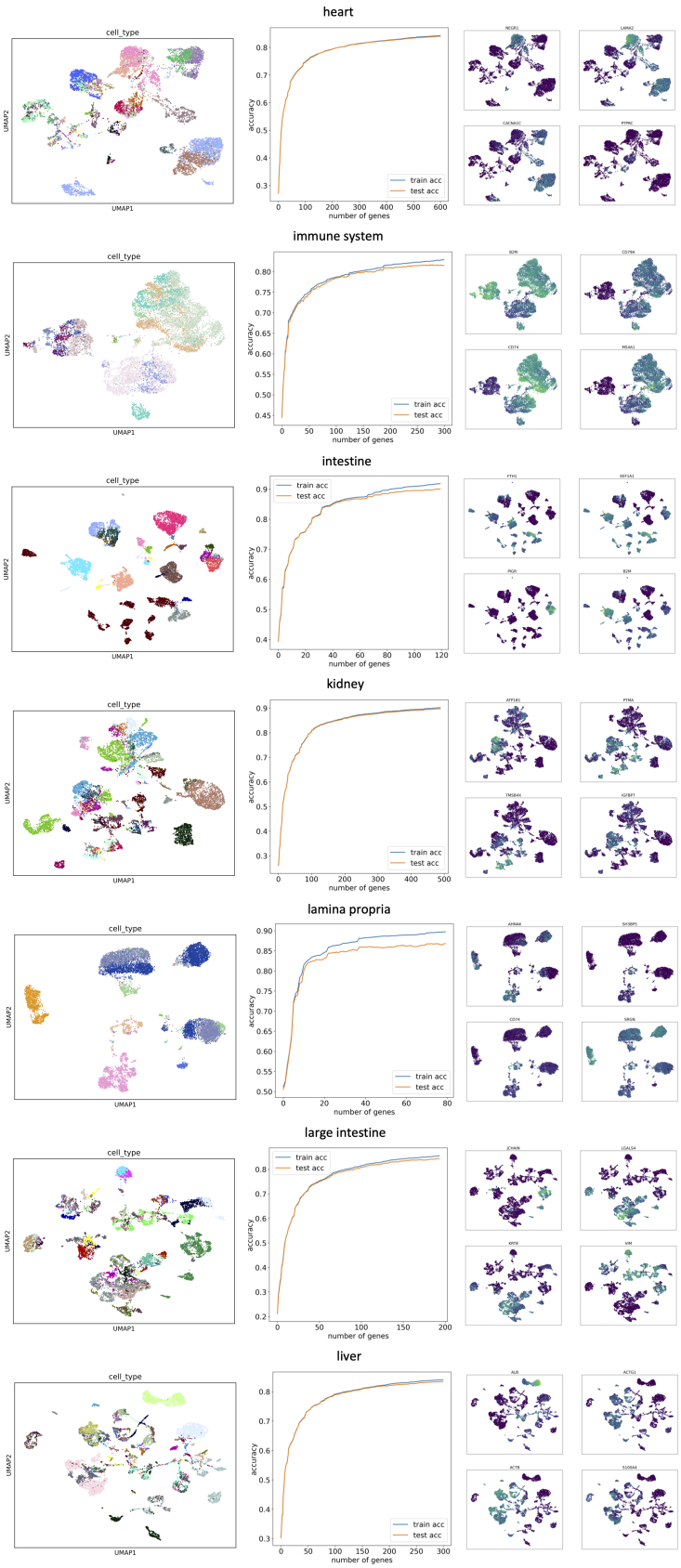


Figure B.4: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

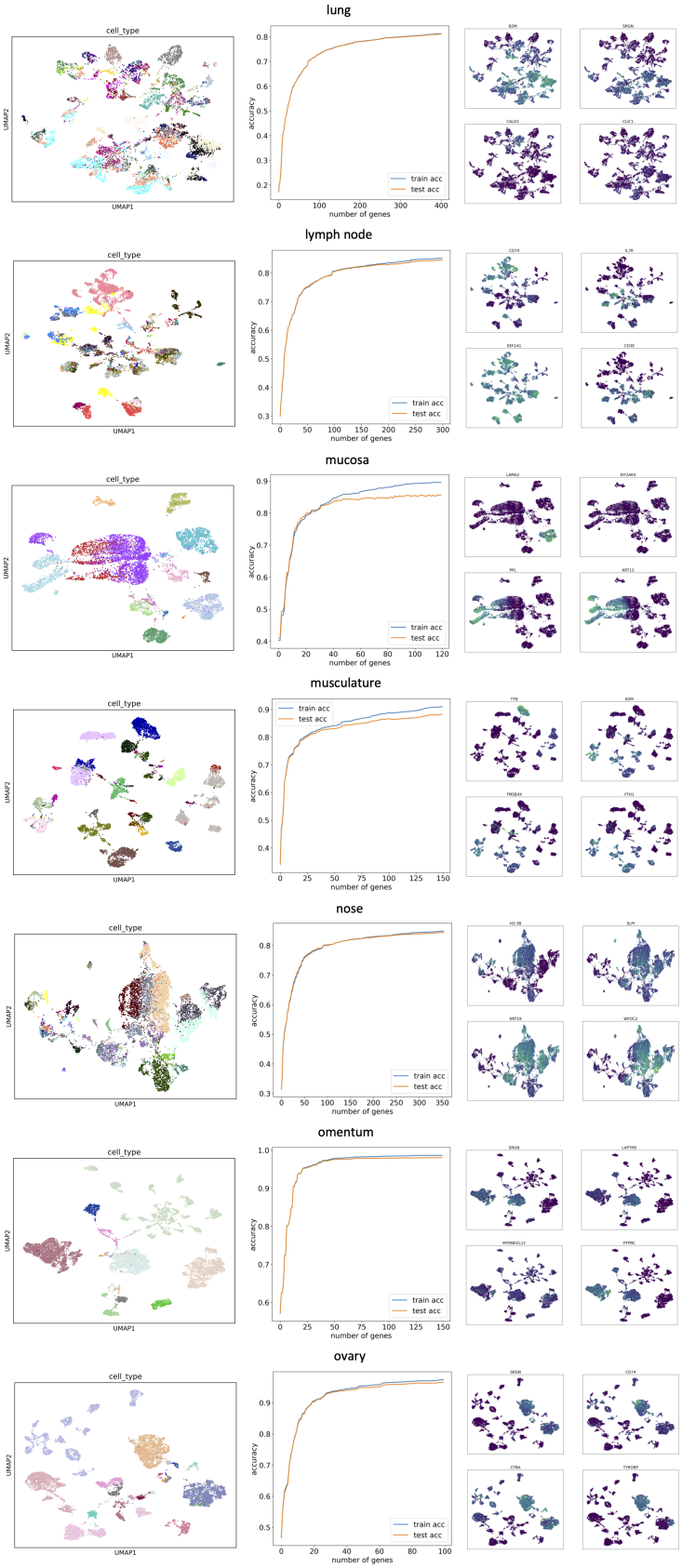


Figure B.5: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

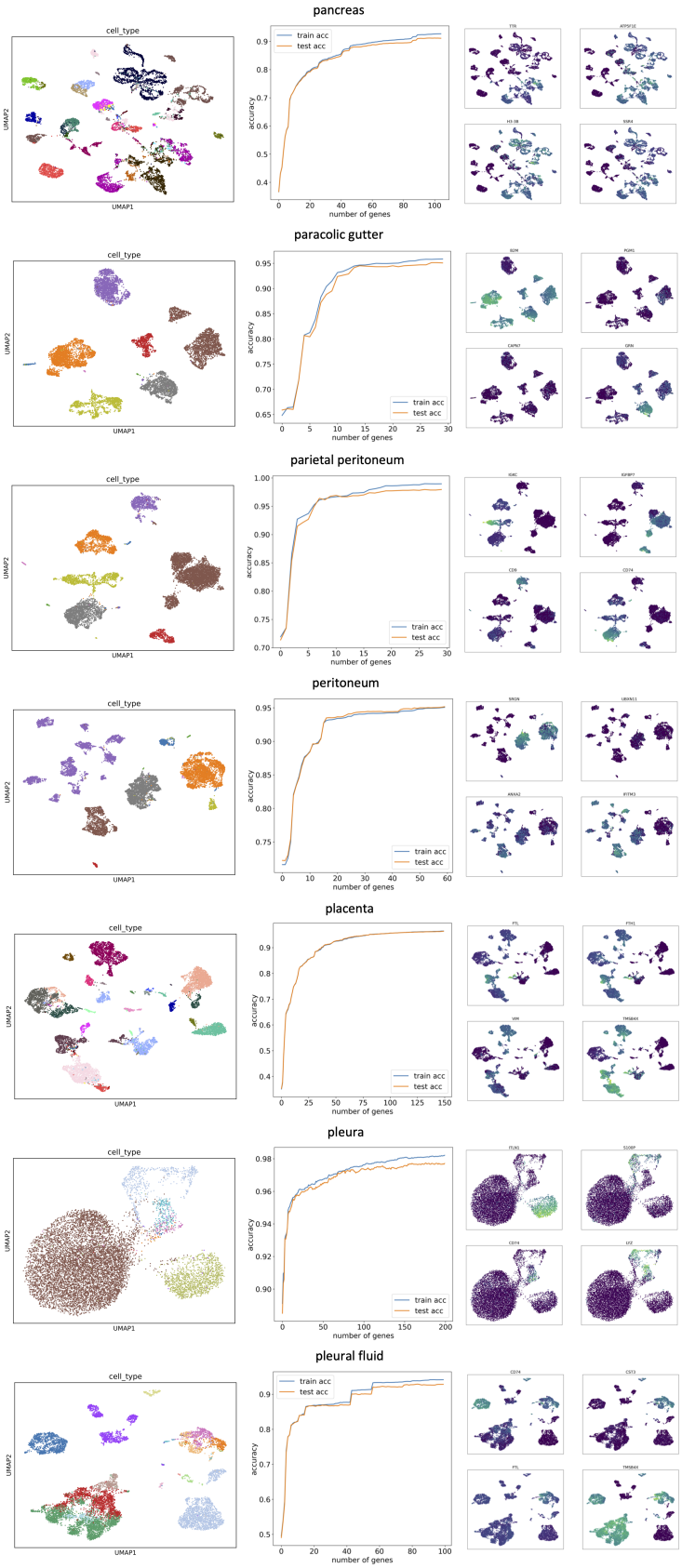


Figure B.6: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

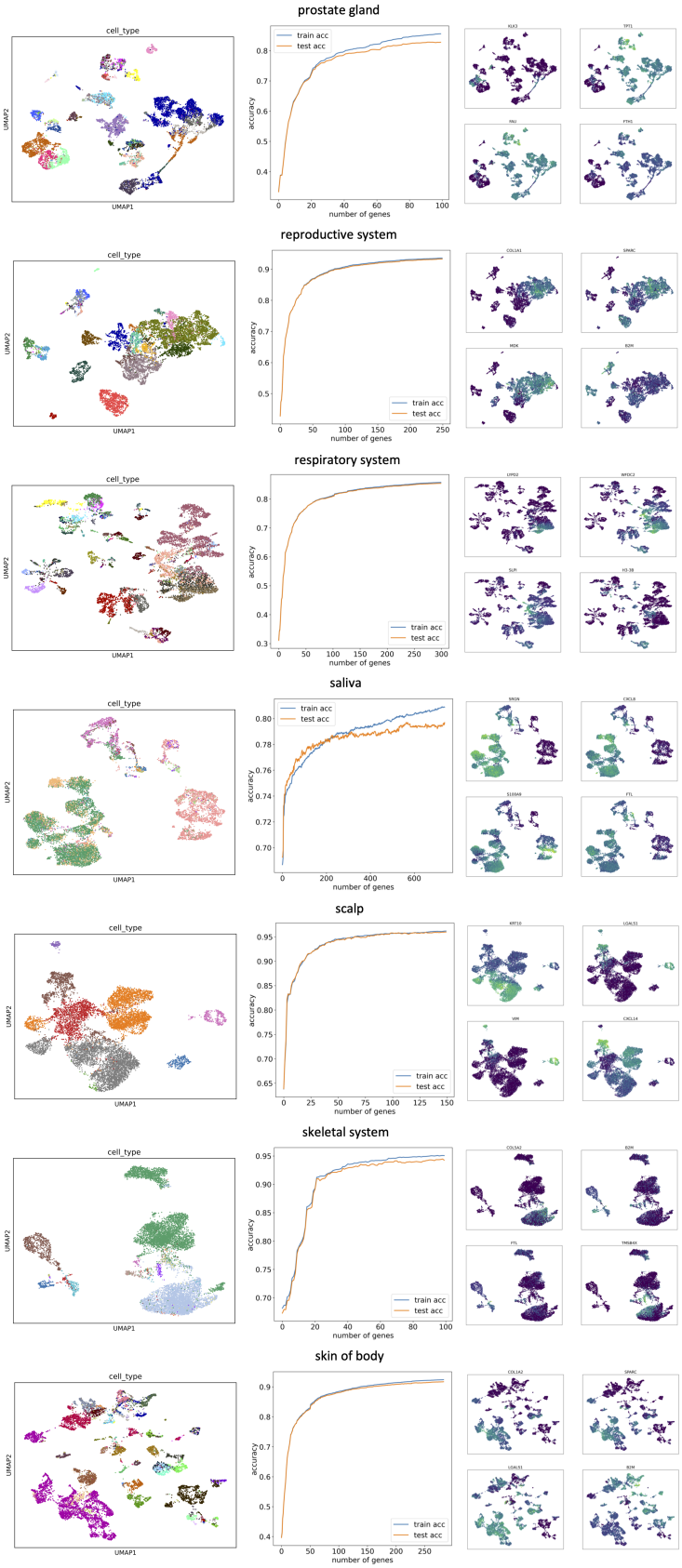


Figure B.7: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

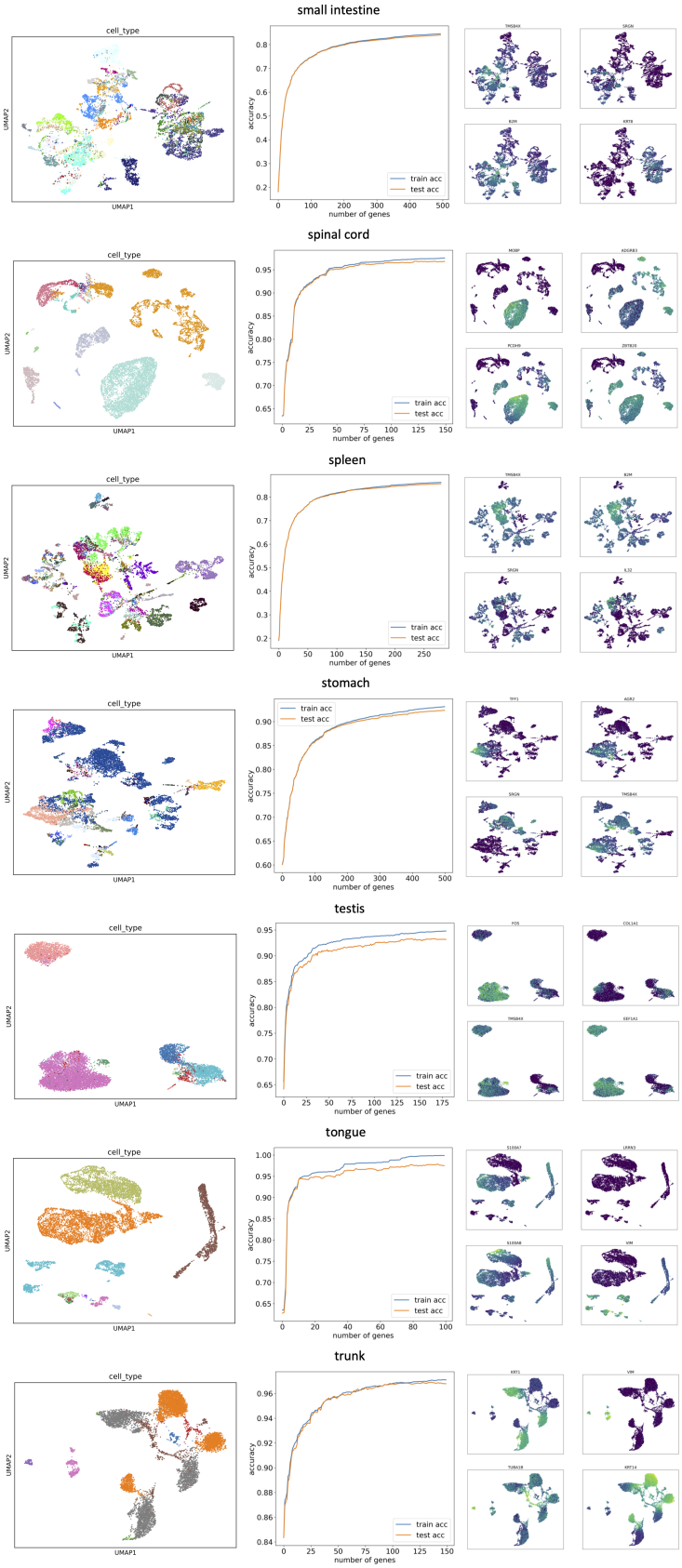


Figure B.8: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

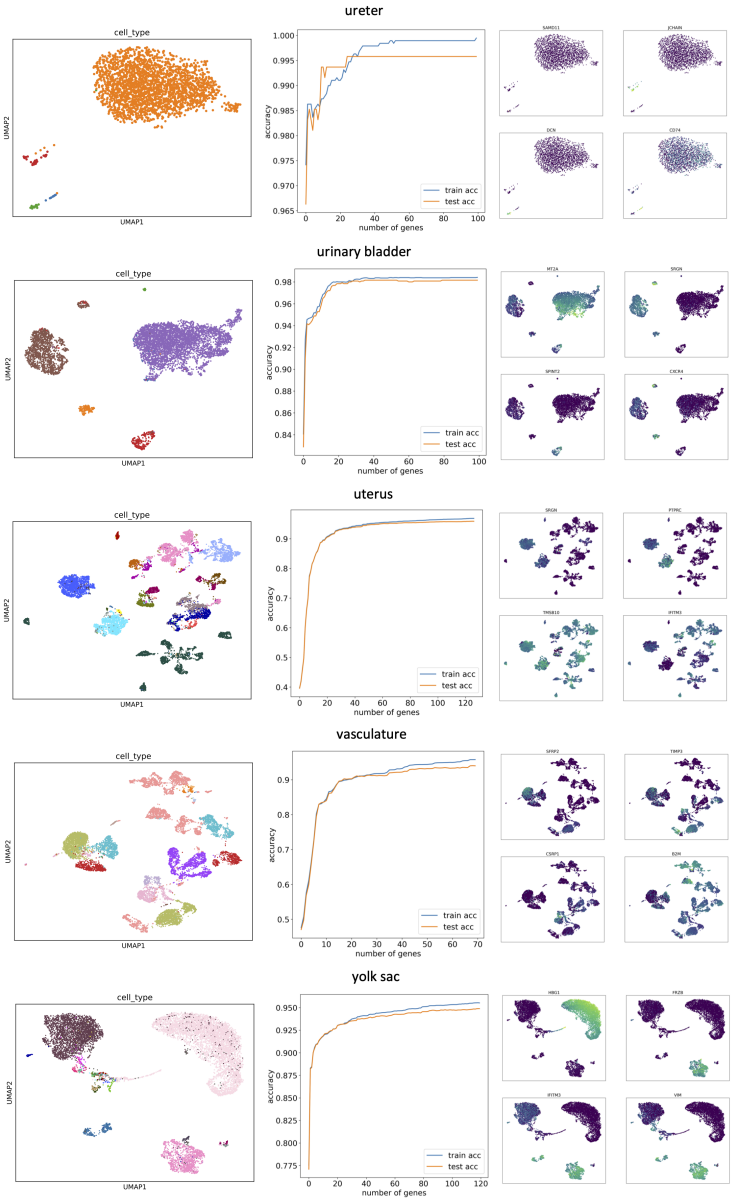


Figure B.9: The cell types umaps (left), the accuracy of ActiveSVM gene set (middle), and the first four genes expression umap (right).

Appendix C

FULL LISTS OF ACTIVESVM GENES SETS FOR HCA DATA

- abdomen: SRGN, CNTNAP1, TYROBP, TCF4, GAPDH, CD63, CD74, IGFBP7, CST3, CD69, B2M, CLU, PTPRC, YBX3, SSR4, EEF1B2, PARP12, NDUFA4, IFITM3, ARHGDIB, FOS, TSC22D3, HS6ST1, HSP90AB1, HPGD, VIM, FTL, SEL1L3, SEC11C, TAMALIN, ANKRD11, ARL4C, SELENOH, CD2, TRABD, BCL11A, NFATC2IP, CD79A, C1orf56, HDAC9, JCHAIN, IGKC, SGSM2, NCF1, C1orf54, SIL1, RAP1GAP2, ADGRG6, BASP1, SCARF1

- abdominal wall: KRT18, JCHAIN, IGKC, TYROBP, FCER1G, CST3, IGHG1, LYZ, WFDC2, COL3A1, GZMB, COL1A1, NEAT1, CXCR4, FTH1, CD79A, GPR183, TIMP1, IGHM, COTL1, IFITM3, COL1A2, CCL5, IGLC2, CD52, MT2A, IFI27, VIM, MS4A1, C1S, MGP, TM4SF1, TPT1, S100A4, TXN, C3, IGHG3, TSC22D3, IL32, CYBA, SPARC, IRF8, TNFAIP2, CD83, TMSB4X, GNLY, SOD2, CD69, KLF2, IDO1, CD63, BASP1, LTB, FOS, TXNIP, CORO1A, MDK, IGFBP7, SLPI, CD74, POU2F2, C15orf48, IER3, EEF1A1, KRT19, PTPRC, JUNB, GSN, CCDC80, JUN, TRAC, HSPB1, BTG1, CLEC2D, PPA1, CXCL10, KLF6, JUND, MT1E, NFKBIA, FYB1, LAPTM5, RNF213, S100A6, SYNGR2, CD79B, NKG7, ATP1B1, RGS1, GBP2, IRF1, FAU, C1R, LCN2, PDLIM1, SKIL, KRT8, CD2, TNFRSF13C, TUBA1B

- adipose tissue: DCN, SERPINF1, APOD, S100A6, H3-3B, FAU, B2M, PTPRC, NEAT1, TAGLN, CD36, ZEB2, UTRN, FRMD4B, RORA, RBMS3, ARHGAP15, IQGAP2, BNC2, COL4A2, TNFAIP3, ANKRD44, NKG7, CHST11, HSPA1A, HBA2, SKAP1, C5AR1, ANKS1B, HSP90AA1, EEF1A1, CLU, SORBS1, TNS3, COX7B, GON4L, ZFP36L2, NRXN1, C6orf62, CALD1, TUBA4A, ABCA6, ATP5PO, COL8A1, LDB2, TPM1, RGCC, FCGR2B, CD53, RAC2, SYNE2, ZNF772, C3, CDR2, TIAM1, HNRNPA2B1, ARMCX3, PFKP, TP53, RGS2, ARHGDIB, CCL4L2, MCTP2, MTRNR2L12, SOX5, SMYD3, SREBF2, USP53, CDC23, RAN, REXO4, ARHGAP30, VWA5A, SPP1, SLC25A6, CALM3, INPP5E, CD48, CRIM1, PRKG1, MTRNR2L6, PABPC1, TMSB4XP4, ADD3, PRM1, PRKCB, S100A8, IN-

SIG1, FAM160B2, EPOR, CNN2, MBNL1, SNHG9, TNF, MARK3, VTRNA2-1, CCL18, RNF19B, CCDC192, PKHD1L1, ELOC, ZC2HC1A, PLPP5, EZR, TFRC, ODF2, RAP1GAP2, DOCK5, FGFBP2, P2RX1, SUPT5H, RNF44, IKZF1, HSD17B11, ADM, VTRNA1-3, TFPI, CHI3L1, LBP, FLJ37453

- adrenal gland: RBFOX1, ATP5F1E, ACTG1, SRGN, EEF1A1, B2M, H19, TMSB10, DNAJB1, FTH1, VIM, TUBA1B, TMSB4X, COL1A2, FTL, CD74, CHGB, COX7C, COL3A1, RACK1, LDB2, ATP5PF, ANXA2, LGALS1, IGFBP7, CALM2, HBG2, S100A4, ACTB, MEG3, IFITM3, CST3, TPT1, PTMA, HSPA1B, CHGA, SAT1, TUBA1A, IGF2, DLC1, APOE, STMN1, SERF2, COL1A1, ZEB2, MYL6, CYP17A1, HBA2, NDUFA4, GAPDH, NPM1, HSP90AA1, S100A10, H3-3B, SPARC, CTSB, FAU, CD36, ZFP36, MYL12A, S100A8, UBA52, GPC6, STAR, HSP90AB1, CD63, TXN, PABPC1, UBB, UBC, RBMS3, GNAS, PFN1, CFL1, ITM2B, PPIA, ATP5MG, SNHG5, ACTA2, CCL2, BTG1, FLT1, EEF1B2, S100A6, LDHB, STMN2, HMGB2, ANXA1, UBL5, GADD45B, COX4I1, ZFP36L1, FOS, DUSP1, H2AZ1, HSPE1, NACA, HMGB1, CALD1, HSPD1, PPIB, SKP1, C1QA, EIF1, NEAT1, HSP90B1, GSTP1, JUNB, HSPB1, CADM1, PSAP, PSMA7, IGFBP5, RBM47, JUN, DLK1, ARHGDIB, TCF4, KLF6, HSPA1A, ZFP36L2, PRDX2, NAP1L1, MT2A, COX7A2, TMEM258, LYZ, HBA1, MBNL1, DCN, ZFAS1, TIMP3, COX5B, CALM1, PRDX1, PCDH9, MGST3, DPP6, PFDN5, NPC2, HSPA8, EEF2, FOSB, THBS1, PLXDC2, OST4, RAN, DNAJA1, MT1G, COX6B1, APP, HNRNPA1, LAPTM5, PEBP1, RGS1, GPX1, MYL12B, SLC2A3, COX6C, EMCN, SRP14, CHCHD2, SNHG29, AIF1, HES1, MTATP6P1, OAZ1, HNRNPA2B1, SLC25A5, HINT1, UQCR10, BEX3, PKM, HNRNPC, YWHAB, LDHA, TMA7, TYROBP, G0S2, S100A11, PLIN2, CXCL8, PLAT, RHOA, TMBIM6, TOMM7, SEC62, COX8A, KCNQ5, ROBO2, DYNLL1, P4HB, ARPC2, NRXN1, TM4SF1, SON, EIF4A2, PDIA6, ARL6IP1, C1QB, IFITM2, HSPA5, HBG1, DDX5, BSG, SOD1, ARHGAP15, FDX1, SEC61B, ARL15, IGKC, ALDOA, ATP5MC2, SUB1, BTF3, TUBB, ARPC3, MIF, COMMD6, SH3BGRL3, HNRNPK, POMP, SOX4, NDUFS5, ENO1, SSR4, TSC22D3, TIMP1, CALR, NDUFB4, SLC25A37, UQCR11, DBI, ROBO1, RGS5, MRPL33, HSBP1, ID2, NUCKS1, CREM, A2M, RBP1, UQCRQ, UQCRH, UQCRB, CYSTM1, IGFBP4, SNRPE, PTPRC, SRSF3, NFKBIA, GAS5, SNRPD2, DHCR24, ITM2C, TENM3, RTN4, CTSD, SLIRP, COX7B, ATP1B3, SAP18, ERH, WSB1, NDUFB1, CALM3, POLR2L, CD164, LAPTM4A, S100A9, CXCR4, RBM39, ELOB, YWHAE, LYVE1,

YWHAZ, QKI, PRKG1, PTP4A1, CD81, ATP5F1B, ATP5MC3, C7, APOA1, MYL9, ATP5ME, RAP1B, CCL4, CLU, CKS2, PRDX3, CLIC1, SOD2, CNBP, PDIA3, EGR1, LSAMP, SLC25A3, PTGES3, TAGLN2, CDKN1C, NDUFA1, FCGRT, SLC25A6, COX6A1, APLP2, TPM4, EEF1D, EIF4G2, PEG10, MGP, APOC1, GPX4, FAM155A, UBE2D3, ATP6V1G1, ATP5MK, NDUFB9, FABP1, PSME1, ID3, MORF4L1, SAMHD1, DDIT4, PF4, EDF1, TRMT112, FKBP1A, CD59, MEF2C, MDK, COX5A, MGST1, HNRNPD, DAD1, SDCBP, SCARB1, MEIS2, HNRNPDL, EIF4A3, SRSF7, CDC42, CBX3, SUMO2, NDUFB2, SPARCL1, GNG5, WDR83OS, ESD, SNX3

- ascitic fluid: CST3, CTDP1, CPT2, TIMP1, IGHM, CD74, MS4A6A, FCER1G, LST1, ARHGDIB, MEF2C, MPEG1, KIT, IGKC, IRF8, CD52, TYROBP, CSDE1, GZMB, CD86, UNC119, IMPA2, CD37, LYZ, EVI2B, SRGN, TBKBP1, PKIB, ABCA7, ALDH2, SFTPD, JCHAIN, PTPRC, TXN, PI4K2A, FLVCR1-DT, VMO1, RHEX, SMOC1, HSD11B1-AS1, CPNE1, SPCS1, CXCR4, MREG, ADCY7, GCSAML, S100A9, RFC5, SPP1, TYMP, CD2, CD163, SYNCRIP, APOBR, DTHD1, ITGAX, IRF7, EIF3F, TNFRSF1B, CD79A, LILRB2, IKZF1, S100A8, ZDHHC18, C9orf139, KIF16B, MMP25, GPRIN3, FTL, NAGPA

- axilla: FAM87B, FAM41C, SAMD11, IGFBP7, FTH1, CD74, IL32, B2M, S100A9, FTL, BTG1, MGP, SRGN, TPT1, VWF, VIM, AIF1, SPP1, CST3, UBB, CCL5, HSP90AB1, CXCR4, SPARC, RACK1, SAT1, IFITM3, H3-3B, MYL6, S100A10, SPARCL1, PCP4, EEF1A1, UBC, JUND, GNG11, EIF1, CALD1, S100A6, HSPB1, PTPRC, HSPA1A, TYROBP, LYZ, FCER1G, S100A4, CALM1, EZR, FKBP1A, TIMP1, BGN, CD52, CCL4, LTB, MS4A1, FOS, HSP90AA1, ADIRF, SOD2, HMGN2, ANXA2, TXN, NACA, CYBA, CD9, CSTB, LGALS1, GAPDH, REL, CD7, ACTB, SOX18, CD63, H3-3A, PSAP, HSBP1, ZFP36L2, CXCL8, CAV1, RAMP3, FABP5, MGST3, TAGLN, ATP6V1G1, DNAJA1, NFKBIA, GNAS, TUBA1B, DAD1, S100A11, CTSB, SSR4, STK17B, ITM2B, NUCKS1, STMN1, BRI3, NR2F2, GPX1, PTMA, HINT1, TIMP3, ID1, A2M, PLAUR, AREG, RGS1, ATP6V0B, IL1B, TYMP, BNIP3L, POU2AF1, ATP5IF1, PTMS, HES6, ACTA2, ATP5F1E, CKB, NDUFS5, GPX4

- bladder organ: EEF1A1, CD81, JUND, SRGN, IFITM3, CD74, NEAT1, TIMP3, SPARCL1, ADIRF, B2M, CST3, MGST2, PSAP, TPM2, ANXA2, CYSTM1, DCN, SH3BP5, GRN, APOLD1, S100P, DDX5, CTSB, PECAM1,

RGL2, ESD, IFI27, AQP1, ATP5ME, MTATP6P1, SLC25A6, SOX18, TCIM, GLUL, PFN1, ACTA2, SPINK1, CD52, CAPG, AGR2, SPTBN1, HES1, TM4SF1, HSPB7, FTLP3, TSC22D1, LAPTM5, ESAM, FGD5, SPIB, TPSB2, MS4A6A, ELF3, PLVAP, CTD-2287O16.1, CSTA, CTSS, TMSB4X, PLN, SERPINA1, GADD45B, FCER1G, SELE, IGHM, RAPGEF4, S100A2, C1orf162, WDR83OS, FLT1, MYH7, ATXN3, MYL12A, INPPL1, ADCY10P1, CD99, NAPSB, PCAT19, AGGF1, TNFSF4

- blood: CD74, CD3D, FCER1G, TMSB4X, TYROBP, B2M, LTB, SNHG29, EEF1G, NKG7, H3-3A, MTRNR2L12, IL32, S100A4, TXNIP, DUSP1, S100A9, H3-3B, BTG1, CCL5, CD52, IFITM1, LYZ, EEF1A1, CD8A, AIF1, FOS, ALDOA, CRIP1, LGALS1, COTL1, S100A8, CYBA, ACTB, CST3, IL7R, JUND, GAPDH, FTL, ARID5B, NEAT1, CD8B, PTPRCAP, CD99, TMSB10, FTH1, GNLY, JCHAIN, JUN, VIM, ANXA1, EIF4A1, KLF2, ZFP36L2, HNRNPA1, GAS5, TLE5, IFITM2, LSP1, ITGB1, NFKBIA, NME2, CD3E, JUNB, TAGLN2, PLAC8, PPBP, TUBA1B, IFITM3, CXCR4, SRGN, GABARAP, MTRNR2L8, H2AZ1, NOP53, SEPTIN7, SAT1, SELL, KLF6, TSC22D3, GZMB, HSP90B1, HSP90AB1, PTPRC, EMP3, LDHB, HCST, S100A10, PNRC1, CD37, AHNAK, TPT1, ZFP36, GZMA, SLC25A6, ENO1, CD69, EIF3E, S100A6, LAPTM5, ZFAS1, SARAF, PABPC1, HINT1, ID2, HSPA8, OST4, EEF1B2, SUB1, TRAC, YWHAZ, IL2RG, DDX5, OAZ1, MYL12A, UBC, YBX1, TNFAIP3, IGKC, IER2, ISG20, FYB1, MT2A, ATP5F1A, ITM2B, NAP1L1, CTSW, DUSP2, PPDPF, ACTG1, RAC2, CCNI, PCBP2, HSPA5, HNRNPC, ZFP36L1, NCL, PNISR, PSAP, UBB, IRF1, LY6E, MIF, HMGB1, HNRNPU, HSP90AA1, FXYD5, FLNA, TUBA1A, ATP5F1D, GSTP1, TPM3, PCBP1, HBB, COMMD6, ALOX5AP, PTMA, RNASEK, CALR, UQCRB, CD7, PPIB, KLRB1, ARPC1B, CD48, ARL4C, EEF2, STK17B, CORO1A, TOMM7, ANP32B, PLAAT4, COX5B, UQCRH, C12orf57, ISG15, S100A11, TRBC2, LDHA, RAP1B, GMFG, HMGN1, H1-4, MBNL1, RBM39, RNF213, EZR, MYH9, SSR4, IGHM, CST7, HMGN2, SF1, COX7C, HNRNPA2B1, FAU, SH3BGRL3, UCP2, SRSF5, GNAS, ITGB2, ANKRD12, PPIA, ARHGDIB, CALM2, CLIC1, PRRC2C, ARPC2, MYL6, GPSM3, RSRP1, CDC42, GPX4, SLC2A3, EID1, LIMD2, EIF1, SEC61B, CALM1, SOD1, SMCHD1, PSME2, SAMHD1, CHCHD2, NPM1, TCF25, H4C3, TPI1, ARL6IP4, H1-10, TMA7, GSTK1, PPP1R15A, SEPTIN9, PRR13, MORF4L1, SON, GADD45B, PKM, PSME1, PSMB9, CD63, CD44, CCND3, UQCR11, TGFB1, ATP5F1E, PDIA3,

PARK7, RNASET2, MCL1, GUK1, FUS, MTDH, PRDX5, ARGLU1, RIPOR2, LCP1, CCNL1, DDX24, CD47, UBE2D3, EIF3K, NACA, XBP1, ANAPC16, ACTR3, EIF5A, CIRBP, PTP4A2, DDIT4, SRSF7, IFI6, LITAF, CELF2, JAK1, TMEM123, DBI, VAMP8, ATP5MK, NDUFA4, ERP29, ITGA4, SKP1, BTG2, SYNE2, FOSB, SERBP1, CSDE1, SSR2, SERF2, SERP1, CNN2, CYTIP, RHOA, COX7A2, SRRM2, DAZAP2, TRBC1, KMT2E, RSL24D1, RGS10, HCLS1, HNRNPDL, COX4I1, GYPC, ELOB, YWHAB, NOSIP, DDX17, GDI2, AKAP13, SNHG5, UXT, ATP6V0C, LEPROTL1, ATP5MC2, COX6C, TXN, PFN1, TAPBP, RAC1, TUBB, SLC25A3, TRIR

- bone marrow: HBG1, HBG2, SLC25A6, B2M, CST3, EEF1A1, CD74, LYZ, AIF1, NKG7, STMN1, CD52, CD63, SRGN, DUSP1, CD99, FTH1, TMSB4X, FOS, S100A4, CXCR4, MPO, S100A8, NPM1, S100A6, TUBA1B, VIM, BTG1, S100A10, PTMA, SNHG29, ZFP36L2, TYROBP, HBB, SELL, GAS5, IGHM, HMGB1, LGALS1, HSP90AB1, EEF1G, ACTG1, ACTB, HSPA1A, FTL, IL32, TAGLN2, H2AZ1, TMSB10, CD37, HSP90AA1, JUND, CYBA, H3-3A, HCST, MTRNR2L12, JUN, COTL1, H3-3B, GAPDH, IGLL1, HSP90B1, IGKC, IL7R, ENO1, S100A9, S100A11, SOX4, DUSP2, IFITM2, GSTP1, GPX1, DDIT4, FAU, FCER1G, YBX1, ITGA4, NEAT1, ZFAS1, HMGB2, TPT1, H4C3, CD8A, PRDX2, TXNIP, SAT1, ANXA2, CD44, JUNB, LTB, ARHGDIB, HERPUD1, CST7, CD69, EEF1B2, NFKBIA, ITGB2, SNHG6, KLF6, UBC, HINT1, PSAP, HSPA1B, TUBB, ALDOA, PLEK, PLAC8, HMGN2, MACROH2A1, CCL5, BTG2, ANXA1, ZFP36L1, SNHG32, RACK1, CALM1, MIF, SEC61B, HBA2, IRF8, TFRC, PTPRC, VCAN, ATP5F1E, LAPTM5, CD164, CD81, YBX3, XBP1, RCSD1, ZFP36, CTSW, RSL1D1, SNHG8, ITM2C, HNRNPA1, H1-10, OAZ1, PABPC1, ITGB1, IER2, MTRNR2L8, DNAJB1, GNLY, HSPA8, NAP1L1, TUBB4B, IFITM3, HSPB1, CLIC1, SERF2, CALR, BLVRB, EEF2, CD24, AREG, ANP32E, LSP1, CD47, ATP5IF1, NACA, CSF3R, AZU1, IGFBP7, FXYD5, CORO1A, HSPA5, ITM2A, LDHA, MSI2, SLC3A2, NUCB2, SH3BGRL3, H1-2, CLC, TSPO, UBA52, ZEB2, ARPC3, PCNA, GZMA, PPIA, DDX5, PPIB, TSC22D3, RETN, TRAC, UBB, EIF3E, GNAS, KLF2, TALDO1, LBR, HNRNPA2B1, CTSS
- brain: ADGRL3, DLG2, CNTNAP2, PTPRD, NPAS3, RACK1, NRXN1, NEAT1, EEF1A1, TUBA1A, FTH1, B2M, PLXDC2, RORA, ERBB4, SLC1A3,

PTPRZ1, MEF2C, PTPRK, NFIB, QKI, GPM6A, LRP1B, CELF2, NLGN1, TMSB10, FTL, ROBO2, APOE, ACTG1, NTM, NFIA, RBMS3, CST3, VIM, TMSB4X, ROBO1, LSAMP, PTN, MAP1B, SAT1, GRID2, CALM2, TCF4, RASGEF1B, MAML2, KCND2, VCAN, SPARCL1, SPP1, TUBA1B, NRXN3, XIST, CD74, PLCG2, ZBTB20, GPC6, UTRN, CLU, ATP5F1E, SNHG14, PRKG1, IGFBP7, FOXP1, HMGB1, RBFOX1, ACTB, SEPTIN7, CADM2, SOX5, MEG3, GRIK2, PHACTR1, PCDH9, SLC8A1, H3-3B, PSAP, MEIS2, KCNIP4, HSP90AA1, AUTS2, ZEB2, EVL, LHFPL3, GAPDH, FRMD4A, CALM1, ALCAM, BTG1, CPE, BASP1, SOX4, CD63, S100A6, DLC1, DOCK4, THSD7A, PDE4B, LRRTM4, LRRC4C, UBC, NACA, DPP10, FOS, CNTN5, ANKRD12, GNAS, ADARB2, DDX5, RTN4, HNRNPA2B1, GRIA4, TPT1, TSC22D1, CTSB, FAU, STMN1, SYT1, DSCAM, ITM2B, GALNTL6, HSPA1A, GSTP1, MT2A, GPM6B, JUND, H3-3A, PTPRG, HSP90AB1, CYRIB, MBNL1, CFL1, PDE4D, HNRNPU, PNISR, RELN, WSB1, DDX17, FABP7, PTMA, JMJD1C, JAK1, PDE1A, PRKCA, NXPH1, NKAIN3, MLLT3, PTPRM, ZEB1, JUN, ELMO1, DST, CTNNA2, STMN2, MACF1, DLGAP1, SRGN, CADM1, SEC62, BPTF, AGAP1, UNC5C, TENM2, SPARC, TRPM3, NUCKS1, ANK2, SFPQ, HNRNPDL, NAP1L1, RTN3, APP, PKM, GLUL, CCNI, APLP2, CTNND2, NRG1, ZNF385D, YWHAE, DCC, IGKC, NKAIN2, MAGOH, LPP, CCDC88A, CACNA2D1, PPP3CA, NCAM2, NDUFA4, CKB, CSMD1, RALYL, MYL6, EEF2, FXYD6, DYNLL1, TCF12, HSP90B1, NAV3, PTGDS, AKAP9, AKAP13, RTN1, CHST11, N4BP2L2, YBX1, GNAQ, CD81, SRGAP1, ZNF804A, FYN, PABPC1, UBB, SYNE2, HNRNPH1, DPP6, TTC3, CIRBP, SEM1, SGCZ, EIF1, EIF4A2, CHD9, PPIA, SON, KMT2E, RAC1, YWHAB, MARCHF1, NELL2, JUNB, SEC61G, SGK1, HNRNPC, CNTNAP5, NRIP1, COX4I1, EEF1D, DPYSL2, MSI2, BICD1, MAP2, CALD1, PRKCB, S100A9, PTPRN2, CUX1, SERF2, PRDX1, ID2, TMEM14B, LUC7L3, TTC28, HDAC9, FUS, C11orf58, NRG3, MYCBP2, RBM39, ANK3, MTCO2P22, GNB1, DBI, BEX3, SNTG1, GRIA2, H4C3, CCND2, KAZN, PBX1, YWHAZ, HMGN1, ATP2B1, COX6A1, SRSF11, PRRC2C, RNF130, MEF2A, C1QB, DMD, NNAT, PEBP1, UBA52, FTX, PCDH15, PAM, OAZ1, MAGI2, NAV2, TNRC6A, FOXN3, ATP5ME, DNAJB1, PPP2R2B, HSPA8, PTK2, SMYD3, CAPZB, CLASP2, SYNE1, ARID1B, ZFAND3, ARL6IP1, RHOB, MED13L, EGR1, FGF14, TMBIM6, ATP5MC2, TNRC6B, SRSF5, ANKS1B, SOX2-OT, CHL1, ATRX, ATP1B1, BAZ2B, CTSD, FABP5, CEP170, CALR,

PSMA7, KCNH7, EPS8, RBPJ, MORF4L1, CCL4, SARAF, CAMTA1, UNC5D, KIDINS220, RUNX1T1, HNRNPA1, EIF4G2, LDLRAD4, TXNIP, PLCB1, SRP14, SELENOW, FMNL2, NOVA1, BTF3, MTSS1, OSBPL8, ARID4B, GRIK1, RERE, BSG, LGALS1, XRCC5, TMTC2, LINGO1, GPC5, KTN1, FAM155A, SOX6, TUBB2B, HSPB1, APOD, OXR1, TOMM7, PITPNC1, ZFP36L1, PFDN5, IL1RAPL1, ARGLU1, PHIP, SETBP1, SKP1, KIRREL3, UCHL1, TIMP1

- breast: CD24, SRGN, CYBA, CD74, B2M, IGFBP7, TM4SF1, PTMA, EEF1G, IL7R, CST3, FTH1, CCL5, ANXA1, TMSB10, S100A4, TPT1, JUN, CD44, JUND, SAT1, MT2A, HSPA1A, VIM, PTPRC, ACTB, SOD2, CALD1, TXNIP, SPARCL1, MGP, ZEB2, FABP4, CEBPB, S100A10, MTRNR2L12, BTG1, EEF1A1, JUNB, TMSB4X, NEAT1, NFKBIA, FTL, PABPC1, DUSP1, APOD, KLF6, TYROBP, IL32, LGALS1, FOS, ZFP36L2, GAPDH, H3-3B, SNHG29, HSP90AA1, TCF4, RGS1, PLCG2, ARID5B, IGKC, S100A11, CEBPD, EEF1B2, ID2, HSP90AB1, DCN, KLF2, CXCL8, CCL4, ACTG1, KRT17, REL, FAU, HMGB1, GNLY, S100A6, ITM2B, LMNA, TXN, ANKRD28, TSC22D3, LDHA, RACK1, UBA52, H2AZ1, EIF1, IFITM2, C11orf96, PLIN2, DDX5, CXCR4, TUBA1B, LRRFIP1, IER2, LAPTM5, CREM, TPM1, HERPUD1, EMP1, FOSB, FABP5, ITGB1, ISG20, HSPA1B, TIMP3, H3-3A, EZR, UBB, NAMPT, SQSTM1, UBC, TSHZ2, ZFP36, YBX3

- colon: SCHLAP1, PIGR, LGALS4, IFITM3, SRGN, CD74, TMSB4X, GAPDH, CST3, VIM, FOS, B2M, FTH1, RACK1, S100A6, LGALS1, H3-3A, CYBA, S100A10, PHGR1, TXNIP, HSPA1A, EEF1A1, TPT1, TFF3, HNRNPA1, IL32, ACTB, JCHAIN, TPM1, NEAT1, TMSB10, BTG1, MTRNR2L12, HSP90AA1, JUND, MARCKSL1, SAT1, KLF6, JUNB, MT2A, FTL, JUN, EEF1D, FABP5, HSPB1, TSPAN8, H4C3, CD63, PFN1, TUBA1A, H3-3B, SLC25A6, CD24, PABPC1, S100A11, GSTP1, KRT8, PTMA, ATP5F1E, IGHA1, EEF1B2, ID2, ZFP36, SH3BGRL3, MDK, LGALS3, TUBB, PPDPF, OAZ1, HMGN2, NPM1, IGFBP7, UBA52, ACTG1, ANXA1, FABP1, UBC, ITM2B, KRT18, ATP5MK, EEF2, HSPA8, ATP5MG, CD9, UBB, ZFP36L1, S100A4, GNAS, FOSB, TUBA1B, TAGLN2, HSPA1B, DUSP1, COL3A1, TXN, CALM2, HSP90AB1, LTB, IFI27, UQCRB, ATP5MC2, FAU, SSR4, H2AZ1, CFL1, CD52, LYZ, MYL12A, MARCKS, DDX5, ZFP36L2, LMNA, ANXA2, IER2, NACA, MYL6, NFKBIA, CKB, SERF2, CCL5, STMN1, FOXP1, HMGB1, HERPUD1, C15orf48, YBX1, PNRC1, RGS1, CALM1,

MUC2, HINT1, BTG2, COX7C, ARPC3, COX8A, ID3, ALDOA, DNAJB1, YWHAZ, EIF3K, EIF1, DBI, SELENOP, COX4I1, TOMM7, PFDN5, IRF1, HES1, IGKC, PSAP, HNRNPA2B1, LITAF, RBM3, HSP90B1, UQCRH, ATP5MJ, DSTN, RAC1, CXCR4, HMGB2, MIF, SRSF7, CXCL14, GNG5, TYROBP, HSPA5, FXYD3, YWHAB, LDHA, ARPC2, ARPC1B, SRP14, COX7B, FCGBP, GSN, CCNI, CSTB, SLC25A5, BTF3, CYCS, LDHB, KLRB1, HSPE1, SRSF5, SOX4, IGLL5, CDC42, TIMP1, PPIA, EIF4A1, MT1G, ENO1, ATP5MC3, HNRNPH1, UQCR11, RBFOX2, ITM2C, PEBP1, COX5B, NOP53, EGR1, DYNLL1, RHOA, HMGN1, EPCAM, NDUFA4, ZG16, SELENOW, SOD1, NAP1L1, FKBP1A, COX6C, NCL, ARHGDIB, EIF4A2, OST4, IFITM2, KRT19, COTL1, COX6B1, CLIC1, RAN, CIRBP, SKP1, EIF3E, TPM4, CD99, CALR, PSME1, GPX1, SLC12A2, PNISR, ATP5F1D, ZFAS1, MCL1, TMA7, HNRNPK, CHCHD2, NDUFA1, SARAF, TPI1, VAMP8, TNFAIP3, TMBIM6, SRSF2, SSR2, CTSD, DDX17, SET, SUB1, ELOB, SLC25A3, UBE2D3, HSPD1, GADD45B, MYL12B, ATP5ME, PCBP2, SUMO2, PLA2G2A, IL7R, HNRNPA3, COMMD6, ATF3, COX6A1, RBM39, POLR2L, GUK1, SOD2, DAZAP2, AGR2, SEC61B, TSC22D3, HNRNPDL, ID1, PPP1R15A, SNU13, ACTA2, COX5A, HNRNPU, SERBP1, CAPZB, GZMA, ATP5IF1, PRDX5, PGK1, AKAP13, HNRNPC, COX7A2, CD44, NBEAL1, CHCHD10, EZR, MICOS10, SON, FUS, TMEM59, NDUFA13, EIF3D

- digestive system: COL3A1, ACTA2, S100A10, IGFBP7, CALD1, MDK, RAMP2, COL1A1, B2M, ZFP36, MEIS2, TPM1, FOS, TUBA1A, COL1A2, FTL, KLF2, SFRP1, PLAT, MCAM, COL4A1, ACTG2, FABP1, IER2, MEST, H2AZ1, MAGED2, IGFBP2, TUBB2B, SEPTIN7, MEF2C, HSPA6, SPARC, HNRNPA1, TMSB4X, CST3, TUBA1B, IGFBP5, EEF1A1, HSPA5, AKAP12, VIM, GADD45B, CENPF, C11orf96, TMSB10, HSPA1B, SAT1, NPM1, HMGB2, CD74, TAGLN2, CXCL12, EGFL7, IRF1, FABP5, IGFBP4, MEG3, PHOX2B, HSPA1A, BEX3, LMNA, JUN, PTMS, KCNQ1OT1, AIF1, TUBB, ID3, PHGR1, MT1H, NEAT1, TSHZ2, H3-3B, CCN1, LGALS1, RGS16, CCN2, ID4, TCF4, PPP1R15A, WSB1, TFPI, PRDX2, MT1G, TAGLN, DDIT4, STMN1, HES4, TOP2A, PLD3, EGR1, TPT1, JUNB, SPARCL1, CD81, LGALS3, COL4A2, EIF3E, SRGN, CTSL, FTH1, COL6A3, HSPB1, EIF5, HNRNPDL, EEF1B2, ATP1B1, HMGN2, UBC, HMGB1, DSTN, HAND2, ZFP36L1, ZFP36L2, S100A11, CCL2, SUMO2, COX4I1, HBG2, MYLK, MARCKS, PFN1, UBA52, NR2F2, OAZ1, PNRC1, NEDD8,

DDX24, TUBB4B, NTS, PDE5A, LRRFIP1, TCEAL9, BTG1, HSP90AB1, HSP90AA1, ATF3, PBX1, PARK7, CCL4, DDX17, CKB, LAPTM4A, ZEB2, TCF21, SERTAD1, EEF1D, NASP, PLP1, FOSB, BSG, CD24, BPTF, RHOA, IRF2BP2, ELAVL4, JUND, CEBPD, TIMP1, MYL9, TMSB15A, PFDN5, MFAP4, KRT8, ELOB, MYH9, PTMA, TYMS, GPC6, PA2G4, OGN, UQCR10, CALM1, H4C3, CNN3, FST, COL6A2, RAC1, ACTG1, CD99, GEM, HSPE1, LTBP4, DUSP2, ITM2C, ATP5F1E, ACTB, EIF1B, H1-0, KLF6, UBE2S, XBP1, DNAJB1, HBA1, BTG2, BASP1, YWHAQ, MAP1B, BGN, ASPN

- embryo: DLX6-AS1, TMPRSS15, MANSC1, TTTY10, C1QTNF5, TPRG1-AS1, PTPRN2, ENC1, SCNN1G, SOX18, DCN, LGALS3, MAGOH, TTTY14, STK38, TAGLN, FABP7, CCND2, LYVE1, CXCL14, ACTA2, AQP1, DEFA3, VTN, WNT2, GPM6A, ZNF302, UGCG, RCN3, ARL4A, ZNF783, LUM, UBB, RFC2, STK36, TGFBR3, HBA1, HSD3BP5, FERMT1, PEBP1, TJP1, SGCA, ESYT2, MPO, XYLT2, RAB27B, DES, FBRSL1, ANXA5, TUBA3D, HGF, ASAH1, NFKBIL1, VWF, HBG2, PDE9A, SOX10, MFSD14B, CDV3, SP140L, CCN2, NMRAL1, HNRNPDL, LGR5, KRIT1, APOD, PIN1, HOXA11, DEK, OLFM3, DHCR24, SPARC, GTF2IRD1, BCAN, COL1A1, MAP2K5, RBP1, CD63, MIF-AS1, CTNNAL1

- endocrine gland: CD8B, CD3E, CD3D, AIF1, CD74, SRGN, BTG1, STMN1, CD52, CXCR4, S100A4, B2M, JUND, EEF1A1, VIM, DNAJB1, LGALS1, ACTB, NEAT1, IGKC, CD99, IFITM2, IL7R, IL32, CST3, ZFP36, HSPB1, GAS5, ARHGDIB, FOS, S100A6, TXNIP, IGFBP7, CD7, MTRNR2L12, EEF1B2, PABPC1, FTH1, SOX4, ITM2A, HSP90AA1, LAPTM5, TMSB10, HSPA1A, TMSB4X, SLC25A6, ZFP36L2, ANXA1, CYBA, FTL, JUN, TRBC2, MZB1, MT2A, H4C3, LTB, GSTP1, LEF1, LDHB, JUNB, H3-3A, NFKBIA, S100A11, ID3, ZFP36L1, ARPC3, IFITM3, DUSP1, RGS1, IFITM1, ZFAS1, TRBC1, TPT1, ID2, TUBA1B, TSC22D3, ALDOA, GPX1, PTPRC, SNHG6, MIF, GAPDH, GPR183, CORO1A, SAT1, HSPA1B, IER2, TCF7, LSP1, EEF2, KLF6, S100A10, ACTG1, H2AZ1, TUBA1A, SPARCL1, GNAS, LDHA, CCL4, IGLC2, XIST, TOMM7, CALR, SARAF, PSAP, HCST, EMP3, PLAC8, DDIT4, NR4A1, HMGA1, MYL12A, DDX5, FOSB, CD63, RGS2, UBB, SATB1, KLF2, SELL, CD37, BTG2, CD69, ITM2C, LYZ, ITM2B, HSPA5, MYL6, CD1E, HSPA8, HMGB1, OST4, SH3BGRL3, NUCB2, TAGLN2, ATP5F1E, HSPE1, TUBB, PRDX2, SOD1, NACA,

GADD45B, HMGN2, FYB1, HSPH1, H3-3B, IL2RG, UBC, PTMA, PPDPF, CLIC1, HSP90B1, UQCRB, TXN, TNFAIP3, MTATP6P1, GLUL, NR4A2, CD44, CALM2, GYPC, TRAC, BIRC3, CALM1, ENO1, COL1A1, PNRC1, SNHG29, SLC2A3, FABP5, PFN1, UBA52, RGS10, JCHAIN, NPM1, YBX1, DNAJA1, HNRNPA1, HSP90AB1, BTF3, HNRNPA2B1, PPIA, PCBP2, FXYD5, DUSP2, YWHAZ, EVL, SRSF5, TPI1, C12orf57, HNRNPDL, HMGB2, CD3G, NCL, LIMD2, GMFG, PPIB, MGP, NOP53, ID1, DYNLL1, FKBP5, TIMP3, NAP1L1, ARL4C, COTL1, LCP1, SOD2, CCL5, EIF3F, VAMP8, EZR, CREM, GNG5, POMP, PFDN5, TSC22D1, CCL2, SUB1, SEPTIN7, MBNL1, TCF4, MARCKSL1, COX7C, EIF3E, FOXP1, PSMA7, MYH9, ELOB, SFPQ, CIRBP, CELF2, PKM, ANXA2, REL, SRP14, PRDX1, ZNF331, TPM4, MS4A1, GSN, GUK1, HSPD1, HINT1, IFI16, CSTB, EIF4A2, GPX4, SERF2, HBG2, CFL1, PCBP1, AREG, NDUFB2, HNRNPC, SON, MCL1, ATP5ME, HERPUD1, FUS, ARPC1B, TIMP1, PSME2, COX7A2, ATP5PO, PRDX6, CD2, RAN, SRSF7, DBI, COX6C, LRRFIP1, PPP1R15A, TMBIM6, GIMAP7, OAZ1, WSB1, CYCS, TYROBP, PTP4A2, ISG15, ELF1, COMMD6, BCL11B, TFDP2, EIF1, DEK, MDK, NDUFB1, CDC42, SNHG8, EGR1, IRF1, RAC2, LY6E, HNRNPM, SH3BGRL, PARK7, SRSF2, TRA2B

- esophagogastric junction: TFF1, CLU, CD74, CST3, B2M, LYZ, TMSB4X, S100A4, SRGN, KRT19, LIPF, CXCR4, FTL, SLPI, ACTB, PCSK1N, IGHA1, CD69, CD9, MT2A, SPARCL1, EEF1A1, CCL5, CYSTM1, KRT13, HBB, KRT8, NEAT1, FTH1, PSCA, GHRL, ANXA1, SSR4, RGS1, IGFBP7, S100A9, AGR2, PTMA, CYBA, JUN, BTG1, CCL4, S100A8, PGC, CSTA, VIM, SAT1, FABP5, S100A2, ITM2B, S100A6, IL32, TMSB10, RACK1, IGLC2, KLF6, CLDN5, PERP, MT1X, ID2, TPT1, HSPB1, KRT14, SLC25A6, S100P, ANXA2, JUNB, TFF2, S100A11, KRT15, PIGR, ADIRF, UQCRB, RHCG, ZFP36L2, COX4I1, KLF2, S100A10, OAZ1, ZFP36, CD24, SLC25A5, ARPC2, LCN2, LGALS3, EEF1D, ATP5MG, CSTB, COX7A2, GAPDH, TCN1, SPINK1, HSPA1A, TXNIP, LGALS1, NFKBIA, S100A14, CD52, GSTP1, ALOX5AP, HSPA8, ID3, GADD45B, ZFP36L1, TACSTD2, PTPRC, FOS, DUSP1, VAMP8, RNASE1, PFDN5, AREG, EIF1, CD63, FXYD3, SPRR3, AQP3, WFDC2, IFI27, HSP90AA1, FAU, SFN, LY6E, IGLC3, UBC, CCNL1, KRT6A, TM4SF1, HERPUD1, MT1G, CIRBP, KRT5, CAPG, ACTG1, BTG2, TMEM59, PPDPF, SRSF7, CITED2, IER2, COX6A1, IFITM2, ENO1, TSC22D3, KRT4, CA2, MT1E, H3-3B, DNAJB1, MYL12A

- esophagus: KRT13, S100A14, TMSB4X, B2M, EEF1A1, VIM, S100A6, LYZ, GAPDH, FAU, CD74, TPT1, NEAT1, HSPB1, S100A8, ANXA1, FTH1, ACTB, EMP1, SPARCL1, FOS, SRGN, H3-3A, EIF1, KRT19, FTL, SAT1, LPP, TXNIP, ZFP36, DUSP1, CST3, S100A2, IGFBP7, BTG1, TMSB10, S100A4, RACK1, TSC22D3, MT2A, UBA52, ITM2B, NFKBIA, PTMA, CD63, CYBA, H3-3B, TFF1, UBC, GSTP1, JUNB, ZFP36L1, ID2, S100A10, CXCR4, CLU, DDX5, ZFP36L2, IGKC, IFITM3, SERF2, JUN, PTPRG, ACTG1, AGR2, FKBP5, S100A9, HSP90AA1, GSN, SSR4, TPM1, SLPI, NACA, MYL6, FABP5, TIMP3, HSPA1A, FOXP1, CFL1, IER2, HMGB1, KLF6, LGALS1, TMA7, PRKG1, TFF3, MYH11, TSHZ2, HNRNPA1, EEF2, PABPC1, S100A11, CCL5, SRSF5, SLC25A6, UBB, ID1, EEF1B2, TUBA1B, DDIT4, CD9, XIST, PSAP, MEF2C, CALM2, ANXA2, MBNL1, CALD1, ATP5F1E, IGHA1, TIMP1, WSB1, OAZ1, NAP1L1, IFI27, SH3BGRL3, GADD45B, A2M, SLC2A3, RAB11A, MT1E, HNRNPDL, RBPMS, RGS2, SCGB3A1, MYL12A, APP, TPSB2, ATP5MG, ADIRF, CD44, CD24, COX7C, SRP14, HNRNPA2B1, EEF1D, HINT1, CALM1, CSTA, AKAP13, CELF2, IFITM2, CYSTM1, HSPA8, ARHGAP26, EZR, ID3, KRT14, PTPRC, IER3, PFDN5, COX4I1, CIRBP, JMJD1C, TOMM7, SERP1, BTF3, LMNA, N4BP2L2, CD59, PPIA, LDHA, TM4SF1, RBM47, KRT8, AHNAK, DSTN, LRRFIP1, SOCS3, RBM39, NPM1, SEPTIN7, MCL1, HBB, SPTBN1, GLUL, PFN1, DCN, H2AZ1, PLXDC2, SLC25A5, PPDPF, ALDOA, IGLC2, ARPC2, LGALS3, SOD2, ATP5ME, TAGLN2, KRT15, CYRIB, EIF4A2, RTN4, CCNI, LAPTM5, HES1, SORBS1, CSTB, ZBTB20, RGS1, JTB, TSC22D1, CCL4, VMP1, NR4A1, KRT5, MGP, YWHAZ, SRSF7, ARPC3, BTG2, HERPUD1, RBMS3, IL32, DYNLL1, ARHGAP15, SRSF2, DDX17, SPRR3, FOSB, DNAJB1, TXN, COX7A2, IGLC3, PSMA7, RORA, GNAS, PNRC1, CALR, UBE2D3, CD69, MIF, AREG, TCF4, ACTN4, HSP90AB1, PNISR, ZFAS1, SIPA1L1, RNASE1, TUBA1A, FRMD4B, PKM, TPI1, EGR1, HNRNPH1, FTX, FN1, HSPA1B, CHCHD2, HNRNPK, SARAF, TACSTD2, TMBIM6, PCBP1, FUS, NDUFA4, TMEM59, EIF3E, HSPE1, MACF1, UQCRQ, HMGN3, ATF3, SKP1, FCHSD2, TNRC6B, COX6B1, HNRNPC, CTNNB1, FXYD3, GPX1, MT1X, ENO1, UQCRH, GNLY, DBI, RBPJ, IGFBP5, ZEB2, DDX3X, COL1A2, POLR2L, PRDX1, RAC1, KRT4, SF1, SRSF3, BPIFB1, PIGR, RSRP1, PTGES3, ARGLU1, CYCS, CEMIP2, DST, SRSF11, CTSB, HSP90B1, TUBB4B

- exocrine gland: MTCO1P12, STATH, CXCR4, CD74, SPARCL1, CD63,

SRGN, CALD1, NEAT1, CYBA, IGFBP7, TMSB4X, NDUFS2, GSN, IGHA1, ACTG1, TM4SF1, LDHA, HMGB2, MTATP6P1, TCF4, PECAM1, FOSB, IGLC2, COL4A2, GNG11, DDIT4, MEF2C, S100A4, EPAS1, MTND1P23, IGFBP4, SLC2A3, SOCS2, TFPI, IL7R, PTP4A3, CELF2, CLEC2B, HILPDA, SLC8A1-AS1, CSRP2, ITGA5, ITGA1, JCAD, PLXDC2, TPM2, APOLD1, ADAMTS4, PLPP3, ZEB2, CSTA, CCL4, A2M, FYN, TNFRSF1B, CXCL12, KIF14, PMP22, THBD, ZAP70, RAMP2, CSF3R, KLRD1, TGFB1I1, KCNE4, CTC-425O23.5, AC073254.1, TRAF1, ADAMTS2, ADAP2, CENPF, TMEM204, GIMAP7, GIMAP8, CD2, THBS2, PDGFRA, CD93, KCTD12, AC108004.3, DOK2, MT1A, PJVK, HOXB2, TGFB3, TDRD6, ZNF366, PRR16, AMY2A, PRDM1, CCL5, XCL1, CETN4P, GIMAP1, TENM1, HAS2, COL5A1, CSPG4P12, RFLNB

- eye: PDE6A, SYT1, CKB, CADM2, ANK2, TRPM3, TMSB4X, PDE4D, NEAT1, NRXN3, LSAMP, B2M, KAZN, TCF4, CLU, VIM, GPM6A, GAPDH, CADM1, PAX6, IGFBP7, FTH1, SAT1, EEF1A1, KIAA1217, DLG2, HES1, MEG3, GLUL, PLCB4, TIMP3, H3-3B, SYNE2, TRPM1, S100A6, HSP90AA1, ANK3, ROBO2, CH507-528H12.1, DST, RORA, CALM2, ZBTB20, CST3, PTPRG, MAP1B, RORB, ANXA1, MSI2, TMSB10, FTL, CRYAB, SLC2A3, ZEB2, PCDH9, TPT1, EPB41L2, DMD, CADPS, MACF1, PLXDC2, ACTB, NRG3, SRGN, FOS, GNAS, MEF2C, BTG1, ANXA2, CALD1, CNTN4, SAG, DCN, RTN4, PRKG1, MTATP6P1, EBF1, FOXP2, MBNL1, CALM1, MYL6, CELF2, UBC, NCKAP5, SPARC, NTM, RIMS2, ENO1, PTGDS, AKAP9, JMJD1C, LRMDA, PTPRD, APP, KCNMA1, TSC22D1, PCP4, UBB, NLGN1, VAMP2, SPP1, MAGI2, HSPB1, IQGAP2, JUND, FHIT, ITM2B, AUTS2, CTNNA2, SYNE1, MAGI1, WWOX, WSB1, SNHG14, SNHG29, IGFBP5, PTMA, SPARCL1, FMN1, TUBA1A, FKBP5, GSTP1, EIF1, TXNIP, FRMD4A, HMGB1, PTPRM, CAMK1D, PKM, ARL15, FSTL5, NFIB, JUN, RCVRN, NEBL, MT2A, HSPA1A, GPC6, FOXP1, ZFP36L1, ZNF385D, DLC1, MGP, ALDOA, XIST, GRID2, S100A4, PRR4, CAMTA1, MACROD2, PRKCA, YBX3, HSP90AB1, TF, N4BP2L2, DDX17, AGAP1, AKAP12, LDHA, NFIA, ACTG1, PARD3, ABLIM1, MAML2, KCNIP4, GADD45B, SON, NDUFA4, GRIA4, DOCK4, DCT, PPP2R2B, KRT5, ST6GALNAC3, SORBS2, CXCL14, APLP2, FOXN3, JUNB, CPE, DSCAM, ARID5B, TUBA1B, GUK1, RERE, ANKRD12, UNC119, THSD7A, ASPH, SPTBN1, PNISR, RABGAP1L, ERBB4, CCSER1

- fallopian tube: KRT18, CLDN4, IGFBP7, SRGN, B2M, CD74, SPARCL1, CYBA, S100A6, PTMA, FTL, DSTN, FTH1, PTPRC, TMSB4X, EZR, NEAT1, LGALS1, ANXA1, IGKC, VIM, TMSB10, DCN, BTG1, MEF2C, HSPA1A, SYAP1, MTATP6P1, CBX3, IGLC2, CLU, CTSS, SLC26A3, TASL, GNAS, SPINT2, CXCL8, EIF1, SLC38A2, STOM, MTRNR2L8, SYNE2, TFF3, CKS2, CTD-2287O16.1, ADAP2, ALOX5AP, IGLC3, MCTP1, BTK, ATP6V1B2, ATF7IP, VTRNA1-3, S100A9, IGHA1, CFLAR, RCSD1, FCER1G, CNOT2, IL1B, HERPUD2, GSAP, TM6SF1, ITPR2, IGSF6, TUBB4B, PCLO, PIK3CG, GPX1, HSPE1P2, KIF5C-AS1, WFDC2, AC016739.2, HIP1R, FCER1A, AP1S3, CHD7, GOLGA8B, TMSB4XP8, RHEX

- gallbladder: SERPINB2, MMP7, SERPINA1, VIM, FCER1G, IGFBP7, TM4SF1, PSAP, TFF3, LAPTM5, CCL21, CRYAB, LGALS1, LYZ, DCN, CTSB, CXCL8, TM4SF4, ADIRF, TMSB4X, EFEMP1, TYROBP, THBS1, REG1A, HSPB1, TAGLN, AGR2, LUM, SPINK1, PLA2G2A, ALOX5AP, MGP, NAMPT, TFF1, HMOX1, GPX2, CLU, SPP1, CALM2, PCAT19, CLEC2B, BCL2A1, C1QA, IFITM3, PLAUR, FOS, UQCRQ, CD74, MT2A, IER3, PTMA, CD59, CD24, CREM, SKP1, CDKN1A, CREG1, C1R, C5AR1, S100A4, ACTG1, GZMB, FABP4, SDCBP, SPARCL1, NNMT, RGS2, ID3, TSC22D1, MGST1, CD55, EREG, SERF2, RNASE1, GNG5, G0S2, TPSB2, GC, PPIA, TRBC2, POMP, CCN1, ATP5MK, HSP90AA1, MMP19, FKBP1A, EEF1D, HSPA6, CD9, CTSD, DDX3X, ARPC2, CNBP, TMSB10, NDUFS5, IFI27, DSTN, CD99, ACTB, FTL

- heart: NEGR1, LAMA2, CACNA1C, PTPRC, UBA52, SAT1, SPARCL1, B2M, NEAT1, CALD1, TTN, IGFBP7, CD36, DLC1, PTPRM, TPM1, RBMS3, ARHGAP15, LDB2, PLXDC2, SORBS1, VIM, PRKG1, FTL, ACTB, IFITM3, MYL2, ZEB2, COL3A1, FRMD4A, TMSB10, CELF2, SRGN, LRMDA, EEF1A1, MYL7, COL1A2, TMSB4X, VWF, UTRN, TCF4, EBF1, FTH1, LINGO1, RYR2, DST, PITPNC1, TXNIP, DCN, ANK3, S100A10, CD74, SLC8A1, ANKRD44, FKBP5, BTG1, MEF2C, CBLB, PTPRG, HBG2, FRMD4B, LGALS1, AKAP13, TIMP3, XIST, DDX5, GNAS, FOXP1, RORA, PLCB1, HSP90AA1, IQGAP2, RBPJ, SYNE2, MBNL1, NRP1, S100A6, DDX17, TPT1, MAML2, ELMO1, BCL2, CCND3, PTMA, TNNT2, MT2A, CDH19, SPTBN1, VCAN, ZBTB20, ZBTB16, ID3, LRP4, ACTG1, GAPDH, ITM2B, LPP, ACTC1, FOS, NFIB, CALM2, S100A4, FCHSD2, MGP, NPPA, ABLIM1, QKI, FOXN3, N4BP2L2, ARGLU1, SRRM2,

PDLIM5, TFPI, MACF1, DMD, PDE3A, CH17-189H20.1, WSB1, FNDC3B, KIAA1217, PLCG2, RABGAP1L, IL32, SSH2, GSN, WWOX, FABP4, SPARC, SON, COL4A1, AUTS2, HMGB1, MED13L, NFIA, ARL15, FTX, FN1, HSPB1, FOXO1, ARHGAP24, HSP90B1, LDLRAD4, NRXN1, MAST4, PSAP, MYL12A, PRKCH, PABPC1, PRKAG2, PDE4DIP, HNRNPA2B1, MAN1A1, MYL6, DPYD, TNRC6B, RBM6, CST3, ARHGAP26, LHFPL6, JMJD1C, PDZD2, DUSP1, HNRNPH1, ANKRD11, SORBS2, SMYD3, ANXA1, TIMP1, FHL2, VMP1, HSPG2, MAGI1, H3-3B, TGFBR2, TRIO, FBXL7, FYN, MAP1B, PDE3B, CHD9, MAP4, DIAPH2, ZFHX3, CFLAR, LUC7L3, UBB, BNC2, MYH9, RBM39, RERE, LRRFIP1, STARD13, SIK3, EIF4G3, SBF2, ENG, TUBA1B, AKT3, PDK4, SEPTIN7, CCNI, JAK1, TSC22D3, MYL9, TBC1D5, APP, COL1A1, RBMS1, INPP4B, ETS1, JUND, SYNE1, ST6GAL1, PBX1, IGFBP5, TCF12, DOCK8, IL6ST, ANKRD1, HMGN3, IMMP2L, CALM1, ASAP1, CRYAB, DLEU2, HSP90AB1, MYCBP2, CUX1, UBC, ANXA2, ARID1B, CEBPD, PTEN, S100A8, EXT1, HIF1A, KMT2E, ATXN1, PFN1, EIF1, PPP1R12A, KTN1, PPP3CA, GPHN, SIPA1L1, MB, RTN4, ZFP36L2, F13A1, HBA2, PPP6R3, CPM, CYBA, HIPK2, RHOA, SOX5, ADAMTS9, CTSB, GRK5, FUS, PDE4D, KANSL1, ZFP36L1, UBE2E2, MYH7, MECOM, PRRC2C, HDAC9, ZFAND3, AOAH, TTC28, CAPZB, EXOC4, LIMS1, RAB11A, RGS5, TM4SF1, A2M, NKTR, ZFP36, TTC17, SRPK2, PLEKHA5, TNNI3, ANKRD12, PALLD, TACC1, CIRBP, ARPC2, NFKBIA, NEXN, LRBA, TSHZ2, GNAQ, AKAP9, COL4A2, AHNAK, SPIDR, IFI27, DOCK4, MEF2A, STAG1, PPP2R5C, PTK2, PNISR, ERC1, CAMK2D, RASAL2, ID2, NAMPT, KMT2C, ITPR2, ZEB1, PPFIBP1, RSRP1, ABCA8, MKLN1, PACS1, JUNB, HSPA1A, CTNNA3, FABP5, SMCHD1, GBE1, NIBAN1, EGFL7, VPS13B, SLC25A4, JUN, NCOA1, ZNF638, CACNB2, ASH1L, DSTN, PLA2G5, ACACB, GOLGA4, LAMB1, SERF2, ROCK1, ATP2A2, ZSWIM6, YBX3, PICALM, CSNK1A1, AFF1, ACTA2, ELF1, PRKCE, PCDH9, FHIT, ZCCHC7, NR3C1, YBX1, PLCL1, CD99, AC007319.1, FRYL, RASGEF1B, HNRNPC, WNK1, EPS8, TCF25, CCNH, ID1, GNB1, PARD3, TBC1D22A, MTSS1, DDIT4, DLG2, H3-3A, FBXW7, SAMHD1, MGST3, STAT3, LMNA, NACA, SLC9A9, ETV6, TRPS1, PARP8, UQCRH, CAV1, EXOC6B, SRSF11, PKM, PAM, EEF1D, CD63, LDHB, CHST11, DEK, SLC2A3, SRSF5, SRSF7, RUNX1, AFF4, DTNA, RBPMS, IGFBP4, PRKN, ITFG1, DES, MAGI2, KLF6, EPB41L2, LARGE1, UBE2D3, DYNLL1, PIP4K2A, RSF1, FOXO3, BIRC6,

PLPP1, HNRNPDL, PDXDC1, SSBP2, UBE2R2, AAK1, SCAF11, CMIP, HNRNPH3, FAF1, HSPA8, BCAS3, MGAT5, MARK3, PDE1C, KLF12, SIK2, ZNF609, AFF3, PEAK1, GPATCH8, USP34, COX4I1, ACTA1, LIMCH1, RAB1A, RBMX, LTBP1, HNRNPU, BACH1, BPTF, HERC1, NCOA3, C7, FNBP1, RAPGEF1, ARID5B, SRSF2, HIP1, PDE7A, PSD3, PCM1, FILIP1, ZNF292, TNNC1, CDC42BPA, CDH13, GNLY, CYRIB, JPX, POSTN, NEBL, ATP5F1E, BTBD9, MTDH, IGF1R, CDC42, MBD5, CTBP2, S100A11, ITGB1, SVIL, SOS1, DENND1A, PAN3, ANAPC16, CARMIL1, REV3L, CH507-528H12.1, ANKRD17, EPS15, IQGAP1, TAGLN, CSGALNACT1, SLC1A3, NPM1, USP15, NRXN3, SRSF3, INSR, NUMB, NCOR1, PIK3R1, RACK1, PHACTR2, CCNY, EHBP1, CD9, TMBIM6, RBM25, GAS7, RNF149, HNRNPA3, ZFPM2, CTNNB1, RAP1B, APBB2, MUC20-OT1, COL6A2, BMPR2, PCNX1, FBXO11, MAP4K4, SPAG9, MAML3, FAU, NFE2L2, NIPBL, YWHAE, TSC22D1, PTGES3, CLIC4, MYBPC3, TRAPPC9, GPBP1, NFAT5, LDB3, RAD51B, TAOK3, FGD4, CTNNA1, GLUL, TJP1, ARID4B, CERS6, NR4A1, HNRNPA1, RNASE1, ADGRB3, CAMK1D, COMMD1, RAP1A, RB1, TMTC1, CDK13, CCDC91, SCN7A, TBL1XR1, MSN, PDE8A, RAB2A, NAALADL2, VPS13D, ATM, SEPTIN2, CAMTA1, PAFAH1B1, AOPEP, TTC3, MEIS2, SRP14, RHOB, ENO1, ATRX, RAPGEF2, ADK, TUBA1A, IFITM2, HERC4, RCSD1, ARHGAP10, SETX, SH3D19, C20orf194, PECAM1, FRY, PTPRE, JARID2, RNF130, FGF12, SUB1, ITSN1, CDKAL1, BAZ2B, CEP350, ACTR2, WAC, NF1, RNF115, EPC1, COX7C, ADD3, NAV1, CFL1, NUCKS1, PHACTR1

- immune system: B2M, CD79A, CD74, MS4A1, JUNB, EEF1A1, VIM, IFITM1, ACTB, NFKBIA, CD3D, TXNIP, TCL1A, KLF6, CD52, IL32, IFITM2, KLF2, CD7, BTG1, TMSB10, SRGN, ZFP36, KLRB1, FTH1, GAPDH, S100A4, HSPA1A, DUSP2, TMSB4X, ZFP36L2, FTL, CD37, LAPTM5, LTB, TPT1, JUN, UBC, H3-3A, ATP5MG, FOS, COTL1, CD83, BTF3, CD3E, FXYD5, TNFAIP3, ISG20, RAC2, HSP90AB1, SELL, RGS1, PPP1R15A, IGHM, HMGB1, CYBA, S100A10, MYL6, CD69, EZR, HSPE1, CD27, PFDN5, SERF2, HCST, TRBC2, CORO1A, PLAC8, TSC22D3, DDX5, LDHB, HSP90AA1, COX4I1, HSPA1B, FCMR, CST3, EEF1B2, ACTG1, SAT1, HMGN2, SUB1, H4C3, PABPC1, CCL5, FYB1, PTPRCAP, SH3BGRL3, TPI1, NPM1, GPR183, CD79B, UCP2, LSP1, EIF1, SARAF, GSTP1, MYL12A, HMGN1, FAU, LIMD2, SNRPD2, PPIA, DNAJB1, EIF3G, CD99, BANK1, PKM, H3-3B, IER2, DUSP1, BCL2A1, C12orf57, MIF,

SUMO2, IFITM3, HERPUD1, IL7R, OAZ1, EEF1D, EEF2, YWHAZ, CXCR4, NOP53, HNRNPA1, TUBA1B, CLIC1, RGS13, EVL, S100A6, ARHGDIB, PGK1, SLC25A5, SSR4, SELENOH, PTPRC, HSPA8, CALM1, STMN1, JAK1, RAN, PFN1, DNAJA1, TOMM7, NCL, SSR2, CCNI, NCF1, CALR, CD44, UBA52, CD48, SYNGR2, GNLY, FDCSP, PSME2, EIF3E, GMFG, YPEL5, MCL1, RNASET2, CAP1, HNRNPA2B1, SRSF5, COMMD6, RACK1, ZFAS1, ITM2B, JUND, DBI, PNRC1, FOXP1, FOSB, LCP1, TCF7, TUBB, CTSH, SYNE2, TPR, SF3B2, RNF19A, C9orf16, MAF, TAGLN2, SNHG8, SEPTIN7, RGS2, ATP5IF1, TRBC1, CD8B, LAT, H2AZ1, ARPC2, MARCKSL1, ARPC3, PSME1, EIF3L, UBB, ZFP36L1, POLD4, CFL1, ENO1, EIF5, SRSF7, TMEM123, ANP32B, LCK, HIGD2A, RGS10, WIPF1, NDUFB11, PCBP2, GADD45B, DDIT4, NACA, YWHAB, HSPB1, PRDX1, SPOCK2, AREG, PPDPF, HSPD1, CYTIP, STK17A, DAZAP2, HNRNPC, EIF3K, CD22, HMGB2, EIF3H, ELF1, CNN2, CD53, CD2, NKG7, YBX1, ID2, NR4A2, MAP3K8, CR2, COX8A, SF1, ANKRD12, ODC1, PNISR, SAP18, PTPN6, ITGB2, IRAG2, GPSM3, CD19, RBM39, COX7C, BTG2, CORO1B, MYL12B, PTMA, ITM2A, HINT1, LITAF, SEPTIN9, GYPC, PPP1CC, COX5B, PDIA6, SLC25A3, EIF3F, GCC2, IKZF3, CSTB, UQCR10, SRSF2, TCEA1, SKP1, CSDE1, ABRACL, MT2A, REL, EIF4A2, EMP3, TPM3, CALM2, SLC2A3, PGAM1, FUS, PARP1, TBCA, PTGES3, RESF1, GABARAP, RIPOR2, POLR2L, CUTA, LDHA, MYH9, DPP7, EGR1, ALOX5AP, DDX17, STAT3, HSPA5

- intestine: FTH1, EEF1A1, PIGR, B2M, LGALS4, CD74, NEAT1, UTRN, ACTG1, S100A6, CFL1, TMSB4X, FOXP1, RBFOX2, ZSWIM6, DYNLL1, ACTN4, TOMM7, PCBD2, TUBA1B, NACA, MAGI1, SEC61G, NFIA, HSPA5, VIM, IFITM2, LPP, HBG2, VAMP8, PLCG2, MDK, COL3A1, CST3, MGAT5, SH3BGRL3, DNAJB1, TMSB4XP4, GAPDH, KRT8, SRGN, CHGA, DPYSL2, SNRPD2, ANKRD11, TMED10, SOX4, ATXN1, SON, PLAC8, COX7A2, FKBP2, RGS2, ZFAS1, CUX1, LILRB1, FYB1, CXCR4, CREB3L2, PHYKPL, AIF1, SAFB2, KLF6, VPS13D, RBX1, ROMO1, COL1A2, RBM5, PBX1, SLC17A5, AUTS2, COL1A1, BTG2, OLR1, COPS9, CCT4, PRKG1, NFKBIA, CTSW, RARRES2, IGF2R, FLRT2, STMN1, ALOX5AP, LAPTM5, ERICH1, UQCR11, JUNB, RBM39, AGAP1, NUDT1, P4HB, TSHZ2, S100A9, GMDS, FAU, RERE, SNHG14, ARL4C, IRF1, TMSB4XP8, ANKRD37, XIST, HP1BP3, NAPSB, HIVEP3, CIRBP, FOS, SAT1, VTI1A, TPT1, GABPB1, ISG15, PFDN5, SLC9A9, CLDN5, TPM4,

ZNF3, RELN, SNHG8

- kidney: ATP1B1, PTMA, TMSB4X, IGFBP7, B2M, CD74, IGFBP5, NEAT1, S100A6, ZBTB20, VIM, CD24, MEIS2, MECOM, SRGN, CST3, HSPA1A, SAT1, EEF1A1, DUSP1, SPP1, IFITM3, FOS, H3-3A, SPARC, CALD1, TXNIP, DDX5, GPX3, PDE4D, FTL, ACTG1, LDHB, APP, S100A4, IL32, PAX8, ACTB, HSPB1, CA12, HNRNPA1, PKHD1, CD9, UBB, JUN, MT2A, HES1, TMSB10, DEFB1, LGALS1, MTATP6P1, GAPDH, TIMP3, MGP, S100A11, DNAJB1, PFN1, JUNB, ID2, FTH1, ITM2B, HSP90AA1, HSPE1, EEF2, SLC8A1, BTG1, COL1A1, TPT1, PABPC1, H3-3B, LRMDA, KLF6, TSC22D1, NACA, FOXP1, WSB1, CALM2, SYNE1, PTPRC, SYNE2, ADIRF, UTRN, IGF2, ATP1A1, NBEAL1, EMCN, XIST, UBC, TUBB, UMOD, ATP5F1E, CD63, PSAP, CYBA, ALDOA, RTN4, ANXA2, APOE, MYL12A, MEF2C, ZFP36L1, ERBB4, FXYD2, GSTP1, NDUFA4, CELF2, HMGB1, PRKG1, ZFP36, RACK1, PTPRG, TPM1, GNAS, IFITM2, MTRNR2L12, ZFP36L2, LDB2, PPDPF, HSPA1B, CFL1, TUBA1B, NPM1, ADGRF5, S100A10, ZEB2, N4BP2L2, ATP5MG, CH507-513H4.1, JUND, DNAJA1, NFKBIA, FOSB, HSP90B1, PNISR, FAU, HSP90AB1, RHOB, CLU, SOX4, BICC1, WNK1, BEX3, TIMP1, COX4I1, TSC22D3, FKBP5, CCL4, UQCRB, WFDC2, RBM39, DDX17, CALM1, COX7A2, HBB, MBNL1, SPTBN1, HNRNPA2B1, HSPA5, HERPUD1, ARHGAP29, EGR1, CTSC, ID1, SERF2, MYL6, ID3, VMP1, MIF, TACC1, KIAA1217, MT1G, EIF1, CIRBP, TPM4, UBA52, CALR, EEF1D, CXCR4, MSI2, SLC25A6, ARL15, ITM2C, COBLL1, DSTN, TMBIM6, ARPC3, GPX1, HSPA8, LAPTM4A, SKP1, PPIA, CRYAB, LRRFIP1, CCSER1, PFDN5, SLC12A1, EIF4G2, SRP14, COX7C, CD52, HSPD1, BCAM, EPS8, TYROBP, YBX1, SON, HINT1, HNRNPC, NAP1L1, UBE2D3, STMN1, NFIA, LUC7L3, RBPMS, ATP5PO, SNHG29, COX7B, HNRNPH1, ST6GAL1, EEF1B2, PKM, TXN, PLCB1, DDIT4, ITGB1, RORA, LDHA, ATF3, NR4A1, PLPP1, ARHGDIB, PLCL1, IER2, DYNLL1, SELENOP, EIF4A2, SRSF5, TCF4, VCAN, IGKC, HNRNPK, GATA3, GLS, GSN, GPX4, RGS2, YWHAB, JMJD1C, TOMM7, AKAP12, MAML2, PBX1, LITAF, NDUFA13, ARGLU1, OAZ1, HMGB2, HNRNPDL, NPC2, KCNQ1OT1, TGFBR2, S100A9, CCNI, RHOA, MAGI1, YWHAZ, SEC62, TMA7, RERE, MYL12B, COL3A1, SLC2A3, ATP6V0E1, MORF4L1, ATP5MC2, SQSTM1, MT1X, PCBP2, GADD45B, TUBA1A, LPP, SRRM2, SUB1, BTF3, CTSD, HIPK2, RAC1, KTN1, NUCKS1, YWHAE, MGST3, AKAP13, ARPC2, NFE2L2, CD81, KMT2E, PRDX1, MACF1, MAGI2, COMMD6,

SH3BGRL3, C12orf75, BTG2, PTGES3, AC013463.2, H2AZ1, A2M, ATP6V0B, CEBPD, ANXA1, COL1A2, HNRNPU, BSG, CD59, MTRNR2L8, HSPH1, HIF1A, SOD2, PEBP1, FTX, KAZN, HBA2, ACTN4, COX6C, RNASE1, RAN, FUS, NDUFA1, NEDD4L, ATP5MJ, PRKCA, AKAP9, CDC42, ZFAND5, TMEM59, CLIC1, NDUFS5, DBI, CLK1, SLC25A5, CREM, ENO1, NFAT5, LMNA, GPC3, CAMTA1, PPP1R15A, COX8A, CHCHD2, SH3BGRL, RAP1B, OST4, ATP1B3, CCN2, HMGN1, UQCRH, NAMPT, ANKRD12, AIF1, RGS1, COX5B, EIF3E, LYZ, PLXDC2, COL4A2, ATP5IF1, ZFAS1, PPIB, CD164, PCDH9, CTSB, SRSF7, INSR, MRPL33, SARAF, OXR1, FOXN3, PSMB1, EIF3H, CHCHD10, COX6B1, ESRRG, SLIT3, ADD3, ARID5B, SEPTIN7, SOD1, GLUL, IER3, CYB5A, PNRC1, SFPQ, RSRP1, TRA2A, MDK, MRPS6, NCOA7, HMGN2, AQP2, RBMS1, UBL5, EMP3, PRRC2C, PTPRM, QKI, SRSF3, PDIA3, TPI1, SEM1, CAPZB, SRSF11, PICALM, SLIT2, COX6A1, S100A8, PSMA7, POLR2L, ATP5PF, SLC25A3, UQCR10, SUMO2, EZR, NCL, ATRX, TNRC6B, UQCRQ, KCNIP4, MYCBP2, THSD7A, CYSTM1, EIF3K, PSME1, TAGLN2, SLC38A2, ZBTB16, NKTR, DST, HMGN3, NR2F2-AS1, JAK1, RASGEF1B, EBF1, BST2, ATP6V1G1, ATP5MK, COTL1, ANXA5, TCEAL9, AHNAK, PLPP3, FHIT, NRP1, POMP, RBM8A, SEC61B, ELOB, ANK3, STAT3, PPP1CB, TTC3, CLIC4, MSN, LIMCH1, MED13L, HSPA6, GABARAP, UQCR11, ATP5ME, ANAPC16, TRA2B, FAM107B, NDUFB2, MCL1, SERBP1, SNRPD2, ATP6AP2, CUBN, CCL5, SSR2, HNRNPA3, TLE5, SSR4, CHD9, TMEM258, RDX, ZNF207, DNAJB6, SEPTIN2, EIF5B

- lamina propria: AHNAK, SH3BP5, CD74, SRGN, COTL1, CCL5, FTH1, JUND, KLRB1, NEAT1, TYROBP, EEF1B2, TMSB10, TPT1, CD52, IFI30, CST3, SEC31A, SLC11A2, PTPRC, FOSB, XBP1, TNFRSF18, MCUR1, JCHAIN, DAPK1, AREG, SOX4, SORL1, FCER1G, FTX, MATN2, IRF8, SMS, SELL, CACFD1, C1orf162, GPR183, ALDH2, FABP5, APOC1, IGHA2, CCAR2, C5orf24, IGHA1, HEBP1, BRIP1, FGR, CPSF4, IGHV3-33, SIRT5, JRKL, FOXC1, TRBV11-2, MLC1, TPSB2, SMC4, FREM1, ZNF154, SMYD4, FXYD7, GARS1-DT, FCGR2A, RBM39, AP000704.5, SGK1, S1PR5, GNA15, BTG1, SMIM2-IT1, AQP3, TOX2, SLC20A1, DPY30, CPM, TBXAS1, PRDM1, TP53INP2, STAM, SAA1

- large intestine: JCHAIN, LGALS4, KRT8, VIM, MTRNR2L12, SRGN, PHGR1, FOS, NEAT1, CD74, FTH1, S100A6, B2M, EEF1A1, IFITM3,

JUN, MT2A, JUND, PIGR, IGHA1, HSPB1, TFF3, ACTB, CST3, SAT1, HSPA1A, TMSB4X, TMSB10, LGALS1, YBX1, DUSP1, FTL, TPM1, EEF1B2, S100A10, BTG1, MIF, CCL5, COL1A2, IGKC, SLC25A6, FABP5, FABP1, GAPDH, KRT18, ZFP36, TXNIP, KRT19, HSP90AA1, H3-3A, SPARCL1, LGALS3, JUNB, PABPC1, HSPA1B, MT1G, ZFP36L2, HES1, IL32, CD9, TPT1, SOX4, H4C3, CD52, MDK, ID1, XIST, ID2, FOSB, H3-3B, SH3BGRL3, IGFBP7, PTMA, COL3A1, UBC, PFN1, KLF6, DDX5, HSP90AB1, ATP5F1E, HSPA5, CD63, ATP5ME, RACK1, CYBA, HMGN2, EZR, HSPA8, ACTG1, PFDN5, NFKBIA, IGLC2, MTATP6P1, IER2, ANXA2, TSC22D3, HSP90B1, ZFP36L1, UBB, TUBA1B, IFITM1, HNRNPA1, CALM2, COX7C, CALM1, SELENOP, FKBP1A, CD24, DNAJB1, DDIT4, S100A11, MARCKSL1, NPM1, PPDPF, MT1E, AKAP13, ITM2B, EEF1D, H2AZ1, NACA, CFL1, LMNA, GSTP1, LDHA, IL7R, AGR2, GSN, PSAP, NR4A1, GNAS, ELF3, WSB1, CKB, PPP1R15A, MUC2, ITM2C, BTG2, EIF1, ID3, ATP5MC2, HSPD1, IGLC3, COX6C, FUS, CD99, TUBA1A, UQCRB, OAZ1, HMGB1, HSPE1, SRRM2, TXN, FAU, MYL6, CLIC1, CXCR4, C15orf48, RHOB, MYH9, EEF2, HINT1, CLDN4, UBA52, COX4I1, TOMM7, HNRNPA2B1, CHCHD2, TPI1, SSR4, FXYD3, TAGLN2, GADD45B, SQSTM1, SELENOW, CALR, DDX17, EPCAM, TMBIM6, COL1A1, TSPAN8, PNRC1, RBM39, PCBP2, C12orf57, IFITM2, SARAF, STMN1, NUCKS1, PPIA, FOXP1, CALD1, PRDX5, AHNAK, PEBP1

- liver: ALB, ACTG1, ACTB, S100A4, TYROBP, NEAT1, S100A11, B2M, CST3, VIM, CD74, JUND, FTL, H3-3A, HP, FOS, HSPA1A, TPT1, MTRNR2L12, JUN, S100A9, HSP90AB1, SERPINA1, STMN1, HBG1, XIST, NKG7, S100A6, IFITM3, TMSB4X, IGFBP7, SAT1, EEF1G, HBG2, PTPRC, HSPD1, HMGB1, RTN4, RUNX1, SRGN, FTH1, DUSP1, CXCR4, DNAJB1, PDE4D, EEF1A1, DNASE1L3, ZEB2, EEF1B2, MT2A, GAPDH, APOA2, HBA2, UBC, TUBA1B, RORA, MIF, SAA1, HNRNPA2B1, ZFP36L2, CCL5, LYZ, APOE, S100A8, HSPA1B, HNRNPA1, APOC1, KLF6, CTSB, SH3BGRL3, HSP90AA1, NFKBIA, ATP5F1E, MEG3, HSPB1, CD52, MTRNR2L8, PTMA, ZFP36, TMSB10, EEF2, LDB2, UBA52, EIF4A1, UTRN, FKBP5, BTG1, ID2, CALR, PSAP, HSP90B1, ZFP36L1, PRDX1, MBNL1, ITM2B, IFITM2, S100A10, GPC6, FCER1G, TXNIP, PCBP2, MYL6, EGFL7, DDX5, PABPC1, PFN1, FAU, JUNB, FOXP1, HSPA8, CCL4, HNRNPU, NAMPT, EIF1, MTSS1, APOB, AIF1, RSRP1, YBX1, ZBTB20, UBB, HNRNPH1, PRDX2, TIMP1, GNAS, APOA1, NPM1, GYPC, SLC25A5, NAP1L1, RBM39,

CD63, IL32, H3-3B, PIP4K2A, IQGAP2, HSPE1, HINT1, ANXA1, TMBIM6, RTF1, CYBA, HBA1, HNRNPDL, H4C3, LRMDA, LGALS1, HNRNPC, RAP1B, DDX17, APOC3, MS4A6A, CD36, PNISR, FGB, GSTP1, FOSB, CALM1, DUSP2, SLC2A3, COX7C, TNFAIP3, RSL1D1, MYL12A, BLVRB, SOD2, CELF2, PRDX6, ARPC2, TUBB, PPIB, ATP5ME, HMGN2, JMJD1C, PLCB1, N4BP2L2, TCF4, CD37, RAC1, PNRC1, MACF1, TAGLN2, ARHGDIB, FUS, PLAC8, HSPA5, CALM2, TSC22D3, CD69, CD99, SRSF7, LDHB, SRSF11, AKAP13, HMGB2, PEBP1, SRRM2, PPIA, GPC3, NCL, TPM4, RELN, UQCRQ, EIF3E, WSB1, LDHA, HBB, VAMP8, SLC25A37, ZFAS1, KLF2, POLR2L, SERF2, LRRFIP1, IGKC, SLC25A6, SNX3, JCHAIN, RTN3, SOD1, RACK1, ATP5IF1, RHOA, IL6ST, P4HB, PPDPF, HNRNPA3, C1QB, LCP1, EEF1D, DDIT4, PARK7, JAK1, SF1, MYH9, TACC1, YWHAZ, NACA, AFF3, SLC25A3, PCBP1, ATRX, SON, UQCRB, OAZ1, ITGA4, H2AZ1, SCP2, PFDN5, LAPTM5, GNLY, TAOK3, NFIA, NME2, ANP32B, GPX1, MCL1, GLUL, SRSF5, CFL1, MGST1, SMCHD1, RSRC2, SFPQ, TMA7, SELENOP, CFLAR, PKM, DNAJA1, CIRBP, TBCA, MTDH, CAST, C3, CPS1, RERE, NDUFA4, ELF1, PHF3, RABGAP1L, ARGLU1, FGA, ENO1, COX6C, MSI2, PSMA7, SLC40A1, HERPUD1, NFKBIZ, LPP, VMP1, PDIA3, HP1BP3, MED13L, UQCR11, ATP5MC2, IER2, RAB11A, ARPC1B, SRP14

- lung: B2M, SRGN, CALD1, CLIC1, CD9, CST3, H3-3B, TYROBP, MTND1P22, CD74, IFITM3, H3-3A, EEF1B2, MTCO2P22, VIM, IGFBP7, SAT1, LGALS1, EEF1A1, S100A10, ZEB2, S100A6, S100A4, SLC25A6, FTL, S100A11, PTPRC, CLU, GABARAP, SFTPB, EZR, TUBA1A, NEAT1, FOS, TMSB4X, ATP5F1E, MGP, MTRNR2L12, MT2A, CRIP1, RGCC, PFN1, ANXA2, DSTN, DCN, IL32, ANXA1, HSPA1A, BTG1, ACTG1, CELF2, SOD2, SLPI, GSTP1, DDX5, JUN, RNASE1, TMSB10, CD99, TXNIP, TPT1, ZFAS1, CYBA, JUND, TIMP1, TUBA1B, FTH1, GAPDH, CD63, LGALS3, ZFP36L1, TCF4, CCL5, CALM2, ACTB, SNHG29, SPARC, DUSP1, FAU, STMN1, IFITM2, CALM1, FN1, PABPC1, MTCO3P18, KLF6, ARID5B, HSP90AA1, EEF1G, IER2, ZFP36, NFKBIA, EPAS1, PLXDC2, AKAP13, ARHGDIB, HSPB1, SCGB3A2, XBP1, PTMA, ZFP36L2, PPDPF, CEBPD, CTSB, SSR4, EEF2, MIF, CD52, LY6E, SFTPC, NAP1L1, UBB, SYNE2, TSC22D3, S100A8, ITM2B, MACF1, JUNB, FOXP1, DYNLL1, MBNL1, HSP90AB1, CD44, RACK1, RORA, KLF2, ID2, YBX1, TAGLN2, MTATP6P29, ALCAM, LDHA, UBA52, SCGB3A1, LPP, CCL4, TIMP3, CFLAR, IFI27,

WSB1, GLUL, PSAP, HNRNPA2B1, FOSB, SPARCL1, UBC, ARGLU1, MYL6, SERF2, HMGB1, HSP90B1, LYZ, CFL1, FKBP5, SCGB1A1, RBM39, APOE, LAPTM5, SON, HINT1, GNAS, CSTB, LMNA, NAMPT, UTRN, HNRNPA1, A2M, CALR, NPC2, KMT2E, ARPC2, TMBIM6, COL1A2, EGR1, IGKC, CXCR4, DST, PNISR, PPIA, VMP1, JMJD1C, TNRC6B, TPM4, DDX17, NKG7, CTSD, ARID4B, TXN, HNRNPU, PDE4D, PCBP2, LRRFIP1, RNF213, PKM, HNRNPH1, NPM1, CAST, SH3BGRL3, HSPA8, MYL12A, CCNI, SPTBN1, EIF1, CAPZB, GADD45B, XIST, LUC7L3, N4BP2L2, RTN4, GSN, SLC2A3, TUBB4B, LITAF, DDIT4, NUCKS1, NACA, UQCRB, S100A9, HMGN2, ARPC3, NBEAL1, EEF1D, KTN1, ENO1, RGS1, MCL1, HNRNPK, CHCHD2, MAML2, NDUFA4, AKAP9, TUBB, SRP14, AUTS2, ZBTB20, APP, KRT19, HNRNPC, FCER1G, SRSF5, SQSTM1, RHOB, ITGB1, RBMS3, MTND2P13, FYB1, DNAJB1, COX7C, HMGN3, IFITM1, DPYD, GPX1, MSN, LDHB, SYNE1, HNRNPDL, SOX4, SRRM2, RBPJ, PRKG1, H2AZ1, CAV1, CIRBP, TMA7, OAZ1, GPX4, SARAF, DNAJA1, HSPA5, PSME1, SRSF11, PNRC1, PRRC2C, MYH9, YWHAZ, CD81, CD55, CCNL1, PFDN5, RSRP1, ALDOA, AREG, RBMS1, CTSC, ANKRD12, PHIP, UBE2D3, RAB11FIP1, MTDH, PDIA3, ACTN4, PRDX1, SMCHD1, EIF3E, FOXN3, COL1A1, PPIB, CEBPB, ATP5MG, TFPI, HIF1A, NKTR, SRSF7, TACC1, LCP1, PPP1R15A, SRSF3, HMGN1, SEPTIN7, ATP5MC2, PTGES3, COX6C, BPTF, YWHAE, SKP1, IGFBP5, SF3B1, RAC1, RHOA, IL7R, CCL2, ACTA2, DDX3X, NCL, TPM3, BTG2, CDC42, LAPTM4A, MORF4L1, KMT2C, EIF4A2, AHNAK, ANXA5, EIF4G2, GNAQ, EMP3, FKBP1A, MT1X, GPBP1, IQGAP1, MYL12B, COX4I1, FNDC3B, MECOM, WWOX, TSC22D1, RAP1B, CYRIB, ATP5ME, APOC1, PCM1, CTNNB1, BTF3, TNFAIP3, HSPE1, ISG15, YWHAB, IGLC2, C1QA, C4orf3, NFE2L2, SLC25A3, CPM, SFPQ, MDK, DOCK4, MTND4P33, HNRNPA3, CHD2, AFF3, IFI16, SCAF11, HERPUD1, CSDE1, FABP5, CSNK1A1, VAMP5, SEC62, COTL1, HOPX, SEPTIN9, NCOR1, LDB2, SH3BGRL, GOLGA4, EDF1, APLP2, CXCL2, RGS2, SUMO2, TRA2B, ARL6IP5, HSPA1B, DEK, NDUFB1, NOP53, ATRX, RUNX1, FUS, GCC2

- lymph node: CD74, IL7R, EEF1A1, CD3E, SRGN, B2M, TMSB4X, EEF1G, MTRNR2L12, CD69, CD52, JUND, TYROBP, NKG7, TXNIP, VIM, FTL, CXCR4, ATP5F1E, SAT1, IFITM1, FOS, CCL5, ACTB, BTG1, NFKBIA, FTH1, LTB, S100A4, SNHG29, TMSB10, PTPRC, RACK1, IL32, IGKC, ANXA1, JUNB, TSC22D3, NEAT1, KLF6, COTL1, CORO1A, KLF2, JUN,

RGS1, CD7, HSP90AA1, DUSP2, TPT1, ZFP36, PABPC1, SH3BGRL3, TNFAIP3, MT2A, LAPTM5, ARHGDIB, MTCO3P18, S100A6, ZFP36L2, FYB1, CST3, IFITM2, SLC25A6, GAPDH, TUBA1B, CD79A, CD3D, HSPA1A, GAS5, AREG, EIF1, DUSP1, GZMK, S100A10, H3-3B, HERPUD1, GPR183, DNAJB1, HSPA8, PTPRCAP, KLRB1, EEF1B2, ZFP36L1, LGALS1, OAZ1, GSTP1, H4C3, ACTG1, UBC, CD37, NPM1, MS4A1, CYBA, SARAF, FAU, CLIC1, CD8A, IFITM3, UBB, IER2, HSP90AB1, NAP1L1, EIF4A1, NOP53, H2AZ1, NBEAL1, PFN1, HSP90B1, ITM2B, LSP1, HNRNPA1, HSPA5, PTMA, S100A11, CD99, ZNF331, TUBB, LIMD2, PPIB, LDHB, RGCC, CRIP1, EEF2, PPIA, IGLC2, TUBA1A, ATP5MC2, CD44, HBB, HSPE1, CCL4, CALM1, HCST, ID2, ISG20, TRAC, NCL, HNRNPA2B1, HSPB1, LCP1, SLC2A3, GMFG, BTG2, SNHG6, ARL4C, TRBC2, HSPA1B, SRSF7, MTCO2P22, PNRC1, MIF, STK4, DDIT4, H1-10, CALM2, DDX5, RAC2, RBM39, ATP5MG, SELL, PFDN5, ARPC1B, HMGB1, LRRFIP1, ARPC2, YBX1, SSR4, CST7, FOSB, MCL1, PPP1R15A, PCBP2, ZFAS1, C12orf57, MYL12A, FXYD5, CFL1, CD27, H3-3A, TOMM7, RNASET2, UBA52, HINT1, ETS1, RESF1, SERP1, DDX24, ARPC3, UCP2, RGS2, PCBP1, EMP3, ALDOA, CD83, MYL6, SQSTM1, ITM2A, UQCRB, STK17A, TAGLN2, TPI1, APOE, COX4I1, LITAF, HMGB2, GIMAP7, SUB1, SON, LDHA, ANKRD12, CYCS, SRSF5, PPDPF, EVL, GNAS, PNISR, STK17B, STAT3, FOXP1, EZR, CCR7, BIRC3, GABARAP, HMGN2, SERF2, IGHM, RNASEK, HNRNPU, ITGB2, AHNAK, IRF1, EEF1D, CYTIP, RSRP1, GPX1, CSTB, CD2, SRRM2, HNRNPDL, GNLY, RAN, CD8B, PSAP, FUS, FKBP5, CD63, CIRBP, SLC25A5, EIF5, PPP2R5C, SUMO2, HSPD1, NME2, TRIR, GADD45B, SOD2, SMAP2, CDC42, N4BP2L2, TRBC1, COX7A2, YWHAB, BTF3, MBNL1, COX6B1, CD81, JAK1, SDCBP, YWHAZ, USP15, CAPZB, IL2RG, ELOB, TXN, COX7C, TPM3, SEC62, ARL6IP5, GPX4, SF1, MTRNR2L8, PRDX1, LYZ, COMMD6, HMGN1, XBP1, XIST, NR4A2, JCHAIN, HNRNPA3, SFPQ, NACA, SSR2, DDX17, RHOB, EIF4A2, SRSF2, SYNE2, RNF213, SRSF11

- mucosa: LAMA2, EIF2AK4, PPL, KRT13, PDZRN3, PARD3, EBF1, ANK3, ANKRD44, CELF2, ELMO1, PTPRG, ZEB2, BCL2, PLXDC2, RIPOR2, SLC8A1, ZBTB20, TNFAIP8, ARHGAP15, RBMS3, CDC42SE2, RNF144A, XKR4, AUTS2, IKZF1, PRKCB, GNG2, PLCL1, STAB1, NFATC2, TPM1, CHST11, IQGAP2, CD53, HIP1, STAT5B, DOCK10, PDE4D, MERTK, PRKG1, TMC8, ADAM12, AFF3, RBPJ, CD96, FLNA, CALD1, KIR-

REL3, FMNL1, CSF2RA, RGL1, LDB3, CD247, CD37, NCKAP1L, TOX, MYL9, LPCAT1, ITGA1, COL6A2, FN1, SLIT3, TPSB2, WDR86, ACTA2, COL6A1, CYTIP, MB21D2, SH3PXD2B, PKNOX2, ATP1A2, LDLRAD4, PIP4K2A, ABTB2, SHROOM1, FCGR2A, JADE2, ABCA6, MITF, AC007193.10, THBS1, RCAN2, MS4A6A, FYN, ST6GALNAC3, ADGRD1, SEMA3C, GABARAPL1, HCK, MRC1, GNAO1, PDE7B, PTPRM, SPI1, AATK, NR2F2, PDE3A, PLCL2, ARHGAP22, ARHGAP29, CPED1, CH17-340M24.3, RCSD1, KANK2, ID2, FHL1, DOK2, AC062021.1, SGIP1, MN1, SLC6A16, RFTN1, SLCO2A1, ABCG1, NPR2, CTB-161M19.4, SFMBT2, MMEL1, CACNA2D2

- musculature: TTN, B2M, TMSB4X, FTH1, NEAT1, MTRNR2L12, IFITM3, SRGN, FKBP5, SNHG6, ATP5MG, PTPRC, HNRNPDL, CD74, ATP5F1E, EEF1B2, S100A4, UBC, IGFBP7, SAT1, SKAP1, CXCR4, TMSB10, HSP90AA1, N4BP2L2, CD36, RBM47, TSC22D3, ARHGDIB, TYROBP, TXNIP, PRKCB, CLK1, CYBA, VIM, JUN, SNX29, MTATP6P1, UBA52, PHC3, BANK1, ELMO1, MYO1F, ZNF451, CFLAR, C1GALT1, SOX4, NFKBIA, RAP1A, IQGAP2, PIP4K2A, ANKRD12, HPS1, RNASE1, STAT4, LPP, MGST3, ITM2C, MRPL1, PTEN, AIP, DEPTOR, MGAT1, TMEM245, DNAJC6, ADCY7, HMGB1, UTRN, FOS, KMT2B, ZFP36L2, LDLRAD4, ARHGAP15, MIF, SCAF8, UBB, OGA, PRKCE, SBF2, FAU, BCAS3, HMGN1, CD63, UBR3, LAMC1, DOCK8, ZDHHC14, DCLRE1C, CPM, CIRBP, NPM1, SOD2, LY6E, PNKD, MTSS1, HSP90AB1, ARHGAP9, DMXL2, GJA4, CAPG, IKZF1, NR4A2, MADD, CNOT10, TAF15, CREM, EIF4H, HCLS1, HERPUD1, C1QA, MAP3K5, BTG1, RAD23A, CPQ, FOXP1, BUD31, PSTPIP2, MARCKS, SORL1, LAPTM5, ADRM1, GPM6B, ING5, ATP1B3, SF3A1, UQCRB, RGS6, CCL3, LAMA2, CAPN2, ST6GAL1, SERPINB9, TNNI1, HBG1, INPP5D, RBM27, PDK2, MT2A, GNG11, SNCA, DDX54, TMEM60, VCPIP1, TGFBR3, ATP6V1B2, ABCD4, CRYZ, ADGRA3, GREB1L, SOX5

- nose: H3-3B, SLPI, KRT19, WFDC2, CD74, IFITM1, EEF1B2, ACTB, S100A6, B2M, SAT1, FTH1, FOS, TMSB4X, CST3, SRGN, AQP3, HSPB1, FTL, S100A4, VIM, EEF1A1, PIGR, RACK1, IFITM3, CLIC1, NEAT1, CD9, ANXA1, CLU, SERPINB3, TPT1, LGALS3, S100A10, TMSB10, BTG1, GAPDH, S100A11, LCN2, S100A2, MTRNR2L12, TXN, LY6E, NACA, LYPD2, EZR, SH3BGRL3, ZFP36L1, TUBA1A, CCL5, LYZ, ACTG1,

PTMA, ENSG00000275110.1, FAU, PRSS23, NFKBIA, H3-3A, S100A8, JUN, CYBA, GSTP1, ITM2B, CTSB, MT2A, BPIFA1, IFI27, HSP90AA1, IFITM2, PFN1, MIF, ALDOA, PABPC1, TXNIP, XBP1, ZFP36, AGR2, HMGB1, EEF1D, S100A9, GLUL, PTPRC, PSAP, CTSD, LGALS1, ATP5MG, TYROBP, ATP5F1E, UBC, CAPS, UBB, MUC5AC, CD44, CALM2, KLF6, ZFP36L2, PKM, HINT1, HSP90AB1, PPDPF, COX6C, JUNB, TPI1, HNRNPA2B1, CALM1, PPIB, SERF2, ARPC2, UBA52, KRT8, SSR4, PFDN5, MYL6, IGHA1, DDX5, BPIFB1, HSPA5, GPX1, EEF2, GSN, COX7A2, TSPAN1, SLC25A5, CD63, SQSTM1, GNLY, IL32, PRDX5, TUBA1B, COX6A1, F3, DUSP1, AHNAK, TFF3, CD52, ANXA2, MTRNR2L8, TMBIM6, COX4I1, COX7C, TUBB4B, HSP90B1, MT1X, PRDX1, NDUFA4, SCGB3A1, PSME1, PPIA, IGFBP7, CXCL8, HSPA8, CCNI, CSTB, TMA7, SLC25A6, C12orf57, PNRC1, ELOB, PSCA, TAGLN2, APLP2, MYL12B, LAPTM5, SRP14, ATP5MC2, ENO1, MYL12A, DDIT4, KRT18, FXYD3, CIRBP, SKP1, FUS, TIMP1, S100P, TPM4, HNRNPA1, ISG15, EIF1, FABP5, OAZ1, FOSB, CFL1, MUC16, IER2, DSTN, TACSTD2, MSMB, BTF3, CIB1, ATP1A1, ARPC3, H2AZ1, UQCRQ, SARAF, TNFSF10, CD55, DUSP2, PCBP2, YBX1, GABARAP, RNF213, ISG20, RHOA, ELF3, HNRNPDL, JCHAIN, NPC2, GADD45B, RBM3, KRT7, CTSC, ZFAS1, RAC1, SPINT2, COTL1, CANX, SON, CD7, PSMB9, COMMD6, DYNLL1, YWHAZ, ARHGDIB, GUK1, ATP6V1G1, PSMA7, PERP, EPAS1, GNAS, TOMM7, MCL1, COX8A, DDX17, SEC62, MYH9, POSTN, CCL4, TMEM59, IL7R, HMGN2, KRT5, COX5B, NPM1, GPX4, VMO1, SYNE2, UBL5, HMGN1, OST4, EID1, RARRES1, SERPINB1, NCL, UQCRB, CTSS, SEC61B, EIF3K, UCP2, ALDH1A1, MUC4, KTN1, IRF1, ATRX, SNRPD2, LRRFIP1, PARK7, SOD2, ATP6V0C, PPP1R15A, YWHAB, TSC22D3, SERP1, CHCHD2, SRSF5, DAZAP2, POLR2L, PDIA3, ATP5ME, C15orf48, TCF25, NKG7, EIF4G2, PEBP1, VAMP8, NAP1L1, CD99, ALOX15, ARL6IP5, ARGLU1, SELENOH, ATP6V0E1, LCP1, SUMO2, ATP6V0B, CALR, TYMP, LDHB, NDUFB4, UQCR11, CSDE1, HERPUD1, NDUFA1, TPM3, CTSH, RBM39, SYNGR2, FXYD5, ANXA5, ADIRF, DBI, H4C3, PCBP1, CDC42, CD164, BTG2, TSPO, SRSF11, PLAAT4, CAST, HNRNPU, ID3, TMEM123, CNBP, SLC25A3, COX6B1, GNG5, ANKRD12, GRN, DNAJA1, GSTK1, UQCR10, HNRNPK, TNFAIP3, RAB11FIP1, ALDH3A1, NOP53, PSME2, SCGB1A1, RNASET2, HNRNPH1, APP, ATP5MF, MS4A1, ATP5MC3, REEP5, COX7B, ANXA11, SGK1, SAA1, EIF5, CD46, P4HB, ITGB1

- omentum: SRGN, LAPTM5, MTRNR2L12, PTPRC, CD74, EEF1G, PTMA, HBB, SNHG29, NEAT1, FTL, FCER1G, VIM, KLRK1, IFITM3, SAMSN1, B2M, ARHGDIB, PTPRCAP, LCP1, GAPDH, SERPING1, MTATP6P1, S100A8, ZFP36, JUN, CTSB, CREM, CCL4, CXCL8, FTH1, CD69, GABARAP, SNHG6, S100A4, TM4SF1, OGN, SNHG1, TMSB4X, LGALS1, TYROBP, ZNF271P, CYBRD1, DNAJB1, ANXA1, MGP, CCDC80, BCL2A1, TCF4, C1orf35, PLAUR, DDX5, TOB1, HP, RGS2, EFEMP1, EGR1, HMGB1, RNF11, SNHG5, SLPI, ADH1B, RNASEK, ITLN1, IGFBP7, CCDC18-AS1, ALOX15, DCN, RALB, CLU, TMSB4XP4, ARGLU1, ZNF16, AQP9, CPB1, SOX4, CMTM2, PLIN2, ALDOA, VNN3, CCNL1, PLA2G2A, PRNP, C3, ATP6V0C, PATL1, CTD-2287O16.1, FCGR3B, SPARCL1, C3orf86, CBY1, SEMA3C, MEDAG, COL1A2, EEF1A1, LRIG3, CLN8, MLLT10, AC004057.1, TMSB4XP8, BDH2, ITPKB, LMCD1-AS1, SMIM3, SCARNA9, SELENOP, MACO1, LGMN, SH3BP5, ARRB1, SDS, TRIM47, RBAK, CD36, SERPINF1, PELATON, JAG1, ARL4A, SOCS6, LYVE1, RASSF10, ERN1, SNHG16, THBD, IL7R, SH2D3C, ADIRF, GEM, INF2, THSD7A, S1PR1, COX7A1, ABCB1, SLC8B1, IL1B, BEGAIN, CROCC, TPSAB1, CACNA1C, ACVR1C, SOST, GREM2, HES1, GPR183, GJB6, GLE1, PLVAP, TFPI2, CXCL2, COL4A1

- ovary: SRGN, CD74, CYBA, TYROBP, RGS1, VIM, IGFBP7, B2M, FTL, EEF1A1, GAS5, NEAT1, CXCR4, TMSB4X, FABP5, TMSB10, UBB, JUNB, XIST, S100A6, TSC22D3, HSPE1, FTH1, SKP1, SNRPE, IFITM3, BEX3, TPT1, FOS, STMN1, RNASE1, DSTN, GPX1, NDUFA4, ANXA1, DUSP6, CST3, ZMYND8, CD36, SOX4, MTATP6P1, PDCD5, HBA2, JUN, PCLAF, TMEM163, ATP5F1E, LGALS1, MTRNR2L12, TBXAS1, TNFAIP3, PIK3R6, HPGDS, CD44, ADIPOR1, EIF4A2, PABPC1, ALAS2, GPR183, CH17-373J23.1, ZNF75A, RASGEF1B, DCN, ARL6IP1, CXCL8, HSPB1, NCF1, HNRNPL, HBB, SLC25A6, RNF213, RSBN1, S100A4, RASGRP2, MEF2C, KLF6, SYNE2, ISG20, PTMA, MAGED2, SULT1B1, SON, GAPDH, LAPTM4A, GTSF1, MRPL42, COX6B1, NDUFA1, SRSF7, DGCR2, NAA16, TMEM128, NONO, CALD1, TBPL1, AGTPBP1, PLCG2, GSTA1, IGKC, MT2A

- pancreas: TTR, ATP5F1E, H3-3B, SSR4, VIM, GMFG, FTL, SPINK1, B2M, DDX5, LGALS1, IFITM3, CD74, TUBA1A, STMN1, PABPC1, EEF1A1, ZFP36L1, SPARC, H2AZ1, SNHG6, EGR1, ZFP36, S100A4, SFPQ, DNAJA1,

INS, DST, HBG2, GAS5, ATP11A, DDB1, RGS2, CD53, FOS, LDHB, SNHG5, RBPJ, HMGB1, S100B, NEAT1, DCN, TCF4, JMJD1C, AIF1, ZMYND8, PRSS2, SOX4, SRGN, LST1, NREP, CXCL13, TIMP2, LSP1, S100A11, AUTS2, SBF2, ACTN4, EVL, EGFL6, NRXN1, RAC2, XIST, FERMT3, ZEB2, PTN, MAML3, IGFBP5, PTMAP5, HCST, TMEM117, FN1, SOX2-OT, ANXA6, CCL4, LAPTM5, TMSB4XP8, FOSB, MBNL1, ZC3H11A, EFEMP1, GNAI2, STMN4, FTX, BMPR2, IGFBP4, SST, REG1A, HBA1, IGF2, HMGA2, EFNA5, LMBR1, RERE, TMSB4XP4, DPYSL3, C1QB, STK32B, CARD8, TRAC, NNAT, MAF, GGTA1, RAD51B, KCTD7

- paracolic gutter: B2M, PGM1, CAPN7, GRN, APP, IRF8, CD69, CST3, FTH1, FAM3C, LAPTM5, CTSB, SOX4, SPARC, TMEM176B, CSF2RA, SERPINB9, IFITM3, TMEM62, ACTN1, NFIA, TLR2, QTRT1, LTC4S, CXCL12, CD83, MTHFD1L, S100A6, TAGLN2, PPFIBP1

- parietal peritoneum: IGKC, IGFBP7, CD9, CD74, TYROBP, CD69, CD63, XBP1, RAC2, PTPRC, MZB1, DYNLL1, IGLC2, CEBPB, JCHAIN, ACTG1, TCF4, RHEX, VIM, FTL, H1-3, PPP4R3A, IGLC3, IGHG3, GID8, RBPJ, IFITM2, CD40LG, SOCS1, SH3BGRL

- peritoneum: SRGN, UBXN11, ANXA2, IFITM3, CD74, CD69, APP, IGKC, PTPRC, A2M, MARCKSL1, FTL, CALR, TYROBP, CXCR4, TCF4, FAU, ACTG1, FCER1G, NAP1L1, PECAM1, C20orf85, KIT, GZMB, PLCG2, ZNF780A, RGS2, CD93, CD37, FOXN2, IFI30, BCL2A1, CD84, FLT4, CD53, CLEC14A, CTSG, CCDC18, C1orf162, CLDN5, MMP25, DOCK3, FCN3, HERPUD1, IRF8, TIE1, LAIR1, SPN, IGLC2, CCDC88A, RALGPS2, CYSLTR1, SLC11A1, FAM169A, ASCL4, ECT2L, AC006129.2, CSF2RA, CCL5, AIF1

- placenta: FTL, FTH1, VIM, TMSB4X, DCN, GNLY, IGF2, HSPB1, EEF1A1, COL3A1, KRT18, IFITM3, B2M, NEAT1, IGFBP7, CD74, LAMA2, TUBB, PTMA, SAT1, HBA2, FOS, SRGN, CGA, S100A4, GPX1, CCL4, TYROBP, ACTB, TPT1, CDKN1C, ADAM12, SPP1, APOE, CST3, LGALS1, S100A6, MEG3, SOD2, FN1, H19, TUBA1B, GPX3, JUNB, NKG7, ID2, DLK1, IFI6, CSH1, COL1A1, TIMP3, TMSB10, ANXA1, BTG1, S100A10, CCL3, S100A11, PLCB1, ACTA2, PSAP, ACTG1, ZBTB20, VCAN, DAB2, IFITM2, MT2A, TIMP1, IGFBP3, COTL1, FLT1, GAPDH, LUM, H3-3A, COL6A2, KLF6, IGKC, UBC, CALD1, GPC3, HPGD, TXN, PEG10, TPM1,

ITM2B, RNASE1, HBG2, EGFL7, LYZ, ZFP36L1, COL4A1, JUN, ZEB2, EPAS1, CTSB, CXCR4, F13A1, CYBA, AIF1, PAPPA, IL32, SLC25A6, COL4A2, C1QA, CXCL14, DNAJB1, UBE2D2, DUSP1, SLC2A1, ZFP36L2, ANXA2, H3-3B, SLC2A3, KRT19, TFPI, HSP90AA1, PAEP, CTSD, HINT1, SELENOP, HBA1, NFKBIA, PLXDC2, MBNL1, MGP, HNRNPA2B1, UBB, HMGB2, CD52, CXCL8, CCL5, ZFP36, GSTP1, YBX1, PFN1, CALM2, RGS1, HSPA1B, BSG, COL6A1, EEF2, IL6ST, SERPINE1, XIST, RTN4, NUPR1, S100A9, SERPINA1, CYP19A1, WSB1, PRKG1

- pleura: ITLN1, S100P, CD74, LYZ, ARHGDIB, FTL, S100A9, DEFA4, TIMP1, PSAP, HP, CTSB, HSPA1A, CD163, B2M, COL3A1, SLPI, IGKC, DCN, COL6A3, S100A8, SRGN, CXCL14, CXCL8, C3, TMSB10, IGFBP7, MT1E, SEMA3C, HSP90AA1, CD52, IFITM3, TPT1, GSN, HSPE1, FTH1, EEF1A1, MTATP6P1, MT1X, SSR4, CCL4, GPNMB, TSC22D3, AIF1, MT2A, ANXA1, EIF1, C1S, HSP90B1, TMSB4X, GLUL, CD63, UBB, AREG, DUSP1, TXN, MS4A6A, RNASE1, VIM, CXCL2, CCDC80, IGLC2, TIMP2, PRG4, LCP1, ANXA2, BCL2A1, ALOX5AP, LCN2, MMP2, PFDN5, ADIRF, SNHG5, IFITM2, HNRNPK, CREM, NNMT, NAMPT, PFN1, PLAC8, TGFBR2, GNLY, CCL21, DEFA3, SH3BGRL3, EFEMP1, COMMD6, RGS2, H3-3B, COX6C, NEAT1, ATP5MG, MGP, IL1B, IGFBP6, HNRNPA2B1, PTMA, NPC2, S100A4, ADH1B, ATP5MK, BTF3, PABPC1, MPO, TIMP3, PLA2G2A, SFRP2, SARAF, UBC, SOD2, SCARA5, HSPA1B, SPARC, CXCL12, GSTP1, TUBA1B, FABP4, ACTG1, FKBP5, S100A10, GNAS, LUM, S100A6, SDCBP, FOSB, IGHG1, EEF1B2, MFAP5, MGST1, HSPA8, DNAJA1, ACTB, EGR1, CXCR4, FAU, NR4A1, YWHAB, ITM2B, CCL20, SPARCL1, CCL5, MRC1, PLAC9, MYL12A, ATP5MC2, EIF4G2, ATP5MJ, TMBIM6, MCL1, S100A12, CTSS, LST1, ATP5ME, RACK1, C1QA, COL1A2, RAN, YBX1, ID2, PNRC1, RHOA, EID1, CFL1, G0S2, EIF5, SAMHD1, COL1A1, MT1G, SNRPD2, HNRNPC, EREG, APP, SLC2A3, SRP14, SAMSN1, TPI1, SELENOP, FSTL1, COX6B1, CXCL3, H2AZ1, LITAF, CST3, NFKBIA, DBI, PPIA, HSPB1, TYROBP, MFAP4, TOMM7, ZFP36L1, UBE2D3, APOD, CALM2, FPR1, ELOB, SELENOK, UBA52, ANXA5, LTF

- pleural fluid: CD74, CST3, FTL, TMSB4X, TYROBP, MTCO2P22, EEF1A1, SSR3, TMSB10, PHACTR1, S100A8, MARCKS, VMO1, ATP1B1, MTCYBP19, B2M, ARHGAP24, TCF4, ZNF93, GTSE1, GSN, DMD, IGHM, BCL11A, RNF40, HSPB1, BCAT1, ATP5F1E, CSF2RA, MOB1B, BEND5,

NKX2-1, CENPE, FABP5, PKIB, CTSL, FCER1G, ALOX5, HAMP, AP000459.7, RNASE1, A2M, PIR, NKG7, PTGR1, PELATON, SMPDL3A, SPP1, SPIB, LYZ, RAD52, MYCN, GRN, ZDHHC23, BOLA2B, ZNF532, CCL5, IGLC3, ZNF736, H4C3, NNAT, MNDA, LGMN, LAPTM4B, BLVRB, SEM1, YBX3, IFT81, IFITM3, DHRS9, CD68, EFNB1, CD300LF, CSF1R, WASH9P, IL32, TUBA1B, TIMP1, IER5L, ZNF124, S100A9, ENPP2, ACRBP, GAB2, ADAMDEC1, PTGS1, C1orf162, GNG11, ETV5, LGALS2, GLUL, APOC1, GSTO1, CD180, CLEC10A, C1QB, CYP2S1, TP63, ACOT9, CD302

- prostate gland: KLK3, TPT1, FAU, FTH1, SRGN, VIM, CD74, LGALS1, IGFBP7, MYL12A, PSAP, HSP90AA1, CXCR4, SAT1, TXNIP, BTG1, S100A4, MBNL1, PTPRC, TIMP1, TMSB4X, CALD1, LAPTM5, NEAT1, TSC22D3, TCF4, LPP, SP100, SSH2, B2M, TGFBR2, CELF2, DOCK8, GPX1, DUSP1, SPARCL1, ZEB2, ELMO1, FTL, STK4, CCL4, CCL5, PIKFYVE, FCER1G, SARAF, JUN, SSR4, PLAC8, GNAQ, HNRNPH1, GMFG, SAMSN1, HSPD1, FYN, TIMP3, UTRN, ZFP36L2, CD37, HNRNPA2B1, APBB1IP, UBE3D, ZBTB20, ACAP1, MT2A, GYPC, KLRD1, NKG7, PTMA, CDC42SE2, RGS1, GADD45B, GIMAP4, REL, EPB41L3, IKZF1, KYNU, MYL9, HGF, LHFPL6, KDM2B, KLF2, WSB1, PTP4A1, UTP11, MEF2C, KLF12, ZCCHC10, NFIA, CAVIN2, RORA, PKHD1L1, TFPI, TYROBP, IER5, CADPS, ARHGAP24, GNLY, FNBP1, FLI1, FMN1

- reproductive system: COL1A1, SPARC, MDK, B2M, CD74, RNASE1, VIM, TMSB4X, S100A11, FTL, HSPB1, IFITM3, MTRNR2L8, LGALS1, SPRR2F, IGF2, ANXA2, DNAJB1, KRT14, IFITM2, EEF1A1, MTRNR2L12, ZFP36L1, MARCKSL1, S100A6, S100A10, CST3, FOS, FTH1, ARHGDIB, STMN1, COL1A2, TMSB10, IGFBP7, BST2, SRGN, DLK1, DCN, JUN, ZFP36, BTG1, EEF1B2, TUBA1B, TYROBP, HSPA1A, APOE, MIF, BEX1, SAT1, CD9, HSP90AA1, JUNB, NFKBIA, IFITM1, TIMP1, ID3, PTMA, MYL9, AIF1, GNAS, FABP5, ACTB, NPY, SOX4, H3-3B, H4C3, NEAT1, MT2A, COL3A1, S100A8, PLP2, PRXL2A, HSPA8, LY6E, IGFBP2, BCAM, KRT10, HMGN2, SNHG8, TXNIP, PRDX2, SH3BGRL3, H1-0, CAV1, NPM1, HNRNPA1, PSAP, S100A4, ITM2B, PPDPF, CD63, BEX3, RBP1, MARCKS, DUSP2, MEF2C, PCBP1, GPC3, HMGB2, NPC2, IGFBP4, HSPA1B, TUBA1A, UBB, DUSP1, SPINT2, MEG3, H1-10, CD99, IER2, TAGLN2, RHOB, CYBA, DDIT4, PERP, ANXA5, PKM, ZFAS1, GADD45B, PFN1, TPT1, SELENOP, NREP, TPM2, LAPTM4A, HSPE1, CKS2, ACTG1,

TPM1, PCBP2, KLF6, UBC, ATP5F1E, KRT8, GYPC, APOA1, GATM, MAGED2, TSPO, CALM2, H2AZ1, CRABP2, EGR1, TSC22D3, NR2F2, ANXA1, VCAN, HSPD1, PPP1R15A, CALR, MGST3, HMGB1, MFAP2, GABARAP, MYL6, SLC25A5, GSTP1, TPM4, GAPDH, SEPTIN7, GRN, FCGRT, CCN1, IGFBP5, DMKN, ENO1, PRDX1, CEBPD, YBX1, HSP90B1, TXN, HOPX, PNRC1, ID1, GSTA1, NCL, TUBB4B, HMGA1, NME2, EMP3, SOD1, PPIB, DUT, S100A9, CALD1, HMGA2, CTSC, LDHA, TPI1, COMMD6, MEST, ZFP36L2, SLC40A1, UQCRB, CITED2, KRT18, ID2, HSP90AB1, C12orf57, DDX5, NDUFS5, FXYD5, DDX24, SPP1, LSP1, TSC22D1, TKT, CD81, HSPA5, COTL1, EEF1G, CYSTM1, GLUL, SLC2A3, HNRNPA2B1, ATP5ME, GSN, DEK, DST, CALM1, CLIC1, POSTN, SRSF9, TUBB, SQSTM1, PLD3, NAP1L1, RAC1, AHNAK, BTG2, BTF3, CCL4, EEF2, MYL12A, PPIA, CD24, H3-3A, SFRP1, COL6A2, COX6B1, COX6C, DYNLL1, CPE, KDELR1, NR4A1, SERPINF1, MT1X, DBI, RGS2, ATP5MK

- respiratory system: LYPD2, WFDC2, SLPI, H3-3B, CD74, VIM, FTH1, B2M, PTMA, NEAT1, CST3, TMSB4X, KRT19, S100A6, S100A4, S100A11, ACTB, SCGB3A1, IFITM3, CD9, H3-3A, SAT1, EEF1B2, CD63, SCGB1A1, HSP90AA1, ANXA1, TPT1, IGFBP7, BTG1, S100A10, RACK1, GAPDH, CLIC1, LYZ, EPAS1, FTL, PTPRC, ZBTB20, MT2A, TMSB10, SRGN, EEF1A1, LGALS1, FOS, JUN, LCN2, UBA52, CSTB, CYBA, SERPINB3, CELF2, CALM2, S100A9, TIMP3, TXN, TXNIP, FAU, PIGR, ACTG1, CLU, TIMP1, KLF6, S100A2, LGALS3, ZFP36L1, ITM2B, MTRNR2L12, IFITM1, DSTN, SOX4, ALCAM, CD44, IFITM2, CALM1, BPIFB1, PRSS23, MSMB, GSTP1, CD55, ANXA2, SOD2, ZFP36, UBC, HSP90AB1, GNAS, PABPC1, NPM1, UBB, JUNB, SFTPC, MIF, TUBA1B, PFDN5, SSR4, LPP, PFN1, ZFP36L2, EZR, FABP5, GSN, GLUL, SAA1, CCL5, H2AZ1, DDX17, ELF3, TFF3, AQP3, HSPB1, MGP, CTSB, RAB11FIP1, EEF2, NPC2, EEF1D, DDX5, CFL1, SLC25A6, YBX1, DYNLL1, HSP90B1, DUSP1, COX7C, SERF2, MBNL1, LY6E, GPX1, ALDOA, NFKBIA, TCF4, MYL6, CXCL8, JUND, TSC22D3, HMGB1, CTSC, SQSTM1, RORA, TUBB4B, ATP5F1E, CALD1, LRRFIP1, SH3BGRL3, MT1X, KRT17, ATP5MC2, HSPA8, HMGN3, IGKC, PPDPF, NAP1L1, COX7A2, PSAP, LITAF, AKAP13, EIF1, LMNA, TUBA1A, IFI27, FOSB, HNRNPA2B1, PKM, RTN4, VMP1, ARPC2, PTPRG, HSPA5, MYL12B, UQCRQ, BTF3, TACSTD2, SYNE2, TMBIM6, S100A8, SPARCL1, TMA7, OAZ1, PSME1, PPIA, GADD45B,

AHNAK, PRDX1, MDK, SRP14, SARAF, IGFBP5, FOXP1, DST, XBP1, CAST, SLC25A5, NACA, PRDX5, FKBP5, RGCC, AGR2, ATP1A1, MUC5AC, HNRNPA1, MYL12A, ZEB2, PNRC1, FUS, NAMPT, MGST1, GPX4, IER2, NDUFA4, ANXA5, CEBPD, CCL4, ADIRF, UQCRB, YWHAZ, TUBB, PNISR, CXCL1, COX6C, HINT1, RHOA, ENO1, APP, TOMM7, PCBP2, MACF1, CD99, C15orf48, N4BP2L2, ID3, CFLAR, SLC38A2, CD24, EIF4A2, ASH1L, SRSF11, SFTPB, DDIT4, COX4I1, UBE2D3, ATP5IF1, APLP2, ARPC3, TYROBP, SKP1, TAGLN2, PLXDC2, JMJD1C, RARRES1, KRT18, CIRBP, NFE2L2, TSPO, CHST9, SON, NCL, LDHB, POMP, UBL5, H4C15, MUC16, SET, SEC61G, HNRNPDL, TNFAIP3, TMEM59, HNRNPU, KRT7, APOD, UQCR11, JAK1, CIB1, ARHGDIB, IL32, TPI1, MYH9, NUCB2, SERPINB1, CDC42, PEBP1, RNF213, CXCL17, UTRN, TPM4, CD52, HERPUD1, COMMD6, GNG5, RAC1, CYB5A, HMGN2, ATP6V0E1, ID2, MCL1, FXYD3, CTNNB1, PPIB, RBM47, SEC62, AKAP9

- saliva: SRGN, CXCL8, S100A9, FTL, CD74, VIM, S100A8, FCER1G, ANXA1, IFITM2, C15orf48, NEAT1, LGALS3, S100A10, SLPI, TMSB4X, FGF23, TIMP1, B2M, NAMPT, TMSB10, S100A11, SAT1, CCL4, S100A6, FTH1, CSTB, IL1RN, WFDC2, MTRNR2L12, CXCL2, TACSTD2, FOS, ACTB, JUNB, EMP1, DUSP1, EEF1A1, CEBPB, FABP5, ZFP36, MARCKS, LGALS1, S100P, PLAUR, PCBP1, TXN, H3-3B, CTSB, PTPRC, IL1B, KRT19, CCL3, KATNBL1, CCL3L1, CTSD, MT2A, JUN, NFKBIA, EMP3, GADD45B, ITM2B, MXD1, ALOX5AP, LYN, TYROBP, LCN2, CD44, SERF2, ZFP36L1, MTRNR2L8, PNRC1, G0S2, CTSS, PFN1, CXCR4, ACTG1, EIF1, H3-3A, EGR1, CD9, CD55, SCGB1A1, S100A4, FOSB, IFITM3, CCL4L2, ISG15, ITGAX, GLUL, EREG, DDX5, SPRR3, PTMA, PPP1R15A, GSTP1, NFKBIZ, LITAF, PHACTR1, TPM4, IFI6, HMOX1, CDC42EP3, TPT1, FAU, IFI27, APOC1, MYL6, CST3, TNFAIP3, F3, TSC22D3, CD24, SQSTM1, BASP1, ZEB2, OAZ1, LMO7, HSP90AA1, MTRNR2L1, AREG, CCL5, SLC25A37, MAFF, TUBA1A, CXCL3, NOP10, HSPA1A, RAB11FIP1, LLNLF-96A1.1, ASAH1, KLF6, DDIT3, HSPA5, JUND, PPIF, ZFP36L2, CXCL1, IFRD1, COTL1, GMFB, UBC, SRSF3, KLF2, IER2, RNF213, CD63, GPCPD1, GPM6A, ARPC3, BCL2A1, KDM6B, PSAP, AIF1, BRI3, IFIT2, SLC11A1, ATP6V0B, TUBB4B, PLEK, MYL12A, CALM1, YBX3, HES4, NPC2, SRSF5, UBA52, ANXA2, ISG20, CLDN4, HSPA1B, SGK1, IER3, HNRNPH1, MIDN, RAC1, LAPTM5, TMBIM6, MUC4, CALR, SDCBP, ATP6V1F, RHOB, SERPINB9, PELI1, PPP1CB,

MTRNR2L11, PLAU, CYBA, ID2, BTG1, MCL1, TOB1, CCL20, PI3, FPR1, SPP1, HIF1A, NIBAN1, BRD2, UBE2D3, RESF1, FAM177A1, ATP6V1G1, ADM, PMAIP1, YWHAZ, NACA, KLF4, FYB1, CYSTM1, IRF1, TALDO1, PTGS2, HNRNPC, KMT2E, LCP2, DDX3X, LCP1, PHLDA1, SERP1, SERPINA1, VMP1, CD83, SUB1, NABP1, FCGR2A, C5AR1, SERPINB1, ATP5F1A, IER5, TPI1, MAP1LC3B, MAP2K3, TMEM59, LRRFIP1, CTSC, ATP2B1-AS1, CD164, SRSF2, BTN3A2, STXBP2, SLC38A2, ATP13A3, RAB7A, SAMSN1, GNLY, TRIB1, TAGLN2, TNFAIP2, APOE, FXYD3, APLP2, CLIC1, POLR2A, RTN4, SELENOK, GRN, ADGRE5, ZFAND5, ATP5F1E, TSPAN1, PRRG3, PHLDA2, GNG5, B4GALT1, PROK2, AQP9, RNF13, CD47, CXCL16, TXNIP, ABHD5, IRS2, GUK1, DDX17, CTSL, CCNL1, KCNQ1OT1, ZFAS1, CPEB4, GK, SERPINB2, ACTR3, DNAJB6, CAPS, WTAP, RILPL2, KRT17, EFHD2, CDC42, ARPC5, ELF3, SKIL, SLC43A2, BPIFB1, RSAD2, GBP1, ANKRD12, CCRL2, ETS2, CLEC2B, MTRNR2L10, LRRC75A, H1-10, CSF3R, DNAJA1, MYL12B, HNRNPA2B1, PABPC1, DNAJB1, S100A12, STK4, RHOA, IRF2BP2, ECM1, GAPDH, PLCG2, CFLAR, CIB1, SLC2A3, MYO10, SH3BGRL3, ANXA5, HSP90AB1, MARCHF6, JARID2, ARRB2, PRDX5, DYNLL1, HM13, SLC16A3, FNTA, RACK1, MAL, CLEC7A, AGR2, PLSCR1, HINT1, HCAR2, EEF1D, JMJD1C, SNX10, CD53, MDM4, HOPX, CHCHD2, HNRNPU, PFDN5, SMCHD1, MYADM, GBP2, HSP90B1, H2AC6, PTPRE, CEBPD, BCL6, UBB, SOX4, RBMS1, GABARAPL2, PPIA, ANP32A, ELOB, ACSL1, ARPC2, ZNF292, GPRC5A, RASGEF1B, CREM, RBM39, SPRR2A, EIF5, PKM, GPX4, RGS2, SRP14, NR4A2, VEGFA, ATP6V0E1, CSNK1A1, PERP, CYRIB, SMG1, CFL1, RGCC, LYZ, PTP4A2, MAP3K2, HSPE1, REL, YWHAB, PPP1R15B, DAZAP2, LDHA, HSPA8, VAPA, CD59, UBE2B, HCST, BTG2, PIM3, ABCA1, SPI1, RNF149, IFIT3, NCALD, FUS, WSB1, TSPO, CALM2, CREBRF, STK17B, HERPUD1, ASAP1, PARP14, ZNF207, TMEM154, ATXN1, IGSF6, PLEKHB2, FOXP1, NFE2L2, NCF1, CPD, BZW1, ATP2B1, UBL5, SLC20A1, HMGB1, TMEM258, COX6B1, SON, AC058791.1, BTBD9, SPAG9, COX8A, FOSL2, CDKN1A, N4BP2L2, GMFG, RALA, YPEL5, RBM47, SYAP1, UBE2S, TPM3, SERPINB3, GSTO1, CD48, CMTM6, LY96, H1-2, EVI2B, CYTH4, DNTTIP2, TMA7, PLAC8, TANK, ERBIN, OPHN1, TNFAIP6, LYPD2, ATF3, HSPH1, CLTC, TNFRSF1B, OSBPL8, ANKRD36C, CD68, GRINA, CTNNB1, AAK1, MGAT1, EZR, RIOK3, ARF1, FAM133B, ATP5MG, SERINC1, IVNS1ABP,

GPBP1, TAX1BP1, UGCG, PPP1R10, LIMS1, EIF4A2, EIF5A, CSRNP1, JPT1, FGR, FNDC3B, CNNM2, ITGB8, ZNRF1, PDLIM5, COX7C, PLIN2, LSP1, PELATON, CD14, CORO1C, EIF4E, RAP1B, RND3, CCR1, TFRC, FAM107B, C4orf3, GSN, RB1CC1, TREM1, SOCS3, SUSD6, MUC16, GLIPR2, MAFB, MBP, ARHGDIB, CCNI, CLEC4E, RBM3, YWHAE, PLEKHA3, SRRM2, RAB21, OGA, KCNJ2, INSIG1, EIF4A1, COMMD6, MAP3K8, ABCC9, MAPK6, COX4I1, HNRNPA3, NCF2, RBPJ, THUMPD3-AS1, LPP, YIPF4, TNFSF14, HSPB1, ACTR2, CHD2, CYTOR, LIMK2, GSTA1, CAST, GNAS, AP1M1, ENO1, TYMP, RAB1A, WBP2, ALPL, INTS6, HNRNPH3, C15orf40, FOXO3, TMPRSS11B, HNRNPF, OXSR1, CLK1, NEDD4L, TOR1AIP2, AKAP13, KCNJ15, TIMP2, RBM23, GCA, CHMP1B, BAZ1A, RSRC2, YBX1, SLC35E3, PPP2R5C, UBALD2, YPEL3, PRR13, RNF141, KLF10, SARAF, MX1, ATP6V1D, FGF13, SP110, RYBP, CAPZA1, AQP3, MTRNR2L6, TAPBP, PFKFB3, SKP1, ATP6AP2, RNMT, ATP6V1B2, RTN3, LPCAT1, LUCAT1, TNFSF13B, SP100, TRA2A, TMEM123, GNAI2, MNDA, HES1, QKI, ARHGAP26, TNFRSF14, TMEM50A, NINJ1, BHLHE40, EIF1B, NDUFV2, IFI16, ZFAND6, CANX, BTF3, LAPTM4A, ANXA11, SEPTIN7, OSM, SMAD2, ZFYVE16, ELOC, H2AZ1, ARID5B, C9orf72, USP15, ELF1, TMBIM4, WAC, USP8, FNIP1, UPP1, ZNF121, ARL6IP1, CSF1, LMNA, DDX60L, CTA-212A2.3, SPG7, RIT1, DYNC1H1, RSRP1, EHD1, NAP1L1, CNBP, ERGIC1, CAPN2, DUSP6, NFAT5, BID, FLNA, PCF11, HSPA6, ARF6, PNPLA8, TOP1, SYNE2, PRRC2C, TXNRD1, CAPZB, P2RX4, DUSP5, UBE2D1, RAB2A, CSTA, PICALM, ANKRD28, NUMB, DDX3Y, BAG1, STAT3, C6orf62, CIR1, AZIN1, DUSP4, KCNK6, LRP10, CITED2, RASSF5, SLAMF7, GBP5, LST1, PLEKHG2, SLA, LEPROT, SPRR2D, KRT23, PRDM1, KRT18, IDS, PSMA7, UQCRB, HNRNPH2, VASP, IQGAP1, HNRNPK, FLOT1, RAB11A, PIGR, PTP4A1, GNS, EIF4A3, YTHDC1, ITGB1, NFKB1, PTBP3, AKIRIN2, RHOG, PPDPF, HEXIM1, GNAI3, HMGA1, CEACAM6, RBBP6, PGK1, PTEN, PLK3, METRNL, TGFB1, ARL4C, NPM1, MYLIP, CAPZA2, FKBP1A, BST2, NKG7, WARS1, SERTAD1, MDM2, NBN, RRAGC, COX6C, FN1, UBE2R2, TMOD3

- scalp: KRT10, LGALS1, VIM, CXCL14, CALML3, S100A2, KRT14, APOE, ATP1B3, S100A9, KRT5, KRT15, AQP3, CALML5, LY6D, SAT1, TUBA1B, CD74, SPINK5, NFKBIA, STMN1, IFI27, S100A6, CSTB, TACSTD2, S100A8, HOPX, DSP, H2AZ1, KRT1, KRTDAP, RBP1, TMSB4X,

TK1, SBSN, HES1, DST, FTL, MGST1, EEF1A1, TMEM45A, HMGB2, KRT2, S100A7, SOSTDC1, HMGN2, ALDH1A1, TXNIP, DCT, ZFP36, IER3, DEFB1, TPPP3, DEK, IFITM3, DUT, CD59, ADM, IGFBP7, GPX4, TMSB10, S100A14, TM4SF1, KRT16, ID1, CCND1, ANXA1, FOS, CCL27, CALD1, CHCHD10, CSTA, CST3, CAV1, PTTG1, HSPA1A, HINT1, FABP5, TPT1, KRT17, ID2, BRD2, HMGB1, SLPI, IGFBP3, KRT6A, CCL2, RGS2, TUBB, HNRNPA1, SOD1, ATP1B1, PMAIP1, LGALS7, CEBPB, DMKN, FGFBP1, NDRG1, SFRP1, JUN, KLF6, S100A16, SOX4, DAPL1, LSM3, PCNA, EEF2, IER2, TYMP, DUSP1, MAFB, ZFP36L2, CXADR, TPM1, ADRB2, PNRC1, B2M, KLK11, MT2A, FRZB, KLF4, DBI, MYC, SFN, ANXA5, SPRR1B, CKS1B, S100A10, CLDN4, PMEL, TUBA1A, COL17A1, EMP2, PCLAF, FTH1, LGALS3, CRNDE, IMPDH2, CD63, PTMS, HSP90AA1, C12orf75, C1orf21, AOPEP, TOMM7, ID3, JUNB, EPHB6, MZT2A, GAPDH

- skeletal system: COL5A2, B2M, FTL, TMSB4X, FTH1, EEF1A1, S100A6, TMSB10, CST3, CD74, SRGN, SPP1, LYZ, PCOLCE, UBB, MIA, COL1A1, LUM, S100A8, HAPLN1, PPIA, MATN1, HBA2, MYL6, LGALS1, NDUFA4, COL3A1, MGP, S100A4, COX6C, ACTB, EPYC, NACA, DCN, ATP5MC2, FOS, MYL12A, AIF1, IFITM3, VIM, HSP90AA1, SAT1, COX7B, SERF2, HBG2, GAS5, HSPB1, TPT1, COL1A2, COX7A2, RACK1, EEF1B2, ATP5MK, OST4, ANXA2, S100A9, SEC61G, ATP5MJ, EIF1, PTMA, MYL12B, TIMP1, DDX5, ANXA1, TMA7, COL2A1, POSTN, COX4I1, STMN1, H3-3B, IGFBP7, COL12A1, CKS2, CALM2, NPM1, HSPA8, YBX1, COX7C, SNHG29, FAU, MTATP6P1, ARPC3, ATP5F1E, RGS2, EMP3, UBC, CFL1, SH3BGRL3, ECRG4, ITM2A, ATP5MF, ACTA2, LAPTM4A, COL5A1, ATP5ME, CSTB, ACTG1, SF3B6, CCL4, ATP5MG

- skin of body: COL1A2, SPARC, LGALS1, B2M, SRGN, NEAT1, VIM, CD74, TMSB4X, FTL, S100A6, EEF1A1, HSPB1, ACTB, NFKBIA, S100A10, DCN, FOS, SAT1, S100A4, CST3, TXNIP, PTMA, DNAJB1, TMSB10, BTG1, ACTG1, MT2A, ZFP36L1, TPT1, H3-3B, DUSP1, FTH1, LINGO1, S100A11, HSP90AA1, NPM1, ANXA1, HSPA1A, TSC22D3, GAPDH, IFITM1, H2AZ1, HBA2, ZFP36, CD99, JUNB, ZFP36L2, GSN, ANXA2, HMGN2, IGFBP7, ARHGDIB, COL3A1, EEF1B2, GSTP1, KLF6, PFN1, TCF4, LTB, STMN1, TUBA1B, IFITM2, CD44, CXCL14, CD52, GLUL, HERPUD1, ITM2B, H3-3A, HSPE1, GADD45B, CD63, HSPA1B, LYZ, TIMP1, IL32, ATP5MC2, CD9, JUN, SOD2, LDHB, UBC, MIF, EIF3E,

TXN, H4C3, HMGB1, SLC25A6, UBB, FKBP5, CXCL8, HSPA8, HNRNPA1, CALM2, PPDPF, IFITM3, EIF1, PABPC1, TUBB, PNRC1, IL7R, DDIT4, COL1A1, DDX5, LMNA, PSAP, NR4A1, CCL4, HMGB2, AIF1, PPIA, HBA1, TIMP3, ID3, SLC2A3, TUBA1A, PCBP2, COL6A2, NACA, PPP1R15A, HSP90AB1, LAPTM4A, KRT14, MYL6, SH3BGRL3, FOXP1, PTPRC, CALM1, LDHA, KRT10, EEF2, JUND, CYBA, SERF2, ARID5B, DUSP2, CALD1, TAGLN2, MEF2C, HSPD1, CIRBP, HSP90B1, SUB1, CXCR4, AHNAK, RASGEF1B, SOX4, FUS, GNAS, TSPO, ZFAS1, CD55, EMP3, ENO1, ARPC2, IER2, GABARAP, SDCBP, HNRNPH1, ITM2A, FXYD5, ZEB2, HNRNPA2B1, HNRNPA3, SQSTM1, LGALS3, SLC38A2, CD37, HMGN3, MTRNR2L12, EEF1D, ACTA2, CLIC1, YBX1, SRSF7, FABP5, RHOA, PPIB, YWHAZ, DDX17, ALDOA, TPM4, WSB1, KMT2E, BTF3, FCER1G, TNFAIP3, LAPTM5, RTN4, NAP1L1, HSPA5, YBX3, PFDN5, CREM, GPX4, TPI1, PRDX1, CFD, RGS1, SKP1, HNRNPC, HINT1, RBPJ, MBNL1, CHCHD2, HNRNPDL, EIF4A2, S100A9, DMKN, DST, COX6C, RORA, EZR, JMJD1C, DSTN, SARAF, SON, DYNLL1, PRDX2, NR4A2, RHOB, TMBIM6, FAU, HNRNPK, SLC25A5, CBX3, SRSF11, RNASE1, ATP5F1E, NAMPT, CALR, SRSF5, GPR183, ANXA5, RSRP1, DNAJA1, CSTB, OAZ1, PTGES3, TMA7, RBM39, CELF2, LSP1, OST4, SNHG8, GYPC, CCL2, AKAP12, RAN, MYL12B, CFL1, UQCRB, CCNI, ID2, SFPQ, PKM, ATP5ME, LY6E, SELENOK, SSR4, RGS2, CXCL2, ATP5MG, SEPTIN7, CAPZB, SLC25A3, ELOB, MYL12A, STK4, CNBP, RAC1, NCL, MORF4L1, DUT, SEC62, CAV1, MCL1, ELF1, HMGA1, UBA52, RACK1, PDE4D, AREG

- small intestine: TMSB4X, SRGN, B2M, KRT8, LGALS4, IFITM3, IL32, VIM, CST3, CD74, TXNIP, HSPA1A, NEAT1, SELENOP, LGALS1, S100A10, S100A6, MTRNR2L12, JUND, PHGR1, ACTB, FTL, MDK, MIF, HSPB1, FOS, TMSB10, SLC25A6, REG1A, STMN1, EEF1A1, HSPA1B, DUSP1, SH3BGRL3, BTG1, MARCKSL1, KLF6, TPM1, JCHAIN, YBX1, ID2, MT2A, TFF3, JUN, HSP90AA1, SAT1, CALM1, JUNB, PTMA, TPT1, COL3A1, CD63, EEF2, EEF1B2, TUBA1A, MT1G, CYBA, FTH1, LYZ, S100A4, OAZ1, HNRNPA1, ZFP36L2, TUBA1B, IGHA1, ITM2B, PABPC1, FABP5, DDX5, UBC, APOA1, UBB, MYL6, ZFP36L1, ZFP36, H3-3B, TSC22D3, GSTP1, GPX1, ALDOA, HMGN2, ATP5ME, FAU, PPDPF, HSPA8, UBA52, FABP1, PPIA, RGS1, FOSB, ID3, LTB, CALM2, S100A11, H3-3A, HSP90B1, ACTG1, TXN, PRDX1, SARAF, NOP53, ATP5F1E,

NFKBIA, PFN1, CLIC1, COL1A2, NDUFA4, EGR1, SSR4, H4C3, GAPDH, IER2, IL7R, DNAJB1, HSPA5, SRP14, CXCR4, ATP5MG, DEFA5, CCL5, IFI27, HINT1, GNAS, HSP90AB1, H2AZ1, EEF1D, ADIRF, ZFAS1, SERF2, SRSF5, IGKC, ANXA2, H1-10, RBP2, PNRC1, MYL12A, EIF4A1, PIGR, NACA, HMGB1, COX7C, NAP1L1, SELENOW, LGALS3, LDHB, HSPD1, CALR, RACK1, TAGLN2, MARCKS, IGFBP7, XIST, SET, IFITM2, CIRBP, TOMM7, MCL1, CD9, HERPUD1, COX4I1, GABARAP, KRT18, OST4, HMGB2, SUB1, LDHA, FABP6, XBP1, UQCR11, COX7A2, RAC1, HNRNPU, TMA7, HSPE1, EIF3E, PNISR, ATP5IF1, SPARCL1, EIF1, ATP5PO, COTL1, SOX4, PPP1R15A, PSMA7, CFL1, ANXA1, SRSF7, DYNLL1, KLRB1, CCNI, TMEM258, SRSF3, UBL5, PFDN5, SLC25A5, ITM2C, YWHAB, BTG2, HNRNPH1, NPM1, MT1X, GNG5, EZR, TPI1, CRIP1, CD69, YWHAZ, CDC42, HBA2, NPC2, HNRNPA2B1, POLR2L, N4BP2L2, SEC61B, UQCRB, AGR2, TYROBP, PCBP2, MYL12B, SKP1, COMMD6, ATP5MC2, TUBB, ARPC1B, COX6A1, ARPC2, PPIB, DBI, EIF4A2, HES1, EIF5, SQSTM1, ATP5MJ, CD52, POMP, ARPC3, FUS, SNHG8, COX7B, ACTA2, HMGN1, PTPRC, SRRM2, COL1A1, PEBP1, RHOB, TMBIM6, EIF4G2, COX6C, NBEAL1, CYCS, COX6B1, ATP5F1D, DDIT4, NR4A1, RGS2, TMEM59, EIF3L, ENO1, ELOB, GPX4, TIMP1, NDUFA1, NCL, EDF1, ATP1A1, ANAPC16, CD99, PSAP, OLFM4, SRSF2, DDX17, BTF3, TNFAIP3, SEPTIN7, SEC62, SUMO2, ATP6V0E1, RBM39, MTRNR2L8, C12orf57, SERP1, PRDX2, PRDX5, GADD45B, SOD1, REG4, MT1E, VMP1, COX5B, PSME1, RAN, CD164, ATP5MK, DCN, SEC61G, SELENOK, CCNL1, FKBP1A, FCGRT, CD44, SON, RBM3, SF1, ARGLU1, CTSD, GUK1, DSTN, SERBP1, PCBP1, IRF1, CHCHD10, TPM4, VAMP2, RHOA, PTMS, EIF3K, UQCRH, HNRNPK, WSB1, SLC25A3, PTP4A2, RTN4, KTN1, LMNA, SNX3, MZT2B, PRR13, NDUFB11, CHCHD2, ID1, UQCRQ, CEBPD, ARHGDIB, PSME2, PDIA3, HNRNPDL, TPM3, NDUFA13, DNAJA1, RSRP1, IFITM1, CSTB, TUBB4B, EEF1G, AHNAK, HNRNPC, CHGA, ELF3, C9orf16, ATP6V1G1, MBNL1, PPP1CB, GZMA, KLF2, RAP1B, ANKRD12, ST13, SFPQ, NDUFB2, C4orf3, NDUFB1, CUTA, TCF4, SNRPG, ATP5F1B, KRT19, NDUFS5, MORF4L1, CD7, RBMX, ARPC5, NDUFC2, HNRNPA3, RRBP1, EIF3F, EID1, CD3D, ATP5PF, P4HB, DAD1, LUC7L3, PDIA6, PRRC2C, MRPS21, ALDH1A1, CALD1, TSPAN8, UBE2D3, NDUFA11, CNBP, COX8A, MYL9, UQCR10, ARL6IP4, PPA1, DAZAP2, PAPOLA, VAMP8, ATF3, LAPTM4A, BRD2, NOP10,

RABAC1, GPR183, DEK, SPCS2, SSR2, ARL6IP1, MTDH, PARK7, TIMP3, KCNQ1OT1, FOXP1, CITED2, ATP5MF, ANP32B, PKM, NUCKS1, SNHG7, TBCA, CANX, ATP5MC3, UBE2B, SRSF11, SNRPD2, AKAP9, PGK1, EIF1AX, PTGES3, TLE5, YWHAE, YPEL5, CKLF, SPINK4, SOD2, FKBP2, SEM1, COX17, MICOS10, DDX3X, TOP1, APOA4, SPCS1, KMT2E, ERH, CD81, ATF4, ARF1, HMGN3, REL, BIRC3, LRRFIP1, ATP1B1, NDUFB4, KRTCAP2, SELENOS, HSBP1, EIF3H, SERPINA1, GSN, SF3B6, HIGD2A, SDCBP, TRA2B, EMP3, MPHOSPH8, C19orf53, DDX24, JTB, SOCS3, LMO4, ANXA5, MYH9, MGST3, NDUFA3, GSTK1, CTNNB1, CAPZB, JPT1, GNAI2, GCC2, GRN, HBB, EIF5A, LSM7, HNRNPF, WDR83OS, RN7SKP176, YWHAH, SELENOH, DUT, NEDD8, TCF25

- spinal cord: MOBP, ADGRB3, PCDH9, ZBTB20, CD74, TXN, UBA52, B2M, AIF1, S100A11, AHSP, DPP10, NEAT1, RBMS3, S100A4, HBG2, HMGB1, LYZ, TPT1, COL3A1, SRGN, PPIA, RORA, LGALS1, S100A6, LSAMP, NDUFA4, SPP1, SPARC, PTGDS, TMSB4X, S100A8, FTL, SNRPG, PTPRG, CFL1, IFITM2, GAS5, VIM, ARHGDIB, TUBA1B, DSCAM, COX6C, GAP43, MTATP6P1, ANXA2, COX7B, NACA, DCN, ACTG1, ATP5MG, UQCR10, ACTB, DYNLL1, HMGB2, FTH1, TMSB10, QKI, UQCRB, NPM1, EEF1A1, PTN, A2M, GAPDH, COL11A1, H2AZ1, UBB, HBA2, SERF2, PFDN5, PTMA, SLC1A3, BEX3, TMSB15A, GMFG, NDUFB6, PLAC8, MGST3, MYL12A, HINT1, CST3, CD52, MGP, NDUFB1, VCAN, CLU, EEF1B2, PABPC1, HSP90AA1, SLIRP, TMA7, SH3BGRL3, CXCL8, ATP5MC2, MLLT11, ATP5PF, DEFA3, UQCRH, PPIB, CALM1, COL1A2, IL1RAPL1, FYN, LHFPL6, FCER1G, SNRPD2, CALM2, TUBB, COX7A2, SNRPE, SKP1, S100A10, NOP10, COX7C, UBL5, HNRNPA1, OST4, S100A9, HBA1, FAU, NDUFA1, UCHL1, PTPRC, SEC62, FN1, COX6B1, CTSB, GLUL, SNRPF, CD63, PTTG1, ATP5MJ, GSTP1, COMMD6, HSP90AB1, CHCHD2, ATP6V0E1, YWHAB, TMEM108, SON, ATP5MK, PSMA7, PLP1, BANF1, SNHG6, GPC5, COX4I1, RBX1, SEM1, HNRNPA2B1
- spleen: TMSB4X, B2M, SRGN, IL32, CD74, EEF1A1, BTG1, TYROBP, H3-3A, FOS, LTB, NKG7, ACTG1, IGKC, VIM, IL7R, AIF1, S100A4, EEF1G, LGALS1, HSPA1A, MS4A1, CCL5, FTL, PTPRC, JUND, S100A6, HBA2, CD52, JUN, CCL4, STMN1, NEAT1, MTRNR2L12, HSPB1, FTH1, GSTP1, ZFP36, NFKBIA, TXNIP, SAT1, LYZ, CD69, ZFP36L1, IGHM, S100A8, DNAJB1, KLF6, GAPDH, IFITM2, SLC25A6, LDHB, XBP1,

TMSB10, H4C3, HSP90AA1, JUNB, CST3, FCER1G, CD7, ACTB, HBA1, ANXA1, KLF2, CYBA, ID2, KLRB1, DUSP1, IFITM1, CD3E, H2AZ1, HSP90B1, EEF1B2, ZFP36L2, SLC2A3, CXCR4, EEF2, IGLC2, TUBA1B, HSPA8, UBB, TPT1, LAPTM5, H3-3B, TNFAIP3, HMGB1, UBC, ZEB2, PLAC8, DUSP2, CORO1A, PPP1R15A, CD3D, HERPUD1, HINT1, LCP1, HNRNPA1, FYB1, FOSB, CALR, SNHG29, DDIT4, ARHGDIB, FXYD5, S100A10, HSPA1B, NAP1L1, CD37, GADD45B, ZFAS1, S100A11, HBG2, PFN1, PPIB, TSC22D3, COL1A1, HMGN2, NPM1, GNLY, CD63, HSPA6, PTMA, S100A9, MYL6, COTL1, IFITM3, CALM1, EIF1, NCL, XIST, AREG, JCHAIN, TUBB, DDX5, HSPD1, HNRNPU, IER2, TRBC2, HSPE1, HSPH1, UQCRB, CD99, SRSF7, ENO1, PNRC1, TAGLN2, EEF1D, CMC1, ITM2B, YWHAZ, BTG2, MYL12A, BTF3, HMGB2, EIF3E, MBNL1, HBG1, HBB, CD79A, HSPA5, YBX1, SARAF, PABPC1, SEPTIN7, LSP1, LIMD2, OAZ1, PCBP2, ALDOA, CD44, PSAP, RAC2, HCST, ARL4C, SH3BGRL3, ISG20, SAMHD1, SSR4, EIF4A2, GAS5, HSP90AB1, NUCKS1, DNAJA1, NFKBIZ, AKAP13, PPDPF, GLUL, RAP1B, CST7, ITGA4, LDHA, CALM2, FUS, IGHA1, RBM39, MT2A, SERF2, SUB1, UCP2, ANP32B, CCL3, TUBA1A, NR4A2, DDX17, SRRM2, FOXP1, ATP5F1E, EMP3, HMGA1, COX7C, MYH9, HNRNPA2B1, MIF, TPM4, ARPC3, TMBIM6, EIF4A1, GNAS, ARPC2, CLIC1, PKM, EVL, IGHG1, ARPC1B, BIRC3, DDX3X, NUCB2, PNISR, SMCHD1, GYPC, NOP53, HNRNPC, TOMM7, UBA52, SYNE2, FAU, UQCRH, PFDN5, RGS1, CSTB, OST4, CTSW, PPIA, SRSF5, GPX4, CCNI, ATP5MC2, FNBP1, HNRNPA3, NR4A1, CIRBP, DYNLL1, ATP5PO, RGS2, GNAI2, CFL1, RHOB, SELL, LITAF, C12orf57, PRDX1, HMGN1, SRSF11, PTGES3, PRRC2C, EIF3L, RACK1, NME2, ANKRD12, EZR, PDIA6, SF1, PPP1R2, CITED2, PSME1, ETS1, PEBP1, SOD1, GABARAP, RNASET2

- stomach: TFF1, AGR2, SRGN, TMSB4X, B2M, ACTG1, EEF1A1, CST3, SSR4, VIM, FTH1, CYSTM1, CALM2, S100A6, CD74, FAU, FTL, NEAT1, JUN, IFITM3, TMSB10, H3-3A, SPINK1, FOS, PGC, TPT1, HSPB1, ACTB, UBA52, SAT1, KRT8, BTG1, CD63, JCHAIN, HNRNPA1, LYZ, PTMA, PABPC1, IGFBP7, EEF1B2, CALM1, JUNB, IGHA1, COL1A1, RACK1, KRT19, CLDN18, TUBA1B, S100A10, HSP90AA1, COX7A2, HSPA1A, SOX4, MT2A, CCL4, UBB, GAPDH, H3-3B, CFL1, DNAJB1, IGFBP5, LGALS1, FOSB, ITM2B, EGR1, MDK, IGKC, GSTP1, IFI27, HBA2, UBC, NACA, TM4SF1, HSPD1, CYBA, NFKBIA, LIPF, FOXP1, CLU, DUSP1,

ANXA1, EZR, COX7C, JUND, KLF6, NPM1, SLC25A6, HSP90AB1, CHGA, GHRL, ATP5F1E, ZFP36L2, ZFP36, FABP5, PSAP, HNRNPA2B1, PFN1, TUBA1A, EEF1D, OST4, SH3BGRL3, HES1, S100A4, MYL6, STMN1, SEC62, DDX5, RAB11FIP1, LMNA, YBX1, ARPC2, ANXA2, SLC25A5, TXN, RGS2, TIMP1, CCL5, TCF4, EEF2, HSPE1, SERF2, OAZ1, PPDPF, IER2, LDHB, PPIA, PTPRC, GAS5, GAST, CXCL8, ZFP36L1, PHGR1, COL1A2, EIF1, HMGB1, S100A11, TFF3, SPARCL1, SON, COX7B, HSP90B1, SOD1, H2AZ1, ZFAS1, GNAS, MT1G, TMBIM6, ATP5MG, HSPA8, COX4I1, NAP1L1, SUMO2, POMP, SRP14, RBM39, ELF3, ID2, KRT18, PFDN5, ARGLU1, TOMM7, IGF2, IFITM2, TXNIP, XIST, SUB1, DCN, LAPTM4A, HNRNPU, YWHAZ, NUCKS1, NDUFA4, DSTN, DNAJA1, GADD45B, KLF2, HSPA5, PEBP1, GLUL, MUC5AC, TFF2, PRDX1, SNHG8, EIF4A2, SYNE2, SDCBP, UQCRQ, UQCRH, BTF3, MYL12A, TSC22D1, SRSF7, NCL, GSN, LMO4, UQCRB, COX8A, PPIB, RAC1, RHOA, CALR, TMA7, HINT1, CDC42, EIF3E, AKAP13, MARCKSL1, RGS1, NDUFA1, ID3, NFIB, CSTB, UBL5, MTATP6P1, ID1, HNRNPH1, LGALS3, HSPA1B, HNRNPK, SQSTM1, JMJD1C, COX6C, MYL12B, H19, IGLC2, FUS, UBE2D3, CREM, SELENOK, PSMA7, TAGLN2, PPP1R15A, NR4A1, HMGB2, CIRBP, EDF1, ANKRD11, ADIRF, PDIA3, PARK7, HNRNPC, COX5B, PRDX2, COMMD6, HERPUD1, COX6A1, TPM4, PNRC1, TIMP3, ARPC3, SRSF5, PDE4D, ARHGDIB, IL32, COX6B1, HBB, ATP5ME, DDIT4, MIF, CXCR4, SKP1, SLC38A2, CCNI, ATF3, RNASE1, ATP1B1, LDHA, TUBB4B, MEG3, ATP5MC2, ATP5MK, SPARC, IGFBP2, ACTN4, WSB1, REG1A, SPTBN1, PTMS, YWHAE, RAP1B, TPM3, APP, LPP, NPC2, ATF4, EIF4G2, VMP1, MEIS2, TTC3, HNRNPDL, TRA2B, TSC22D3, BTG2, TMEM59, MGST3, RAN, IGLC3, ATP6V1G1, ENO1, ESRRG, SLC2A3, SARAF, POLR2L, CAPZB, ITGB1, SST, TBCA, STAT3, CD44, MAGI1, CHCHD2, PPP1CB, MT1E, SEC61B, CLIC1, USP34, SF3B6, BSG, SERBP1, RBM47, SAP18, FKBP1A, YWHAB, HMGN2, NCOR1, AKAP9, PNISR, ELOB, SERP1, TPM1, PCBP2, ZBTB20, PSCA, CCNL1, GPX4, ATP1B3, DYNLL1, RBMX, CD59, LRRFIP1, ATP5PO, SRSF3, CYCS, ARF4, LUC7L3, PKM, COTL1, COL3A1, CAST, NDRG1, SELENOW, MTDH, TMEM258, ANP32B, ALDH1A1, CD9, PRRC2C, HMGN1, NFE2L2, CTNNB1, NDUFB1, AUTS2, SEC61G, MCL1, PTGES3, ATP5MJ, KMT2E, XBP1, LAMTOR5, NDUFS5, IRF1, TSPO, ANXA5, HBG2, IGHM, TGFBR2, SCAF11, CTSB, DDX17, DSP, EIF3H, KTN1, TUBB, GNG5, SFPQ, GKN1,

MECOM, ALDOA, SRSF11, CALD1, HSPH1, RTN4, SNRPD2, NDUFB2, C12orf57, SELENOP, TRA2A, MGP, HNRNPF, CMIP, SRRM2, CSDE1, ARID4B, MT1X, MUC6, VAMP2, MZT2B, TRAM1, EIF3K, EIF5B, GNB1, N4BP2L2, MORF4L1, IER3, GPBP1, GUK1, ARPC1B, AKAP12, RAB1A, PGA5, TNFAIP3, ATP5F1B, ANAPC16, HSPA9, DDX3X, TYROBP, PCBP1, UQCR11, DBI, MYH9, C4orf3, CD24, EMP1, H4C3, LUM, RRBP1, APOA1, ATP5MC3, SRSF2, MORF4L2, IL7R, ATP6V0E1, SSR2, PPP3CA, ITM2C, SET, S100P, NDUFC2, RBM25, CD52, SNRPG, EVL, RERE, MIDN, LAPTM5, SOCS3, RHOB, GABARAPL2, CKS2, DEK, EPCAM, SNX3, ANKRD12, RBM3, ARL6IP1, NAMPT, PHIP, NDUFB9, NDUFA13, RORA, ERH, NEDD8, DST, MUC1, ANXA10, CTSD, SKAP2, HPGD, PSME1, PAP-OLA, HNRNPUL1, SPRY1, EIF2S2, PDLIM1, SLC25A3, DNAJB6, EMP3, HNRNPA3, CNBP, GNAI2, HNRNPA0, SOD2, ELF1, TNRC6B, ESD, TRMT112, HNRNPM, TPI1, ST13, UBXN4

- testis: FOS, COL1A1, TMSB4X, EEF1A1, HSP90AA1, COL1A2, B2M, EEF1B2, FTL, TMSB10, NACA, COL3A1, IGFBP7, IFITM3, CALM2, TUBA1A, GPC3, GNAS, RACK1, S100A8, ACTG1, HBA1, SAT1, GAS5, VIM, DCN, NPM1, PRDX1, ACTB, TUBB, S100A10, A2M, RNASE1, S100A9, STMN1, HMGB2, BEX3, CST3, TMEM123, TM4SF1, LGALS1, SRP14, ITM2C, ATP5MG, CFL1, TPT1, MYL6, GSTP1, NREP, BEX1, HNRNPA1, SEM1, SUB1, HMGN1, PABPC1, CD81, H2AZ1, ACTA2, ZFP36L1, SEC61G, SNRPE, EIF4G2, FTH1, EEF2, MDK, UBB, LAPTM4A, SNHG29, EIF1, IFITM2, CTSB, SRGN, HINT1, UBA52, PTMA, COX6C, DUSP1, ARPC3, TCEAL9, FAU, DDX5, ITM2B, BTF3, JUNB, COX7C, SOX4, SOD1, TUBA1B, SERF2, RAN, HSPA1A, ENO1, EGR1, RGS2, SELENOP, UQCRQ, SPARC, PFN1, JUN, HNRNPA2B1, TMA7, DYNLL1, HBG2, RBP1, HNRNPDL, MEG3, PSMA7, DDX17, ATP5MC3, MARCKS, SLC25A5, FN1, SRSF3, HBE1, HMGB1, CALM1, CIRBP, PSAP, PCP4, SLC25A3, DLK1, GLUL, WSB1, LAMTOR5, UBC, GNG5, YWHAE, TSC22D1, GNG11, NPY, NDUFA4, ATP5F1E, MGST3, MYL12A, ZFP36, CD99, ID3, PPIA, COX7B, NDUFB6, S100A11, ATP5F1A, UQCRB, SKP1, PGK1, ATP6V1G1, MEST, CKS2, GAPDH, SLC25A6, NDUFB9, NDUFS5, NDUFB4, PEG3, UBE2C, NOP10, C11orf58, S100A6, HSPB1, TIMP1, LDHA, ATP5F1C, ATP5PF, PRDX2, AIF1, HNRNPK, JUND, TMEM258, ZFAS1, EIF4A2, COL4A2, UBL5, H3-3B, ATP5MJ, SDCBP, NDFIP1, FC-GRT, HSPD1, HSBP1, HSPE1

- tongue: S100A7, LRRN3, S100A8, VIM, SPARCL1, CAV1, PCAT19, IGFBP7, EPAS1, MSRB3, ANXA1, MT1X, C11orf96, FN1, BHMT2, NPR1, NLGN4X, LGALS1, ACTG1, SORBS2, LAMA2, CCDC68, IL6, IFI44L, NBEA, DEPP1, FAM107A, ANTXR2, PTPRB, HOXD1, ITGA1, HRCT1, NES, SFN, FAM110D, SOCS3, SBSN, SRGN, IGKC, ZFPM2, APBB1IP, LYPLAL1-AS1, SLC2A3, LAPTM5, SLA, XIST, LIMD2, SYTL2, CRACR2B, TNIP3, SPOCD1, ANXA9, PTPN7, PTPRC, DEFB4B, BMX, PDK4, GIMAP6, CXCL12, FCER1A, HSPB7, MYO7B, DBI, GPR65, TNFRSF9, ITGAM, KRT14, TGM2, CA4, NOTCH4, RAMP2, S1PR1, LPIN2, KRT23, CHCHD2, CHCHD10, RBP1, CRCT1, KRT4, ALS2CL, KRT1, GAK, IFFO2, U2AF2, LZTFL1, TCF7L2, PISD, ALDH6A1, ENDOD1, MROH1, RABEP2, MCAM, BOLA2B, TBL1XR1, PTTG1, SAT2, LSM10, SESN3, ZCCHC10, NDUFA2

- trunk: KRT1, VIM, TUBA1B, KRT14, CXCL14, SAT1, CD74, DMKN, APOE, ATP1B3, KRT15, KRT5, S100A10, DUT, CALML5, LY6D, GAPDH, CALML3, CCL27, TMEM45A, S100A2, LGALS1, TPPP3, KRT2, KRT-DAP, HMGB2, ZFP36L2, MT1X, GADD45B, KRT17, KRT10, S100A6, SYT8, LYPD3, S100A4, PCNA, DCT, EEF1A1, COL17A1, SPINK5, GATA3, GPX4, RACK1, STMN1, HSPA1B, CRABP2, B2M, FTL, DDIT4, SBSN, HMGN2, KRT6A, HOPX, CEBPB, TYMS, SLPI, CST3, FGFBP1, FTH1, RND3, DEK, MT2A, FABP5, TXNIP, TACSTD2, POSTN, PERP, TM4SF1, KLF4, UBR4, DEFB1, HES1, PTTG1, NPM1, TGFBI, SFN, HEXIM1, MYC, H2AZ1, CSTB, LSP1, IRF1, PTMA, RANBP1, ADM, SGK1, EEF2, CITED2, DSP, TMSB4X, LGALS7B, CAV1, PCLAF, RBP1, CD59, CSTA, H3-3B, DST, DEGS1, RRM1, HSPA1A, NUCKS1, ID1, FXYD3, TYRP1, BRD2, S100A14, DUSP1, CLEC2B, TSC22D3, ENO1, NDUFA4L2, PRXL2A, CA2, JUN, APP, HSPB1, CHCHD10, SOSTDC1, CCND1, PPDPF, NCL, ID4, MT1E, TUBA4A, ARL4D, HINT1, PKP1, C19orf33, GLUL, DSC3, ETS2, H1-0, MCM3, LDHB, AREG, ATP1A1, TUBB4B, MZT2B, CLDN4, DSC1, SOCS3, AHNAK2, EPHB6, ZFAS1, RNH1, KLF6, MLANA, CD63, EGFR

- ureter: SAMD11, JCHAIN, DCN, CD74, IGKC, IGLC2, IGHG1, IGHA1, SRGN, MGP, CXCL8, IGHG2, LGALS1, LUM, GPR183, SFRP4, TMSB10, TMSB4X, ADIRF, FOS, S100A4, FCER1G, FTL, RGS1, CST3, BCL2A1, NFKBIA, IL1B, MS4A6A, IGKV3-15, CFD, CTSC, S100A2, SPINK1, FAU, SKP1, DNAJB1, MTND2P28, RNASE6, C1QB, APOC1, CCL4, EEF1A1,

FXYD3, UBC, TMEM59, SOD2, CD63, B2M, IFITM3, EEF1D, ATRAID, EGR1, CSTB, COMMD6, ATP5MC2, CALM2, PSAP, LY86, TMEM258, SDS, C1QA, HSPB1, GPNMB, VAMP8, UPK1B, COX6C, CXCL1, SPCS1, CPVL, STX7, DUSP4, CD83, ATP1B3, FAM219B, WDR83OS, HSPH1, NDUFA1, PLAUR, ICAM1, CDC42, ID3, HNRNPR, LAPTM4A, ATP5IF1, GNAI3, APH1A, UROD, PCBP1, MZT2B, UQCRH, GNG12, ATP6V0B, ODC1, CSDE1, MDH1, TAGLN2, CFH, SLC20A1, OST4

- urinary bladder: MT2A, SRGN, SPINT2, CXCR4, SLPI, FTH1, WFDC2, NEAT1, CD69, VEGFA, LGALS1, FTL, CCL4, CD74, KRT7, IFITM3, B2M, CCL5, CXCL8, SPARC, ACTB, S100A6, NDRG1, IFI27, BTG1, IGFBP7, ACTG1, FOS, RARRES1, S100A9, MT3, COL4A1, HILPDA, CST3, TMSB4X, LCN2, CLU, MT1E, H3-3B, ERRFI1, NUPR1, HP, SAT1, MT1X, APLP2, IGFBP3, NFKBIA, XIST, ERO1A, VIM, HSPA5, CCL20, PGK1, PFN1, CA12, ANXA2, NAMPT, HINT1, LOX, CALM1, IER3, IFITM2, HSPG2, MMP7, HSP90B1, EGFL7, GNG11, CD81, RHOA, C15orf48, N4BP2L2, C3, SELENOS, S100A10, SERPINB4, P4HA1, SQSTM1, SKP1, CRIP2, IGFBP2, PABPC1, HERPUD1, TXNIP, FKBP1A, GLUL, SOD2, LY6E, HSP90AB1, DSTN, TIMP1, A2M, HSPB1, TPI1, WSB1, SELENOW, CXCL3, ENG, CD9, DDX5, FN1

- uterus: SRGN, PTPRC, TMSB10, IFITM3, FTH1, C11orf96, CXCL8, LGALS1, CD74, B2M, NEAT1, MTATP6P1, MT2A, ACTB, PTMA, IGFBP7, GAS5, FOS, SNHG6, TPM2, GADD45B, TSC22D3, SLPI, FTL, LCP1, THBD, TXN, ITM2C, SPARCL1, HSPB1, SNHG29, TCF4, GEM, SNHG5, CREM, RGS2, IFI27, CLEC2B, TMSB4X, MT1A, FXYD5, PKM, GABPB1-IT1, TNFAIP8, SNHG1, S100A4, POLR2J3, FTLP3, PRNP, CALD1, MTCO1P12, ADIRF, TYMP, CCL3L1, PLCG2, JUN, DANCR, ADAMTS4, TXNIP, XIST, EEF1A1, CXCR4, SLC2A3, CHASERR, RAB33A, BST2, CCDC3, FCER1G, FCHSD2, SNHG3, FCER1A, A2M, MGP, SGTB, IRF7, TNFAIP6, GMFG, DDIT4, SNHG17, KRT14, IGKC, ACTA2-AS1, VEGFA, NBPF8, CCL2, SAMHD1, ACAP1, SERPINE1, CCL3, FZD10-AS1, KLF2, DLEU2, S100A6, ALOX5, TIMP3, CHSY1, BMP2, CD37, ALOX5AP, TGFBR3, S100A2, IL18BP, SNCG, RNU6ATAC39P, APOD, FAM3D, GXYLT1, GJA4, TRAF1, XXbac-BPG283O16.9, FLNA, B3GAT1, PLAC8, AC004057.1, UBBP4, ATM, GPR183, ARHGAP4, CAV1, ID2, ERF, TAGLN, LY86, TPT1, ID3, AQP1, USP44

- vasculature: SFRP2, TIMP3, CSRP1, B2M, MGP, SRGN, GPX1, GAS5, TUBA4A, TSC22D3, SERPING1, MTATP6P1, PPP2R5C, C3, TPT1, SPARCL1, PRG4, CXCR4, HSPA8, YBX3, CEACAM6, RGS1, SNHG29, BIRC3, TSPAN5, DCAF12, RGS2, RASSF6, SLC25A37, CXCL16, MTCO1P12, RBM12, RHBDF2, PLAU, SNHG5, CD74, BZW1P2, FBXO48, GLUL, STK4, MT1F, MYL9, CPB1, FCER1G, GMFG, AREG, RBM38, CPA3, IQCB1, CHST11, EDEM1, CCL5, TMEM71, FTH1, RHOH, IGHG2, KLHL6, POU2AF1, IGKC, PUM2, IGLC3, IGLC2, ACTB, PCED1B-AS1, HSPG2, CD48, TGM1, ADIRF, ZBED2, LEF1

- yolk sac: HBG1, FRZB, IFITM3, VIM, FTL, CST3, SRGN, HSPA1A, CD74, CCL4, S100A6, LYZ, FOS, ZFP36L1, MT1G, EEF1A1, JUNB, LSP1, HBA1, SPARC, RNASE1, EGFL7, NR4A2, HSPA1B, NFKBIA, MARCKSL1, S100A11, PRSS57, TTR, CD52, JCHAIN, LGALS1, H4C3, FTH1, TMSB4X, CCL4L2, DLK1, MPO, STMN1, S100A4, JUN, HBE1, SPP1, IFITM2, COL3A1, BTG2, LTB, MEG3, DDIT4, S100A10, CD164, GATA2, HSPB1, NR4A1, ACTB, ANXA2, CD99, ACY3, TYROBP, ACTG1, APOE, UBE2C, S100A9, DNAJB1, IL1B, TUBA1B, TMSB10, MDK, C1QC, IGFBP7, HBA2, SELENOP, SAT1, IER2, PPP1R15A, TAGLN2, CD34, HMGB2, FCGRT, MTRNR2L12, FXYD5, GYPC, SAMHD1, ID3, APOA1, HBD, CORO1A, TUBB, CXCL3, DUSP2, CYTL1, TYMS, MT2A, ZFP36L2, TPM1, EEF1B2, DAB2, GNG11, IFITM1, CD37, ITM2C, TIMP3, PTMA, GAPDH, ZFP36, SOX4, RETN, CNRIP1, FCER1G, PTTG1, H2AC6, NACA, APOA2, CALM1, B2M, BLVRB, EGR1, HMGA1, CCL3, KLF6

POCKET MATERIAL: MAP OF CASE STUDY SOLAR SYSTEMS