Algorithms for modeling gene regulation and determining cell type using single-cell molecular profiles

Hannah Andersen Pliner

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Jay Shendure, Chair

Cole Trapnell, Chair

William S. Noble

Program Authorized to Offer Degree:
Genome Sciences

University of Washington

Abstract

Algorithms for modeling gene regulation and determining cell type using single-cell molecular profiles

Hannah Andersen Pliner

Co-Chairs of the Supervisory Committee:
Professor Jay Shendure
Genome Sciences

Assistant Professor Cole Trapnell
Genome Sciences

Single-cell genomic technologies are helping us answer key biological questions that have long remained elusive. How do a single cell and a single genome generate such complex multicellular organisms as humans? More specifically, how do these cells orchestrate specific transcriptional programs depending on their cell type? New technologies like single-cell RNA-seq and single-cell ATAC-seq allow us to examine the transcription and regulation of individual cells as they develop; however, these methods have important limitations. A primary limitation with all single-cell data is data sparsity, which must be overcome computationally to extract useful information from these experiments. In this dissertation, I present two algorithms designed to overcome the sparsity of single-cell data and allow biological discovery.

I first introduce Cicero for single-cell chromatin accessibility data, which is both an algorithm that calculates co-accessibility scores to assign distal regulatory elements to genes, and a software system that adapts existing single-cell RNA-seq analysis techniques for use with single-cell chromatin accessibility data. In Chapter 2, I apply Cicero to an *in vitro* myoblast differentiation assay and find evidence for the use of ”chromatin hubs” during myogenesis. In Chapter 3, I apply Cicero to single-cell ATAC-seq data from mouse bone marrow and recapitulate known patterns of hematopoiesis and known *cis*-regulation of the β-globin locus. In Chapter 4, I introduce a second

algorithm, Garnett, which uses single-cell expression data to train and apply automated cell type classifiers. The accuracy of this technology is demonstrated with data from various single-cell RNA-seq methods and tissue sources. In a final chapter, I reflect on the development of software for biological applications and future directions for this work.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am deeply grateful to the many people who supported this work. First, I would like to thank Cole Trapnell and Jay Shendure, my two superb advisors and mentors, for their incredible support and dedication to my training. To Cole, thank you for your dedication, your honest mentorship, and your sense of humor when things were going less well. To Jay, thank you for your steady guidance, and your unwavering enthusiasm for the science.

I would also like to thank the other members of my supervisory committee, William Noble, Daniela Witten, and Stephen Tapscott for your advice, guidance, and for making yourselves and your expertise available to me.

In addition to having two wonderful advisors, I am lucky enough to also have two wonderful labs. I want to thank both the Trapnell and Shendure labs for your advice, support, and friendship. I would specifically like to thank José McFaline, Darren Cusanovich, Riza Daza, and Dana Jackson, who generated the majority of the data described in this dissertation. In addition to the above names, I would like to thank Jonathan Packer, Andrew Hill, Xiaojie Qiu, and Delasa Aghamirzaie for their tremendous scientific contributions to this work. I am grateful to my two labs for creating an incredibly positive atmosphere in which to work and learn. In particular, thank you to Lauren Saunders, José McFaline, Sanjay Srivatsan, Dana Jackson, Andrew Hill, Xiaojie Qiu, and Molly Gasperini for your friendship and support.

Thank you as well to my past scientific advisors, especially Bryan Traynor, Alan Renton, Elena Casey, Maria Donoghue and Doreen Cunningham.

I have been very fortunate to have both made and kept great friendships during my time here. In particular, I would like to thank Lindsay Pino, Lauren Saunders, Alison Bae, Anne Musica, Holly Tao, Jared Dominguez, and Alex Dewald. Thank you to Maizie for both secretarial and emotional

assistance.

Thank you to my entire family, particularly my parents and brother, for your endless support and for always fielding my phone calls, you've kept me going when I needed it. Lastly, thank you to Mike Quist for being there, for always being interested in my work, and for the late-night Chipotle deliveries.

Chapter 1: INTRODUCTION

Each cell in a multicellular organism carries the same genome. However, the morphologies and functions of different cell types can be as distinct as neurons sending electrical impulses and macrophages engulfing pathogens. Given that all cells contain the full set of instructions for the vast array of potential cell types and states, what causes different cells to behave in such different ways? The phenotype a cell displays is governed by what genes are expressed or 'on' in the given cell and at what levels. A major goal in biological research is understanding the causes and effects of varying gene expression, with the ultimate aim of being able to manipulate cell types and functions to effect health and other outcomes.

The first step in studying the causes and effects of varying gene expression is measuring the expression itself. Traditionally, gene expression was measured a single transcript at a time with visual methods like *in situ* hybridization or molecular methods like quantitative polymerase chain reaction (qPCR). Because mammalian genomes have tens of thousands of expressed genes, and a near infinite combination of cell types and responses to stimuli, the field has quickly moved towards higher throughput and more global expression measurement such as microarrays and RNA-seq. These methods have the benefit of sampling the total expression of a tissue or sample.

In addition to measuring gene expression, genomics aims to also measure potential mechanisms by which expression might be regulated. One mechanism of particular interest is chromatin accessibility – how available a given segment of DNA might be to transcriptional machinery. DNase-seq and ATAC-seq are two assays that can measure chromatin accessibility genome-wide.

Genome-wide measurement with RNA-seq and ATAC-seq have allowed major strides in our understanding of many biological systems. However, biological samples are seldom homogeneous and so any bulk measurement of gene expression is necessarily an average of the expression across the cell population. This fact has prompted the recent transition towards single-cell measurement.

Single-cell genomic measurements avoid the drawback of averaging the features of the population, but because of the low levels of starting material in each cell, the data produced is generally a very sparse sampling of the total picture. This sparsity has required the development of new computational methods to analyze single-cell data. In this dissertation, I describe two new computational algorithms designed to address the sparsity of single-cell genomic data in order to mine meaningful information from these new genomic techniques. In this introduction, I will review the motivations, methods and challenges of single-cell genomic data.

1.1 Towards a quantitative understanding of gene expression and regulation

In a multicellular organism, different cell types perform different critical functions for the survival of the whole. Each of these cell types derives from a single progenitor, which must contain all of the instructions for all of the cells that make up the adult organism. These instructions are contained in the genome, which every cell in the organism (with a few exceptions) contains. From the single progenitor, cells divide and differentiate with precise timing, cell numbers, and intricate organization. The mystery of how this occurs has been of paramount interest to biologists for centuries.

1.1.1 Differing gene expression explains differing cell structure and function

The central dogma of biology states that the information encoded in DNA is transcribed into RNA and then translated into proteins, the collection of which defines a cell's structure and function. A gene can therefore be considered 'on' if it is transcribed into RNA. Gene expression – which genes are transcribed, when, and at what levels – seems to be the most straightforward answer to the question of cell types and development. Different cell types execute different transcriptional programs that govern their morphology, function, and both cell division and differentiation. But how is gene expression controlled? Many of the technological advances in modern genomics hope to advance our understanding of this question.

1.1.2 Defining cell types

The challenge of measuring and understanding how a single genome can define hundreds of cell types is further complicated by the fact that what constitutes a single cell type is far from black and white. Traditionally, cell types have been defined by the expression of a handful of marker genes, which in turn were generally discovered by immunohistochemistry staining that was restricted to cells of a certain morphology within the field of view of a microscope. As more global methods of measuring gene expression have emerged, it has become clear that even within traditionally defined cell types, there can be massive variation in both gene expression and cell function. To illustrate this variation, consider the cell type of 'macrophage'. Macrophages range from Kupffer cells, which break down red blood cells in the liver, to osteoclasts, which break down and maintain bone (Mosser and Edwards, 2008). How much variation constitutes a new cell type versus a new cell state is still a matter for debate, with some going so far as to suggest that the field abandon the notion of the cell type altogether (Clevers et al., 2017; Trapnell, 2015).

1.1.3 Expression in the context of chromatin

One more complicating factor to our understanding of the causes and effects of gene expression is the three-dimensional nature of the genome. In the case of human cells, approximately 2 meters of DNA must fit in the approximately 10 micron diameter nucleus. This requires considerable compaction. However, this compaction must also be specific, so that the transcriptional machinery still has access to the genes the cell needs to express.

The first level of compaction has been described by the "beads on a string" model. Genomic DNA is wrapped around histone octamers at approximately two hundred base pair intervals forming nucleosomes (Kornberg, 1974). These histones can then be modified with, for example, methylation or acetylation, which can alter the strength of DNA-histone interactions (Li et al., 2007). The positioning of nucleosomes has been shown to affect transcription *in vitro* (Knezetic and Luse, 1986; Lorch et al., 1987) and *in vivo* (Han and Grunstein, 1988; Kayne et al., 1988). On a larger scale, how tightly regions of chromatin are compacted can influence whether genes in that region

are expressed (Li et al., 2007).

For a gene to be expressed, a series of DNA-binding proteins must bind at a sequence upstream of the gene's coding region, termed the promoter. Whether and to what extent these proteins transcribe a given gene is determined by a complex mixture of the promoter sequence itself, which transcription-associated proteins (transcription factors) are available, and the activation of distal DNA sequences called enhancers.

Because many DNA-binding proteins can only bind to DNA that is uncompacted (Zaret and Carroll, 2011) – termed accessible – measuring the chromatin accessibility of a region or whole genome has been used extensively to point researchers towards important regulatory regions (Gross and Garrard, 1988; Felsenfeld et al., 1996; Thurman et al., 2012). Thus, the combination of measuring gene expression and chromatin accessibility has led to a better understanding of both gene expression and the mechanisms that regulate it.

1.2 Measuring expression and accessibility in bulk

The first step in understanding how varying gene expression contributes to varying cell morphology is to measure gene expression and potential influencers of gene regulation like chromatin accessibility.

1.2.1 Measuring gene expression

Gene expression is assayed by attempting to quantify the number of RNA transcripts from a particular gene present in a cell. This can be done for a single gene using various molecular techniques, for example using reverse transcriptase qPCR or northern blotting. Assays of gene expression went transcriptome-wide (measuring the complete set of RNA transcripts in a sample) with the introduction of hybridization- (for example, microarrays) and sequencing- (for example, RNA-seq) based techniques (Lowe et al., 2017).

Currently, the dominant method of measuring the transcriptome is RNA-seq, which uses next-generation sequencing to measure a library of complementary DNA (cDNA) that has been con-

verted from RNA using reverse transcriptase (Mortazavi et al., 2008; Nagalakshmi et al., 2008; Lister et al., 2008). The output of RNA-seq is a set of sequenced reads, which can be mapped back to the genome. The number of transcript reads that map to each gene in the genome can be used to quantify the level of expression of that gene in the sample.

1.2.2 Measuring chromatin accessibility

Since the late 1970s, researchers have used cleavages by nucleases as indicators of DNA accessibility, reasoning that if nucleases could reach DNA, transcription factors could as well (Weintraub and Groudine, 1976; Varshavsky et al., 1978; Scott and Wigmore, 1978). In the mid-2000s, researchers began measuring chromatin accessibility genome-wide using DNase-chip and DNase-seq assays (Crawford, 2006; Crawford et al., 2006; Boyle et al., 2008). These assays combined next-generation sequencing with nuclease digestion to determine DNA's sensitivity to nucleases genome-wide.

An alternative to DNase-seq, which provides similar chromatin accessibility data, is assay for transposase-accessible chromatin using sequencing (ATAC-seq) (Buenrostro et al., 2013). ATAC-seq leverages the activity of the enzyme Tn5 transposase, which cuts DNA and simultaneously inserts the oligonucleotides it is loaded with onto the cut ends. Tn5 transposase preferentially acts on open chromatin, so we can identify chromatin accessibility by mapping where Tn5 has inserted.

Similar to RNA-seq, the output of ATAC-seq is a set of sequenced reads. In the case of ATAC-seq, by mapping these reads back to the reference genome and quantifying where the ends of the reads (and thus the insertions of Tn5) occurred, we can determine what parts of the genome were most accessible to Tn5.

1.2.3 Limitations of bulk genomics

Until recently, molecular genomic measurements were restricted to large bulk populations. Bulk RNA-seq has provided invaluable insights into differences in expression across populations of normal and perturbed tissues and samples, most recently represented by projects like the Genotype-

Tissue Expression (GTEx) project (Carithers et al., 2015). Similarly, bulk DNase-seq and ATAC-seq assays have provided comprehensive maps of regulatory elements across many cell types, exemplified by the ENCODE project (The ENCODE Project Consortium, 2012) and the NIH Roadmap Epigenomics project (Kundaje et al., 2015). Accessibility is routinely used as a filter when searching for regulatory elements (John et al., 2011), and alterations to chromatin remodelers have been implicated in cancer (Schwartzentruber et al., 2012).

However, because bulk measurements are, by necessity, averages across the cells of interest, and because tissues are generally heterogenous mixtures of different cell types, bulk genomic measurements can be insufficient for studying cell level phenomena. In addition, data from bulk assays can sometimes be misleading. A toy example of this is illustrated in Figure 1.1. In this illustration, what from bulk data would look like an increase in expression of the gene of interest across samples (in blue) might actually be a shift in the cell type populations present, with no increase in gene expression happening on the cell level.

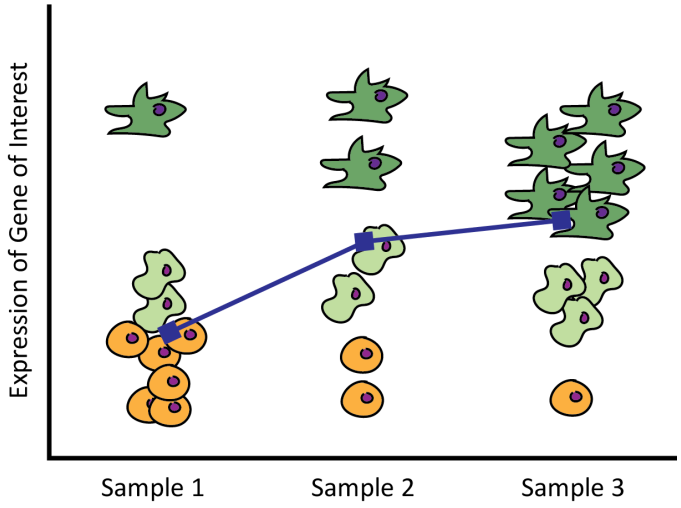

Figure 1.1: Diagram demonstrating how bulk genomic data can be misleading when made up of mixed populations. In this toy example, a bulk expression assay would show an increasing trend of expression across the samples (blue line and boxes). However, when viewed as single cells, it is clear that expression among the cell types remains fairly constant, and instead the composition of the samples changes.

1.3 MEASURING EXPRESSION AND ACCESSIBILITY AT SINGLE-CELL RESOLUTION

To avoid some of the limitations of bulk genomic assays like those demonstrated in Figure 1.1, there has been a recent shift towards single-cell genomic assays that allow measurements to be assigned to individual cells. The major modification necessary to convert an assay from bulk to single-cell is isolation of the cells for barcoding. This has been done in multiple ways, from limiting dilutions or hand-picking individual cells, to microfluidics that isolate cells in droplets. An alternative to physically isolating cells is single-cell combinatorial indexing (sci-), which was developed in the Shendure Lab and has been extended to several technologies (Adey et al., 2014; Amini et al., 2014; Cusanovich et al., 2015; Vitak et al., 2017; Ramani et al., 2017; Cao et al., 2017; Mulqueen et al., 2018) (see Figure 1.2 for an illustration of combinatorial indexing in the context of sci-ATAC-seq).

1.3.1 Single-cell RNA-seq

Since single-cell RNA-seq (scRNA-seq) was first described in Tang et al. (2009), there has been a surge in new techniques aimed at increasing both the throughput and the depth of single-cell transcriptional profiles. In 2009, individual cells were isolated by hand under a microscope, but we can now use single-cell combinatorial indexing to measure the transcriptomes of 50,000 cells in a single experiment (Cao et al., 2017).

In all contemporary methods, an entire experiment comprising many cells is sequenced together and subsequently "demultiplexed" using a barcode sequence attached to each read that corresponds to the read's cell of origin. After assigning reads *in silico* to specific cells, the remainder of each read can be mapped to the reference transcriptome, and the single-cell transcriptome can thus be quantified.

1.3.2 Single-cell ATAC-seq

During 2015, the Shendure Lab and the Greenleaf Lab at Stanford University independently designed single-cell ATAC-seq assays (Cusanovich et al., 2015; Buenrostro et al., 2015). The Green-

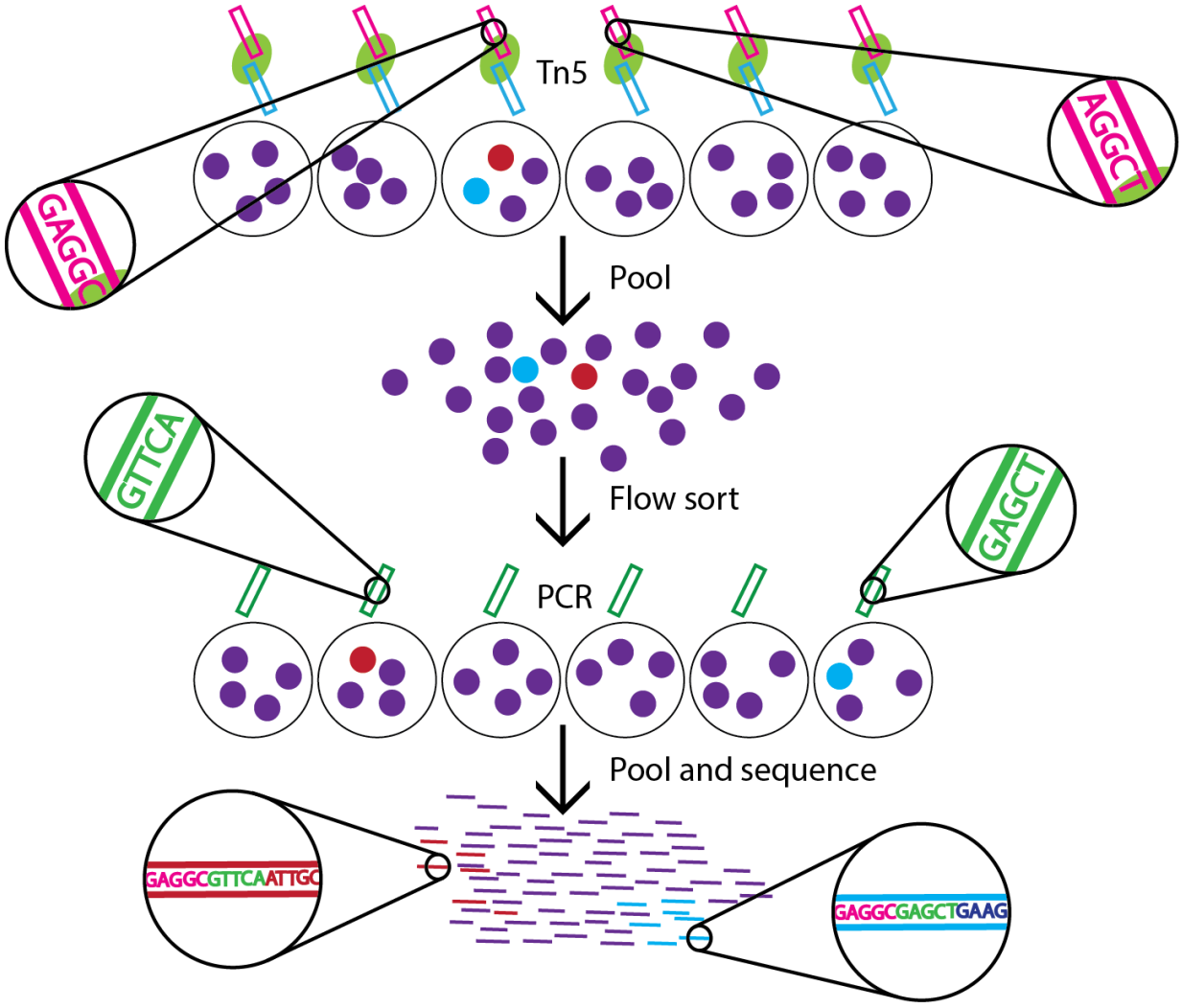

Figure 1.2: Diagram of single-cell combinatorially indexed ATAC-seq. Tn5 transposase is added to wells of divided cell nuclei such that each well's Tn5 is loaded with a different molecular barcode. Nuclei are then pooled and flow sorted into a second set of wells. Next, nuclei are lysed and PCR is performed such that the primers in each well have a different molecular barcode. Lastly, reads from all wells are pooled and sequenced. With combinatorial indexing, nuclei/cells are never physically isolated, however any cells that travel unique paths through the sequence of wells will be uniquely barcoded.

leaf lab performed ATAC-seq on individually isolated cells, acquiring data on 200-300 cells per experiment and collecting an average of 74,000 reads per cell (Buenrostro et al., 2015). In contrast, we published an assay called sci-ATAC-seq that uses combinatorial indexing to tag reads from

single cells without ever isolating the cells themselves. This method allowed us to acquire data on up to 1,500 cells per experiment, but with relatively few reads (a median of up to 3,000 reads per cell) (Cusanovich et al., 2015). Since this original paper, the sci-ATAC-seq method has improved such that we are able to isolate more than 10 times as many reads per cell without sacrificing our high cell count (Cusanovich et al., 2018b).

Similar to single-cell RNA-seq, a library of reads can be demultiplexed using the barcode sequence and then each cell's reads can be mapped and quantified similarly to bulk data.

1.4 Challenges of single-cell data

Single-cell data avoids some of the limitations of bulk data, but also comes with some unique computational challenges that must be overcome to take full advantage of these new technologies (Stegle et al., 2015; Bacher and Kendziorski, 2016; Hwang et al., 2018).

1.4.1 Addressing sparsity in single-cell data

The primary challenge when dealing with single-cell genomic data is sparsity. Because of the low input material present in a single cell, single-cell assays are likely to miss many true values. In the case of scRNA-seq data, this leads to high variation in expression profiles even among very similar cells. This includes complete dropout where a gene known to be expressed is missing entirely from the cell transcriptome. Similarly, in sci-ATAC-seq, where there are generally only two potential Tn5 insertions to be captured at a given site (one from each homologous chromosome), profiles of accessibility are missing reads from most of the expected accessible sites.

In both scRNA-seq and sci-ATAC-seq, zeroes in a cell-feature matrix can occur for two reasons; first, a true zero represents an actual lack of the feature in that cell – either the gene is not expressed or the site is not accessible – and second, a dropout zero represents a feature that does occur in that cell, but the data was not captured by the assay. Dropout is more likely to occur in features that are rarer – for example highly expressed genes are less likely to dropout than more lowly expressed ones. In scRNA-seq, this observation has been used to develop more robust statistical

models (Kharchenko et al., 2014), and has also underpinned various methods aimed at imputing the missing information (Azizi et al., 2017; Li and Li, 2018; van Dijk et al., 2018; Zhu et al., 2018; Gong et al., 2018).

In this work, we address sparsity in a few ways. In Chapters 2 and 3, we move our binary sci-ATAC-seq data into the binomial count regime by aggregating or bagging similar cells to create pseudo-bulk data. This avoids the limitations of bulk data by not averaging across different cell types and instead averaging across only similar cells. However, this also decreases the sparsity of our single-cell data. In addition, with sufficiently large counts, sites can be modeled by the Gaussian approximation, which makes adjusting for various technical artifacts trivial. In Chapter 4, we acknowledge the fact that false negatives are much more likely than false positives in sparse data due to dropout by using a term frequency-inverse document frequency (TF-IDF) transformation when weighting the presence of marker genes. This procedure will up-weight specific genes that are rarely seen, and down-weight genes that are more widely captured. In addition, we develop profiles of cell types with large numbers of gene features rather than relying on a few marker genes that may be susceptible to dropout.

1.4.2 Addressing technical variability

Related to the challenge of sparsity in single-cell data is the challenge of increased technical variability. Single-cell data is prone to higher technical variability due to the low levels of input material and the low data capture rate of single-cell assays. Various approaches have been developed to model (Grün et al., 2014) and address the high technical variability in single-cell data, including comparing variability in genes to variability in spike-ins (Brennecke et al., 2013; Vallejos et al., 2015; Kim and Marioni, 2013) and attempting to correct for variation associated with the cell cycle (Buettner et al., 2015; Leng et al., 2015).

In this dissertation, we address increased technical variability in a few different ways. In our methods described in Chapters 2 and 3, we allow for easy correction of any batch or technical effects, and also rely on aggregation to boost signal and lower noise. In Chapter 4, we use a

heuristic marker cutoff such that more highly expressed genes must overcome a higher threshold to be considered expressed, which will help address technical variability related to highly expressed genes "leaking" into the profiles of other cells.

1.4.3 Addressing a lack of 'gold standards'

Lastly, single-cell analysis often suffers from the lack of a gold standard. As with most new technologies, especially those that greatly expand the amount of data that can be captured in a single experiment, we do not yet have many good positive controls that guarantee that a new method is "working". In the case of single-cell chromatin accessibility, we do not have a gold standard expectation of accessibility within a single cell, nor do we have many examples of the types of *cis*-regulatory interactions we attempt to predict in Chapters 2 and 3. In the case of cell type identification, gold-standard genetic markers are often only specific to a cell type within a given tissue, but not across the entire organism, and markers may be more variably expressed than was realized when examining bulk tissues. Chapter 4 of this dissertation addresses the lack of a gold standard by attempting to expand the definition of a cell type beyond the initial literature derived markers that are available.

1.5 ORGANIZATION OF THIS DISSERTATION

In the following chapters, I describe three projects that aimed to address some of the challenges of the analysis of single-cell genomic data and to generate tools that could be made available to the genomics community.

Chapters 2 and 3 describe the development of a new algorithm called Cicero. Cicero uses sci-ATAC-seq data to identify putative cis-regulatory interactions between enhancers and promoters. In addition, Cicero facilitates the overall analysis of sci-ATAC-seq data by providing methods that address the sparsity of the data, which allows the application of existing analysis techniques designed for other data types. The manual pages for the Cicero software are provided in Appendix A. In Chapter 2, Cicero is applied to sci-ATAC-seq data from myoblast differentiation and iden-

tifies dynamically accessible elements and chromatin hubs that act to regulate gene expression during differentiation. In Chapter 3, Cicero is applied to investigate chromatin dynamics in mouse hematopoiesis to recapitulate known interactions of the locus control region and beta-globin during erythropoiesis.

Chapter 4 describes a second new algorithm called Garnett, which aims to automate cell type identification. The manual pages for the Garnett software are provided in Appendix B. Garnett is applied to several available datasets including both single-cell RNA-seq and sci-ATAC-seq data. Lastly, I present closing remarks and future directions in Chapter 5.

Chapter 2: CICERO PREDICTS *CIS*-REGULATORY DNA INTERACTIONS FROM SINGLE-CELL CHROMATIN ACCESSIBILITY DATA

Chapter 2 is adapted with minimal modification from:

Pliner, H.A., Packer, J., McFaline-Figueroa, J.L., Cusanovich, D.A., Daza, R., Aghamirzaie, D.A., Srivatsan, S., Qiu, X., Jackson, D., Minkina, A., Adey, A., Steemers, F.J., Shendure, J., and Trapnell, C. (2018) Cicero predicts *cis*-regulatory DNA interactions from single-cell chromatin accessibility data. Molecular Cell. *71*, 858–871.

2.1 ABSTRACT

Linking regulatory DNA elements to their target genes, which may be located hundreds of kilobases away, remains challenging. Here, we introduce Cicero, an algorithm that identifies co-accessible pairs of DNA elements using single-cell chromatin accessibility data and so connects regulatory elements to their putative target genes. We apply Cicero to investigate how dynamically accessible elements orchestrate gene regulation in differentiating myoblasts. Groups of Cicero-linked regulatory elements meet criteria of "chromatin hubs"—they are enriched for physical proximity, interact with a common set of transcription factors, and undergo coordinated changes in histone marks that are predictive of changes in gene expression. Pseudotemporal analysis revealed that most DNA elements remain in chromatin hubs throughout differentiation. A subset of elements bound by MYOD1 in myoblasts exhibit early opening in a PBX1- and MEIS1-dependent manner. Our strategy can be applied to dissect the architecture, sequence determinants, and mechanisms of *cis*-regulation on a genome-wide scale.

2.2 INTRODUCTION

Chromatin accessibility is a powerful marker of active regulatory DNA. In eukaryotes, chromatin accessibility at both promoters and distal elements delineates where transcription factors (TFs)

are bound in place of nucleosomes (Felsenfeld et al., 1996). Genome-wide analyses of chromatin accessibility as measured by DNaseI hypersensitivity have found that the repertoire of accessible regulatory elements constitutes a highly specific molecular signature of cell lines and tissues (Thurman et al., 2012). Furthermore, genome-wide association studies (GWAS) show that a substantial proportion of genetic risk for common disease falls within accessible regions in disease-relevant tissues or cell types (Gusev et al., 2014; Maurano et al., 2012).

Despite its importance, we continue to lack a quantitative understanding of how changes in chromatin accessibility relate to changes in the expression of nearby genes. A prerequisite for such an understanding is a map that links distal regulatory elements with their target genes. To this end, we developed Cicero, an algorithm that generates such linkages on a genome-wide basis based on patterns of co-accessibility in single-cell data.

We demonstrate Cicero's capabilities through an analysis of skeletal myoblast differentiation, which remains one of the best characterized models of gene regulation in vertebrate development. Myoblast differentiation is orchestrated by a core set of TFs, including MYOD1 and MEF2 (Molkentin et al., 1995), which regulate the expression of thousands of genes as cells exit the cell cycle, align, and fuse to form myotubes. Here, we used single-cell combinatorial indexing ATAC-seq (sci-ATAC-seq) on 13,367 cells to identify 329,020 accessible elements in myoblasts, nearly 22,000 of which open or close during differentiation. When applied to these data, Cicero linked most dynamic sites to one or more putative target genes. From the resulting *cis*-regulatory map, we can predict changes in gene expression based on the chromatin accessibility dynamics of the linked distal elements.

2.2.1 Design

In contrast with previous approaches that rely on a large compendium of bulk chromatin accessibility data generated across many cell lines or tissues (Thurman et al., 2012; Budden et al., 2014), we sought a method that would work with single-cell chromatin accessibility data from a single experiment and that was robust to the sparsity of that data. Cicero uses sampling and aggregation

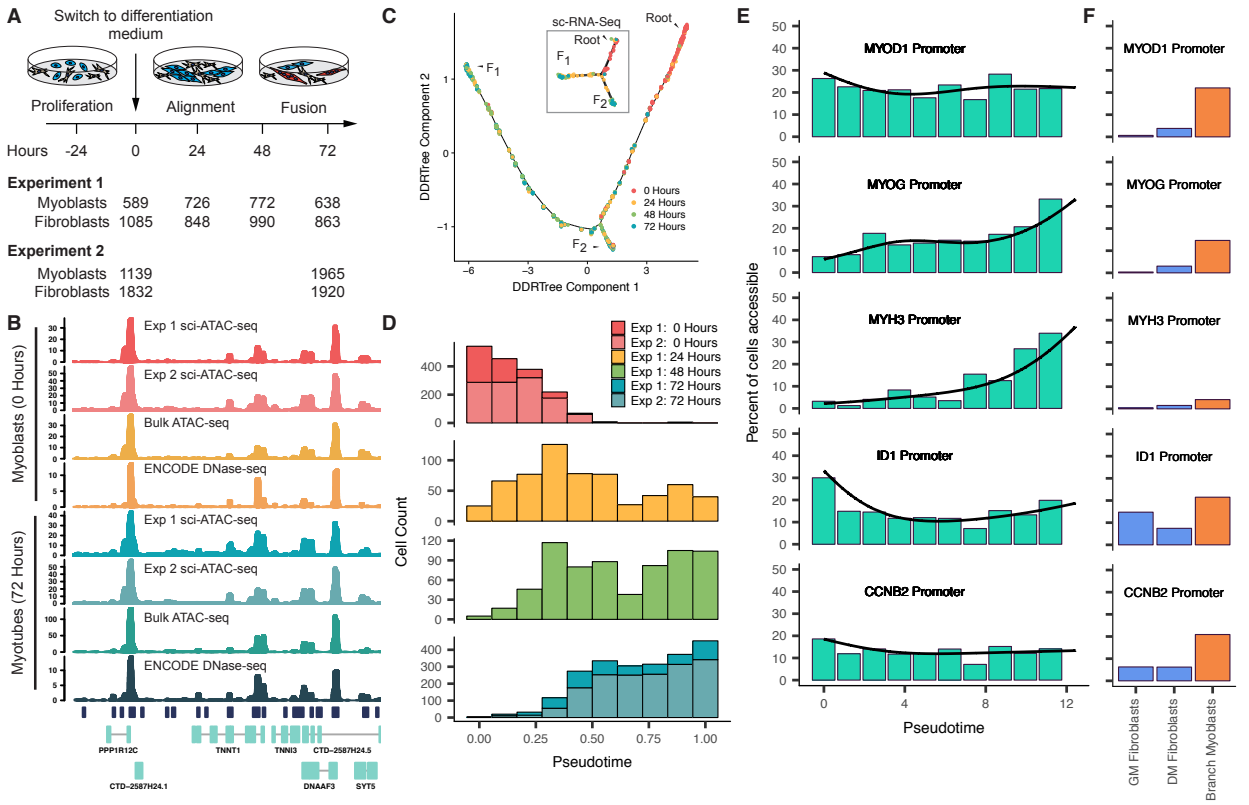

Figure 2.1: Differentiating myoblasts follow similar single-cell chromatin accessibility and gene expression trajectories. A) Single-cell chromatin accessibility profiles for human skeletal muscle myoblasts (HSMM) were constructed with sci-ATAC-seq. Contaminating interstitial fibroblasts (common in HSMM cultures) were removed informatically prior to further analysis. B) Aggregated read coverage from sci-ATAC-seq experiments in the region surrounding TNNT1 and TNNI3 in myoblasts (0 hours) and myotubes (72 hours). Bulk ATAC-seq prepared from the same wells as experiment 2 are shown alongside DNase-seq from ENCODE for comparison (ENCODE experiments ENCSR000EOO and ENCSR000EOP) (The ENCODE Project Consortium, 2012). C) The single-cell trajectory inferred from 2,725 myoblast sci-ATAC-seq profiles from experiment 1 by Monocle (see Section 2.5, Methods). In subsequent panels and throughout the paper, we exclude cells on the branch to outcome F2 unless otherwise indicated. Inset: the sci-RNA-seq trajectory reported for HSMMs (reproduced from Figure 2 of Qiu et al. (2017b), cells were from the same lot and were cultured under identical conditions to those for sci-ATAC-seq). D) Distribution of cells in chromatin accessibility pseudotime from the root to trajectory outcome F1. E) Percent of differentiating cells whose promoters for selected genes are accessible across pseudotime. Black lines indicate the pseudotime-dependent average from a smoothed binomial regression. F) Percent of cells whose promoters for selected genes in (E) are accessible in fibroblasts collected in growth medium (GM) or differentiation medium (DM), as well as myoblasts localized to the branch to F2. See also Figure 2.2.

of groups of similar cells to adjust for technical confounders and to quantify correlations between putative regulatory elements. Based on these correlations, Cicero links regulatory elements to target genes using unsupervised machine learning. The algorithm can be applied to any cell type and organism for which a sequenced genome and single-cell chromatin accessibility data are available. Because it accepts single-cell data as input, Cicero can in principle work on complex mixtures of different cell types as are found in tissues.

2.3 RESULTS

2.3.1 The trajectories of chromatin accessibility and gene expression during myoblast differentiation are highly similar

We performed a differentiation time course on human skeletal muscle myoblasts (HSMM), harvesting cells at 0, 24, 48, and 72 hours after the switch from growth media to differentiation media Figure 2.1A. With optimized sci-ATAC-seq (Cusanovich et al., 2018b), we profiled chromatin accessibility in 13,367 cells across 2 independent experiments. Aggregated single-cell ATAC-seq data were highly concordant with both bulk ATAC-seq and published DNaseI hypersensitivity data from myoblasts and myotubes (Figure 2.1B and Figure 2.2A) (The ENCODE Project Consortium, 2012). To define accessible regions, we pooled reads from all cells from each experiment and called peaks with MACS 2 (Zhang et al., 2008). The vast majority of peaks were shared between experiments (Figure 2.2B), so we used a single merged set of peaks for all downstream analyses. After excluding 7,538 cells flagged as likely interstitial fibroblasts based on the absence of promoter accessibility in any of several known muscle markers (56%, a proportion similar to our estimate from single-cell RNA-seq in this system) (Qiu et al., 2017b), we identified 329,020 sites accessible in muscle cells. Each cell had reads overlapping with an average of 3,466 promoter-proximal accessible sites and 9,055 distal accessible sites (Figure 2.2C).

We next sought to characterize changes in chromatin accessibility as myoblasts differentiated. However, analyzing differentiation from time series data is confounded by asynchronicity, i.e., Simpson's Paradox (Simpson, 1951). To overcome this, we recently developed the technique of

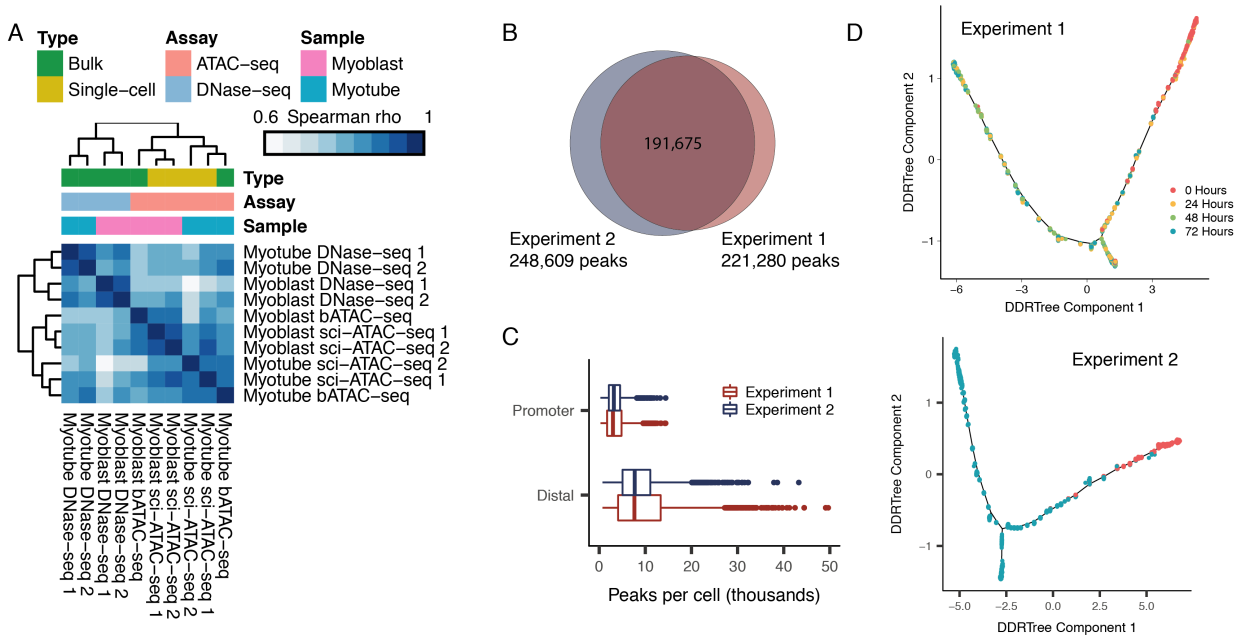

Figure 2.2: Chromatin accessibility profiles of differentiating myoblasts are highly reproducible, related to Figure 2.1. A) Spearman correlation heatmap between pairs of chromatin accessibility profiles as measured by bulk ATAC-seq, DNase-seq, and aggregated sci-ATAC-seq from 0 and 72 hours. MACS was used to call new peaks on each dataset; these peaks were merged, and reads were counted in each peak from each dataset. These counts were used to calculate Spearman correlations. B) Venn diagram illustrating reproducibility in MACS 2 peaks calls between independent sci-ATAC-seq experiments. Peaks in the intersection correspond to DNA elements called in both experiments. C) Boxplot of the number of MACS-called sci-ATAC-seq peaks per cell. Promoter-proximal peaks are peaks intersecting the first 500 base pairs upstream of a transcription start site (see Section 2.5, Methods). Distal peaks are all other peaks. D) Monocle trajectories on each of the sci-ATAC-seq experiments. The top panel is identical to Figure 2.1C and included for comparison purposes.

"pseudotemporal reordering" (or "pseudotime") that uses machine learning to organize cells according to their progress through differentiation (Trapnell et al., 2014). Although our algorithm, Monocle 2, was designed for single-cell transcriptomes (Qiu et al., 2017b), we were able to adapt it to sci-ATAC-seq data with straightforward modifications (see Section 2.5, Methods).

Monocle independently placed the cells from each experiment along similar trajectories with two outcomes (denoted F1 and F2) (Figure 2.1C and Figure 2.2D). These trajectories are similar to the trajectory constructed from single-cell transcriptomes in our previous work (Qiu et al.,

2017b) (Figure 2.1C, inset). Cells harvested from growth media fell almost exclusively near the beginning of the trajectories, while cells from later time points were distributed over their length (Figure 2.1D). Over the path to F1, promoters for well-known myogenic regulators and structural components of muscle opened (became more accessible), whereas the promoter of *ID1*, a well-characterized repressor of myoblast differentiation (Benezra et al., 1990), closed (Figure 2.1E). Similar to the single-cell RNA sequencing (scRNA-seq) trajectory (Qiu et al., 2017b), a number of cells were positioned on a branch leading to the alternative outcome F2. That these cells are accessible at the *MYOD1* promoter, but not the *MYH3* promoter, suggests they represent "reserve myoblasts" that did not fully differentiate (Yoshida et al., 1998) (Figure 2.1F). The similar trajectories constructed by Monocle from three independent experiments, as well as the close correspondence between the kinetics in expression and chromatin accessibility for key muscle genes, support the accuracy of Monocle's pseudotime ordering.

2.3.2 Distal DNA elements are dynamically accessible during myoblast differentiation

Differential analysis revealed significant pseudotime-dependent changes in accessibility at 21,678 of 329,020 (6.6%) sites during myoblast differentiation (Figure 2.3A and Figure 2.4A). In addition, we conducted a similar differential analysis on previously published scRNA-seq data from the same system (Trapnell et al., 2014). Of the "dynamic" accessible sites, only 1,324 (6.1%) were promoters (Figure 2.3B), of which 92 overlapped with 1,464 differentially expressed transcripts (false discovery rate [FDR] $<5\%$) by scRNA-seq. Of the 64 promoters with nontransient changes in both accessibility and gene expression, 62 (97%) were directionally concordant. Of the 20,354 distal, dynamically accessible sites, 68% were annotated as enhancers in myoblasts or myotubes (Libbrecht et al., 2018), as compared with only 36% of all accessible sites (Figure 2.3B).

Using gene set enrichment analysis, we found that genes associated with contraction and other muscle-related functions were strongly enriched among genes with significantly opening promoter regions. In contrast, promoters for genes associated with the cell cycle, which are downregulated early in differentiation, were only marginally enriched among the differentially accessible sites

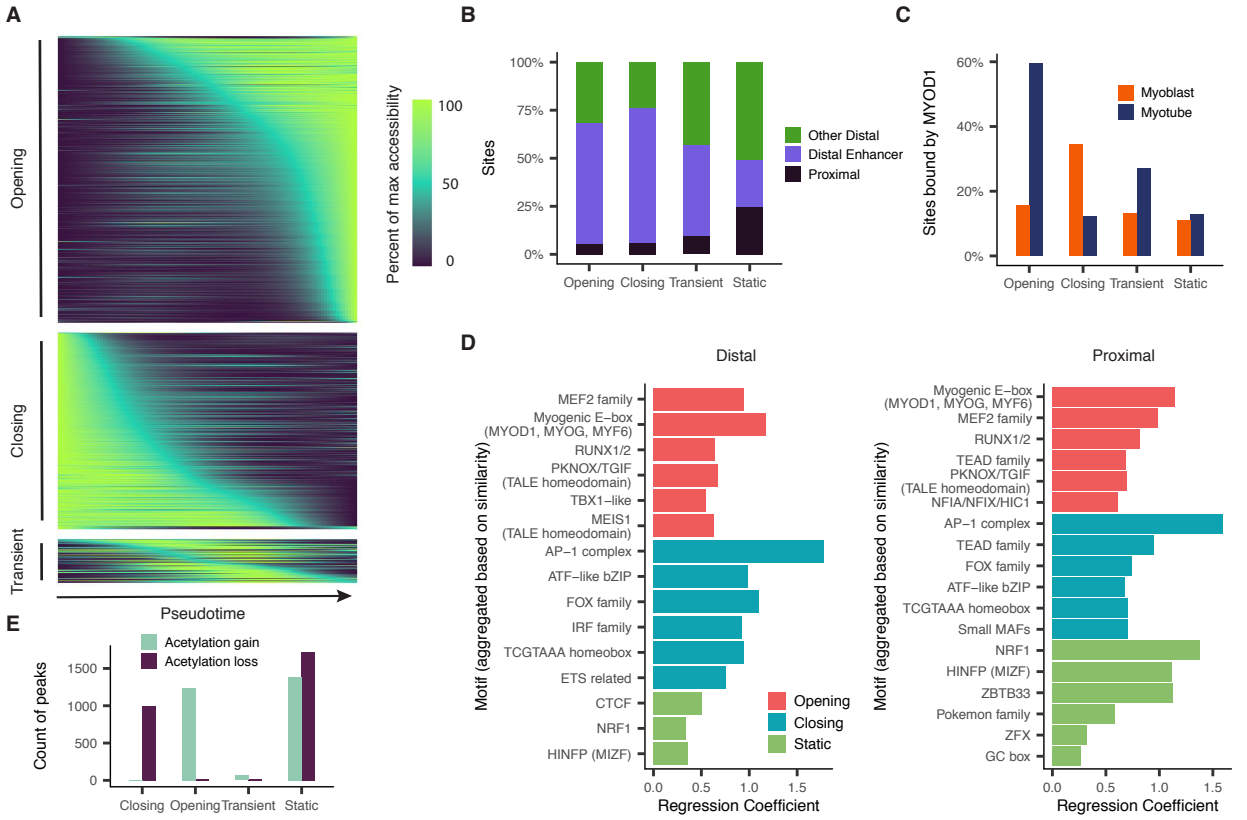

Figure 2.3: Thousands of DNA elements are dynamically accessible during myoblast differentiation. A) Smoothed pseudotime-dependent accessibility curves generated by a negative binomial regression and scaled as a percent of the maximum accessibility of each site. Curve regressions are the same as regression for differential accessibility (see Section 2.5, Methods). Each row indicates a different DNA element. Sites are sorted by the pseudotime at which they first reach half their maximum accessibility. B) Proportions of dynamic and static sites by site type. Color indicates whether a site is promoter-proximal (see Section 2.5, Methods), a distal enhancer (defined as peaks that are not promoter-proximal and are annotated by Segway as enhancers in either myoblasts or myotubes), or other distal (remaining sites). C) Percent of sites reported as bound by MYOD1 in either myoblasts or myotubes by Cao et al. (2010). D) Motif enrichments in accessible sites. p values result from logistic regression models that use the presence or absence of a given motif in each site to predict whether the site has a given accessibility trend (opening/closing/static). Plots show up to the top 6 Bonferroni-significant motifs by p value. E) Counts of sites undergoing significant changes in H3K27 acetylation as measured by chromatin immunoprecipitation sequencing (ChIP-seq) (Tang et al., 2015). See also Figure 2.4.

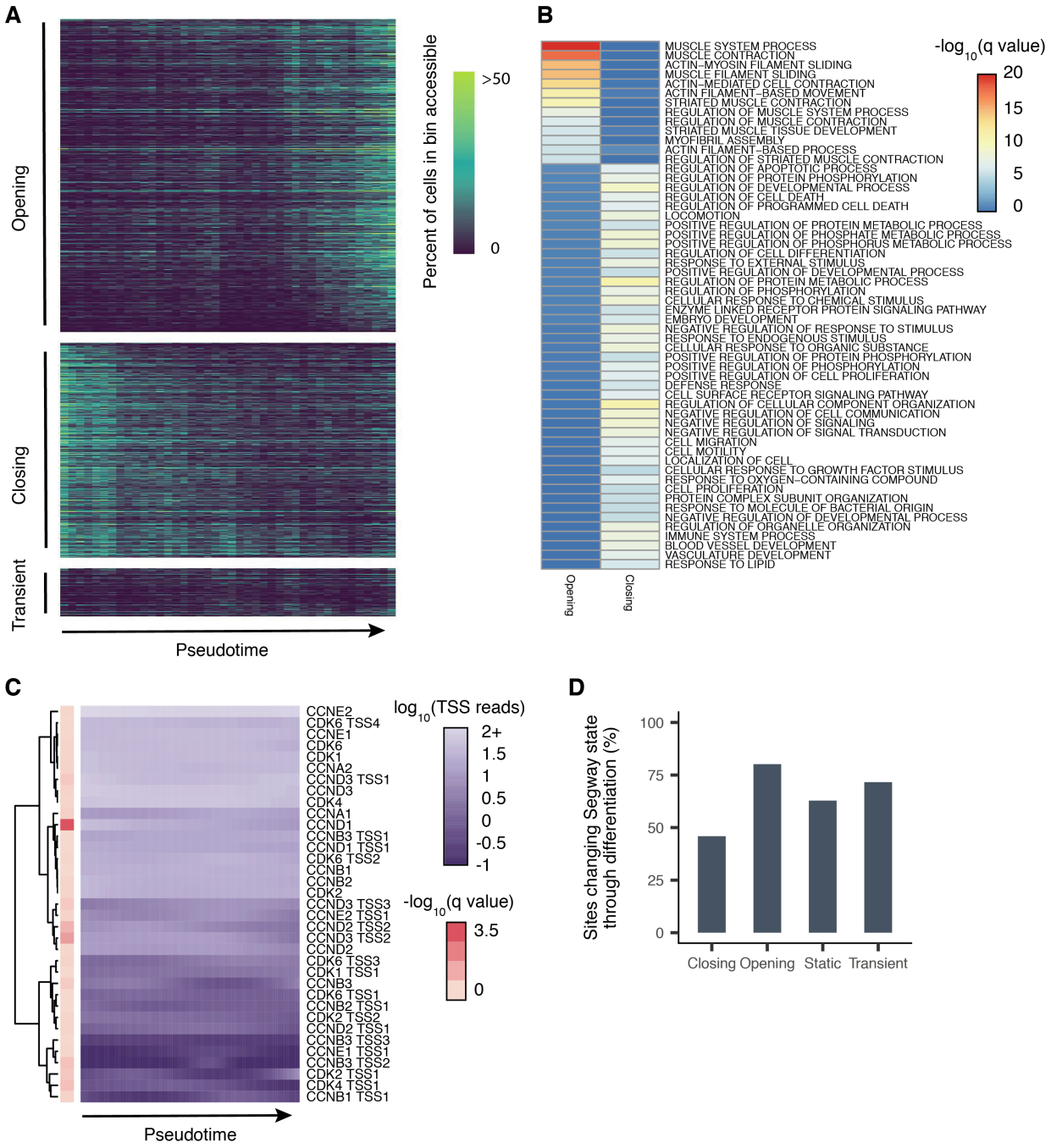

Figure 2.4: DNA elements that open during differentiation are enriched for muscle related promoters, related to Figure 2.3. A) Heatmap of accessibility across pseudotime. Color represents the percent of cells per bin that are accessible at a given DNA element. (*Legend continued on the following page*)

Figure 2.4: (*continued*) Each row indicates a different DNA element; each column represents a bin of approximately even numbers of cells divided by pseudotime. Rows are in the same order as Figure 2.3A. B) Gene set enrichment analysis of significantly opening and closing accessible sites. Adjusted p values were computed using a hypergeometric test. Terms shown are all sites with an adjusted p value $< 1 \times 10^{-6}$ in either the opening set or the closing set. Color represent the $-\log_{10}$ adjusted p value. Sites are ordered by the $-\log_{10}$ adjusted p value of the opening set. C) Smoothed pseudotime dependent accessibility curves, generated by a negative binomial regression of each for a set of selected cell cycle relevant genes. Each row indicates a different DNA element. Annotation column represents the $-\log_{10}$ adjusted p value for the test of differential accessibility across pseudotime. For visualization, fitted curve range was capped at 100. D) Percent of dynamic and static sites with changing Segway state assignment from myoblast to myotube.

(Figure 2.4B). Most markers of actively proliferating cells did not show significant changes in promoter accessibility (Figure 2.4C).

Comparison to ChIP-seq data (Cao et al., 2010) revealed that 59% of opening sites and 34% of closing sites are bound by MYOD1 in myotubes and myoblasts, respectively (Figure 2.3C). In contrast, only 16% of static sites (those without significant changes in accessibility) were MYOD1-bound in either myoblasts or myotubes. Dynamically accessible distal elements and promoters were also strongly enriched for binding motifs for MYOD1, MYOG, and MEF2 family members and other TFs with central regulatory roles in myogenesis (Figure 2.3D).

Many TFs recruit enzymes that mark histones near regulatory DNA elements. For example, MYOD1 recruits p300, whose histone acetyltransferase activity is required for its role in activating gene expression (Dilworth et al., 2004; Puri et al., 1997; Sartorelli et al., 1997). A comparison with ENCODE data for myoblasts and myotubes showed overwhelming directional concordance between sites that were gaining or losing H3K27 acetylation (H3K27ac) versus sites that were opening or closing in chromatin accessibility, respectively (Figure 2.3E). However, most changes in histone marks during differentiation occurred at sites that did not undergo significant changes in chromatin accessibility (Figure 2.4D). Thus, myoblast differentiation is characterized by changes in H3K27ac at hundreds of thousands of sites, only a minority of which were accompanied by changes in their chromatin accessibility, at least to the extent that they are detectable by the methods employed here.

2.3.3 Cicero constructs genome-wide cis-regulatory models from sci-ATAC-seq data

We next sought to exploit patterns of co-accessibility between distal elements and promoters to build a genome-wide *cis*-regulatory map. This is challenging for several reasons. First, the raw correlations are driven in part by technical factors such as read depth per cell. Second, we have insufficient observations to accurately estimate correlations between billions of pairs of sites. Third, single-cell ATAC-seq data are very sparse. Finally, while the accessibility of distal elements might be correlated with their target promoters, very distant or interchromosomal pairs of sites will also be correlated by virtue of being part of the same regulatory program.

To address these challenges, we developed a new algorithm, Cicero, that subtracts technical and genomic distance effects while constructing a global *cis*-regulatory map from single-cell chromatin accessibility profiles (Figure 2.5A). Briefly, the user provides Cicero with cells as input that have been clustered or pseudotemporally organized. The algorithm creates many groups, each comprised of 50 cells similarly positioned in clustering or trajectory space. This helps to overcome the sparsity of the data while avoiding Simpson's paradox (Simpson, 1951; Trapnell, 2015). It then aggregates accessibility profiles for cells in each group to produce counts that can be readily adjusted to subtract the effects of technical variables. Finally, it computes the correlations in adjusted accessibilities between all pairs of sites within 500 kb. To calculate robust correlations, we use Graphical LASSO (Friedman et al., 2008), which estimates regularized correlation matrices. Cicero penalizes correlations in a distant-dependent manner, preserving local patterns at the expense of very long-range ones. The output of Cicero consists of the co-accessibility scores for all pairs of sites within 500 kb of one another. Full details are provided in Section 2.5 (Methods).

We applied Cicero to generate a genome-wide *cis*-regulatory map from our myoblast sci-ATAC-seq data. As the first step, for example in experiment 1, Cicero aggregated differentiating myoblasts into 277 groups and identified 6.5 M pairs of sites with positive co-accessibility scores, including 1.8 M comprising a distal element and promoter. As the co-accessibility threshold is raised, promoters are connected to fewer regulatory elements with higher confidence. For example, at a cutoff of 0.25, promoters were connected to a median of 2 distal elements in experiment 1 (Fig-

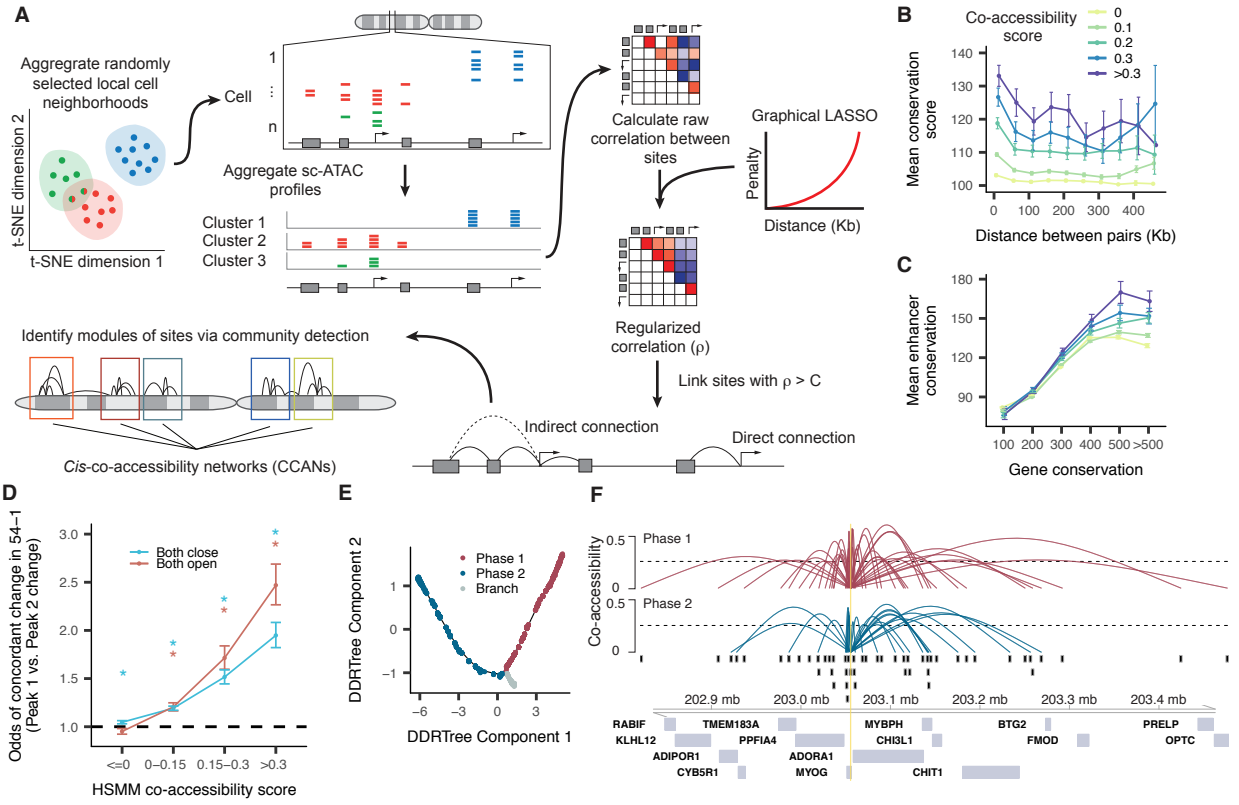

Figure 2.5: Cicero constructs *cis*-regulatory models genome-wide from sci-ATAC-seq data. A) An overview of the Cicero algorithm (see Section 2.5, Methods, for details). B) Mean phastCons 46-way placental conservation scores of distal peaks connected to promoters. Peaks were stratified by distance from the promoter and co-accessibility score between the promoter and the distal peak. C) Mean distal site conservation score versus connected gene conservation score stratified by co-accessibility score. D) Odds ratios of concordant accessibility dynamics across differentiation in 54-1 myoblasts between pairs of sites that are co-accessible in HSMM. For each bin of co-accessibility in HSMM, pairs of peaks that overlapped peaks in 54-1 non-targeting controls were assessed for concordant dynamics (>2 $\log_2$ fold change in both peaks or <-2 $\log_2$ fold change in both peaks). Error bars indicate 95% confidence intervals calculated using Fisher's exact test. Asterisks represent estimates significantly different than 1 (p values <0.05 by Fisher's exact test). E) Two "phases" of myoblast differentiation illustrated. F) A summary of the Cicero co-accessibility links between the *MYOG* promoter and distal sites in the surrounding region. The height of connections indicates the magnitude of the Cicero co-accessibility score between the connected peaks. The top set of (red) links were constructed from cells in phase 1, while the bottom (in blue) were built from phase 2. See also Figure 2.6 and Figure 2.7.

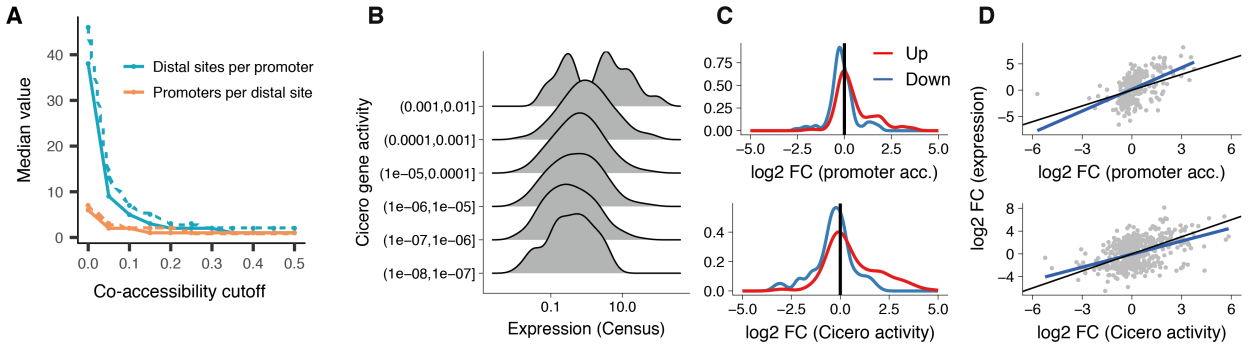

Figure 2.6: Cicero gene activity scores correlate with gene expression, related to Figure 2.5. A) The median number of linked distal sites per promoter and promoters per distal site as a function of the co-accessibility threshold of the links considered. Dashed lines indicate experiment 2. Solid lines indicate experiment 1. B) Average Cicero gene activity scores across cells in phase 1 compared to their average expression from scRNA-seq libraries from myoblasts. Cicero gene activity scores were computed by summing the reads falling in distal sites linked to a gene's promoter (see Section 2.5, Methods, for details). C) Top panel: $\log_2$ fold changes in mean accessibility at gene promoters for genes that are significantly up- (red) or down-regulated (blue) between 0 and 72 hours as measured by scRNA-seq. Bottom panel: corresponding changes in Cicero gene activity scores. D) Top panel: comparison of $\log_2$ fold changes between expression and promoter accessibility. Bottom panel: fold changes in expression versus changes in Cicero gene activity scores. Black lines indicate perfect concordance between $\log_2$ fold changes, while blue lines indicate linear regressions between Cicero activity or promoter accessibility and expression.

ure 2.6A). Distal sites that were highly co-accessible with promoters were more conserved across vertebrates (Figure 2.5B). This trend was more pronounced for sites linked to highly conserved genes, which tended to be co-accessible with more conserved distal elements (Figure 2.5C). To verify that co-accessibility between sites was not confined to our specific primary myoblast culture, we performed bulk ATAC-seq in myoblasts from another donor ("54-1"), before and after differentiation. Reassuringly, highly co-accessible sites were 2.2-fold more likely than unlinked sites to be undergoing directionally concordant changes in accessibility across differentiation in the 54-1 cells (Figure 2.5D). To explore how accessibility corresponded with gene regulation during differentiation, we devised a composite "gene activity score" of accessibility at both promoters and linked distal sites (Section 2.5, Methods). Accessibility-based gene activity scores were positively correlated with expression (Figure 2.6B, Figure 2.6C and Figure 2.6D).

As co-accessible elements tended to cluster, we post-processed Cicero's output with a community detection algorithm to identify "*cis*-co-accessibility networks" (CCANs): modules of sites that are highly co-accessible with one another. The majority of dynamically accessible sites were included in CCANs even using a high co-accessibility threshold (Figure 2.7A, Figure 2.7B, Figure 2.7C, Figure 2.7D, and Figure 2.7E). To assess the reproducibility of Cicero maps, we adapted a maximum weighted bipartite matching method to identify pairs of CCANs from the two experiments that share DNA elements in common (Section 2.5, Methods). This algorithm matched 1,868 of the CCANs between the experiments, accounting for 84% and 91% of the sites in CCANs in experiments 1 and 2, respectively (Figure 2.7F). Most pairs of sites linked in one experiment were also linked in the other (score $>$0.25; 81% of experiment 1 sites also linked in experiment 2; 64% of experiment 2 sites also linked in experiment 1; Figure 2.7G and Figure 2.7H).

To further investigate chromatin dynamics during differentiation, we constructed Cicero maps on the two "phases" of the pseudotime trajectory (before versus after the F2 branch) and computed CCANs for each map (Figure 2.5E). The general structure of Cicero connections was often maintained around genes of interest. For example, a similar set of distal elements are linked to the promoter of *MYOG* in both the early and late phases (Figure 2.5F). To identify CCANs that were maintained, gained, or lost during differentiation, we applied our matching algorithm to compare CCANs between the first and second phases. For experiment 1, this algorithm matched 1,945 CCANs, accounting for 88% and 91% of the sites in CCANs in the first and second phases, respectively. However, although the general structure of CCANs was stable (few sites switched CCANs), many sites joined or left CCANs during differentiation (Figure 2.7I and Figure 2.7J). Intriguingly, we identified 60 sequence motifs that were predictive of whether a site would join, leave, or remain within a CCAN, including CTCF, which strongly predicted that an accessible site would remain within a CCAN (Figure 2.7K).

We hypothesized that the CCANs identified by Cicero constitute "chromatin hubs." Chromatin hubs, which are thought to involve looping interactions between distal regulatory elements and the genes they target, may act to coordinate the assembly of transcription complexes (de Laat and Grosveld, 2003; Tolhuis et al., 2002). To satisfy the definition of a chromatin hub, we expect

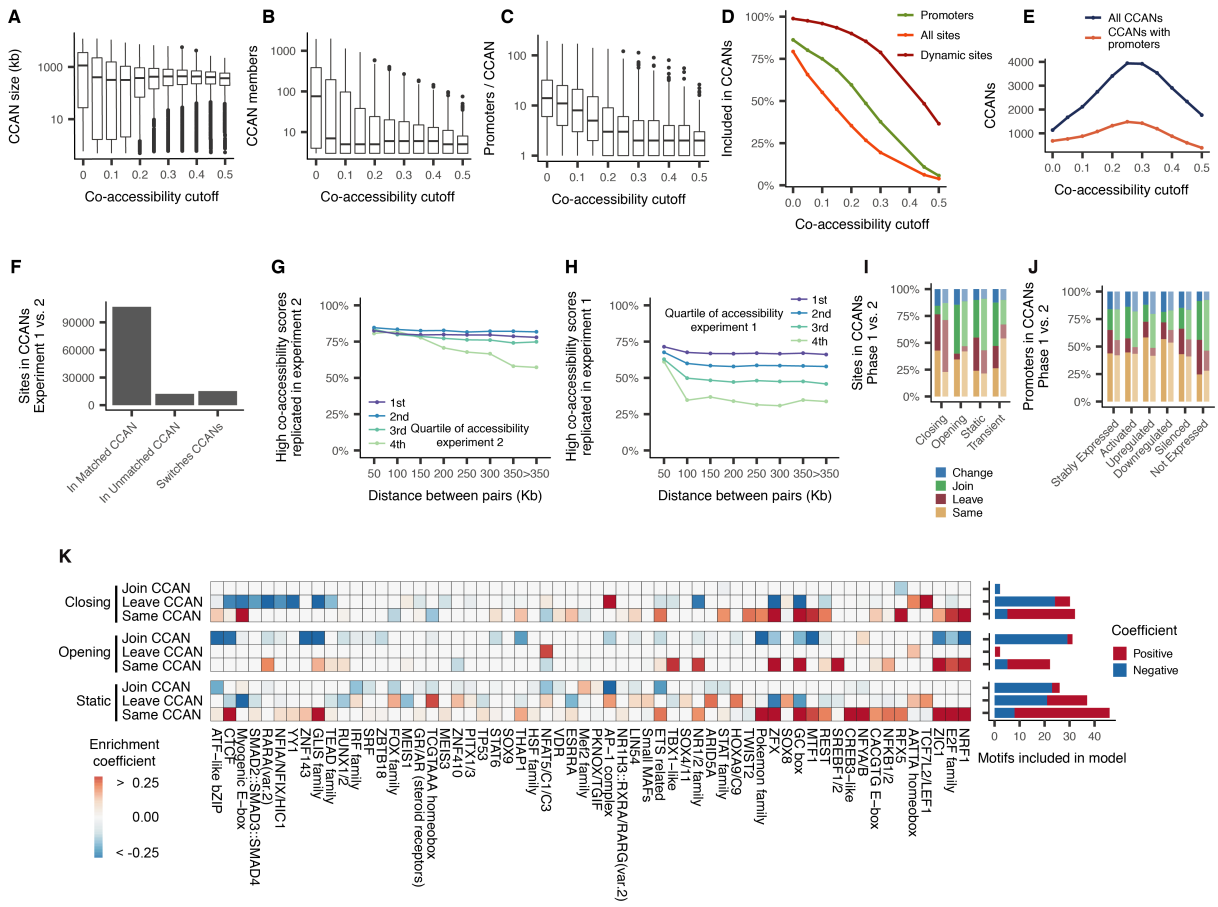

Figure 2.7: *Cis*-co-accessibility networks (CCANs) maintain properties at varying cutoffs, related to Figure 2.5. A) Boxplots of the length in the linear genome (in kilobases) of CCANs formed at varying thresholds of co-accessibility in experiment 1. CCANs are defined as groups with 3 or more co-accessible DNA elements identified with Louvain community detection. Prior to running Louvain, connections below the indicated Cicero co-accessibility score are excluded (see Section 2.5, Methods, for details). B) Boxplots of the number of sites in CCANs formed at varying thresholds of co-accessibility. C) Boxplots of the number of expressed gene (at a level of 10 transcripts per cell on average in scRNA-seq) promoters per CCAN at increasing co-accessibility score cutoffs. D) Percent of sites recruited into a CCAN at increasing co-accessibility score cutoff. Colors represent subsets of sites: green represents promoters for genes that are expressed; orange and red represent sites that are accessible and differentially accessible across pseudotime, respectively. E) Number of CCANs identified with varying co-accessibility cutoffs. Blue series shows the total number of CCANs, orange shows the number of CCANs that include a promoter of at least 1 detectably expressed gene. F) Number of sites that linked into CCANs that are matched in experiment 1 and experiment 2 by a maximum weighted bipartite matching method (see Section 2.5, Methods). Also shown are sites that are linked into CCANs that are not matched at all and sites that are linked into CCANs that are matched in experiment 1 and experiment 2, but not to one another. (*Legend continued on the following page*)

Figure 2.7: (*continued*) G) Fraction of pairs of sites linked in experiment 1 at co-accessibility <0.25 that are also linked in experiment 2 at co-accessibility <0. Colors indicate the quartile of accessibility in experiment 2. H) Reciprocal plot to panel (G), examining sites linked at co-accessibility <0.25 in experiment 2 and <0 in experiment 1. Colors indicate the quartile of accessibility in experiment 1. I) Sites linked into CCANs found in both phase 1 and phase 2 by maximum matching, subdivided by those that were linked into the CCAN in both phases (yellow), those linked in phase 1 but unlinked in phase 2 (red), those unlinked in phase 1 but linked in phase 2 (green), and those linked into different (i.e. non-matched) CCANs in the two phases. The four groups of sites from Figure 2.3B, Figure 2.3C, Figure 2.3E are considered. The left bar in each group corresponds to experiment 1, while the right bar corresponds to experiment 2. J) Similar to panel (I) but considering only promoters: groups are promoters of genes that are "stably" expressed at an unchanged level, those that are silent in myoblasts but expressed in myotubes ("activated"), and those expressed in both myoblasts and myotubes, but higher in myotubes ("upregulated"). Similarly, we show promoters of genes that are downregulated or fully silenced in myotubes, as well as those that are not detectably expressed (at a level of 10 transcripts per cell on average) in either cell type. K) A heatmap of regression coefficients from three multinomial elastic net regression analyses that predict whether a site will join, leave, or remain linked into its CCAN during differentiation on the basis of varying sequence motifs. Coefficients were capped at -0.25 and 0.25 for visualization. Only sites with consistent CCAN dynamics across both experiments were included in the models. The number of positive and negative coefficients surviving regularization in each model are shown in the barplot to the right. Regression was performed using the glmnet package in R and the regularization parameter was chosen that produced the minimum mean cross-validated error after 10-fold cross validation.

CCANs should meet four criteria. First, they should exhibit greater physical proximity than expected based on their distance in the linear genome. Second, they should interact with a common set of protein complexes. Third, they should be epigenetically modified in concordant ways and at similar times. Finally, they should substantially contribute to regulating genes with promoters within the hub.

2.3.4 Co-accessible DNA elements exhibit physical proximity

To test whether co-accessible sites are closer together in the nucleus than unlinked sites at similar distances in the linear genome, we generated and applied Cicero to sci-ATAC-seq chromatin profiles from 889 human lymphoblastoid cells (GM12878), for which ChIA-PET and promoter-capture Hi-C data are available.

We observed strong concordance between Cicero-based linkages and DNA elements in RNA

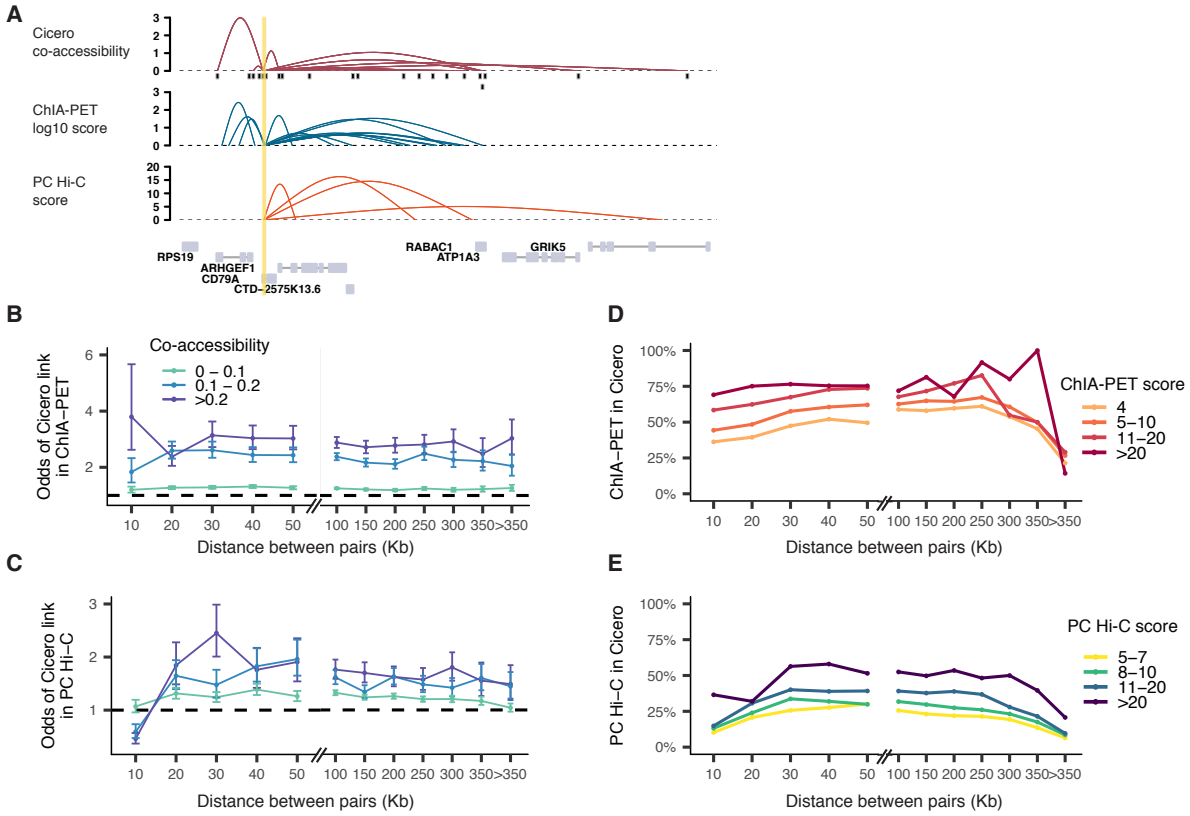

Figure 2.8: Co-accessible DNA elements linked by Cicero are physically proximal in the nucleus. A) Cicero connections for the *CD79A* locus compared to RNA polymerase II ChIA-PET (Tang et al., 2015) and promoter capture Hi-C (Cairns et al., 2016). Link heights for ChIA-PET are log-transformed frequencies of each interaction PET cluster and for promoter capture Hi-C are soft-thresholded log-weighted p values from the CHiCAGO software. B) Odds ratio of pairs of sites within a given co-accessibility and distance bin found in RNA polymerase II ChIA-PET compared to pairs of sites with co-accessibility ≤ 0. Color represents the co-accessibility bin. Error bars indicate 95% confidence intervals calculated using Fisher's exact test. All points shown were significantly different than 1 (p values $<$0.05 by Fisher's exact test). C) Similar to (B) but comparing Cicero links to sites ligated in promoter-capture Hi-C. All points shown were significantly different than 1 (p values $<$0.05 by Fisher's exact test) except for at the 10 kb bin that may be impacted by the resolution of Hi-C consequent to the use of a 6-cutter restriction enzyme. For (B) and (C), we fit a linear model that included co-accessibility score and overall accessibility and found in both cases that co-accessibility had a significant effect on presence in comparison datasets even after correcting for this potential confounder (see Section 2.5, Methods for details). D) Fraction of ChIA-PET contacts found in Cicero connections as a function of distance, stratified by multiplicity of ligation product detections. E) Promoter-capture contacts detected in Cicero CCAN connections as a function of distance, stratified by CHiCAGO score. See also Figure 2.9.

polymerase II-mediated contacts captured via ChIA-PET (Tang et al., 2015) as well as contacts found by promoter-capture Hi-C (Cairns et al., 2016; Mifsud et al., 2015), e.g., at the *CD79A* locus (Figure 2.8A). Approximately half of DNA elements ligated via ChIA-PET ("anchors") overlapped with accessible sites in our data, with greater overlap between anchors that were supported by multiple ChIA-PET reads and sites that were accessible in many cells (Figure 2.9A and Figure 2.9B). Pairs of sites reported by Cicero to be co-accessible were up to 2- to 3-fold more likely to be found in ChIA-PET and promoter-capture Hi-C than unlinked sites separated by similar distances (Figure 2.8B and Figure 2.8C). Reciprocally, pairs of sites linked by many independent ChIA-PET or Hi-C ligation fragments were more likely to also be reported as co-accessible by Cicero (Figure 2.8D and Figure 2.8E), e.g., 75% of high-confidence ChIA-PET connections were found in Cicero's map. Although proximity ligation frequencies should not be taken as a direct measure of physical distance, these analyses show that Cicero-linked sites exhibit greater than expected physical proximity, even when very distant in the linear genome. We also found that Cicero connected sites were more likely to occupy the same topologically associated domain (TAD) than unlinked sites at the same distance (Fisher's exact test, p value $<2 \times 10^{-5}$ for all distance bins, TADs derived from 1 kb-resolution Hi-C analysis of GM12878 (Rao et al., 2014)). Similarly, Cicero-linked sites were 1.5-fold more likely than unconnected pairs at the same distance to be found in the same A/B compartment (Figure 2.9C and Figure 2.9D).

2.3.5 Co-accessible DNA elements carry pairs of motifs for interacting TFs

We next investigated whether Cicero links might be mediated by interacting TFs. We searched for known sequence motifs within each peak in the HSMM data that could accurately predict whether Cicero would link other sites to it. Promoters with DNA binding motifs for one or more core myogenic TFs were significantly more likely to be connected (co-accessibility score >0.25) to an opening distal site than promoters without them. For example, promoters containing at least one MYOD1, MYOG, or MYF6 motif were 3.6-fold more likely to be connected to an opening distal site than promoters with none of these motifs ($p = 8.7 \times 10^{-270}$; likelihood ratio test for logistic

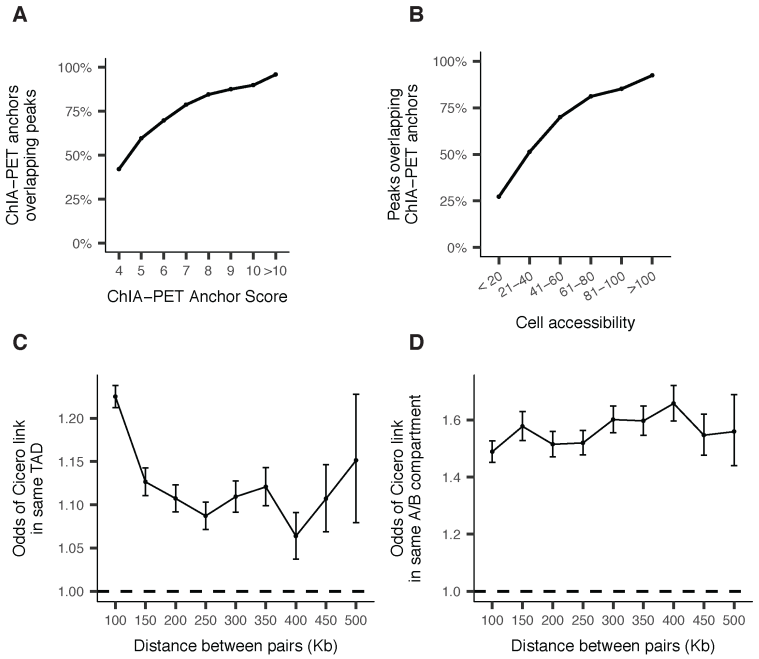

Figure 2.9: ChIA-PET anchors are concordant with sci-ATAC-seq peaks, related to Figure 2.8. A) Percent of polymerase II ChIA-PET anchors within 1 kb of a sci-ATAC-seq peak as a function of ChIA-PET anchor score provided by Tang et al. (2015). B) Percent of sci-ATAC-seq peaks within 1 kb of polymerase II ChIA-PET anchors as a function of overall cell accessibility (number of cells where the peak is accessible). C) Odds ratio that both members of Cicero linked pairs (co-accessibility >0) are in the same TAD, compared with unlinked pairs at the same distance. GM12878 TAD calls are from Rao et al. (2014). D) Odds ratio that both members of Cicero linked pairs (co-accessibility >0) are in the same A/B compartment, compared with unlinked pairs at the same distance. A/B compartment calls are from Fortin and Hansen (2015).

regression model), and similarly, promoters with at least one MEF2 family motif were 2.8-fold more likely to be connected to an opening distal site ($p = 7.9 \times 10^{-119}$).

We hypothesized that these correlations resulted from direct, TF-mediated interactions. To explore this further, we focused on promoters linked to exactly one dynamically accessible distal site (co-accessibility score >0.05) and used Graphical LASSO to identify pairs of motifs where the presence of a motif in the promoter predicted the presence of the paired motif in the dynamically accessible distal site (Section 2.5, Methods). We identified a number of motif pairs corresponding to TFs known to physically interact. For example, opening distal elements were significantly more

likely to have a MEF2 or RUNX1 motif if they were linked to a promoter with a MYOD1 motif (Figure 2.10A). Myogenic regulatory factors (MRFs) are known to interact physically with MEF2 and RUNX1 (Knoepfler et al., 1999; Molkentin et al., 1995; Philipot et al., 2010).

2.3.6 MYOD1 coordinates histone modifications in cohorts of co-accessible sites

The physical proximity of co-accessible sites suggested that recruitment of histone-modifying enzymes to one site might induce changes in physically proximate sites. Indeed, pairs of sites were more likely to be undergoing significant, concordant gains in H3K27ac if they were linked by Cicero (Figure 2.11A). Sites that themselves exhibited static accessibility, but were linked to a dynamic, opening site, showed strong gains in H3K27ac, while static sites that were linked to dynamic, closing sites showed strong losses (Figure 2.11B). The gains in acetylation might be driven by *de novo* binding of MYOD1 at the opening site followed by recruitment of a histone acetyltransferase (e.g., p300). Supporting this, of the 2,050 sites with significant gains in H3K27ac but static accessibility, only 46% were bound by MYOD1 in myotubes. However, 97% were linked by Cicero to a MYOD1-bound site (Figure 2.11C). Moreover, equipping a regression model with information about linked sites improved its accuracy in predicting changes in a site's histone marks (Figure 2.10B).

We next considered whether gains in MYOD1 binding were concentrated in a few CCANs or widely distributed. Of the 2,323 hubs containing promoters, 74% contained at least one site undergoing a change in MYOD1 binding. For the subset of 431 hubs with a differentially expressed gene, 92% contained at least one site changing in MYOD1 binding. For example, within the single hub that includes myosin heavy chain isoforms 1, 2, 3, 4, 8, and 13 and numerous other genes, 15 sites underwent significant increases in accessibility. Of these, all were bound by MYOD1 in myotubes (Figure 2.11E). Interestingly, however, two sites very near *MYH3* (marked with asterisks) opened substantially earlier in pseudotime than others and were bound by MYOD1 in myoblasts as well.

We wondered, more generally, whether sites bound by MYOD1 in myoblasts and through-

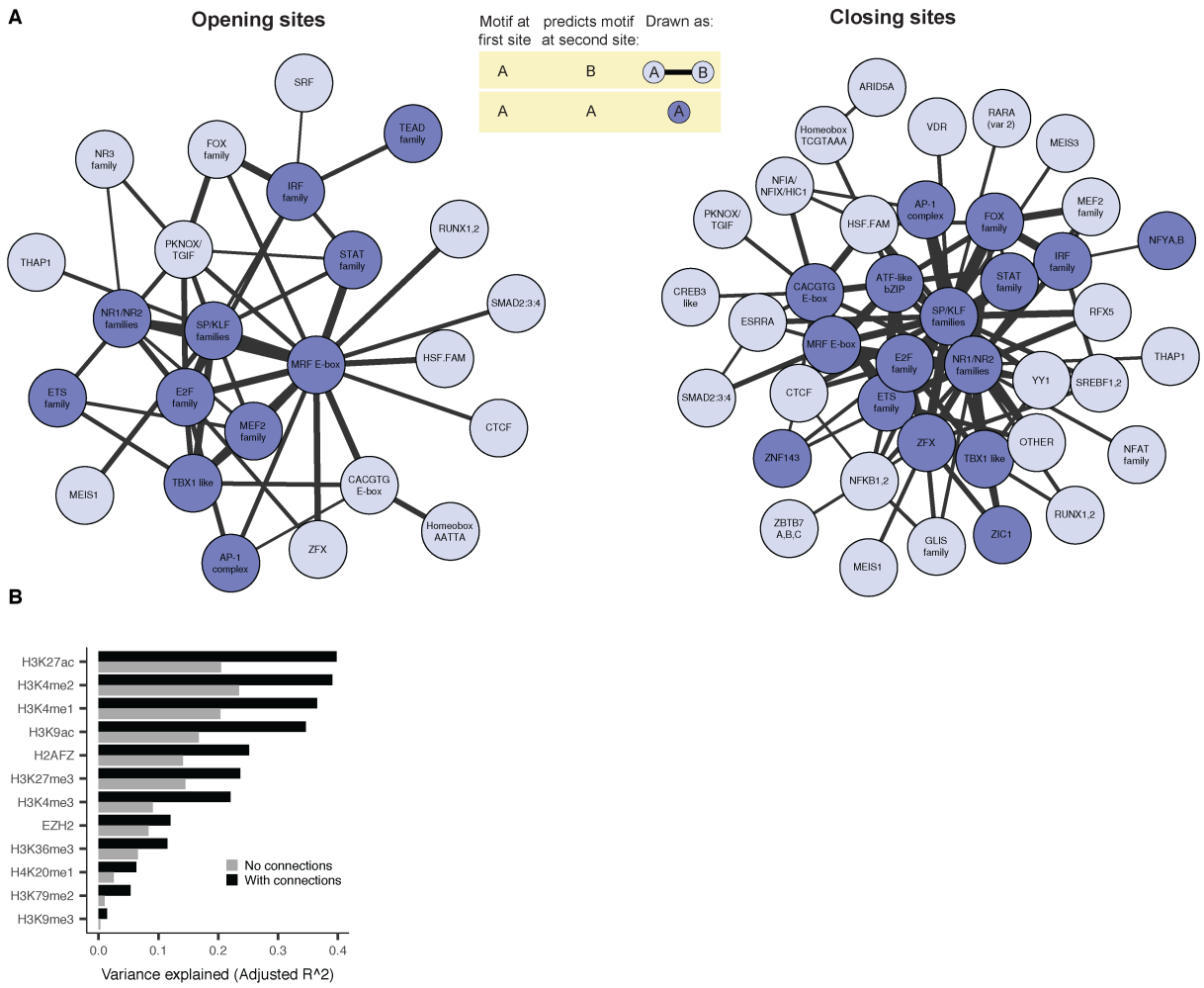

Figure 2.10: DNA motifs predict motifs in Cicero-linked sites, related to Figure 2.11. A) Motifs in accessible sites predict motif content of Cicero-linked sites. The network summarizes a graphical model that captures how occurrences of motifs in pairs of sites predict whether they are connected. Each motif is connected to the motifs it suggests will exist in one or more connected sites. A motif that predicts itself in a connected site is shown in dark blue. If motif "A" at a distal site predicts that "B" will be found at a promoter, and symmetrically "B" at a distal site suggests "A" will be found at a promoter, they are connected with a black line, with a width proportional to the strength of the co-accessibility. Asymmetric motif relationships are not shown. B) Variance explained by a linear model that aims to predict $\log_2$-transformed fold changes in the listed ChIP-seq read counts between myoblasts and myotubes. Two models are considered. The first, with performance indicated as gray bars, uses a site's accessibility and MYOD1 binding status. The second, indicated as black bars, augments the first with accessibility and MYOD1 at linked sites. The predictor for MYOD1 at linked sites was significant (p-value <0.05) for all augmented models. See Section 2.5, Methods, for more details.

out differentiation opened earlier than sites that gained MYOD1 binding during differentiation. A changepoint analysis using PELT (Killick et al., 2012) revealed that sites bound by MYOD1 throughout differentiation opened significantly earlier in pseudotime than those that gained MYOD1 (Mann-Whitney test p value 1.2×10^{-122}) or were never bound by it (Mann-Whitney test p value 8.0×10^{-223}) (Figure 2.11F). Moreover, rather than being enriched in whole hubs that open early as a group, constitutively MYOD1-bound sites opened significantly earlier than sites linked to them that either gained MYOD1 (two-sided paired Student's t test p value = 1.0×10^{-182}) or were never bound by it (two-sided paired Student's t test p value = 4.5×10^{-318}) (Figure 2.11G). Constitutively MYOD1-bound sites were enriched for the MEIS1 and AP-1 motifs (Figure 2.11H) compared with sites that gain MYOD1 later in differentiation. Altogether, sites with MEIS1 motifs were linked to 69% of dynamically opening sites compared with only 16% of sites genome-wide (co-accessibility score $>$0.25). Murine Meis1, in conjunction with Pbx1, has been reported to act as a complex required for the MYOD1-mediated activation of the myogenin promoter, and mutations in MYOD1 that prevent interaction with PBX1 resulted in loss of many binding sites and regulatory targets (Berkes et al., 2004; Fong et al., 2015). Our results suggest MEIS1 recruitment of MYOD1 may be pervasive throughout the genome and could nucleate activation of other sites within a chromatin hub.

2.3.7 Sequence features of active chromatin hubs predict gene regulation

We wondered whether Cicero's putative maps could be used to predict changes in gene expression. As a first test, we asked whether two genes with co-accessible promoters exhibited greater correlation in expression across individual cells than genes that were nearby but whose promoters were not linked by Cicero. Indeed, differentially expressed genes showed greater correlation in expression as a function of their co-accessibility score (Figure 2.12).

We next sought to develop a linear regression model to predict changes in either gene expression or changes in "barrier region" histone marks associated with promoter activation (Figure 2.13A). Our first model takes as input a binary map of the TF binding motifs present at the

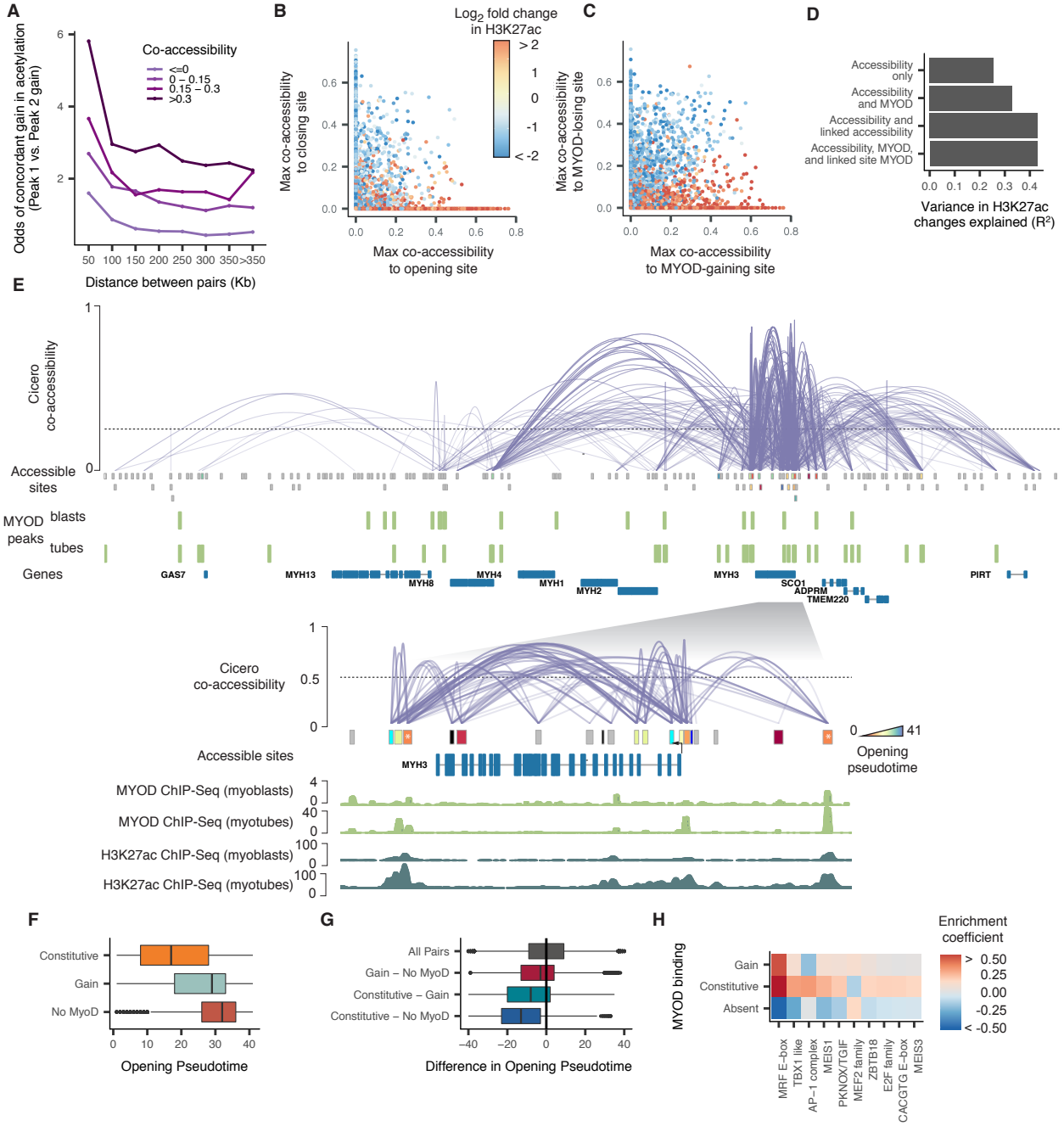

Figure 2.11: Co-accessible DNA elements linked by Cicero are epigenetically co-modified. A) Odds ratio of a site gaining H3K27ac during myoblast differentiation, given that it is linked to a site that is doing so. Color indicates the strength of the Cicero co-accessibility links. The lightest color indicates pairs of sites that are unlinked by Cicero. (*Legend continued on the following page*)

Figure 2.11: (*continued*) B) Correspondence between a statically accessible site's gain or loss of H3K27ac and its maximum co-accessibility score to a site that is opening (x axis) or closing (y axis). Sites that are not linked to an opening or closing site are drawn at x = 0 or y = 0, respectively. C) Similar to (B) but describing the correspondence between a site's gain or loss of H3K27ac and its maximum co-accessibility score to a site that is gaining or losing MYOD1 binding. D) The variance explained in a series of linear regression models in which the response is the $\log_2$ fold change in H3K27ac level of each DNA element and the predictors are whether that site is opening, closing, or static, whether it gains or loses MYOD1 binding, and whether it is linked to neighbors that are doing so. See Section 2.5, Methods, for details on model specifications. E) The Cicero map for the 755 kb region surrounding *MYH3* along with called MYOD1 ChIP-seq peaks from Cao et al. (2010). Sites opening in accessibility are colored by their opening pseudotime (see Section 2.5, Methods), sites not opening in accessibility are shown in gray. Inset: 60 kb region surrounding *MYH3* along with MYOD1 ChIP-seq and H3K27ac ChIP-seq signal tracks from Cao et al. (2010) and the ENCODE Project Consortium (The ENCODE Project Consortium, 2012). Only protein-coding genes are shown. F) Opening pseudotimes for all opening sites, subdivided by whether MYOD1 is bound in myoblasts and myotubes, myotubes alone, or neither. G) The difference in opening pseudotimes between pairs of linked DNA elements. The pairs are grouped based on whether one or both sites is constitutively bound by MYOD1. H) TF binding motifs selected by an elastic net regression (alpha = 0.5), with a response encoding the MYOD1 binding status of each site. See also Figure 2.10.

promoter upstream of each TSS. We then train it to predict how much of a gene's observed expression change is attributable to each TF motif using elastic net regression and 50-fold cross-validation (Section 2.5, Methods). The promoter-based model explained only 17% of the variance in expression and performed similarly in predicting a panel of histone marks (Figure 2.13B and Figure 2.13C). Augmenting this model with TF motifs at linked distal sites improved its ability to explain changes in expression by 2.27-fold (Figure 2.13C). The TF motifs identified by the model included the MRF E-box, the MADS box bound by MEF2 family proteins, and the MEIS1 binding site, which were associated with upregulation, along with motifs for factors that drive cell proliferation such as AP-1, which were associated with downregulation (Figure 2.13D). Thus, when tasked with predicting which factors are important for gene regulation, our regression identified the major myogenic TFs using only the sequences in sites linked together by Cicero.

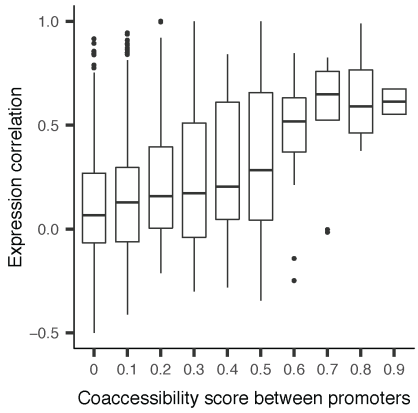

Figure 2.12: Expression correlation increases with increasing co-accessibility, related to Figure 2.13. Correlation in expression among linked differentially expressed genes. Boxplots of the cell-wise correlation between gene expression among pairs of differentially expressed genes whose promoters have different Cicero co-accessibility scores.

2.3.8 MEIS1 and PBX1 are required for coordinated myoblast chromatin hub activation

We hypothesized that MYOD1, which recruits p300/PCAF and the BAF complex upon myoblast differentiation, might act to nucleate histone modification and nucleosome remodeling throughout a chromatin hub. To test this hypothesis, we genetically ablated MEIS1 or PBX1 (that forms a heterodimer with MEIS1) with CRISPR/Cas9 in 54-1 cells and then performed bulk ATAC-seq as they differentiated. Both ΔMEIS1 and ΔPBX1 myoblasts differentiated markedly less efficiently, with fewer and smaller myotubes than cells transduced with non-targeting control (NTC) single guide RNAs (sgRNAs) (Figure 2.14A). Of 14,321 sites that underwent significant changes in accessibility in the 54-1 NTC, 7,868 (55%) and 12,520 (87%) failed to do so in ΔMEIS1 or ΔPBX1 cells respectively, and nearly 25% of sites that failed to open overlapped sites bound by MYOD1 in both normal myoblasts and myotubes (Figure 2.14B).

Having observed that pairs of sites Cicero identified as highly co-accessible in HSMMs were more likely to open or close concordantly in the 54-1 cells upon differentiation (Figure 2.5D), we asked whether co-accessible pairs would be concordantly perturbed in the mutants. Indeed, pairs

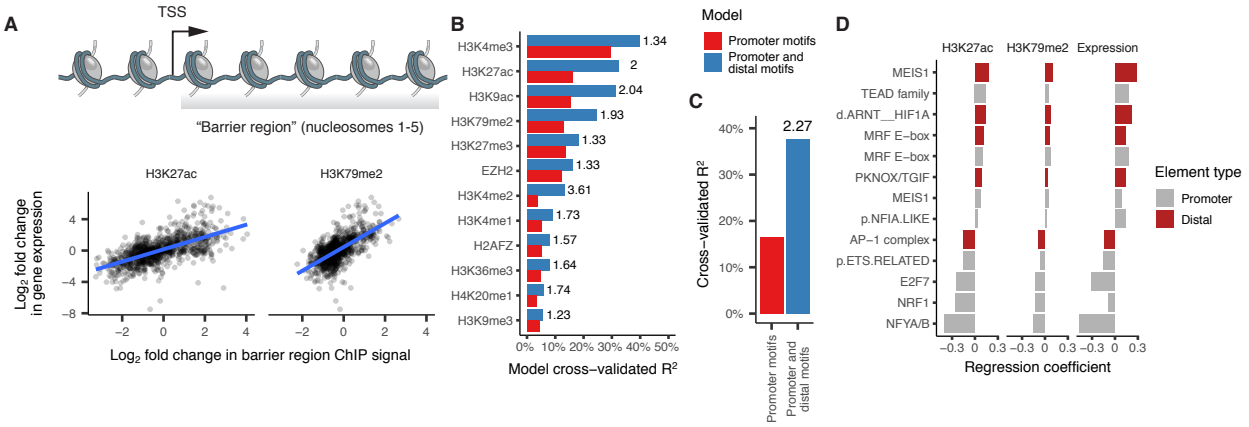

Figure 2.13: Chromatin dynamics at distal DNA elements predict gene regulation. A) Changes in histone acetylation in the first 1 kb downstream of each gene's TSS, corresponding to the "barrier" to RNA polymerase II elongation posed by nucleosomes, are correlated with changes in the gene's expression. B) Two regression models predict changes in the histone marks deposited throughout each gene's barrier region. The first model predicts changes on the basis of TF binding motifs in gene promoters. The second model adds variables encoding the strength of co-accessibility with linked sites containing the motif. See Section 2.5, Methods, for details on the various models. Adjusted R^2 is computed as the fraction of null deviance explained. The number to the right of each bar indicates the ratio of variance explained between the first and second model. C) Similar to (B), with changes in expression as the response. D) Coefficients from the model incorporating sequence at distal sites for each motif surviving model selection via elastic net. Note that the model considers each motif twice: once at promoters and again at distal sites, and both can be selected by elastic net. See also Figure 2.12.

of sites that opened in the 54-1 NTC and were linked by Cicero tended to both fail to open in the mutants. For example, pairs of sites that Cicero linked with a co-accessibility score >0.3 were 2.3-fold more likely to both fail to open in ΔPBX1 and 1.6-fold more likely in ΔMEIS1 than pairs of sites Cicero deemed not co-accessible, suggesting that co-accessibility is often maintained even when cells fail to differentiate (Figure 2.14C).

We next assessed whether constitutively MYOD1-bound sites might nucleate changes throughout hubs by first dividing normally opening sites into those that were constitutively MYOD1-bound and those that were not. We then tested whether other sites linked to these groups at varying levels were more or less likely to coordinately fail to open in ΔMEIS1 (Figure 2.14D). Consistent with our hypothesis, highly co-accessible sites were 1.5-fold more likely to both fail to open in

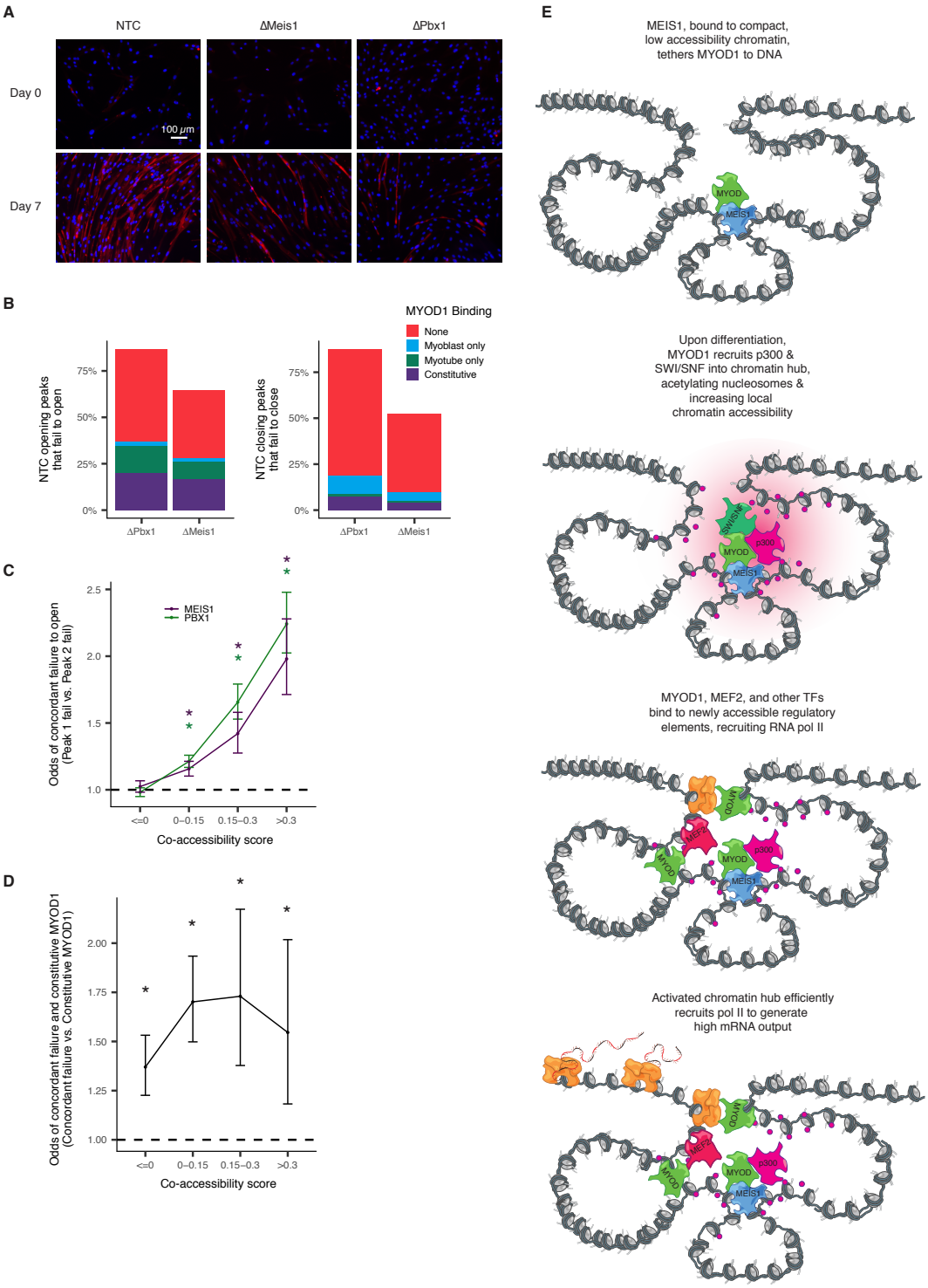

Figure 2.14: *(Legend continued on the following page)*

Figure 2.14: (*continued*) MEIS1 and PBX1 knockout myoblasts fail to differentiate and show coordinated accessibility defects. A) Immunofluorescence microscopy images of non-template control, MEIS1 knockout and PBX1 knockout 54-1 cells at day 0 and day 7 post induction of differentiation. Nuclei are stained using DAPI, MYH3 is stained using anti-myosin MF20 and Alexa Fluor 594. B) Percent of peaks that open during differentiation in the NTC that fail to open in PBX1 and MEIS1 knockouts. Colors indicate the presence of MYOD1 binding in myoblasts and myotubes by ChIP-seq in HSMM. C) Odds ratios of concordant failure in accessibility gain across differentiation in 54-1 knockout myoblasts between pairs of sites that are co-accessible in HSMM. For each bin of co-accessibility in HSMM, pairs of peaks that overlapped peaks in 54-1 non-template controls were assessed for concordant failure to open (peaks that open in NTC but do not do so in knockouts). Color indicates knockout. Error bars indicate 95% confidence intervals calculated using Fisher's exact test. Stars represent estimates significantly different than 1 (p values <0.05 by Fisher's exact test). D) Similar to (C). Odds ratios of concordant failure in accessibility gain given constitutive MYOD1 binding in one of the peaks. For each bin of co-accessibility in HSMM, pairs of peaks that overlapped peaks in 54-1 were assessed for presence of constitutive binding of MYOD1 in one or both sites as well as coordinated failure to open. Only pairs of sites where both open in NTCs were included. Data are for MEIS1 knockout only due to a lack of sufficient power in PBX1. E) A model of how chromatin hub activation could be nucleated by a subset of "precociously" opening DNA elements. Such sites are occupied by TFs competent to bind relatively closed, inactive DNA elements, such as MEIS1, which may tether less competent factors such as MYOD1 to the hub. Subsequent recruitment of p300 and the BAF complex, possibly through intermediary factors (e.g., MYOD1), leads to remodeling and acetylation of histones throughout the hub. These newly available sites are then bound by other transcriptional activators (e.g., MEF2), leading to the recruitment of RNA polymerase II. Moreover, acetylation of the histones downstream of assembled preinitiation complexes reduces the barrier they pose to elongation, enhancing efficient transcription of genes within the hub.

ΔMEIS1 myoblasts when one of the sites was constitutively bound by MYOD1 than when neither was constitutively bound by MYOD1 (p value 0.0011, Fisher's exact test). These findings suggest that MEIS1 may be important for proper MYOD1-mediated recruitment of chromatin remodeling complexes to specific sites that subsequently act on others in 3-D proximity.

2.4 Discussion

Despite their paramount importance, maps that comprehensively link distal regulatory sequences to their target genes are still lacking. Toward addressing this, we developed Cicero, which constructs putative *cis*-regulatory maps from single-cell chromatin accessibility data. We anticipate these maps will guide downstream validation by other scalable methods such as massively parallel

reporter assays and CRISPR-mediated (epi)genome editing. In contrast with other approaches like ChIA-PET and promoter-capture Hi-C, Cicero operates on single-cell data and therefore avoids averaging effects that can confound bulk assays. As described here for a model of skeletal muscle differentiation, downstream analyses of Cicero-based links can advance our quantitative understanding of eukaryotic gene regulation and may also facilitate the identification of the target genes of noncoding variants underlying GWAS signals.

Pseudotemporal ordering of chromatin accessibility profiles from differentiating myoblasts revealed dynamic changes in thousands of DNA elements. Although changes in promoter accessibility were a poor predictor of gene expression dynamics, distal sites linked to genes by Cicero improved these models, particularly when sequence motifs were incorporated.

Our analyses show that the CCANs defined by Cicero meet the definition of chromatin hubs: they are physically close in the nucleus, their histone marks change in a coordinated fashion, and their interactions are likely mediated by a common set of TFs, some lineage-specific. For myogenesis, our results support a model of gene activation in which a subset of "precocious" enhancers recruit chromatin remodeling enzymes and other epigenetic modifiers to the hub, which mediates increases in accessibility of other binding sites (Figure 2.14E). For such a mechanism to work, chromatin hubs enclosing genes silent in myoblasts and activated during differentiation would need to be largely established prior to its onset. Indeed, Cicero linked more than half of activated or upregulated genes into such "pre-established" chromatin hubs. Sites that join or leave a hub are distinguished from those that remain part of it by specific TF motifs.

In differentiating myoblasts, MYOD1 is widely understood to recruit the BAF complex and p300/PCAF to activate enhancers of muscle genes (Serra et al., 2007; Simone et al., 2004). Although the role of MYOD1 in recruitment is well appreciated, how MYOD1 is itself recruited is less clear. We find that early opening sites are enriched for MEIS1 motifs and constitutive MYOD1 binding. Meis1 has previously been reported to tether Myod1 to the inactive myogenin promoter prior to the onset of differentiation and is required for myogenin activation and chromatin remodeling that permits the binding of MYOD1 to nearby MRF E-boxes that were previously inaccessible (Berkes et al., 2004; Maves et al., 2007; de la Serna et al., 2005). Whether MEIS1/PBX1 acts to

tether MYOD1 to inactive chromatin more generally throughout the genome has remained an open question.

Our analyses suggest that MEIS1 and its cofactor PBX1 are required for chromatin remodeling at a large fraction of sites that normally open during myoblast differentiation by serving as initial recruitment sites for epigenetic remodeling enzymes. Binding of p300 to MEIS1/PBX1-tethered MYOD1 could then acetylate histones at all DNA elements physically nearby in the chromatin hub. This model may help explain the pervasive gains and losses of histone acetylation throughout the accessible genome, despite the smaller number of differentially accessible or MYOD1-bound elements. Although we cannot exclude the possibility that some of the defects in chromatin remodeling are due to secondary effects downstream of MEIS1/PBX1, our genome-wide analysis taken together with biochemical and genomic data from previous studies support a direct role for MEIS1/PBX1 in recruiting factors that activate chromatin hubs.

Cicero provides an effective, genome-wide means of generating candidate links between regulatory elements and target genes in a tissue or cell type of interest using data from a single experiment. The chromatin hubs that it defines will facilitate the construction of quantitative models of epigenetic and gene expression dynamics, as well as the identification of genes whose dysregulation underlies GWAS associations. As the field pursues organism-scale cell atlases that comprehensively define each cell type and its molecular profile, such regulatory maps will be essential for understanding the epigenetic basis of each cell type's gene expression program, in both health and disease.

2.4.1 Limitations

The primary limitation of Cicero is the putative nature of the regulatory connections it identifies. Determining whether a distal DNA element is necessary or sufficient to exert regulatory influence on the genes Cicero links to it requires downstream experimentation. We note that proximity ligation-based methods for linking DNA elements to genes such as ChIA-PET or promoter capture Hi-C also have this limitation; proximity does not definitively mean regulatory interaction. More-

over, although we have found that our overall comparisons with available proximity ligation-based data are concordant, some individual connections are missing from one or both sets of putative interactions, and it is not clear which should be considered more reliable. On the one hand, ligation is a molecular measurement (albeit an indirect one) of physical proximity, while Cicero's links are based on computational inference. One the other, the ligation assays discussed here operate on bulk cell populations and are therefore subject to the artifacts introduced by averaging cells of different types or states, while Cicero operates at single-cell resolution.

2.5 METHODS

2.5.1 Experimental model and subject details

Human skeletal muscle myoblasts (HSMM)

HSMM, derived from quadriceps biopsy (Lonza, catalog #CC-2580, lot #257130: healthy, age 17, female, of European ancestry, body mass index 19; cells were used within 5 passages of purchase), were cultured in skeletal muscle growth media (GM) using the SKGM-2 BulletKit (Lonza). The cells and differentiation protocol are those from Trapnell et al. (2014). Cells were seeded in 15 cm dishes, media was replenished every 48 hours and cells were allowed to reach 80%–90% confluence. Differentiation was induced at time 0 via a switch to differentiation medium (DM) composed of alpha-mem (Thermo Fisher Scientific) and 2% horse serum. Cells in GM (time 0) or DM were then harvested at the specified times and processed as described below. HSMM tested negative for mycoplasma contamination within 3 weeks of the experiment.

GM12878

GM12878 (purchased from Coriell Cell Repository) was cultured in RPMI 1640 medium (GIBCO 11875) supplemented with 15% FBS, 100 U/ml penicillin and 100 mg/ml streptomycin. Cells were cultured in an incubator at 37°C with 5% CO_2 and were split to a density of 300,000 cells/ml three times a week.

54-1 immortalized human myoblasts

54-1 human myoblasts (Krom et al., 2012; Snider et al., 2010) were a kind gift from Dr. Robert Bradley and Dr. Silvere van der Maarel. For expansion, 54-1 cells were cultured in high serum media containing 20% FBS, 1% penn-strep, 10 ng/mL recombinant human FGF and 1 mM dexamethasone in F-10 media. For myoblast differentiation, media was replaced with low serum media containing 1% horse serum, 1% penn-strep, 10 mg/mL insulin and 10 mg/mL transferrin in F-10 media.

2.5.2 Method details

sci-ATAC-seq library construction

We prepared sci-ATAC-seq libraries using an improved version of the original protocol (Cusanovich et al., 2015), with improvements reported in the related paper (Cusanovich et al., 2018b). Briefly, HSMM cells were harvested at defined times post switch to DM, washed and cells were lysed to obtain nuclei by resuspending cells in cold lysis buffer (CLB, 10 mM Tris HCL pH 7.4, 10 mM NaCl, 3 mM $MgCl_2$ and 0.1% IGEPAL CA-130) supplemented with protease inhibitors (Sigma). For each time point, 2.75×10^5 nuclei were resuspended in a mix of 990 mL of CLB supplemented with protease inhibitors and 1.1 mL of Tagment DNA buffer (Illumina), and divided evenly among the wells of a 96 well LoBind plate (Eppendorf). 1 mL of uniquely barcoded Tn5 (Illumina) was added to each well followed by incubation at 55°C for 30 minutes. Following Tn5 incubation, 20 mL of a solution containing 40 mM EDTA and 1 mM spermidine were added to each well and incubated at 37°C for 15 minutes. Tagmented nuclei were pooled, stained by addition of DAPI to a final concentration of 3 mM and 25 DAPI positive nuclei were sorted into the wells of 96 well LoBind plates containing 12.5 ml of 0.8 mg/mL BSA and 0.04% SDS in EB buffer (QIAGEN). Nuclei were lysed by incubation at 55°C for 15 minutes. ATAC libraries were PCR amplified by addition of unique combinations of P5 and P7 primers for each well of sorted nuclei and PCR conditions were such that amplification did not reach saturation. For each sorted 96 well plate ATAC libraries were pooled and products cleaned using the Zymo Clean & Concentrator kit (Zymo). Li-

braries were quality controlled by analyzing on PAGE gels and quantified using the Qubit broad range DNA quantitation kit (Thermo Fisher Scientific).

Bulk ATAC-seq library construction

Bulk ATAC-seq experiments were performed as previously described (Buenrostro et al., 2013). Briefly, cells were trypsinized, washed with PBS and resuspended in cold-lysis buffer (CLB: 10 mM Tris-HCl pH 7.4, 10 mM NaCl, 3 mM $MgCl_2$ and 0.1% IGEPAL CA-630) supplemented with protease inhibitors (Sigma) to obtain nuclei. 100,000 nuclei were pelleted, resuspended in CLB and the equivalent of 20,000 nuclei were transferred, mixed with Tagment DNA buffer and Tn5 enzyme (Illumina). Reactions were incubated at 37°C for 30 minutes and purified using the MinElute kit (Zymo). Sequencing adapters and indices were added via PCR using standard Nextera P5 and P7 primers with excess primers removed using a 1x Ampure cleanup (Agencourt). Libraries were quality controlled by examining on a PAGE gel and quantified using the Qubit broad range DNA assay (Thermo Fisher Scientific).

54-1 knockout construction

Oligos containing sequences for single guide RNAs (sgRNAs) targeting *PBX1*, *MEIS1* and non-targeting controls were designed as follows:

50-tatcttGTGGAAAGGACGAAACACC[G]-[20bp sgRNA]-gttttagagctaGAAAtagcaagttaaaataagg-3

corresponding to the following form:

[U6 homology]-[sgRNA]-[sgRNA backbone homology]

The sgRNA sequences targeting *MEIS1*, *PBX1* and non-targeting controls are shown in Table 2.1.

Oligos were ordered from IDT and made double stranded by PCR using primers that bind the U6 and sgRNA backbone homology sequences. Oligos corresponding to each sgRNA were then ligated using the In-Fusion HD kit (Clontech) into BsmBI and alkaline phosphatase digested lentiCRISPRv2-Blast (Addgene, #83480). Ligations were then transformed into Stellar Competent

Table 2.1: sgRNA sequences targeting *MEIS1*, *PBX1* and non-targeting controls

sgRNA	**Gene targeted**	**sgRNA sequence**
MEIS1_1	*MEIS1*	TACTTGTACCCCCCGCGAGC
MEIS1_2	*MEIS1*	CACAGCTCATACCAACGCCA
MEIS1_3	*MEIS1*	GGTGGCCACACGTCACACAG
MEIS1_4	*MEIS1*	ACTCGTTCAGGAGGAACCCC
PBX1_1	*PBX1*	GATCCTGCGTTCCCGATTTC
PBX1_2	*PBX1*	TGGTCCGGCTTTGCTCTCGC
PBX1_3	*PBX1*	CCTGCGCCTCATCCAAACTC
PBX1_4	*PBX1*	CGGCCATCCCGACCCCAGCA
PBX1_5	*PBX1*	TGTGAAATCAAAGAAAAAAC
NTC_1	Non-targeting control	ACGGAGGCTAAGCGTCGCAA
NTC_2	Non-targeting control	CGCTTCCGCGGCCCGTTCAA
NTC_3	Non-targeting control	ATCGTTTCCGCTTAACGGCG
NTC_4	Non-targeting control	GTAGGCGCGCCGCTCTCTAC
NTC_5	Non-targeting control	CCATATCGGGGCGAGACATG

cells (Clontech), bacteria grown overnight in LB containing ampicillin and plasmids recovered using the QIAGEN MiniPrep kit. Lentivirus was generated by transfection into HEK293T using the ViraPower packaging mix and viral containing supernatant was filtered using a 45mm steriflip vacuum filter (Fisher Scientific). 54-1 cells were transduced with filtered virus, cultured for 48 hours and sgRNA containing cells selected by incubation with 5 mg/mL blasticidin. Cells were expanded for 21 days post-selection to allow for genome editing prior to myoblast differentiation experiments and bulk ATAC-seq.

Differentiation was induced in 54-1 lines as described for HSMM. Bulk ATAC-seq was conducted on day 0 and day 7 after induction.

54-1 staining

At various times along the differentiation protocol, 54-1 cells were washed with PBS, fixed by incubating for 20 minutes in 4% PFA (Electron Microscopy Sciences) in PBS and an additional 10 minutes in 100% methanol (Sigma). Samples were washed twice with IF buffer (0.2% Triton X-100 and 5% w/v bovine serum albumin in PBS) and incubated with anti-myosin MF20 antibody (eBioscience) overnight at 4°C with rotation. After primary antibody incubation, samples were washed twice with IF buffer and incubated with donkey anti-mouse Alexa Fluor-594 secondary antibody (Molecular Probes) and 5 mM DAPI. Finally, samples were washed twice with IF buffer, PBS added, and myosin staining assessed by imaging on a Zeiss Axio Observer (Carl Zeiss Microimaging).

2.5.3 Quantification and statistical analysis

Processing of raw data

For sci-ATAC-seq, raw reads were processed identically to those in Cusanovich et al. (2018b). Details are reproduced here for clarity. Briefly, BCL files were converted to fastq using bcl2fastq v2.16 (Illumina). Barcodes were corrected using a custom python script such that if a barcode component (tagmentation or PCR barcode separately) was within 3 edits from an expected barcode component, and the next best matching barcode component was at least 2 further edits away, the barcode component was corrected to the best match. Reads of barcode components that were not unambiguously assignable to an expected barcode were discarded.

Reads were mapped to the hg19 reference genome using bowtie2 with non-default options '-X 2000 -3 1' (Langmead and Salzberg, 2012). Reads with mapping quality less than 10 were filtered. PCR duplicates were removed using a custom python script. Cells with low read counts were removed. The read count cutoff was determined by identifying the trough between the peaks of the bimodal distribution of read counts using the mclust package in R (Scrucca et al., 2016).

For the bulk ATAC-seq datasets, processing was done as above, but without barcode correction.

Defining accessible sites

To define peaks of accessibility across all sites, we used the MACS (version 2.1.1) (Zhang et al., 2008) peak caller. Specifically, we used macs2 callpeak, with the following non-default options: –nomodel –extsize 200 –shift -100 –keep-dup all. Reads mapped to the ENCODE blacklist (ENCFF001TDO) were excluded from peak-calling. Promoter peaks were further defined as the union of the annotated transcription start site (TSS) (Gencode v17) minus 500 base pairs, and MACS defined peaks upstream of the TSS. Cells were determined to be accessible at a given peak if a read from that cell overlapped the peak. Peaks were called as above for both of the HSMM experiments separately, and then the union of the two peak sets was used.

For the GM12878 and HL60 mixed dataset, preliminary peaks were called by MACS and used to separate the cell types using multi-dimensional scaling by Jaccard distance. The subset of reads from GM12878 cells was then used to recall peaks for GM12878 as above.

After accessible peaks were defined, a matrix was generated for each dataset with the count of reads from each cell or time point (in bulk) that overlapped each accessible peak.

Pseudotemporal ordering

For the HSMM dataset, contaminating interstitial fibroblasts were removed *in silico* based on the absence of promoter accessibility in any of several known muscle markers (*MYOG*, *MYOD1*, *DMD*, *TNNT1*, *MYH1*, *MYH3*, *TPM2*). In addition, cells with fewer than 1,000 accessible sites were excluded due to low assay efficiency. Finally, peaks present in less than 1% of cells were excluded during pseudotemporal ordering steps.

Despite improvements to the sci-ATAC-seq protocol that delivered a substantial increase in the number of sites detected per cell, sci-ATAC-seq data remain zero-inflated. The quality and efficiency of transposition, which varies between cells and across batches, is likely to be a major technical source of variation in the data. Simple dimensionality reduction techniques such as MDS show that a poorly assayed cell is often more similar to other poorly assayed cells of a different type than to well-assayed cells of the same type. In order to accurately group cells with similar

chromatin accessibility profiles, we first clustered peaks that were within 1 kb and summed the reads overlapping them to create an integer-valued count matrix M.

To order the cells by progress through differentiation, we determined which aggregated peaks were relevant to the HSMM time course by fitting the following model:

$$ln(M_i) = \beta_0 + \beta_T T + \beta_S S$$

where M_i is the mean of a negative binomially distributed random variable for the number of reads overlapping the aggregate region i, T encodes the times at which each cell was harvested and S is the total number of accessible sites in each cell. We compared this full model to the reduced model:

$$ln(M_i) = \beta_0 + \beta_S S$$

by likelihood ratio test. This approach has been shown to improve power for scRNA-seq transcript counts compared to simple two-group tests comparing cells at the beginning and the end of a trajectory (Qiu et al., 2017a,b). Sites determined by this method to be time dependent and which were accessible in less than 10% of cells were then used to reconstruct the pseudotime trajectory using Monocle 2 (parameters ncenter and param.gamma set to 100, see Qiu et al. (2017b)). To remove any bias created by different assay efficiency in different cells, total sites accessible was included as a covariate in the tree reconstruction. Each cell was assigned a pseudotime value based on its position along the trajectory tree. Cells that mapped to the F_2 branch were excluded from downstream analysis.

Differential accessibility analysis

When testing for differential accessibility across cells at a particular site, it is important to exclude technical variation due to differences in assay efficiency as discussed above. We first grouped cells

at similar positions in pseudotime. We did this by k-means clustering along the pseudotime axis (k = 10). These clusters were further subdivided into groups containing at least 50 and no more than 100 cells. Next, we aggregated the binary accessibility profiles of the cells in each group into a matrix A, so that A_{ij} contains the number of cells in group j for which DNA element i is accessible. The average pseudotime ψ_j and average overall cell-wise accessibility S_j for cells in each group i were preserved for use during differential analysis.

To determine which peaks of accessibility were changing across pseudotime, we fit the following model to the binned data:

$$ln(A_i) = \beta_0 + \beta_{\tilde{\psi}}\tilde{\psi} + \beta_{\tilde{S}}\tilde{S}$$

where A_i is the mean of a negative binomial valued random variable of cells in which site i is accessible, and the tilde above ψ and S indicates that these predictors are smoothed with natural splines during fitting. This model was compared to the reduced model:

$$ln(A_i) = \beta_0 + \beta_{\tilde{S}}\tilde{S}$$

by the likelihood ratio test. Peaks with an adjusted p value of less than 0.05 were determined to be dynamic across pseudotime.

Gene set enrichment analysis

Gene set enrichment analyses was conducted using the R package piano (Väremo et al., 2013), using a hypergeometric test. We tested against the Human GO Biological Processes gene set from Merico et al. (2010).

Cicero

Cicero aims to identify all pairs of co-accessible sites. The algorithm takes as input a matrix of m by n binary accessibility values A, where A_{ij} is zero if no read was observed to overlap peak i in cell j and one otherwise. The algorithm also requires either a pseudotemporal ordering of the cells along a developmental trajectory (e.g., with Monocle 2) or the coordinates of the cells in some sufficiently low dimensional space (e.g., a t-SNE map) that the cells can be readily clustered using k-nearest neighbors (while avoiding the curse of dimensionality (Weber et al., 1998)). The algorithm then executes the following steps, which are detailed in the sections below: first, groups of highly similar cells are sampled using the clustering or pseudotemporal ordering, and their binary profiles are aggregated into integer counts. Second, these counts are optionally adjusted for user defined technical factors, such as experimental batch. Third, Cicero computes the raw covariances between each pair of sites within overlapping windows of the genome. Within each window, Cicero estimates a regularized correlation matrix using the graphical LASSO, penalizing pairs of distant sites more than proximal sites. Fourth, these overlapping covariance matrices are "reconciled" to produce a single estimate of the co-accessibility across groups of cells. These co-accessibility scores are reported to the user, who can extract modules of sites that are connected in co-accessibility networks by first specifying a minimum co-accessibility score and then using the Louvain community detection algorithm on the subgraph induced by excluding edges below this score.

Grouping cells:

In principle, Cicero could analyze the sample covariance computed between the vectors x_i and x_j of binary values encoding accessibility across cells for a pair of sites i and j. However, rather than working with the binary data directly, Cicero groups similar cells and aggregates their binary accessibility profiles into integer count vectors that are easier to work with in downstream steps. Under the grouping discussed below, the number of cells in which a particular site is accessible can be modeled with a binomial distribution or, for sufficiently large groups, the corresponding

Gaussian approximation. Modeling grouped accessibility counts as normally distributed allows Cicero to easily adjust them for arbitrary technical covariates by simply fitting a linear model and taking the residuals with respect to it as the adjusted accessibility score for each group of cells.

In order to control for technical variation as discussed above, Cicero operates on a grouped cell count matrix, *C*. *C* is constructed by first mapping cells into low dimensions by either Monocle 2 or tSNE. Within this space, Cicero constructs a k-nearest neighbor graph, via the FNN package (Beygelzimer et al., 2018), which is based on KD-trees and is highly efficient, scaling to large numbers of cells. The dimensionality reduction prior to constructing the k-nearest neighbor graph helps avoid the curse of dimensionality (Weber et al., 1998). Cicero next iterates through the following procedure to create groups (default k=50):

1. Initialize list *L* to contain all cells, and list *G* to empty.
2. Choose a random cell *r* from *L* and create a new group *g* of its k-nearest neighbors.
3. If *g* more than 90% shared cells with any existing group in *G* discard *g*, else add *g* to *G*.
4. Remove *r* from *L*.
5. Return to step 2 until *L* is empty.

Accessibility counts are next summed across all cells in each group in *G* to create count matrix *C*. Cicero's grouping procedure can be viewed as a type of bootstrap aggregation, or "bagging" (Breiman, 1996), which has been shown to substantially improve the stability of a variety of algorithms in machine learning. Note that with these parameter settings in a typical experiment, a cell will be part of more than one group and therefore the groups will sometimes contain some of the same cells, which could in principle inflate co-accessibility scores across cells. However, in practice in our analyses of both GM12878 and HSMM, the median number of cells shared between pairs of groups is zero.

Adjusting accessibility counts for technical factors:

To normalize for variations in assay efficiency across groups, matrix *C* is divided by a group-

wise scaling factor (computed using the standard Monocle 2 method for library size calculations (estimateSizeFactors()) to create an adjusted accessibility matrix R. Because the entries of C are integer counts that can reasonably be approximated by Gaussian distributions, this matrix can be readily adjusted for arbitrary technical covariates (e.g., using the Limma package's removeBatchEffect() function). In this study we did not adjust for factors beyond library size.

Computing co-accessibility scores between sites:

Cicero next analyzes the covariance structure of the adjusted accessibilities in R. Given enough data, Cicero could in principle simply compute the raw covariance matrix U. However, because the number of possible pairs of sites is far larger than the number of groups of cells, Cicero uses the Graphical Lasso to compute a regularized covariance matrix to capture the co-accessibility structure of the sites. The Graphical LASSO computes the inverse of the sample covariance matrix, which encodes the partial correlations between those variables as well as the regularized covariance matrix (Friedman et al., 2008). These constitute a statistically parsimonious description of the correlation structure in the data: informally, two variables are partially correlated when they remain correlated even after the effects of all other variables in the matrix are excluded. The Graphical LASSO expects a small fraction of the possible pairs of variables to be partially correlated, preferring to select a sparse inverse covariance matrix over a dense one that fits the data equally well. Those pairs of sites that lack sufficient partial correlation to be worth the penalty term are assigned zero partial correlation in the inverse covariance matrix reported by Graphical LASSO. Formally, Cicero uses Graphical LASSO to maximize:

$$\log\det\Theta - \mathrm{tr}(U\Theta) - \|\Theta * \rho\|_1$$

where Θ is the inverse covariance matrix capturing the conditional dependence structure of p accessible sites, and U is the sample covariance matrix computed from their values in R. In order to ensure stability of Graphical LASSO, which can hang on poorly conditioned input, we add a

small conditioning constant of 1×10^{-4} to the diagonal of U prior to running it. The matrix ρ contains penalties that are used to independently penalize the covariances between pairs of sites, and $*$ denotes component-wise multiplication.

In Cicero, we aim to find local *cis*-regulatory interactions, rather than the global covariance structure that might be expected due to overall cell state. To achieve this, we set each penalty term in ρ such that peaks closer in genomic distance had a lower penalty term. Specifically, we used the following equation to determine ρ:

$$\rho_{ij} = \left(1 - d_{ij}^{-s}\right) \alpha$$

where d_{ij} is the distance in the genome (in kilobases) between sites i and j and s is a constant that captures the power-law distribution of contact frequencies between different locations in the genome as a function of their linear distance. A complete discussion of the various polymer models of DNA packed into the nucleus is beyond the scope of this paper, but we refer readers to Dekker et al. (2013) for a discussion of justifiable values for s. We use a value of 0.75 by default in Cicero, which corresponds to the "tension globule" polymer model of DNA (Sanborn et al., 2015). The scaling parameter α controls the distance at which Cicero expects no meaningful *cis*-regulatory contacts, and its value is calculated automatically from the data. To calculate α, Cicero selects 100 random 500 kb genomic windows, and determines the minimum α value such that no more than 5% of pairs of sites at a distance greater than 250 kb (a user-adjustable value) had non-zero entries in Θ and less than 80% of all entries in Θ were nonzero. The mean of these values of α is then used to set the penalties for the whole genome. Cicero then applies Graphical LASSO to overlapping 500 kb windows of the genome (windows are spaced by 250 kb such that each region is covered by two windows).

Reconciling overlapping local co-accessibility maps:

Cicero calculates correlation values (co-accessibility scores) from the resulting estimated sparse

covariance matrix for each pair of peaks within 500 kb of each other. Because the genomic windows are overlapping, the majority of pairs of peaks have two calculations of co-accessibility. To consolidate these sites and create a genome-wide map of the accessible regulome, Cicero considers the co-accessibility scores for each pair of peaks to determine if they are in qualitative agreement (both calculated scores in the same direction). The qualitative agreement in our two test datasets were both $>95\%$. Pairs of peaks not in qualitative agreement are considered undetermined. For peaks in qualitative agreement, the mean score of the two values is assigned.

Extracting *cis*-co-accessibility networks (CCANs):

Positive Cicero co-accessibility scores indicate that a pair of peaks is connected, with the magnitude of the co-accessibility corresponding to Cicero's confidence in the link. To identify hubs of co-accessibility, Cicero can create a graph where each node is a peak of accessibility, and edges are the co-accessibility scores above a user-defined threshold. Communities within this genome-wide graph can be found using the Louvain community finding algorithm. Cicero can then assign peaks to *cis*-co-accessibility networks (CCANs) based on these communities.

Calculating gene activity scores:

Cicero calculates an overall measure of the accessibility of sites linked to each gene k by first selecting rows of the binary accessibility matrix A that correspond to sites proximal to the gene's transcription start sites or to distal sites linked to them. These rows are weighted by their co-accessibility and then summed to produce a vector of accessibility scores R_k, where the overall accessibility of gene k in cell i is:

$$R_{ki} = \sum_{p \in P} \sum_{j \in P_p} A_{ji} \frac{u_{pj}}{\sum_{k \in D_p} u_{pk}} + A_{pi}$$

where P indexes the promoter proximal sites of k, D_p indexes distal sites linked to proximal site p, u is the Cicero co-accessibility score linking distal site j to proximal site p, and A is the binary

score for accessibility at site j or p in cell i. In principle, D_p could include all distal sites linked to p, but here we restrict the set to distal sites that are differentially accessible (FDR $<1\%$) across pseudotime.

Because the magnitude of these aggregate accessibility values will depend on overall sci-ATAC-seq library depth in each cell, we capture this relationship via a linear regression:

$$\log\left(\sum_k R_k\right) = \beta_0 + \beta_A \log\left(\sum_j A_{ji}\right)$$

The aggregate accessibility for each gene k in cell i is then scaled using the output of this model r_i for cell i:

$$\tilde{R}_{ki} = R_{ki} \cdot \frac{\sum_i r_i}{r_i}$$

Gene expression values measured by RNA-seq are typically approximately log-normally distributed. We therefore transform aggregate accessibility values to gene "activity" scores G_{ki} for each gene k in each cell i by simply exponentiating them. We also scale them by the total (exponentiated) gene accessibility values to produce "relative" activities:

$$C_{ki} = \frac{e^{\tilde{R}_{ki}}}{\sum_k e^{\tilde{R}_{ki}}}$$

Comparing two Cicero maps:

Different Cicero CCAN maps were matched using the push-relabel algorithm for maximum matching in a weighted bipartite graph (Goldberg and Tarjan, 1986). Specifically, we used the maxmatching package in R to calculate the matching. Maximum matching in weighted bipartite graphs is a one-to-one matching such that the edge weights are maximized. In the case of comparing Cicero CCAN maps, the maximum matching is the one-to-one match of CCANs from map 1 to CCANs from map 2 such that the largest number of peaks is shared across the maps overall.

Analysis of 54-1 immortalized myoblasts

Bulk ATAC-seq libraries from 54-1 cells were processed as above. Data from the multiple guides at each time point and targeting each gene were merged for peak calling as described above. The resulting peaks were merged to create a master peak list. Reads per peak were then counted for each guide and time point separately. DESeq2 (Love et al., 2014) was used to test for differential accessibility between day 0 and day 7 across each of the three conditions (non-template control, *MEIS1* targeted and *PBX1* targeted). Two libraries (NTC guide 5 day 0 and *PBX1* guide 4 day 7) were removed as major outliers by PCA. Peaks with a greater than 2-fold moderated fold change were considered to be dynamic. When comparing 54-1 peaks to HSMM peaks, overlap was determined by overlapping coordinates with a maximum gap of zero.

Motif enrichment analysis

Transcription factor motifs from the JASPAR 2016 database (Mathelier et al., 2016) were located in the sci-ATAC-seq peaks using FIMO (Grant et al., 2011). Motifs for TFs not expressed at ≥ 2 transcripts per million in bulk RNA-seq (HSMM myoblasts or myotubes) were excluded from downstream analysis. Many TF motifs are similar or identical to each other. To prevent this correlation from confounding regression analyses, we clustered motifs into motif families. For each pair of motifs, A and B, we computed the conditional probability that given motif A is called at a genomic location with a FIMO p value $<2\times10^{-5}$ (a stringent threshold), an overlapping instance of motif B will be called at p $<1\times10^{-4}$ (a permissive threshold). We constructed an undirected graph of motifs where there is an edge between motifs A and B if $\mathrm{P}\left(\mathrm{B\ at\ p}<1\times10^{-4}\mid\mathrm{A\ at\ p}<2\times10^{-5}\right)\geq 0.5$ or $\mathrm{P}\left(\mathrm{A\ at\ p}<1\times10^{-4}\mid\mathrm{B\ at\ p}<2\times10^{-5}\right)\geq 0.5$. Edges in this graph are assigned weights equal to the greater of these two conditional probabilities minus 0.5. We clustered the motifs on this graph using Louvain clustering (Blondel et al., 2008) and manually assigned names to each cluster. For downstream regression analyses, a genomic location is considered to have an instance of a motif family if any motif in the family is called at that location at p $<5\times10^{-5}$ (an intermediate threshold).

To generate the motif co-accessibility networks shown in Figure 2.10A, we computed two sets of binary variables for each protein coding gene that had at least one sci-ATAC-seq peak in its promoter(s). The first set of variables are indicators of whether or not at least one instance of a motif family is present in any promoter peak for the gene. The second set of variables are indicators of whether or not at least one motif instance is present in any distal peak (excluding promoters of other genes) connected to the gene's promoter(s) with a co-accessibility score greater than 0. We constructed a matrix where rows are genes and columns are these two sets of motif indicator variables. This matrix was provided as input to the Graphical LASSO subject to the constraint that partial correlations between two promoter motif variables or two distal motif variables are fixed to zero. The regularization parameter ρ for the Graphical LASSO was set as the smallest value that could achieve an estimated false discovery rate (FDR, the proportion of truly zero partial correlations that are estimated as non-zero) of less than 5%. The FDR for a given value of ρ was estimated by running the Graphical LASSO with that value of ρ on versions of the motif indicator matrix with the distal variables row-shuffled (essentially assigning each gene to a random other gene's set of distal motifs) and counting the proportion of motif pairs that are assigned a nonzero partial correlation (ideally, all should be zero in a shuffled matrix).

In Figure 2.10A, an edge is drawn between a pair of motif families A and B if both 1) the co-accessibility of the indicator variable for A being at a distal site to the indicator variable of B being at a linked promoter site is $>$0.02, and 2) the same is true if B is in the distal position and A is in the promoter.

Analysis of ChIA-PET and Hi-C data

To compare our Cicero connections to promoter-capture (PC) Hi-C, we used publicly accessible GM12878 data (Cairns et al., 2016). We used the provided ChICAGO score as our indicator of physical proximity.

To compare our data to PC Hi-C, we first overlapped our peaks with peaks from PC Hi-C. Peaks were considered to overlap if they were within 1 kb of each other. For this analysis, we only

considered pairs of sci-ATAC-seq peaks where at least one was a promoter represented in the PC Hi-C data. In addition, we only considered pairs of peaks that were within the same A/B compartment to avoid potential confounding of cross-compartment connections (Fortin and Hansen, 2015). To check that the effect of a pair of peaks' co-accessibility on presence in the dataset in Figure 2.8C was beyond the effect of the overall accessibility of the peaks, we ran a logistic regression predicting presence of the pair in PC Hi-C using binned co-accessibility and the geometric mean of the accessibility of the two peaks in the pair and found the coefficient for co-accessibility to be significant (p value $< 2 \times 10^{-16}$).

As a second comparison dataset, we used publicly accessible GM12878 polymerase II ChIA-PET data (Tang et al., 2015) (GEO: GSE72816). To compare these data to Cicero's connections, we first looked for overlap between our peaks and ChIA-PET anchors. Because ChIA-PET anchors often overlap each other, we first merged overlapping anchors to create comparable ChIA-PET "peaks." We considered accessible peaks within 1 kb of ChIA-PET peaks to be overlapping. To generate Figure 2.8B and Figure 2.8D, we considered the subset of ChIA-PET and Cicero connections where the peaks were present in both datasets. Similarly as for PC Hi-C, to check that the effect of a pair of peaks' co-accessibility on presence in the dataset in Figure 2.8B was beyond the effect of the overall accessibility of the peaks, we ran a logistic regression predicting presence of the pair in ChIA-PET using binned co-accessibility and the geometric mean of the accessibility of the two peaks in the pair and found the coefficient for co-accessibility to be significant (p value $< 2 \times 10^{-16}$).

Analysis of ChIP-seq data (MYOD1 and histone)

To compare our accessible peaks to the known myogenesis master regulator MYOD1, we used publicly accessible MYOD1 ChIP-seq in human myoblast and human myotube (MacQuarrie et al., 2013) (GEO: GSE50413). We considered our peaks to be bound by MYOD1 if they overlapped one of the annotated MacQuarrie et al. (2013) ChIP-seq peaks.

To compare our accessible peaks to histone modifications, we used publicly accessible EN-

CODE datasets in HSMM and HSMMtube (The ENCODE Project Consortium, 2012) (ENCFF000BKV, ENCFF000BKW, ENCFF000BMB, ENCFF000BMD, ENCFF000BOI, ENCFF000BOJ, ENCFF000BPL, ENCFF000BPM). We counted both HSMM and HSMMtube histone ChIP-seq reads in each accessible peak. To determine whether sites were changing in accessibility between HSMM and HSMMtube, we used DESeq2 differential analysis (Love et al., 2014) (FDR $<5\%$). To determine whether the barrier regions of genes were differentially histone modified, we similarly used DESeq2 to compare the read counts in the first 1000 base pairs of each GENCODE v17 transcript in HSMM and HSMMtube datasets.

To compare agreement between H3K27 acetylation marks of peaks connected by Cicero, we divided the odds of a site gaining acetylation if its connected site gained acetylation by the odds of a site gaining acetylation is it is connected to a site that is not gaining acetylation (Figure 2.11A).

Modeling H3K27 acetylation changes

To model changes in acetylation among linked sites (Figure 2.11D), we compared four linear regression models:

$$\ln(a_i) = \beta_0 + \beta_1 A_{icl} + \beta_2 A_{iop}$$

$$\ln(a_i) = \beta_0 + \beta_1 A_{icl} + \beta_2 A_{iop} + \beta_3 m_{ig} + \beta_4 m_{il} + \beta_5 m_{ic}$$

$$\ln(a_i) = \beta_0 + \beta_1 A_{icl} + \beta_2 A_{iop} + \beta_3 \Theta_{op} + \beta_4 \Theta_{cl}$$

$$\ln(a_i) = \beta_0 + \beta_1 A_{icl} + \beta_2 A_{iop} + \beta_3 m_{ig} + \beta_4 m_{il} + \beta_5 m_{ic} + \beta_6 \Theta_{mg} + \beta_7 \Theta_{ml} + \beta_8 \Theta_{mc}$$

where a_i is the $\log_2$ fold-change in H3K27 acetylation from myoblast to myotube at site i, A_{icl} and A_{iop} are indicator variables for whether site i is closing or opening across pseudotime, m_{ig}, m_{il}, and m_{ic} are indicator variables for whether site i is gaining, losing, or constitutively bound by MYOD1 from myoblast to myotube according to ChIP-seq, Θ_{op} and Θ_{cl} are the highest Cicero co-accessibility scores that connect site i to another opening or closing site respectively, and Θ_{mg}, Θ_{ml} and Θ_{mc} are the highest Cicero co-accessibility scores that connect site i to another MYOD1

gaining, MYOD1 losing or MYOD1 constitutive site. For each of the fitted models, we used elastic net regression (Zou and Hastie, 2005) to estimate the effect of each predictor.

Similarly, in Figure 2.10B, we predict the $\log_2$ fold-change in each of the 12 ENCODE histone mark ChIP-seq datasets described above using only indicator variables for whether a site is gaining losing or constitutively bound by MYOD1, or using these variables and the highest Cicero co-accessibility scores connecting a site to an opening or closing site.

Regression models for barrier region histone marks and gene expression

For each of the 12 ENCODE histone mark ChIP-seq datasets described previously, we fit two regression models that predict, for each transcription start site, the log fold change in the number of reads from the given ChIP-seq dataset that fall in the barrier region of that TSS (first 1000 bp downstream) for myotubes versus myoblasts. We exclude TSSs that do not have a significantly different number of barrier region reads in myotubes versus myoblasts for any of the 12 datasets ($p > 0.01$), leaving 5,563 TSS included in the model.

In the first set of models ("promoter motifs"), the features are a set of binary indicator variables that have value 1 if any promoter sci-ATAC-seq peak for the TSS has at least one instance of a motif from a given motif family. In the second set of models ("promoter and distal motifs"), the features are the promoter motif indicator variables plus a second set of real-valued variables that encode the presence of distal sequence motifs. For a given motif family and TSS, the corresponding distal motif variable has a value equal to the highest co-accessibility score from any promoter sci-ATAC-seq peak for that TSS to any connected distal peak that has at least one instance of a motif from the motif family. If no such distal peak exists (the motif is absent in all connected distal sites), the distal motif variable is assigned a value of 0. The models were trained using elastic net regression.

We additionally fit models with the same features ("promoter motifs" and "promoter and distal motifs") to predict the expression of the subset of the above TSSs (n = 937) that were additionally expressed in at least 4 cells in scRNA-seq and which were predicted by smoothed average across pseudotime to be expressed at above 1 copy per cell at some pseudotime.

2.6 Data availability

The accession number for the sci-ATAC-seq data reported in this chapter is GEO: GSE109828.

2.7 Code availability

Cicero is available as an R package at http://cole-trapnell-lab.github.io/cicero-release. The manual pages for the Cicero software are in Appendix A.

2.8 Project acknowledgments

We gratefully acknowledge S. Tapscott, W. Noble, and D. Witten, as well as members of the Shendure and Trapnell labs, for advice. This work was supported by NIH (U54DK107979 to J.S. and C.T., DP2HD088158 to C.T., DP1HG007811 and R01HG006283 to J.S., T32HL007828 to J.L.M.-F., and R35GM124704 to A.C.A.) and The Paul G. Allen Frontiers Group (to J.S. and C.T.). J.S. is an Investigator of the Howard Hughes Medical Institute. C.T. is partly supported by an Alfred P. Sloan Foundation Research Fellowship. D.A.C. was supported in part by the National Heart, Lung, and Blood Institute (T32HL007828). J.L.M.-F. and A.M. are supported by NIH (5T32HG000035). H.A.P. is supported by the NSF Graduate Research Fellowship (DGE-1256082).

Chapter 3: CHROMATIN ACCESSIBILITY DYNAMICS DURING MOUSE HEMATOPOIESIS

Section 3.3, Figure 3.1, and relevant methods sections are adapted with minimal modification from:

Cusanovich, D.A., Hill, A.J., Aghamirzaie, D., Daza, R.M., Pliner, H.A. Berletch, J.B., Filippova, G. N., Huang, X., Christiansen, L., DeWitt, W.S., Lee, C., Regalado, S.G., Read, D.F. Steemers, F.J., Disteche, C.M., Trapnell, C., and Shendure, J. (2018a) A Single-Cell Atlas of *In Vivo* Mammalian Chromatin Accessibility. Cell. *174*, 1309–1324.

3.1 ABSTRACT

As described in Chapter 2, the Cicero R package aims to overcome some of the limitations of sparse single-cell data to allow the application of existing analysis techniques. In addition, Cicero estimates co-accessibility scores between nearby sites in the genome to assign regulatory elements to genes. In this excerpt, Cicero is applied to sci-ATAC-seq data of bone marrow from the mouse sci-ATAC-seq atlas. Cicero is able to generate a pseudotime trajectory matching the hierarchy of hematopoiesis, and to recapitulate the known regulatory dynamics of the β globin locus during erythropoiesis.

3.2 INTRODUCTION

Cusanovich et al. (2018b) describes the collection and analysis of single-cell chromatin accessibility data from over 100,000 cells from 13 tissues of an 8-week-old male C57BL/6J mouse and four additional replicate tissues from a second mouse. This work represents the first single-cell atlas of *in vivo* mammalian chromatin accessibility.

Nuclei were isolated and processed using an optimized single-cell combinatorial indexing ATAC-seq protocol (Cusanovich et al., 2018b). 30 major cell type clusters were identified using t-distributed stochastic neighbor embedding (tSNE) and Louvain clustering and an additional 85 sub clusters were identified using iterative clustering.

Next, Cicero was used to predict *cis*-regulatory interactions and assign distal sites to genes. In addition, Cicero connections were aggregated to compute Cicero gene activity scores for each gene. These gene activity scores correlated better with gene expression than promoter accessibility alone. In the following excerpt, we analyze the bone marrow tissue sample of the mouse sci-ATAC-seq atlas described in Cusanovich et al. (2018b).

3.3 RESULTS

3.3.1 Chromatin accessibility dynamics during hematopoiesis

We examined chromatin accessibility in the bone marrow, the site of adult hematopoiesis. Although tSNE resolved several subpopulations (Figure 3.1A), a few clusters were large and did not cleanly separate cells by accessibility at genes expressed in a mutually exclusive manner in differentiated cells. We reasoned that, similar to RNA-seq, differentiating blood cells might be organized along a continuous "trajectory" of chromatin accessibility states. We therefore applied Monocle 2, which can pseudotemporally order cells based on chromatin accessibility (Pliner et al., 2018) to the marrow, resulting in a tree-like trajectory with a prominent "root" and five major branches (F1–F5) (Figure 3.1B).

To explore which parts of this tree correspond to various stages of blood development, we projected accessibility at previously defined sets of hematopoietic enhancers (Lara-Astiaso et al., 2014) onto the tree (Figure 3.1B). Enhancers specific to erythroid or lymphoid cells were more accessible on branches F4 and F2, respectively. Myeloid-specific enhancers were more accessible on F5 and the two small branches (F1 and F3) and modestly accessible on the root. We also examined gene activity scores for lineage markers along each branch. Gene activity scores for lineage-specific markers (*Cd3e*, *Cd19*, *Hbb-b1*, and *Cd11b/Itgam*) were at or near zero on the root, but each rose sharply on one of the five branches (Figure 3.1C). In contrast, *Cd34*, a marker of multipotent hematopoietic progenitors, was highly active on the root but decreased to near zero at the termini of all branches except F1. These observations are broadly consistent with specification of hematopoietic progenitors into B cells (F2), T cells (F3), erythrocytes (F4), and monocytes (F5).

We note, however, that although we would expect approximately 37% of cells from marrow to be neutrophils (Yang et al., 2013), we were unable to identify any cluster or branch of cells with a consistent activity score for neutrophil markers (e.g., *Elane*). Neutrophil nuclei are more fragile than other cell types (Olins et al., 2008) and may not have survived fluorescence-activated cell sorting (FACS), or possibly, they are present in our dataset, but we are simply failing to identify them.

We next visualized Cicero connections for cells in different regions along the trajectory of erythropoiesis (F4). We identified sci-ATAC-seq peaks corresponding to the six hypersensitive sites (HSs) in the $\beta-$globin locus control region (LCR; HS1-6) along with several others known to play a role in establishing the 3D chromatin conformation critical for developmental control of $\beta-$globin expression (Dostie et al., 2006). In the erythroblasts of adult mice, the LCR is positioned close to $\beta-$globin subunits *Hbb-b1* (bMaj) and *Hbb-b2* (bMin), while during development, these genes are looped away from the LCR, which contacts subunits βh1 or εy instead (Noordermeer and de Laat, 2008; Tolhuis et al., 2002). Consistent with this, in cells at the root of the tree, Cicero reported modest co-accessibility between elements of the LCR and the more distal flanking noncoding elements, as well as limited linkages between noncoding elements and the adult globin genes (Figure 3.1D). Cicero did not link the fetal and embryonic globins to the LCR, as expected. At intermediate stages through to the terminus of the erythroid branch, the Cicero maps have increasingly strong linkage of the adult globin genes and the LCR, the downstream 30'HS1, and both the -62/60 and -85 upstream HS (Figure 3.1D). In contrast, we observe only light links between the LCR and the other distal sites or the globin genes on the lymphoid and myeloid lineages, confirming that the robust association of the globin LCR with its targets is specific to the erythroid lineage.

3.4 Discussion

The $\beta-$globin locus is perhaps the best understood *cis*-regulatory region in complex organisms. Over 30 years of research has developed a complex theory around the regulation of embryonic, fetal

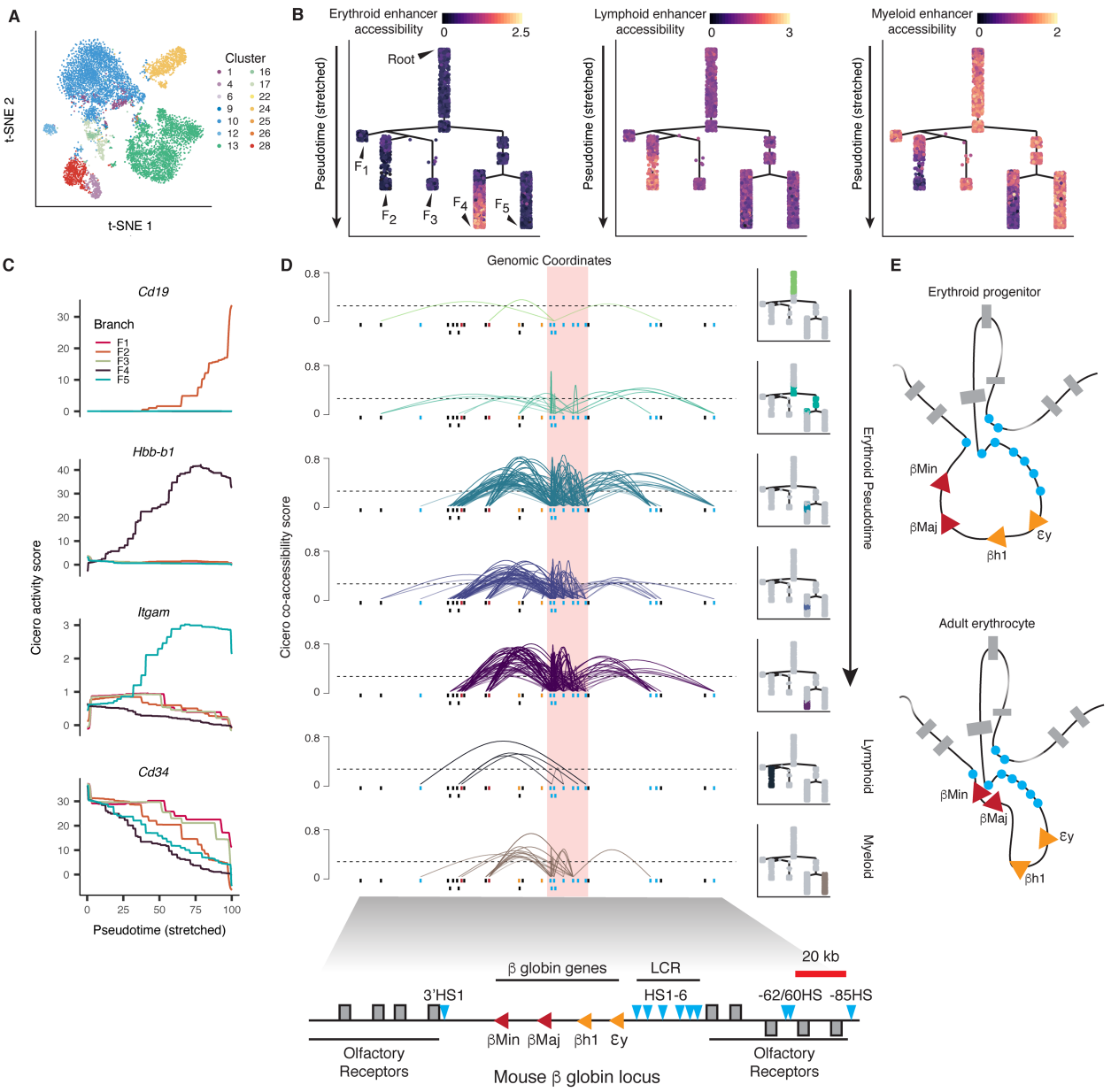

Figure 3.1: Chromatin accessibility dynamics during hematopoiesis. A) tSNE of bone marrow cells colored by major cluster from Figure 2B of Cusanovich et al. (2018a). B) Branched hematopoietic trajectory colored by accessibility of lineage-restricted enhancers (Lara-Astiaso et al., 2014). Color values represent normalized mean accessibility of peaks overlapping known enhancers (top: erythroid and erythroid progenitor, middle: lymphoid and lymphoid progenitor, bottom: myeloid and myeloid progenitor). C) Cicero gene activity scores of selected marker genes (*Cd19*, *Hbb-b1*, *Itgam*, and *Cd34* for B cells, erythroid, myeloid, and hematopoietic stem cells, respectively) across pseudotime in each branch. Each line includes cells from the root to the named branch (from B). (*Legend continued on the following page*)

Figure 3.1: (*continued*) Activity scores are plotted as a moving average over pseudotime (percent of total distance from the root). D) Cicero co-accessibility at the $\beta-$globin LCR along erythroid differentiation (roughly equal-size groups). Cells used to generate each plot are highlighted (right). Lymphoid and myeloid plots are included for comparison. Boxes below each track indicate sci-ATAC-seq peaks (colored by overlap with elements in the $\beta-$globin locus diagrams below and in (E). Arcs connecting peaks represent co-accessibility (height indicates strength of co-accessibility). Only connections originating in the LCR with co-accessibility above 0.25 (dashed line) are shown (LCR is the red highlighted region). E) Model of the $\beta-$globin locus adapted from Noordermeer and de Laat (2008).

and adult globin genes (reviewed in Levings and Bungert (2002)). Because of this, the $\beta-$globin locus represents an ideal 'gold standard' against which to test a new technology like Cicero. In this short section, we are able to demonstrate that Cicero not only connects the locus control region with the adult globin genes, but also skips over the embryonic and fetal genes, which are not expressed in the adult mouse. These results, and the accompanying demonstration of chromatin accessibility pseudotime and Cicero gene activity scores, suggest that Cicero can reliably identify *cis*-regulatory dynamics in new systems as well.

3.5 METHODS

All methods related to this project are published in Cusanovich et al. (2018b). Those methods that are directly relevant to this chapter are reproduced below.

3.5.1 Linking distal sites to putative target genes

We defined a site as distal if it was located >5 kb upstream and >1 kb downstream of any transcription start site (TSS) reported in GENCODE mm9, release M1. We only considered sites that were accessible in at least 1% of the cells in each cluster as open in that cluster. We then ran Cicero on open sites for each cluster separately to identify co-accessible sites using the following parameters: aggregation k = 30, window size = 500 kb, distance constraint = 250 kb. The aggregation value k is the number of cells that are aggregated using k-nearest neighbors prior to calculating co-accessibility scores, the window size parameter controls the size of each model window in the

genome, and the distance constraint parameter is the distance at which the distance penalty is trained to regularize the majority of connections. Using Cicero, we were able to find sites that were co-accessible in aggregated groups of cells within each cluster. Cicero assigns a regularized co-accessibility score to each pair of "open" sites in each cluster using a Graphical Lasso model, penalized by genomic distance (Pliner et al., 2018).

We first used Cicero maps to find a global *cis*-regulatory view of genome in different clusters with a co-accessibility cutoff of 0.2. Using this threshold and the windows around any TSS defined above, we were able to define pairs of sites that were co-accessible into proximal-to-proximal, distal-to-distal, and distal-to-proximal linked sites. Second, we used these co-accessibility maps to perform enrichment tests to help inform our biological interpretations of each cluster. For this purpose, we focused on distal differentially accessible (DA) sites and proximal DA sites in each cluster. In order to assign DA distal sites to target genes, we devised the following linking policy: i) distal DA sites were associated to any proximal site if they had a co-accessibility cutoff $>$0.2; ii) if a distal DA site was not linked to any proximal site with a co-accessibility cutoff of 0.2, it was assigned to the proximal site with highest co-accessibility, provided that this co-accessibility score was greater than a relaxed cutoff of 0.1. The union of genes linked to distal DA sites under the relaxed policy and proximal DA sites were used for annotation enrichment tests.

3.5.2 Computing gene activity scores

For each gene in each cell, we compute "activity scores" which summarize the degree to which chromatin surrounding the gene is accessible. We do so using Cicero, which includes a procedure to summarize overall chromatin accessibility of all sites it links to a given gene (Pliner et al., 2018). Details are reproduced here for clarity. Briefly, the method works as follows: first, Cicero calculates an overall measure of the accessibility of sites linked to each gene k by first selecting rows of the binary accessibility matrix A that correspond to sites proximal to the gene's transcription start sites or to distal sites linked to them. These rows are weighted by their co-accessibility and then summed to produce a vector of accessibility scores R_k, where the overall accessibility of gene k in cell i is:

$$R_{ki} = \sum_{p \in P} \sum_{j \in P_p} A_{ji} \frac{u_{pj}}{\sum_{k \in D_p} u_{pk}} + A_{pi}$$

where P indexes the promoter proximal sites of k, D_p indexes distal sites linked to proximal site p, and u is the Cicero co-accessibility score linking distal site j to proximal site p, and A is the binary score for accessibility at site j or p in cell i.

Because the magnitude of these aggregate accessibility values will depend on overall sci-ATAC-seq library depth in each cell, we capture this relationship via a linear regression:

$$\log\left(\sum_k R_k\right) = \beta_0 + \beta_A \log\left(\sum_j A_{ji}\right)$$

The aggregate accessibility for each gene k in cell i is then scaled using the output of this model r_i for cell i:

$$\tilde{R}_{ki} = R_{ki} \cdot \frac{\sum_i r_i}{r_i}$$

Gene expression values measured by RNA-seq are typically approximately log-normally distributed. We therefore transform aggregate accessibility values to gene "activity" scores G_{ki} for each gene k in each cell i by simply exponentiating them. We also scale them by the total (exponentiated) gene accessibility values to produce "relative" activities:

$$C_{ki} = \frac{e^{\tilde{R}_{ki}}}{\sum_k e^{\tilde{R}_{ki}}}$$

The distribution of activity values of all genes in a given cell typically resembles a bimodal distribution, the lower peak of which corresponds to genes that are very lowly or not expressed in the population of cells. In contrast, genes in the second peak are typically expressed at appreciable levels in the population of cells. To ensure that non-expressed genes receive an activity score of

zero, we first compute the mean activities for each gene across all cells in each cluster, and then fit a mixture of two Gaussians to the mean values. The larger of the location parameters is used as a threshold; all activity values in all cells in the cluster are divided by the threshold and rounded to the nearest integer. Finally, we normalize these activity values by computing "size factors" using Monocle 2 with previously described methods for normalizing scRNA-seq data (Qiu et al., 2017b).

3.5.3 Trajectory analysis of hematopoiesis

To further investigate the chromatin accessibility landscape of hematopoiesis, we took the subset of cell clusters from Cusanovich et al. (2018a) where a high percentage of cells were derived from the bone marrow (10, 13, 17.4, 24 and 28) for further analysis. We set out first to order cells in a branched trajectory, as we expected blood progenitors at various stages to be present in the marrow. To order cells, we used a modification of the Monocle 2 pseudotime ordering algorithm, as was used in Pliner et al. (2018). Briefly, peaks of accessibility within 10 kb of each other were aggregated from the binary matrix to create a count matrix. Cells were clustered using density peak clustering after tSNE dimensionality reduction, and then a differential accessibility test was performed for each site, including the cluster labels as indicator variables (full model of cluster and reduced model of 1), to find differentially accessible sites across clusters. The top 3,000 differentially accessible sites by adjusted p value were chosen as ordering genes for trajectory inference. We then used Monocle 2's DDRTree dimensionality reduction algorithm to align cells to a branched trajectory as shown in Figure 3.1.

To validate the trajectory inferred using Monocle 2, we compared our tree with H3K4me1 ChIP-seq data on sorted hematopoietic cell populations from Lara-Astiaso et al. (2014). To do this, we summed a TF-IDF normalized matrix of read counts from sci-ATAC-seq peaks that overlapped H3K4me1 ChIP-seq from each sorted cell population. The resulting 'Enhancer accessibility' from each cell population was plotted in Figure 3.1B. The myeloid, erythroid, and lymphoid accessibilities resulted from the sum of scores from these populations and their progenitors.

We next wanted to examine the well-studied β-globin locus at various points along the ery-

throid lineage. Cells were divided into 5 time points with equal numbers of cells based on the assigned pseudotime and co-accessibility was calculated on each set of cells separately using Cicero (Pliner et al., 2018).

3.6 Data availability

The accession number for the sequencing data and some processed data files reported in this chapter is GEO: GSE111586.

3.7 Project acknowledgments

We thank D. Prunkard and L. Gitari for exceptional assistance with flow sorting; the Neale lab for access to UK Biobank GWAS; R. Walters for advice on partitioned LDSC; W. Noble for GPU cluster access; and W. Noble, A. Adey, G. Findlay, M. Gasperini, M. Spielmann, V. Ramani, R. Chawla, and X. Qiu for valuable discussions and feedback. Funding is from the Paul G. Allen Frontiers Group (J.S. and C.T.), the W.M. Keck Foundation (C.T. and J.S.), an Alfred P. Sloan Foundation Research Fellowship (C.T.), the NIH (DP1HG007811 and R01HG006283 to J.S., DP2HD088158 to C.T., and R01GM046883 to C.M.D.). D.A.C. was supported in part by the NHLBI (T32HL007828), A.J.H. and H.A.P. were supported by NSF Graduate Research Fellowships, and W.S.D. was supported in part by the NHGRI (5T32HG000035-23). J.S. is a Howard Hughes Medical Institute Investigator.

Chapter 4: SUPERVISED CLASSIFICATION ENABLES RAPID ANNOTATION OF CELL ATLASES

Chapter 4 is adapted with minimal modification from::

Pliner, H.A., Shendure, J. and Trapnell, C. (2019) Supervised classification enables rapid annotation of cell atlases. bioRxiv.

4.1 ABSTRACT

Single-cell technologies for profiling tissues or even entire organisms are rapidly being adopted. However, the manual process by which cell types are typically annotated in the resulting data is labor-intensive and increasingly rate-limiting for the field. Here we describe Garnett, an algorithm and accompanying software for rapidly annotating cell types in scRNA-seq and scATAC-seq datasets, based on an interpretable, hierarchical markup language of cell type-specific genes. Garnett successfully classifies cell types in tissue and whole organism datasets, as well as across species.

4.2 MAIN

Single-cell transcriptional profiling (scRNA-seq) has emerged as a powerful means of cataloging the myriad cell types present in complex animal tissues (methods reviewed in Svensson et al. (2018)). The computational steps of constructing a cell atlas typically involve unsupervised clustering of cells based on their gene expression profiles, followed by the annotation of known cell types amongst the resulting clusters (Han et al., 2018; Tabula Muris Consortium et al., 2018). With respect to the latter task, there are at least four challenges that are proving rate-limiting for the field. First, cell type annotation is labor-intensive, requiring extensive literature review of cluster-specific genes (Zhang et al., 2019). Second, any revision to the analysis (e.g. additional data, adjustment of parameters) necessitates manual reevaluation of all previous annotations. Third, cell type an-

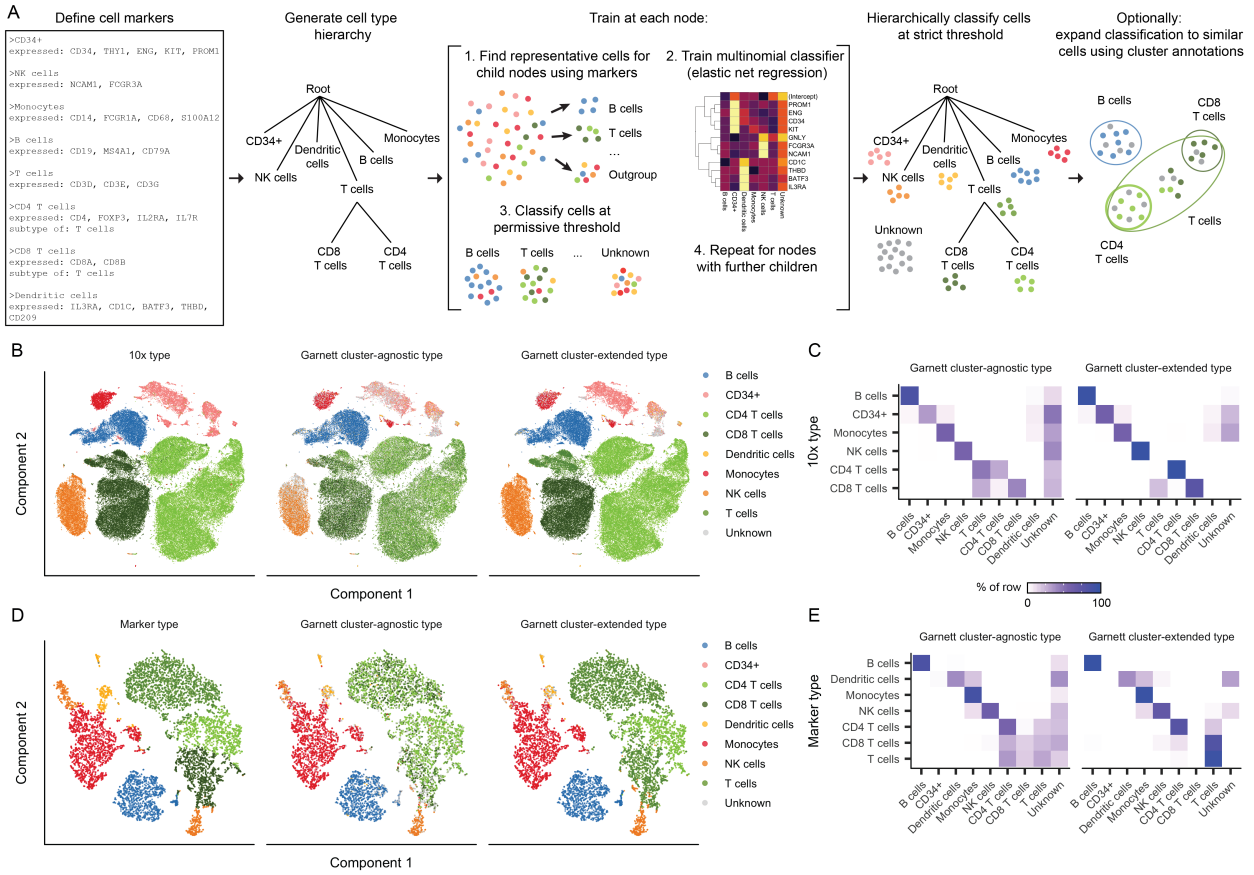

Figure 4.1: Garnett accurately classifies peripheral blood mononuclear cells (PBMCs). A) Overview of Garnett algorithm. See Section 2.5, Methods, for algorithmic details. Briefly, Garnett takes as input a marker file that defines cell types using marker genes, and builds a cell type hierarchy that can include cell subtypes. Next, Garnett trains a classifier using elastic net multinomial regression (Zou and Hastie, 2005) at each node beginning at the root of the tree by comparing cell type representative cells. Lastly, Garnett hierarchically classifies all cells and optionally provides a second cluster-extended classification. B) tSNE plots of 10x Genomics' 100,000 cell PBMC dataset. The first panel is colored by cell type based on FACS sorting, the second panel is colored by cluster-agnostic cell type according to Garnett classification, and the third panel is colored by the Garnett cluster-extended type, which labels cells based on the composition of their cluster or community. C) A heatmap of data in (B) comparing the labels based on FACS (rows) with the cluster-agnostic (left) and cluster-extended (right) cell type assignments by Garnett (columns). Color represents the percent of cells of a certain FACS type labeled each type by Garnett. D) tSNE plots of 10x Genomics V2 chemistry applied to 8,000 PBMCs from a healthy donor. The first panel is colored by type determined manually using known gene markers. The second and third panels are colored by Garnett cluster-agnostic and cluster-extended cell type assignments by a classifier trained on the data shown in panels (B) and (C). E) Similar to panel (C), a heatmap of data in (D).

notations are not easily transferred between datasets generated by independent groups on related tissues, resulting in wasteful repetition of effort. Finally, cell type annotations are typically *ad hoc*; although ontologies of cell types exist (Bard et al., 2005; Smith et al., 2007; Diehl et al., 2016), we lack tools for systematically applying these ontologies to annotate new scRNA-seq datasets. Collectively, these challenges are strongly hindering progress towards a consensus framework for cell types and the features that define them.

Towards addressing these challenges, we devised Garnett (Figure 4.1A). Garnett consists of four components. First, Garnett defines a markup language for specifying cell types using the genes that they specifically express. The markup language is hierarchical in that a cell type can have subtypes (e.g. CD4+ and CD8+ are subsets of T cells). Second, Garnett includes a parser that processes the markup file together with a single-cell dataset, identifying representative cells bearing markers that unambiguously identify them as one of the cell types defined in the file. Third, Garnett trains a classifier that recognizes additional cells as belonging to each cell type based on their similarity to representative cells, similar to an approach that our groups recently developed for annotating a single-cell mouse atlas of chromatin accessibility (Cusanovich et al., 2018a). Importantly, Garnett does not require that cells be organized into clusters, but it can optionally extend classifications to additional cells using either its own internal clustering routines or those of other tools, such as Monocle (Qiu et al., 2017b) or Seurat (Satija et al., 2015). Finally, Garnett provides a method for applying a classifier trained on one dataset to rapidly annotate additional datasets.

We tested Garnett on a benchmark single-cell RNA-seq dataset generated using the 10x Genomics Chromium platform. This dataset is comprised of 94,571 peripheral blood mononuclear cells (PBMCs) that were individually immunophenotyped via flow cytometry and therefore have a "gold standard" annotation of cell type (Zheng et al., 2017). Garnett requires at least one marker gene for each cell type, ideally one that is specifically expressed and readily detectable only in that cell type. As a supervised method, Garnett's accuracy will be dependent on these markers, so we devised a measure of each marker's usefulness for the purposes of Garnett classification. This marker score combines the number of cells that a marker nominates with an estimate of how many cells inclusion of the marker will render ambiguous. To classify the PBMCs, we populated

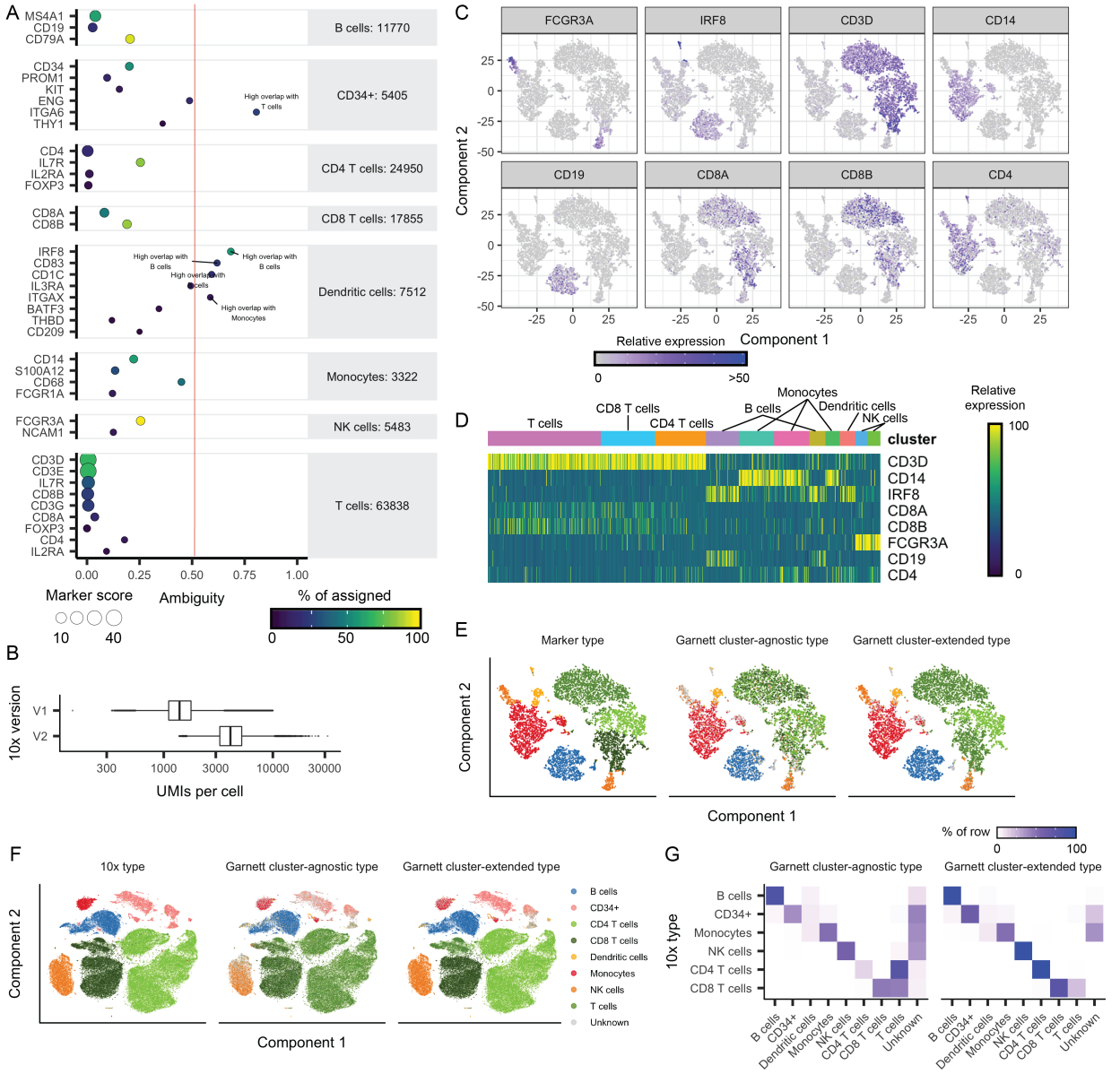

Figure 4.2: Garnett accurately classifies peripheral blood mononuclear cells (PBMCs). A) PBMC marker quality plot. X-axis corresponds to the ambiguity score, defined as the ratio of the number of ambiguous cells when the marker is included over the number of cells in which the marker is expressed. Color represents the percent of nominated cells for that cell type that were nominated by that marker, and the number next to the cell type names is the total number of nominated cells in that cell type. Markers with an ambiguity score greater than 0.5 (indicated by the red line) were excluded from the marker file. B) Boxplots of the number of unique molecular indexes (UMIs) per cell in 10x Genomics version 1 (V1) PBMC dataset versus version 2 (V2) (Boxplot elements: center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range; points, outliers). (*Legend continued on the following page*)

Figure 4.2: (*continued*) C) tSNE plots of 10x Genomics V2 PBMC dataset. Color represents the relative expression of marker genes for each expected cell type (*FCGR3A*: NK cells, *IRF9*: Dendritic cells, *CD3D*: T cells, *CD14*: Monocytes, *CD19*: B cells, *CD8A* and *CD8B*: CD8 T cells, *CD4*: CD4 T cells). D) Correspondence between markers of interest and cell clusters in 10x Genomics V2 PBMC dataset with manually assigned cell type labels. Heatmap of relative expression, rows are marker genes and columns are cells sorted by tSNE cluster assignment. E) tSNE plot of Garnett cluster-agnostic and cluster-extended type assignments for 10x Genomics V2 PBMCs, also trained on V2. F) tSNE plot of Garnett cluster-agnostic and cluster-extended type assignments for 10x Genomics V1 PBMCs, trained on V2. G) Correspondence of Garnett cluster-agnostic and cluster-extended type assignments with FACS assignments for data from (F). Color represents the percent of cells of a certain FACS type labeled each type by Garnett.

a marker file including each of the expected cell types using commonly used markers in the literature. We then used Garnett's quality metric to exclude poorly scoring markers (ambiguity >0.5) before proceeding with classification (Figure 4.2A).

Garnett assigned 71% (3% incorrect, 26% unclassified) of the cells to the correct type ("cluster-agnostic type"), with 34% of T cells also receiving a correct subtype classification (41% not subclassified, 23% unclassified, 2% incorrect) (Figure 4.1B, Figure 4.1C). Cells that remained unlabeled were comparably distributed amongst immunophenotypes, suggesting that the algorithm was not failing to recognize one or more of the cell types entirely. Moreover, by expanding cell type assignments to nearby cells using Louvain clustering (Levine et al., 2015) ("cluster-extended type"), correct assignments increased to 94% (2% incorrect, 4% unclassified), with 91% of T cells also receiving a correct subtype classification (8% not subclassified, <1% unclassified, <1% incorrect).

We next evaluated Garnett's ability to classify data not seen during training by analyzing PBMCs that were generated with a second generation of the 10x Genomics Chromium system ("V2"). These cells were unsorted and profiled using a different library preparation method that yields much greater molecular depth per cell (Figure 4.2B). Because the V2 cells were unsorted, we manually assigned cell types to clusters based on classic markers (Figure 4.2C, Figure 4.2D). Despite being trained on sparser molecular data from a different version of the Chromium chemistry, classification accuracy remained high, with 80% (3% incorrect, 17% unclassified) of cells correctly labeled with cluster-agnostic type and 95% (3% incorrect, 2% unclassified) with cluster-extended

Table 4.1: Consensus cell types for lung datasets

Consensus type	**Tabula muris type**	**MCA type**	**Lambrechts type**
Alveolar	epithelial cell of lung	AT2 Cell AT1 Cell Alveolar bipotent progenitor	Alveolar
B cells	B cell	Ig-producing B cell B Cell	B cells
Ciliated cells	ciliated columnar of tracheobronchial tree	Ciliated cell Clara cell Dividing cells	Epithelial
Endothelial	lung endothelial cell	Endothelial cell_Tmem100 high Endothelial cells_Vwf high Endothelial cell_Kdr high	Endothelial
Fibroblasts	stromal cell	Stromal cell_Inmt high Stromal cell_Dcn high Stromal cell_Acta2 high	Fibroblasts
Granulocytes		Neutrophil granulocyte Eosinophil granulocyte Basophil	
Leukocytes	leukocyte		

Macs/Monos/DC	classical monocyte monocyte	Conventional dendritic cell_H2-M2 high Dendritic cell_Naaa high Dividing dendritic cells Conventional dendritic cell_Tubb5 high Conventional dendritic cell_Gngt2 high Plasmacytoid dendritic cell Conventional dendritic cell_Mgl2 high Monocyte progenitor cell Interstitial macrophage Alveolar macrophage_Ear2 high Alveolar macrophage_Pclaf high	
Myeloid	myeloid cell		Myeloid
Natural killer cells	natural killer cells	NK Cell	
T cells	T cell	T Cell_Cd8b1 high Dividing T cells Nuocyte	T cells
Tumor			Tumor

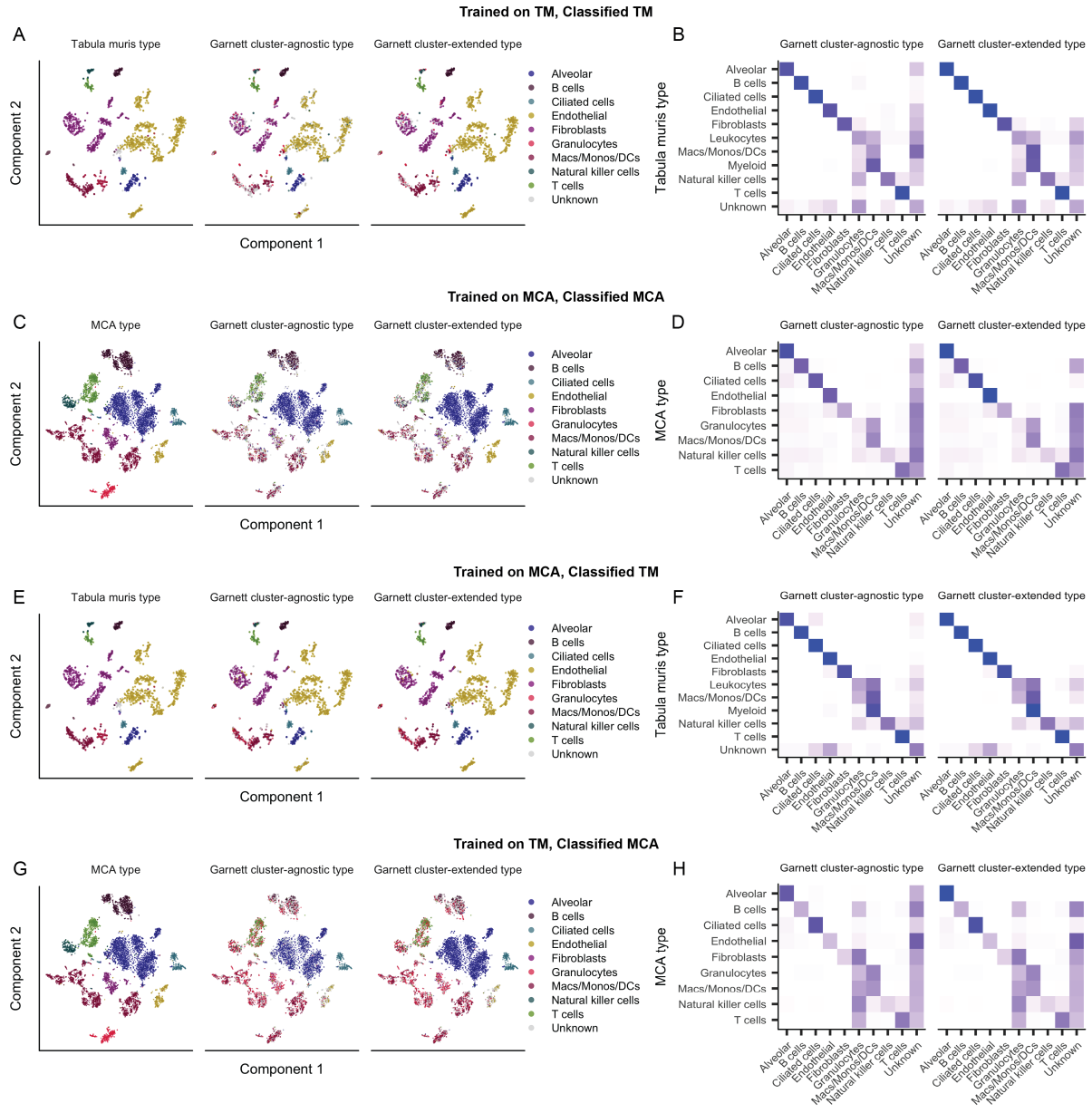

Figure 4.3: Garnett accurately classifies lung cell types from recent mouse cell atlases. Panels (A), (C), (E), and (G) are tSNE plots of Tabula Muris (TM) (Tabula Muris Consortium et al., 2018) and Mouse Cell Atlas (MCA) (Han et al., 2018) lung subsets colored by reported cell type versus Garnett cluster-agnostic and cluster-extended types. Panels (B), (D), (F) and (H) are heatmaps comparing the reported cell types (rows) versus the Garnett cluster-agnostic and cluster-extended types (columns). Color represents the percent of cells of a certain reported type labelled as each type by Garnett. (*Legend continued on the following page*)

Figure 4.3: (*continued*) Panels (B), (D), (F) and (H) correspond with (A), (C), (E), and (G) respectively. Panels (A) and (B) are TM data, and were classified using the TM-trained classifier. Panels (C) and (D) are MCA data, and were classified using the MCA-trained classifier. Panels (E) and (F) are TM data, and were classified using the MCA-trained classifier. Panels (G) and (H) are MCA data, and were classified using the TM-trained classifier.

type (Figure 4.1D, Figure 4.1E).

We also assessed the impact of training on deeply profiled cells in order to classify more sparsely sequenced ones. When trained on V2 cells, Garnett classified V1 cells accurately (83% correct with cluster-agnostic type and 95% correct with cluster-extended) (Figure 4.2E, Figure 4.2F, Figure 4.2G).

To evaluate Garnett's ability to catalog cell types in complex solid tissues, we analyzed lung tissue data from two recently reported "molecular atlases" of mouse organs. The Mouse Cell Atlas (MCA) (Han et al., 2018) and Tabula Muris (TM) (Tabula Muris Consortium et al., 2018) projects collected single-cell RNA-seq data using microwell and droplet-based sequencing platforms, respectively. We defined a single hierarchy of cell types expected to be found in the lung based on those studies and compiled marker genes from literature to recognize them in each dataset (all marker files available in Section 4.4, consensus cell type names in Table 4.2). Overall, Garnett's classifications agreed with both the MCA (58% correct, 29% unclassified with cluster-agnostic type; 65% correct, 23% unclassified with cluster-extended type; Figure 4.3A, Figure 4.3B) and TM (71% correct, 22% unclassified with cluster-agnostic type; 87% correct, 8% unclassified with cluster-extended type; Figure 4.3C, Figure 4.3D) annotations, which were derived by manual inspection of genes enriched in each cluster. Moreover, a Garnett model trained on the MCA accurately classified the TM cells and vice versa (trained on MCA: 82% correct, 5% unclassified with cluster-agnostic type; 86% correct, 2% unclassified with cluster-extended type; trained on TM: 46% correct, 30% unclassified with cluster-agnostic type; 56% correct, 21% unclassified with cluster-extended type; Figure 4.3E, Figure 4.3F, Figure 4.3G, Figure 4.3H).

We next sought to evaluate whether Garnett was similarly useful for annotating single-cell chro-

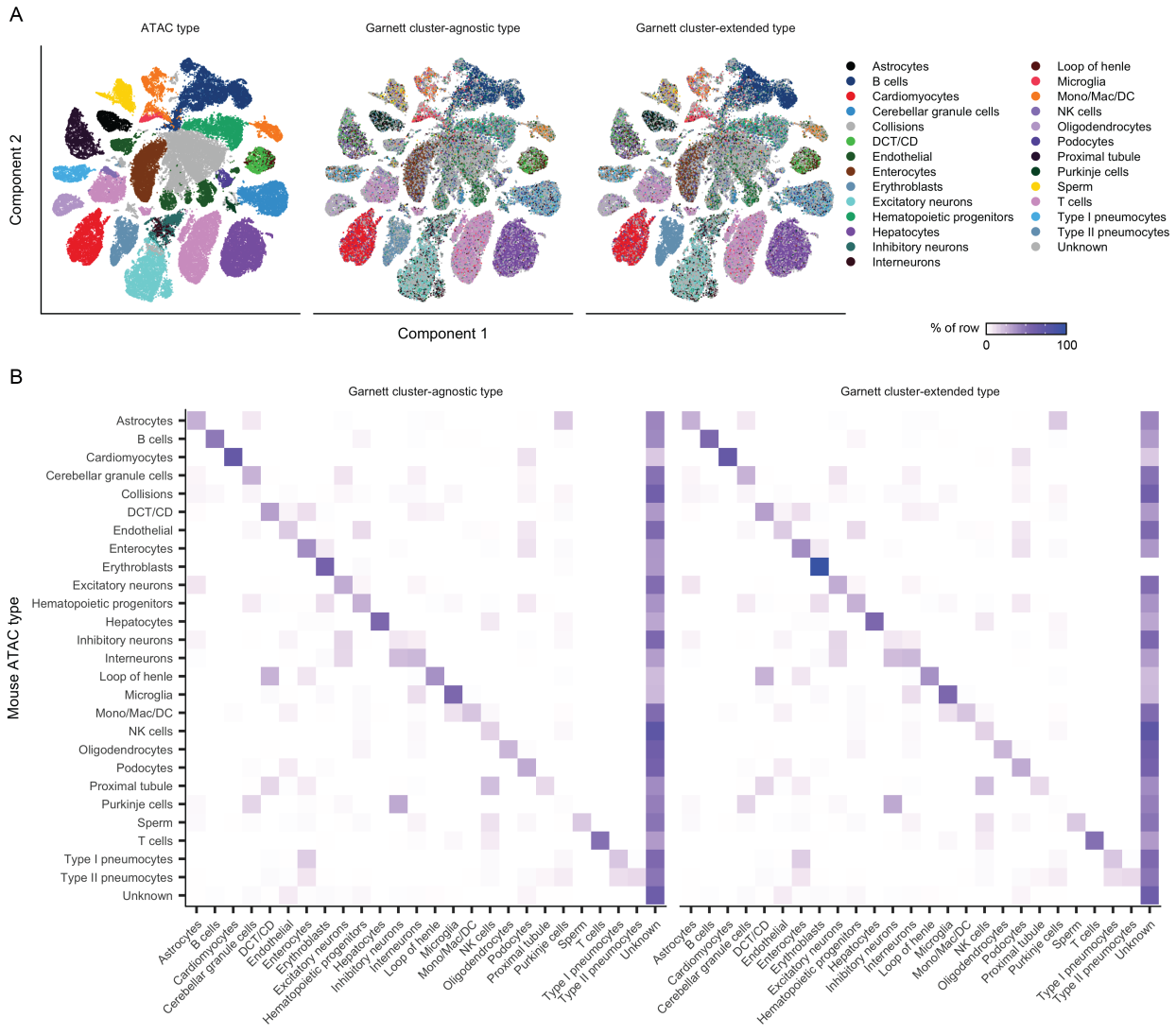

Figure 4.4: Garnett can classify cells from single-cell chromatin accessibility datasets. A) tSNE plot of the Cusanovich et al. (2018a) mouse single-cell ATAC-seq atlas. Garnett used publicly available Cicero (Pliner et al., 2018) gene activity scores in place of expression data to classify cell types. The first panel is colored by Cusanovich et al. (2018a) manually assigned cell type labels. The second and third panel are colored by the Garnett cluster-agnostic and cluster-extended types respectively. B) Heatmaps comparing the reported cell types versus the Garnett cluster-agnostic and cluster-extended types. Color represents the percent of cells of a certain reported type labelled as each type by Garnett.

matin accessibility (scATAC-seq) datasets, which we have generally found to be more challenging to manually annotate than scRNA-seq datasets. We and colleagues recently used regularized, multinomial regression to classify clusters of cells based on chromatin accessibility (Cusanovich et al., 2018a). We adapted Garnett to classify cells based on scATAC-seq-derived "gene activity scores", a measure of open chromatin around each gene (Pliner et al., 2018). Applying it to our recent scATAC-seq atlas of the mouse (Cusanovich et al., 2018a), Garnett labeled 39% of cells concordantly with our previous assignments (cluster-extended; 22% incorrect; 39% unclassified) (Figure 4.4). A caveat is that the marker file was informed by our previous literature-based annotation of the dataset by a related method, but these analyses nonetheless illustrate the potential of Garnett to enable the rapid annotation of not only scRNA-seq but also scATAC-seq datasets.

We next sought to apply Garnett to the task of discriminating all the cell types of a whole animal, focusing on our recent transcriptional atlas of the L2 stage *C. elegans* nematode (Cao et al., 2017). We originally assigned broad cell identities to each of 29 major clusters, and then subtyped the neurons using a second level of markers. We defined a cell hierarchy that discriminated the major cell types, as well as subtypes of neurons, using the marker genes from the original study. Of cells that were previously assigned, Garnett labeled 87% of cells concordantly in terms of major cell type (cluster-extended; 8% incorrect; 5% unclassified), with rectum cells being frequently mislabeled as non-seam hypodermis (Figure 4.5A, Figure 4.5B, Figure 4.6, Figure 4.7). Of the 4,186 neurons assigned subtypes in the original study, 53% were subtyped correctly, and a further 18% were labeled as neurons of unknown subtype (cluster-agnostic; 8% incorrect) (Figure 4.5C). Together, these analyses demonstrate that Garnett can scale to classifying the cell types found in a whole animal.

Finally, as tissue–specific patterns of gene expression are largely conserved across vertebrates (Merkin et al., 2012), we wondered whether Garnett models trained on mouse data could be used to classify human cell types. To evaluate this, we applied the Garnett model trained on the MCA lung dataset to scRNA-seq data from human lung tumors described in Lambrechts et al. (2018) (Figure 4.5D, Figure 4.5E, Figure 4.8, Table 4.2). Over 92% of the alveolar, B cells, T cells, epithelial (ciliated) cells, endothelial cells, and fibroblasts were accurately assigned by the Garnett

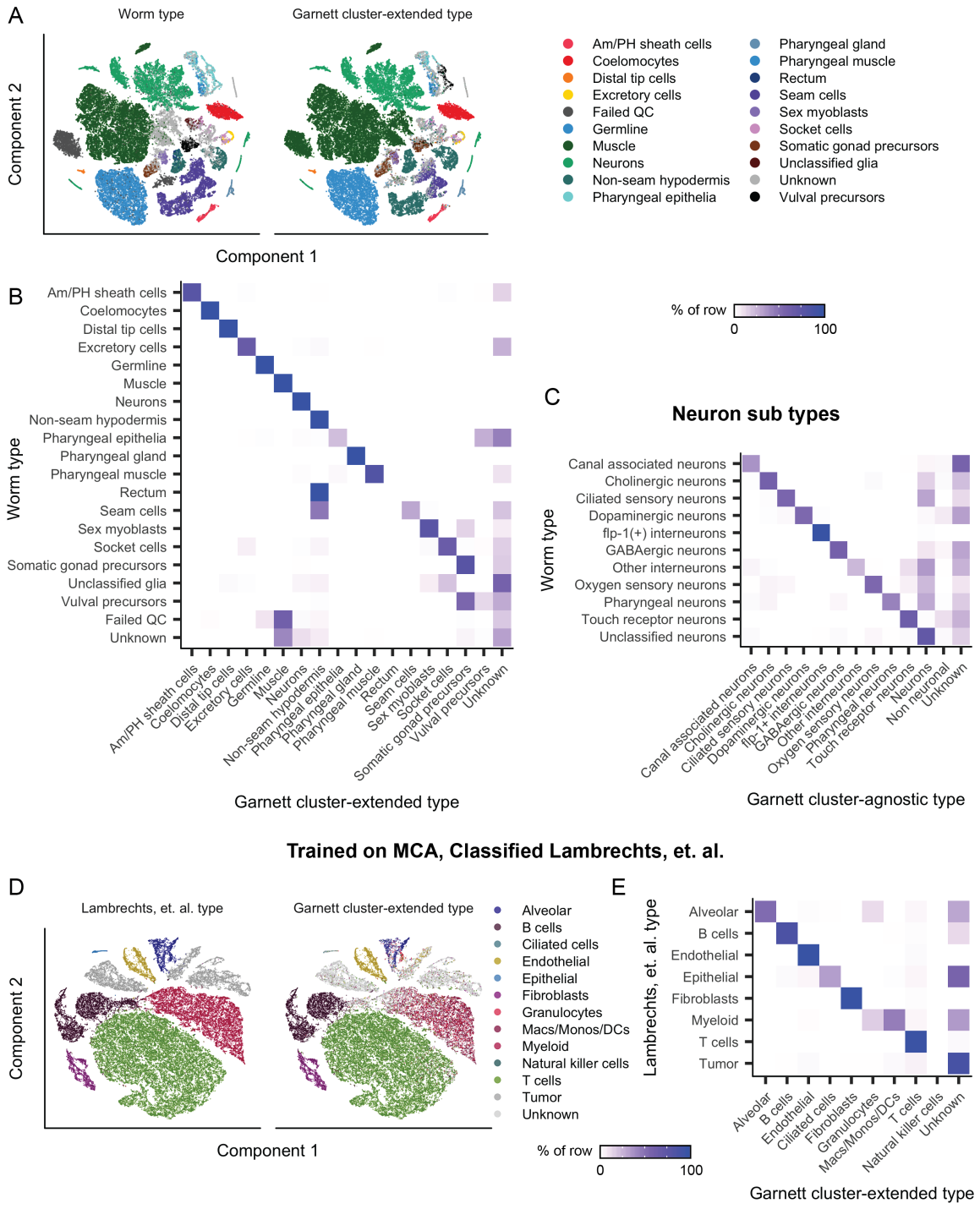

Figure 4.5: Garnett can discriminate among cell types across a whole animal, across species and between normal and pathological tissue. (*Legend continued on the following page*)

Figure 4.5: (*continued*) Garnett classification results for sci-RNA-seq data from whole *C. elegans*, published in ref (Cao et al., 2017). A) tSNE plots of the whole worm dataset. First panel is colored by published type from Cao et al. (2017), second panel colored by the major (top level) Garnett cluster-extended classification. Garnett cluster-agnostic type is available in Figure 4.7. B) Heatmap comparing the reported cell types versus the Garnett cluster-extended cell types. Color represents the percent of cells of a certain reported type labelled as each type by Garnett. C) Heatmap comparing the reported neuron subtypes versus the Garnett cluster-agnostic neuron subtypes. D) Garnett cluster-extended results for human lung tumors from Lambrechts et al. (2018) classified based on a Garnett classifier trained on lung cells from the Mouse Cell Atlas. tSNE plots of the human lung tumor dataset. First panel is colored by published type from Lambrechts et al. (2018), second panel is colored by the Garnett cluster-extended classification. E) Heatmap comparing the reported cell types versus the Garnett cluster-extended cell types from (D). Color represents the percent of cells of a certain reported type labelled as each type by Garnett.

MCA model. Of the 9,756 cells annotated as myeloid (Lambrechts et al., 2018), Garnett labeled 44% as monocyte/macrophage/dendritic cell and a further 16% granulocytes, leaving 34% unclassified. 22% of the dataset was labeled "unknown" by Garnett, of which 55% were identified as tumor cells in the original study. As expected, given that they are not represented in the original marker file nor in the MCA lung dataset, 88% of all cells annotated as tumor cells in the original study were labeled as "unknown" by Garnett. These analyses demonstrate that Garnett can operate across species, and is not necessarily confounded by the presence of pathological cell states when trained on normal healthy tissue.

The annotation of cell types based on their molecular signatures is a critical step for the construction of a human cell atlas. It is also increasingly the rate limiting step, as illustrated by recent studies that resorted to labor-intensive, *ad hoc* literature review to achieve this end (Cao et al., 2017; Cusanovich et al., 2018a; Han et al., 2018; Rosenberg et al., 2018; Tabula Muris Consortium et al., 2018; Zeisel et al., 2018). Garnett is an algorithm and accompanying software that automates and standardizes the process of classifying cells based on marker genes. A key point is that the hierarchical marker files on which Garnett is based are interpretable to biologists and explicitly relatable to the existing literature. Furthermore, together with these markup files, Garnett classifiers trained on one dataset are easily shared and applied to new datasets, including across single-cell methods and chemistries.

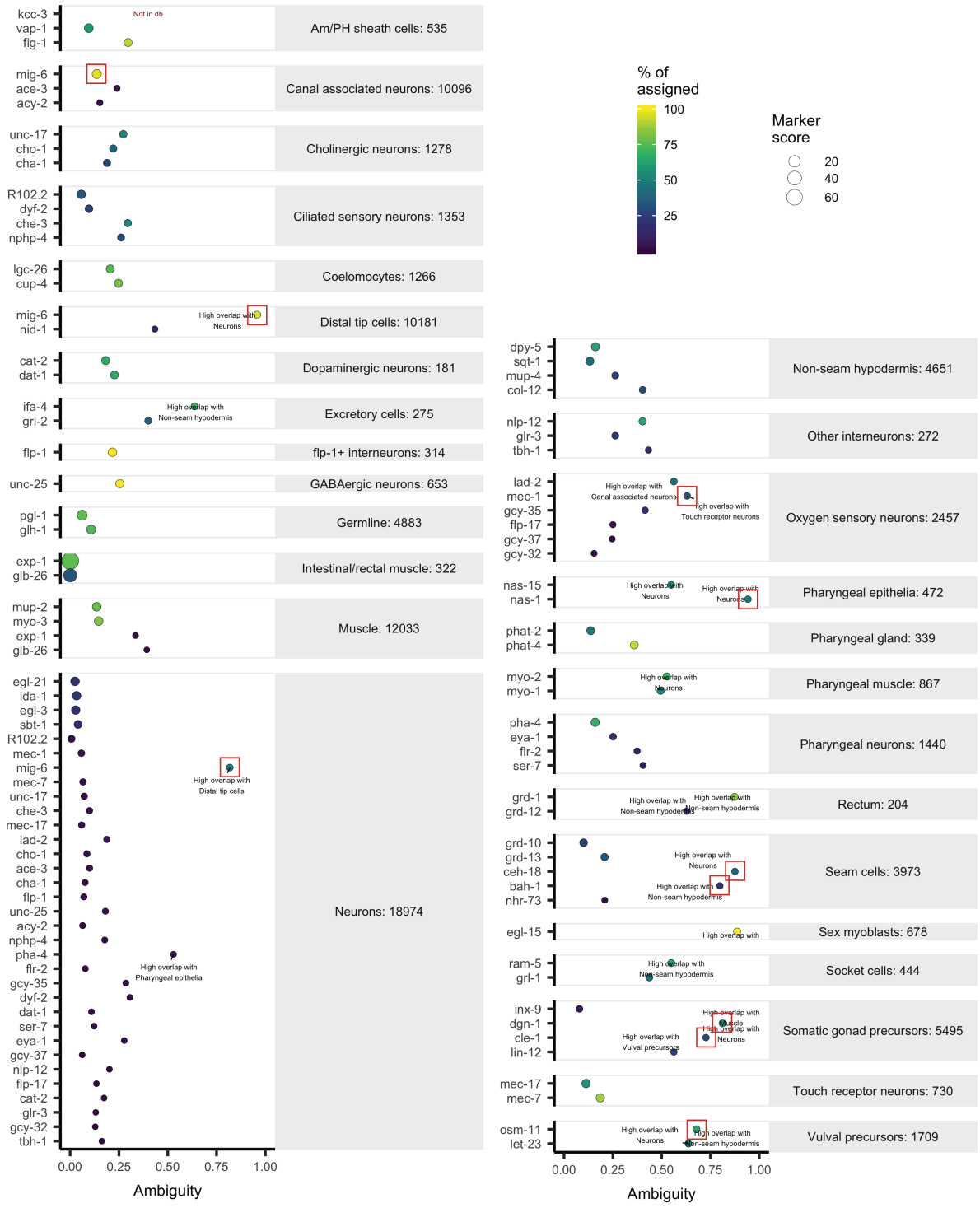

Figure 4.6: Marker quality chart for *C. elegans*. (*Legend continued on the following page*)

Figure 4.6: (*continued*) X-axis represents the ambiguity score, defined as the ratio of number of ambiguous cells when the marker is included over the number of cells the marker is expressed in. Color represents the percent of nominated cells for that cell type that were nominated by that marker, and the number next to the cell type names is the total number of nominated cells in that cell type. Markers were initially chosen directly from Cao et al. (2017). Markers excluded because of high ambiguity are marked with red boxes.

We anticipate the potential for an "ecosystem" of Garnett marker files and pre-trained classifiers that: 1) enable the rapid, automated, reproducible annotation of cell types in any newly generated dataset, 2) minimize redundancy of effort, by allowing for marker gene hierarchies to be easily described, compared, and evaluated, and 3) facilitate a systematic framework and shared language for specifying, organizing, and reaching consensus on a catalog of molecularly defined cell types. To these ends, in addition to releasing the Garnett software, we have made the marker files and pre-trained classifiers described in this manuscript available at a wiki-like website that facilitates further community contributions, together with a web-based interface for applying Garnett to user datasets (https://cole-trapnell-lab.github.io/garnett).

4.3 METHODS

4.3.1 Garnett

Garnett is designed to simplify, standardize, and automate the classification of cells by type and subtype. To train a new model with Garnett, the user must specify a cell hierarchy of cell types and subtypes, which may be organized into a tree of arbitrary depth; there is no limit to the number of cell types allowed in the hierarchy. For each cell type and subtype, the user must specify at least one marker gene that is taken as positive evidence that the cell is of that type. Garnett includes a simple language for specifying these marker genes, in order to make the software more accessible to users unfamiliar with statistical regression. Negative marker genes, i.e. taken as evidence against a cell being of a given type, can also be specified. In addition, Garnett includes tools for selecting and checking the quality of markers. Garnett uses the marker information provided to select cells that are then used to train a regression-based classifier, similar to the approach taken in Cusanovich

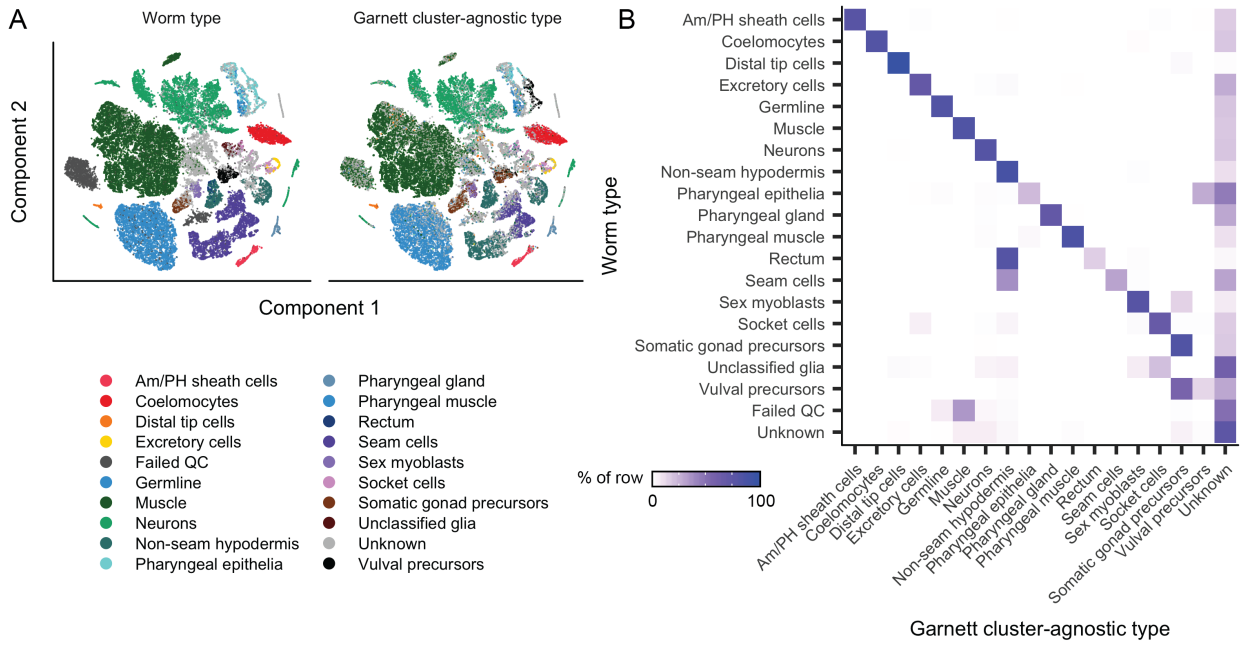

Figure 4.7: Garnett classification results for sci-RNA-seq data from whole L2 stage *C. elegans*. A) tSNE plots of the whole worm dataset (Cao et al., 2017). First panel is colored by published type from Cao et al. (2017), second panel is colored by the major (top level) Garnett cluster-agnostic classification. B) Heatmap comparing the reported cell types (rows) versus the Garnett cluster-agnostic cell type (columns). Color represents the percent of cells of a certain reported type labelled as each type by Garnett.

et al. (2018a). After a classifier is trained, it can be applied to other single-cell datasets run on the same or different platforms. Algorithmic details are provided below.

Constructing marker files

Garnett uses a marker file to allow users to specify cell type definitions. These definitions are then used to choose representative cells for each cell type to use when training the classifier. Full details describing the syntax of the marker file are provided as part of the software package. Briefly, the marker file consists of a series of cell type entries, beginning with a cell type name, followed by lists of expressed markers and metadata. In addition, cell types can be specified to be a subtype of another defined type, i.e. hierarchical definitions. Marker files also have the capability to hold

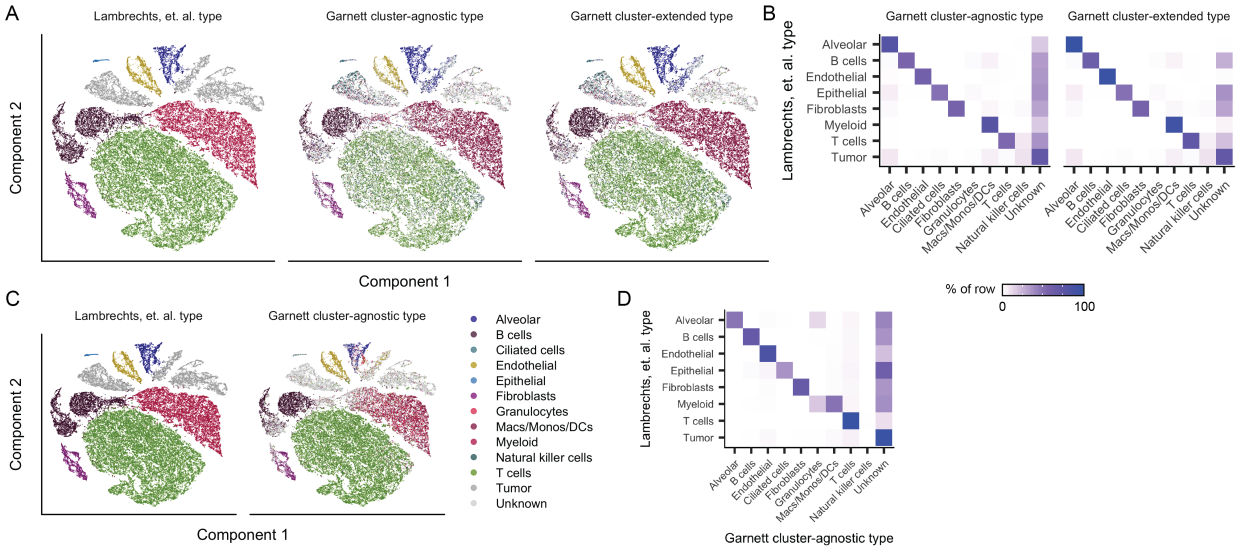

Figure 4.8: Garnett classification of single-cell RNA-seq data from lung tumors. A) tSNE plots of lung tumor scRNA-seq dataset (Lambrechts et al., 2018). First panel is colored by published type from Lambrechts et al. (2018), second panel colored by Garnett cluster-agnostic cell type, and third panel colored by Garnett cluster-extended cell type, based on a model trained using this same dataset. B) Heatmaps comparing the reported cell types versus the Garnett cluster-agnostic and cluster-extended cell types from (A). Color represents the percent of cells of a certain reported type labelled as each type by Garnett. C) Garnett cluster-agnostic results for human lung tumors from Lambrechts et al. (2018) classified based on a Garnett classifier trained on lung cells from the Mouse Cell Atlas (Han et al., 2018). tSNE plots of the human lung tumor dataset. First panel is colored by published type from Lambrechts et al. (2018); second panel colored by the Garnett cluster-agnostic classification. D) Heatmap comparing the reported cell types versus the Garnett cluster-agnostic cell types from (C). Color represents the percent of cells of a certain reported type labelled as each type by Garnett.

literature references for the chosen marker genes that are then included as metadata in the classifier.

Because only markers that are expressed specifically in a given cell type are useful for Garnett classification, we also provide functions for assessing the value of each of the provided marker genes. These functions estimate the number of cells that a given marker nominates for their cell type, the number of cells that become "ambiguously" nominated to multiple cell types in a given level of the hierarchy when the marker is included, and an overall marker score G, defined as:

$$G = \frac{1}{(a+p)} * \frac{b}{n}$$

where a is the fraction of cells nominated by the given marker that are made ambiguous by that marker, p is a small pseudocount, b is the number of cells nominated by the marker, and n is the total number of cells nominated for that cell type. In addition to estimating these values, Garnett will plot a diagnostic chart to aid the user in choosing markers (e.g. Figure 4.2A).

Training the classifier

Choosing cells:

Garnett's first step in training a cell type classifier is to choose representative cells to train on. Let M be an m by n matrix of input gene expression data. First, $M_{i,j}$ is normalized by size factor (the geometric mean of the total UMIs expressed for each cell j) to adjust for read depth, resulting in a normalized m by n matrix N. In addition, the gene IDs of the expression data are converted to Ensembl IDs using correspondence tables from a Bioconductor AnnotationDbi-class package (Pagès et al., 2018) . Next, the input marker file is parsed and the gene IDs are also converted to Ensembl IDs as above. Finally, a tree representation of the marker file is constructed, with any designated subtypes placed as children of the parent cell type in the tree. In addition to the tree, a dataset-wide size factor is generated and saved to the tree to allow normalization to new datasets for later classification (see classifying cells section below).

For each parent node in the tree, the following steps are taken: First, cells are scored as "expressed" or "not expressed" for each of the provided markers and an aggregate marker score is derived for each cell type for each cell (details on scoring below). Next, any metadata or hard expression cutoffs are applied to exclude a subset of cells from consideration. Lastly, outgroup samples are chosen (see below). After choosing the training sample, the classifier is trained (see below), and a preliminary classification is made in order to further train downstream nodes.

Aggregated marker scores:

We devised an aggregated marker scoring system to address two challenges of single-cell RNA-

seq data for the purposes of identifying representative cell types based on markers. The first challenge when choosing cells is that of differing levels of expression of different markers. If a lowly expressed but specific marker is found in a cell profile, this is better evidence of cell type than a highly expressed and less specific marker. To address this, we use the term frequency-inverse document frequency (Jones, 1972) (TF-IDF) transformation when generating aggregate marker scores. The TF-IDF transformed matrix is defined by,

$$T_{i,j} = \frac{N_{i,j}}{\sum_{i=1}^{m} N_{i,j}} * log\left(1 + \frac{n}{\sum_{j=1}^{n} N_{i,j}}\right)$$

where $N_{i,j}$ is the m by n normalized gene expression matrix defined above.

The second challenge we addressed in our aggregate marker score calculation was that highly expressed genes have been known to leak into the transcriptional profiles of other cells. For example, in samples including hepatocytes, albumin transcripts are often found in low copy numbers in non-hepatocyte profiles. To address this, we assign a cutoff above which a gene is considered expressed in that cell. To determine this cutoff, we use a heuristic measure defined as

$$c_i = 0.25 * q_i$$

where c_i is the gene cutoff for gene i and q_i is the 95th percentile of T for gene i. Any gene i in cell j with a value $T_{i,j}$ below c_i is set to 0 for the purposes of generating aggregated marker scores.

After these transformations, the aggregated marker score is defined by a simple sum of the genes defined as markers in the cell marker file,

$$S_{c,j} = \sum_{k \in G_c} T_{k,j}$$

where $S_{c,j}$ is the aggregated score for cell type c and cell j, and G_c is the list of marker genes for cell type c. Cells in the 75th percentile and above for aggregated marker score S in only 1 cell type

are chosen as good representatives. Any metadata specifications (e.g. the requirement that a cell type have come from a particular tissue), provided in the marker file are then used to exclude cells and generate a final training dataset.

Choosing outgroup cells:

When choosing outgroup samples for training, we wanted to make sure that the outgroup set is not dominated by the most abundant cell type. To do this, we cluster a random subset of potential outgroup cells and choose equal numbers of random cells from each cluster to make up the outgroup. Specifically, we first calculate the first 50 principal components using principal components analysis (PCA) as implemented by the irlba R package (Baglama et al., 2018). Next, we calculate jaccard coefficients on a k-nearest-neighbors (kNN) graph generated using k = 20. Lastly, we generate clusters using Louvain community detection on the resulting cell-cell map of jaccard coefficients. A random set of cells from each resulting community is then combined to create the outgroup.

Training with GLMnet:

The classifier is trained on the normalized expression matrix N for cells chosen as representatives and for all genes expressed in greater than 5% of cells in at least one training set and not expressed in the 90th percentile of TF-IDF transformed expression in all cell types. This last filter prevents ubiquitously expressed genes from being chosen as features. The classifier is trained using genes as features and cells as observations with a grouped multinomial elastic net regularized (alpha = 0.3) generalized linear model using the package GLMnet (Friedman et al., 2008) in R. Observations are weighted by the geometric mean of the counts in each of the training groups. The GLMnet regularization parameter λ is chosen using 3-fold cross validation. Genes provided in the marker files are required to be included in the model and are not regularized.

Classifying cells

Because we wished to be able to use pre-trained classifiers to classify cells across datasets and platforms, we include a dataset size factor D for the training data with the classifier object. D is the geometric mean of the total read counts per cell divided by the median number of genes expressed above zero per cell. Formally, D is defined by

$$a = \sum_{i=1}^{m} M_{i,j}$$

$$D = \exp\left[\frac{1}{p}\sum_{k=1}^{p} \ln a_k\right] * \frac{1}{\text{median}\{g\}}$$

where g is the number of genes expressed above zero per cell. When applying an existing classifier to a new dataset, we can then transform the new expression data, an m' by n' matrix M', to the scale of the training data using D,

$$f_j = \frac{\sum_{i=1}^{m'} M'_{i,j}}{D * \text{median}\{g'\}}$$

$$N' = \frac{M'}{f_j}$$

where g' is the number of genes expressed above zero per cell in the new data.

After normalization, gene IDs for the new dataset are also converted to Ensembl IDs. At each internal node in the classifier, the multinomial model for that node is applied to the data, the output probabilities of each class are normalized by dividing by the maximum probability for each cell, and the ratio of the top scoring cell type to the second-best scoring cell type is calculated. If this odds ratio is greater than the user-specified rank probability ratio (in this paper and by default, we use 1.5), the top type is assigned, otherwise the cell type is set to "Unknown". Optionally, Garnett

will add a second set of classifications which classify an entire cluster of cells if: greater than 90% of assigned cells within a cluster are the same type and greater than 5% of all cells in the cluster are classified (not "Unknown"), and greater than 5 cells in the cluster are classified. Cluster labels can be provided by the user or generated by Garnett using Louvain community detection in the top 50 principal components of the expression matrix.

4.3.2 10x Genomics peripheral blood mononuclear cell (PBMC) analysis

10x PBMC datasets from both version 1 (V1) and version 2 (V2) chemistry were downloaded from the 10x Genomics website. The V1 cells are a combination of each of the pure cell type populations isolated by 10x Genomics using FACS sorting (CD14+ Monocytes, CD19+ B cells, CD34+ cells, CD4+ Helper T cells, CD4+/CD25+ Regulatory T cells, CD4+/CD45RA+/CD25- Naive T cells, CD4+/CD45RO+ Memory T cells, CD56+ Natural killer cells, CD8+ Cytotoxic T cells and CD8+/CD45RA+ Naive cytotoxic T cells), preprocessed using CellRanger 1.1.0 and published in Zheng et al. (2017). The V2 cells are the V2 chemistry distributed demonstration dataset labelled "8k PBMCs from a healthy donor", preprocessed using CellRanger 2.1.0. Markers for PBMCs were those often cited in the literature. Using Garnett's marker scoring system, we excluded the markers with high ambiguity ($>$0.5). The final PBMC marker file used is available in Section 4.4.1. Garnett classification for V1 and V2 was run using default parameter values defined in the preceding sections.

4.3.3 Tabula muris and mouse cell atlas (MCA) lung analysis

The Tabula Muris FACS dataset from Tabula Muris Consortium et al. (2018) was downloaded from their figshare website. The MCA dataset from Han et al. (2018) was downloaded from their figshare website. For the purposes of this analysis, only data derived from lung tissue from both datasets were used. To facilitate comparisons between each of the lung datasets used, a set of consensus cell type names was used as described in Table 4.2. The marker file used is available in Section 4.4.2. Garnett classification was run using default parameter values for both datasets.

4.3.4 sci-ATAC-seq analysis

The sci-ATAC-seq data was downloaded from the website associated with Cusanovich et al. (2018a). The input to Garnett was the previously calculated Cicero gene activity scores presented in the original publication. The final marker file used is available in Section 4.4.3. Garnett classification was run using default parameter values.

4.3.5 C. elegans analysis

The worm data was downloaded from the website associated with Cao et al. (2017). Markers were those used by the original publication to identify cell types. Using Garnett's marker scoring system, we excluded the markers with high ambiguity (Figure 4.6). The final marker file used is available in Section 4.4.4. Garnett classification was run using default parameter values.

4.3.6 Human lung tumor analysis

The human lung tumor data was downloaded from the ArrayExpress database entry associated with Lambrechts et al. (2018). Because expression data were log-transformed, we first exponentiated the expression data before classification. To allow for cross-species classification, we first converted the human expression data to mouse gene labels by creating a correspondence table using the biomaRt hsapiens_gene_ensembl and mmusculus_gene_ensembl databases. Only unique rows (one-to-one correspondences) were used. Ultimately 15,336 of the original 22,180 human genes could be converted to mouse labels including 89 percent of the genes in the MCA classifier with non-zero coefficients. The final marker file used is available in Section 4.4.5. Garnett classification was run using default parameter values.

4.4 SUPPLEMENTAL FILES

4.4.1 PBMC marker file

>CD34+ expressed: CD34, THY1, ENG, KIT, PROM1

references: https://www.stemcell.com/human-hematopoietic-stem-and-progenitor-cell-phenotyping-panels.html, https://www.rndsystems.com/research-area/hematopoietic-stem-cell-markers

>NK cells

expressed: NCAM1, FCGR3A

references: https://www.biolegend.com/essential_markers

>Monocytes

expressed: CD14, FCGR1A, CD68, S100A12

references: https://www.biolegend.com/essential_markers

>B cells

expressed: CD19, MS4A1, CD79A

>T cells

expressed: CD3D, CD3E, CD3G

>CD4 T cells

expressed: CD4, FOXP3, IL2RA, IL7R

subtype of: T cells

>CD8 T cells

expressed: CD8A, CD8B

subtype of: T cells

>Dendritic cells

expressed: IL3RA, CD1C, BATF3, THBD, CD209

references: https://www.biolegend.com/essential_markers, https://www.cell.com/pb-assets/products/nucleus/nucleus-phagocytes/rnd-systems-dendritic-cells-br.pdf

4.4.2 Mouse lung marker file

>Endothelial
expressed: Lyve1, Tek, Kdr, Ramp2, Flt1
references: https://www.abcam.com/primary-antibodies/endothelial-cell-markers

>B cells
expressed: Cd19, Ms4a1, Cd79a

>T cells
expressed: Cd3d, Cd3e, Cd3g

>Fibroblasts
expressed: Des, Pdgfra, Pdgfrb, Fap
references: http://biocc.hrbmu.edu.cn/CellMarker/browse.jsp

>Alveolar
references: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1356-9597.2004.00712.x
expressed: Sftpa1, Sftpb, Ager, Aqp4

>Natural killer cells
expressed: Ncam1, Cd160, Itga2, Klrb1a
references: https://www.biolegend.com/essential_markers, https://www.bio-rad-antibodies.com/nk-cell-receptor-antibodies.html

>Granulocytes
expressed: Psg26, Fut4, Siglece, Itgam, Ccr3, Il5ra
references: https://www.researchgate.net/post/What_are_the_best_markers_for_the_identification_of_Neutrophils_eosinophils_and_circulating_dendritic_cells_in_whole_blood_by_flow_cytometry

>Ciliated cells
expressed: Cyp2f2, Ccdc153
references: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/club-cell

>Macs/Monos/DCs
expressed: Cd14, Cd68, Cd33, Tlr7, Tlr9, Itgax, Flt3
references: https://www.bio-rad-antibodies.com/macrophage-m1-m2-tam-tcr-cd169-cd-markers-antibodies.html, https://www.bio-rad-antibodies.com/monocyte-cd-markers-antibodies.html

4.4.3 Mouse ATAC-seq marker file

>Astrocytes
expressed: Slco1c1, Slc1a2
references: https://www.abcam.com/neuroscience/astrocyte-markers-and-functions

>B cells
expressed: Cd19, Ms4a1, Cd79a

>Cardiomyocytes
expressed: Myl2, Myl3, Irx4

>Cerebellar granule cells
expressed: Zic1, Zic2
references: http://www.pnas.org/content/104/8/2997

>DCT/CD
expressed: Prom2, Cdh16
references: https://www.ncbi.nlm.nih.gov/pubmed/20333396,
https://www.ncbi.nlm.nih.gov/pubmed/15696118,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5810970/

>Endothelial
expressed: Cdh5, Sele, Kdr, Gata2, Tek
references: https://www.abcam.com/primary-antibodies/endothelial-cell-markers

>Enterocytes
expressed: Slc26a3, Cdh1, Slc5a1, Vil1

>Erythroblasts
expressed: Hbb-b1, Hbb-bs, Gypa

>Excitatory neurons
expressed: Slc17a7, Grin2b, Kcnj6

>Hematopoietic progenitors
expressed: Cd34, Cd93

>Hepatocytes
expressed: Alb, Cyp7a1, Fabp1, Serpina1c

>Inhibitory neurons

expressed: Gad1, Gad2

>Interneurons
expressed: Dlx2, Calb1, Calb2, Npy

>Loop of henle
expressed: Slc12a1, Cldn19

>Microglia
expressed: Cx3cr1, P2ry12
references: https://www.biolegend.com/microglia, https://www.abcam.com/neuroscience/microglia-markers

>Mono/Mac/DC
expressed: Cd80

>NK cells
expressed: Cd160, Siglece, Sh2d1b1

>Oligodendrocytes
expressed: Mag, Mog

>Podocytes
expressed: Pde3a

>Proximal tubule
expressed: Lrp2

>Purkinje cells
expressed: Aldoc

>Sperm
expressed: Pebp4, Prm1, Tmco5, Hist1h2ba, Ldhc, Tnp1

>T cells
expressed: Cd3e, Cd3g, Cd3d

>Type I pneumocytes
expressed: Ager

>Type II pneumocytes

expressed: Sftpa1

4.4.4 C. elegans marker file

>Neurons
expressed: egl-21, egl-3, ida-1, sbt-1, acy-2, ace-3

>Germline
expressed: glh-1, pgl-1

>Muscle
expressed: mup-2, myo-3

>Intestinal/rectal muscle
expressed: exp-1, glb-26
subtype of: Muscle

>Vulval precursors
expressed: osm-11, let-23

>Coelomocytes
expressed: cup-4, lgc-26

>Seam cells
expressed: grd-10, grd-13, nhr-73, bah-1

>Non-seam hypodermis
expressed: sqt-1, dpy-5, col-12, mup-4

>Pharyngeal epithelia
expressed: nas-15

>Distal tip cells
expressed: nid-1

>Am/PH sheath cells
expressed: vap-1, fig-1, kcc-3

>Pharyngeal muscle
expressed: myo-1, myo-2

>Somatic gonad precursors

expressed: lin-12, inx-9, cle-1

>Pharyngeal gland
expressed: phat-2, phat-4

>Sex myoblasts
expressed: egl-15

>Excretory cells
expressed: ifa-4, grl-2

>Socket cells
expressed: grl-1, ram-5

>Rectum
expressed: grd-12, grd-1

>Ciliated sensory neurons
expressed: R102.2, dyf-2, che-3, nphp-4
subtype of: Neurons

>Cholinergic neurons
expressed: cho-1, cha-1, unc-17
subtype of: Neurons

>Other interneurons
expressed: glr-3, tbh-1, nlp-12
subtype of: Neurons

>GABAergic neurons
expressed: unc-25
subtype of: Neurons

>Touch receptor neurons
expressed: mec-17, mec-7
subtype of: Neurons

>Pharyngeal neurons
expressed: flr-2, ser-7, eya-1, pha-4
subtype of: Neurons

>Oxygen sensory neurons
expressed: flp-17, gcy-35, mec-1, lad-2, gcy-32, gcy-37
subtype of: Neurons

>flp-1+ interneurons
expressed: flp-1
subtype of: Neurons

>Canal associated neurons
expressed: acy-2, ace-3
subtype of: Neurons

>Dopaminergic neurons
expressed: dat-1, cat-2
subtype of: Neurons

4.4.5 Human lung marker file

>Endothelial
expressed: LYVE1, TEK, KDR, RAMP2, FLT1, SELE
references: https://www.abcam.com/primary-antibodies/endothelial-cell-markers

>B cells
expressed: CD19, MS4A1, CD79A

>T cells
expressed: CD3D, CD3E, CD3G

>Fibroblasts
expressed: DES, PDGFRA, PDGFRB, FAP
references: http://biocc.hrbmu.edu.cn/CellMarker/browse.jsp Thy1,

>Alveolar
references: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1356-9597.2004.00712.x
expressed: SFTPA1, SFTPB, AGER, AQP4

>Natural killer cells
expressed: CD160, ITGA2, KLRB1
references: https://www.biolegend.com/essential_markers,
https://www.bio-rad-antibodies.com/nk-cell-receptor-antibodies.html

>Granulocytes

expressed: CEACAM8, FUT4, SIGLEC8, CCR3, IL5RA
references: https://www.researchgate.net/post/What_are_the_best_markers_for_the_identification_of_Neutrophils_eosinophils_and_circulating_dendritic_cells_in_whole_blood_by_flow_cytometry

>Ciliated cells
expressed: CYP2F1, CCDC153
references: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/club-cell

>Macs/Monos/DCs
expressed: CD14, CD68, CD33, TLR7, TLR9, ITGAX, FLT3, CCL18
references: https://www.bio-rad-antibodies.com/macrophage-m1-m2-tam-tcr-cd169-cd-markers-antibodies.html, https://www.bio-rad-antibodies.com/monocyte-cd-markers-antibodies.html

4.5 Code availability

Garnett is available as an R package at http://cole-trapnell-lab.github.io/garnett. The manual pages for the Garnett software are in Appendix B.

4.6 Project acknowledgments

We gratefully acknowledge Stephen Tapscott, William Noble, and Daniela Witten as well as members of the Shendure and Trapnell labs, particularly Andrew Hill, for their advice. Zachary Pliner named the software. This work was supported by the following funding: NIH grant U54DK107979 to JS and CT; NIH grant DP2HD088158, RC2DK114777 and R01HL118342 to CT; NIH grants DP1HG007811 and R01HG006283 to JS; and the Paul G. Allen Frontiers Group to JS and CT. JS is an Investigator of the Howard Hughes Medical Institute. CT is partly supported by an Alfred P. Sloan Foundation Research Fellowship. HAP was supported by the National Science Foundation Graduate Research Fellowship under Grant No. (DGE-1256082).

Chapter 5: CLOSING REMARKS

As new single-cell genomic technologies become more accessible to the wider research community, the need for robust methods to analyze the resulting data will grow. In particular, it is important that the analysis challenges of this new data are thoroughly explored and discussed. This will allow researchers not up-to-date with the latest in technology development to understand the caveats and pitfalls as well as the enormous potential for discovery of these new data sources. In this dissertation, I have presented two algorithms that aim to simplify and standardize analysis workflows with two of the most exciting single-cell technologies, single-cell ATAC-seq and single-cell RNA-seq. In Chapters 2 and 3, I described Cicero, which proposes modifications to more standard single-cell RNA-seq analysis methods to allow for the analysis of single-cell ATAC-seq as well. In Chapter 4, I described Garnett, which aims to automate cell type classification – usually one of the first steps in any single-cell analysis task, with both single-cell RNA-seq and single-cell ATAC-seq data. In this final chapter, I reflect on the importance of providing software that is accessible to biologists, and on some of the future directions of this work.

5.1 WRITING SOFTWARE FOR BIOLOGISTS

As a latecomer to computational biology, I have a particular perspective on the task of writing analysis software that aims be of general use to biologists. I can remember the frustration of attempting to use buggy and poorly documented programs and being unable to "hack" a fix for myself. More often than not, I would abandon a software package entirely before getting anything useful out of it. Because of these early experiences while learning to code, I have made great efforts with the packages presented in this dissertation not to make the same mistakes.

Specifically, I have attempted to simplify the workflows as much as possible, to document the code thoroughly, to provide informative error messages, and to provide tutorials, examples, and a polished documentation website. In the case of Garnett, I have also developed a web application

that will allow users with no coding experience to classify their data against available pre-trained classifiers. Especially when simplifying the workflow, this approach can come with a cost. In order to limit the number of decisions/user inputs the software requires to produce a decent result, I have potentially limited the usefulness of the packages on the extremes, for example by excluding edge use-cases and limiting available features. However, I believe these choices will make the software much more accessible to the scientists who might use it towards discovery.

Ultimately, I suspect that the most highly adopted single-cell technologies will be those with the easiest to use computational tools to accompany them. This likely includes developing graphical user interfaces like those provided by the Galaxy platform (Afgan et al., 2018) to grant access to researchers with very limited computational experience.

5.2 Increasing data utilization

In addition to making the lives of biologists easier, the development of better computational methods for analyzing genomic data will increase the usefulness of the data itself. Often, large scale genomic experiments seem to result in only a few insights. However, I believe that we are likely only scratching the surface with these datasets. With the development of each new computational tool, previously generated data may have more to tell us. This is particularly exciting because of the current expectation that data be submitted to a public repository after publication. As new tools become available, we should be sure that we revisit already published high quality datasets to mine new biological insights.

5.3 Future directions

5.3.1 Cicero

Future directions for Cicero include expanding the suite of tools and developing a set of standard pipelines for the increasing number of researchers utilizing single-cell ATAC-seq methods. I expect this number to increase significantly as commercial single-cell ATAC-seq kits become available. Standardized pipelines might include differential expression-type analyses for accessibility and for

Cicero gene activity scores, dimensionality reduction and visualization, and motif-based analyses.

5.3.2 Garnett

In the case of Garnett, the primary future direction is to build up a set of cell type classifiers for various human and model organism tissues. I have implemented a method using Github for community members to submit classifiers to be hosted on the Garnett website to make them available to the community. In addition, methods to reconcile and combine existing classifiers in order to classify data from mixed tissues or even more complex whole organisms would be useful. Hopefully, Garnett classifiers will also help to identify better whole-organism markers for cell types, as a consistent difficulty with marker selection on a large scale is that classical markers of cell types are often only restricted within a single tissue type.

Garnett would also benefit from further testing of the limits of its classification. For example, will classifiers trained on healthy tissue accurately classify perturbed or diseased tissue? In addition, what are the limits of sequencing depth and data quality at which a trained Garnett classifier will remain accurate?

5.3.3 Novel single-cell analysis

It is difficult to imagine the potential future directions in data analysis when developing computational tools for cutting-edge genomic technologies, because we will not know the challenges of the newest data type until that data is generated. However, it seems likely that many of the current downsides of single-cell data will remain - sparsity, false negatives and noise are unlikely to disappear. Future analysis development should focus on addressing these challenges while focusing on robust pipelines that answer the biological questions of interest.

5.4 Conclusion

New single-cell genomic technologies will never reach their full potential for discovery without robust and accessible computational analysis tools. This dissertation describes two new algorithms

aimed at filling some of this need. As wet-lab techniques become more widely available, we must ensure that researchers with the biological expertise to make important discoveries also have the analysis tools necessary to support their progress.

BIBLIOGRAPHY

Adey, A., Kitzman, J. O., Burton, J. N., Daza, R., Kumar, A., Christiansen, L., Ronaghi, M., Amini, S., Gunderson, K. L., Steemers, F. J. and Shendure, J. (2014). *In vitro*, long-range sequence information for *de novo* genome assembly via transposase contiguity. Genome Research *24*, 2041–2049.

Afgan, E., Baker, D., Batut, B., van den Beek, M., Bouvier, D., Čech, M., Chilton, J., Clements, D., Coraor, N., Grüning, B. A., Guerler, A., Hillman-Jackson, J., Hiltemann, S., Jalili, V., Rasche, H., Soranzo, N., Goecks, J., Taylor, J., Nekrutenko, A. and Blankenberg, D. (2018). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Research *46*, W537–W544.

Amini, S., Pushkarev, D., Christiansen, L., Kostem, E., Royce, T., Turk, C., Pignatelli, N., Adey, A., Kitzman, J. O., Vijayan, K., Ronaghi, M., Shendure, J., Gunderson, K. L. and Steemers, F. J. (2014). Haplotype-resolved whole-genome sequencing by contiguity-preserving transposition and combinatorial indexing. Nature Genetics *46*, 1343–1349.

Azizi, E., Prabhakaran, S., Carr, A. and Pe'er, D. (2017). Bayesian inference for single-cell clustering and imputing. Genomics and Computational Biology *3*, 46.

Bacher, R. and Kendziorski, C. (2016). Design and computational analysis of single-cell RNA-sequencing experiments. Genome Biology *17*, 63.

Baglama, J., Reichel, L. and Lewis, B. W. (2018). irlba: Fast truncated singular value decomposition and principal components analysis for large dense and sparse matrices. R package version 2.3.2.

Bard, J., Rhee, S. Y. and Ashburner, M. (2005). An ontology for cell types. Genome Biology *6*, R21.

Benezra, R., Davis, R. L., Lockshon, D., Turner, D. L. and Weintraub, H. (1990). The protein Id: a negative regulator of helix-loop-helix DNA binding proteins. Cell *61*, 49–59.

Berkes, C. A., Bergstrom, D. A., Penn, B. H., Seaver, K. J., Knoepfler, P. S. and Tapscott, S. J. (2004). Pbx marks genes for activation by MyoD indicating a role for a homeodomain protein in establishing myogenic potential. Molecular Cell *14*, 465–477.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D. and Li, S. (2018). FNN: fast nearest neighbor search algorithms and applications. R package version 1.1.2.2.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment *2008*, P10008.

Boyle, A. P., Davis, S., Shulha, H. P., Meltzer, P., Margulies, E. H., Weng, Z., Furey, T. S. and Crawford, G. E. (2008). High-resolution mapping and characterization of open chromatin across the genome. Cell *132*, 311–322.

Breiman, L. (1996). Bagging predictors. Machine Learning *24*, 123–140.

Brennecke, P., Anders, S., Kim, J. K., Kołodziejczyk, A. A., Zhang, X., Proserpio, V., Baying, B., Benes, V., Teichmann, S. A., Marioni, J. C. and Heisler, M. G. (2013). Accounting for technical noise in single-cell RNA-seq experiments. Nature Methods *10*, 1093–1095.

Budden, D. M., Hurley, D. G. and Crampin, E. J. (2014). Predictive modelling of gene expression from transcriptional regulatory elements. Briefings in Bioinformatics *16*, 616–628.

Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. and Greenleaf, W. J. (2013). Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. Nature Methods *10*, 1213–1218.

Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., Chang, H. Y. and Greenleaf, W. J. (2015). Single-cell chromatin accessibility reveals principles of regulatory variation. Nature *523*, 486–490.

Buettner, F., Natarajan, K. N., Casale, F. P., Proserpio, V., Scialdone, A., Theis, F. J., Teichmann, S. A., Marioni, J. C. and Stegle, O. (2015). Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. Nature Biotechnology *33*, 155–160.

Cairns, J., Freire-Pritchett, P., Wingett, S. W., Várnai, C., Dimond, A., Plagnol, V., Zerbino, D., Schoenfelder, S., Javierre, B.-M., Osborne, C., Fraser, P. and Spivakov, M. (2016). CHiCAGO: robust detection of DNA looping interactions in capture Hi-C data. Genome Biology *17*, 127.

Cao, J., Packer, J. S., Ramani, V., Cusanovich, D. A., Huynh, C., Daza, R., Qiu, X., Lee, C., Furlan, S. N., Steemers, F. J., Adey, A., Waterston, R. H., Trapnell, C. and Shendure, J. (2017). Comprehensive single-cell transcriptional profiling of a multicellular organism. Science *357*, 661–667.

Cao, Y., Yao, Z., Sarkar, D., Lawrence, M., Sanchez, G. J., Parker, M. H., MacQuarrie, K. L., Davison, J., Morgan, M. T., Ruzzo, W. L., Gentleman, R. C. and Tapscott, S. J. (2010). Genome-wide MyoD binding in skeletal muscle cells: a potential for broad cellular reprogramming. Developmental Cell *18*, 662–674.

Carithers, L. J., Ardlie, K., Barcus, M., Branton, P. A., Britton, A., Buia, S. A., Compton, C. C., DeLuca, D. S., Peter-Demchok, J., Gelfand, E. T., Guan, P., Korzeniewski, G. E., Lockhart, N. C., Rabiner, C. A., Rao, A. K., Robinson, K. L., Roche, N. V., Sawyer, S. J., Segrè, A. V., Shive, C. E., Smith, A. M., Sobin, L. H., Undale, A. H., Valentino, K. M., Vaught, J., Young, T. R. and and, H. M. M. (2015). A novel approach to high-quality postmortem tissue procurement: The GTEx Project. Biopreservation and Biobanking *13*, 311–319.

Clevers, H., Rafelski, S., Elowitz, M., Klein, A., Shendure, J., Trapnell, C., Lein, E., Lundberg, E., Uhlen, M., Martinez-Arias, A. et al. (2017). What is your conceptual definition of "cell type" in the context of a mature organism? Cell Systems *4*, 255–259.

Crawford, G. E. (2006). Genome-wide mapping of DNase hypersensitive sites using massively parallel signature sequencing (MPSS). Genome Research *16*, 123–131.

Crawford, G. E., Davis, S., Scacheri, P. C., Renaud, G., Halawi, M. J., Erdos, M. R., Green, R., Meltzer, P. S., Wolfsberg, T. G. and Collins, F. S. (2006). DNase-chip: a high-resolution method to identify DNase I hypersensitive sites using tiled microarrays. Nature Methods *3*, 503–509.

Cusanovich, D. A., Daza, R., Adey, A., Pliner, H. A., Christiansen, L., Gunderson, K. L., Steemers, F. J., Trapnell, C. and Shendure, J. (2015). Multiplex single cell profiling of chromatin accessibility by combinatorial cellular indexing. Science *348*, 910–914.

Cusanovich, D. A., Hill, A. J., Aghamirzaie, D., Daza, R. M., Pliner, H. A., Berletch, J. B., Filippova, G. N., Huang, X., Christiansen, L., DeWitt, W. S., Lee, C., Regalado, S. G., Read, D. F., Steemers, F. J., Disteche, C. M., Trapnell, C. and Shendure, J. (2018a). A single-cell atlas of *in vivo* mammalian chromatin accessibility. Cell *174*, 1309–1324.

Cusanovich, D. A., Reddington, J. P., Garfield, D. A., Daza, R. M., Aghamirzaie, D., Marco-Ferreres, R., Pliner, H. A., Christiansen, L., Qiu, X., Steemers, F. J., Trapnell, C., Shendure, J. and Furlong, E. E. M. (2018b). The *cis*-regulatory dynamics of embryonic development at single-cell resolution. Nature *555*, 538–542.

de la Serna, I. L., Ohkawa, Y., Berkes, C. A., Bergstrom, D. A., Dacwag, C. S., Tapscott, S. J. and Imbalzano, A. N. (2005). MyoD targets chromatin remodeling complexes to the myogenin locus prior to forming a stable DNA-bound complex. Molecular and Cellular Biology *25*, 3997–4009.

de Laat, W. and Grosveld, F. (2003). Spatial organization of gene expression: the active chromatin hub. Chromosome Research *11*, 447–459.

Dekker, J., Marti-Renom, M. A. and Mirny, L. A. (2013). Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nature Reviews Genetics *14*, 390–403.

Diehl, A. D., Meehan, T. F., Bradford, Y. M., Brush, M. H., Dahdul, W. M., Dougall, D. S., He, Y., Osumi-Sutherland, D., Ruttenberg, A., Sarntivijai, S., Van Slyke, C. E., Vasilevsky, N. A.,

Haendel, M. A., Blake, J. A. and Mungall, C. J. (2016). The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. Journal of Biomedical Semantics *7*, 44.

Dilworth, F. J., Seaver, K. J., Fishburn, A. L., Htet, S. L. and Tapscott, S. J. (2004). *In vitro* transcription system delineates the distinct roles of the coactivators pCAF and p300 during MyoD/E47-dependent transactivation. Proceedings of the National Academy of Sciences *101*, 11593–11598.

Dostie, J., Richmond, T. A., Arnaout, R. A., Selzer, R. R., Lee, W. L., Honan, T. A., Rubio, E. D., Krumm, A., Lamb, J., Nusbaum, C., Green, R. D. and Dekker, J. (2006). Chromosome Conformation Capture Carbon Copy (5C): A massively parallel solution for mapping interactions between genomic elements. Genome Research *16*, 1299–1309.

Felsenfeld, G., Boyes, J., Chung, J., Clark, D. and Studitsky, V. (1996). Chromatin structure and gene expression. Proceedings of the National Academy of Sciences *93*, 9384–9388.

Fong, A. P., Yao, Z., Zhong, J. W., Johnson, N. M., Farr, G. H., Maves, L. and Tapscott, S. J. (2015). Conversion of MyoD to a neurogenic factor: binding site specificity determines lineage. Cell Reports *10*, 1937–1946.

Fortin, J.-P. and Hansen, K. D. (2015). Reconstructing A/B compartments as revealed by Hi-C using long-range correlations in epigenetic data. Genome Biology *16*, 180.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics *9*, 432–441.

Goldberg, A. V. and Tarjan, R. E. (1986). A new approach to the maximum flow problem. In proceedings of the eighteenth annual ACM symposium on theory of computing - STOC '86 ACM Press.

Gong, W., Kwak, I.-Y., Pota, P., Koyano-Nakagawa, N. and Garry, D. J. (2018). DrImpute: imputing dropout events in single cell RNA sequencing data. BMC Bioinformatics *19*.

Grant, C. E., Bailey, T. L. and Noble, W. S. (2011). FIMO: scanning for occurrences of a given motif. Bioinformatics *27*, 1017–1018.

Gross, D. S. and Garrard, W. T. (1988). Nuclease hypersensitive sites in chromatin. Annual Review of Biochemistry *57*, 159–197.

Grün, D., Kester, L. and van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. Nature Methods *11*, 637–640.

Gusev, A., Lee, S. H., Trynka, G., Finucane, H., Vilhjálmsson, B. J., Xu, H., Zang, C., Ripke, S., Bulik-Sullivan, B., Stahl, E., Schizophrenia Working Group of the Psychiatric Genomics Consortium, SWE-SCZ Consortium, Kähler, A. K., Hultman, C. M., Purcell, S. M., McCarroll,

S. A., Daly, M., Pasaniuc, B., Sullivan, P. F., Neale, B. M., Wray, N. R., Raychaudhuri, S., Price, A. L., Schizophrenia Working Group of the Psychiatric Genomics Consortium and SWE-SCZ Consortium (2014). Partitioning heritability of regulatory and cell-type-specific variants across 11 common diseases. American Journal of Human Genetics *95*, 535–552.

Han, M. and Grunstein, M. (1988). Nucleosome loss activates yeast downstream promoters *in vivo*. Cell *55*, 1137–1145.

Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F., Huang, D., Xu, Y., Huang, W., Jiang, M., Jiang, X., Mao, J., Chen, Y., Lu, C., Xie, J., Fang, Q., Wang, Y., Yue, R., Li, T., Huang, H., Orkin, S. H., Yuan, G.-C., Chen, M. and Guo, G. (2018). Mapping the mouse cell atlas by Microwell-Seq. Cell *173*, 1307.

Hwang, B., Lee, J. H. and Bang, D. (2018). Single-cell RNA sequencing technologies and bioinformatics pipelines. Experimental & Molecular Medicine *50*.

John, S., Sabo, P. J., Thurman, R. E., Sung, M.-H., Biddie, S. C., Johnson, T. A., Hager, G. L. and Stamatoyannopoulos, J. A. (2011). Chromatin accessibility pre-determines glucocorticoid receptor binding patterns. Nature Genetics *43*, 264–268.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation *28*, 11–21.

Kayne, P. S., Kim, U.-J., Han, M., Mullen, J. R., Yoshizaki, F. and Grunstein, M. (1988). Extremely conserved histone H4 N terminus is dispensable for growth but essential for repressing the silent mating loci in yeast. Cell *55*, 27–39.

Kharchenko, P. V., Silberstein, L. and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. Nature Methods *11*, 740–742.

Killick, R., Fearnhead, P. and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association *107*, 1590–1598.

Kim, J. and Marioni, J. C. (2013). Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. Genome Biology *14*, R7.

Knezetic, J. A. and Luse, D. S. (1986). The presence of nucleosomes on a DNA template prevents initiation by RNA polymerase II in vitro. Cell *45*, 95–104.

Knoepfler, P. S., Bergstrom, D. A., Uetsuki, T., Dac-Korytko, I., Sun, Y. H., Wright, W. E., Tapscott, S. J. and Kamps, M. P. (1999). A conserved motif N-terminal to the DNA-binding domains of myogenic bHLH transcription factors mediates cooperative DNA binding with Pbx-Meis1/Prep1. Nucleic Acids Research *27*, 3752–3761.

Kornberg, R. D. (1974). Chromatin structure: a repeating unit of histones and DNA. Science *184*, 868–871.

Krom, Y. D., Dumonceaux, J., Mamchaoui, K., den Hamer, B., Mariot, V., Negroni, E., Geng, L. N., Martin, N., Tawil, R., Tapscott, S. J., van Engelen, B. G. M., Mouly, V., Butler-Browne, G. S. and van der Maarel, S. M. (2012). Generation of isogenic D4Z4 contracted and noncontracted immortal muscle cell clones from a mosaic patient: a cellular model for FSHD. American Journal of Pathology *181*, 1387–1401.

Kundaje, A., , Meuleman, W., Ernst, J., Bilenky, M., Yen, A., Heravi-Moussavi, A., Kheradpour, P., Zhang, Z., Wang, J., Ziller, M. J., Amin, V., Whitaker, J. W., Schultz, M. D., Ward, L. D., Sarkar, A., Quon, G., Sandstrom, R. S., Eaton, M. L., Wu, Y.-C., Pfenning, A. R., Wang, X., Claussnitzer, M., Liu, Y., Coarfa, C., Harris, R. A., Shoresh, N., Epstein, C. B., Gjoneska, E., Leung, D., Xie, W., Hawkins, R. D., Lister, R., Hong, C., Gascard, P., Mungall, A. J., Moore, R., Chuah, E., Tam, A., Canfield, T. K., Hansen, R. S., Kaul, R., Sabo, P. J., Bansal, M. S., Carles, A., Dixon, J. R., Farh, K.-H., Feizi, S., Karlic, R., Kim, A.-R., Kulkarni, A., Li, D., Lowdon, R., Elliott, G., Mercer, T. R., Neph, S. J., Onuchic, V., Polak, P., Rajagopal, N., Ray, P., Sallari, R. C., Siebenthall, K. T., Sinnott-Armstrong, N. A., Stevens, M., Thurman, R. E., Wu, J., Zhang, B., Zhou, X., Beaudet, A. E., Boyer, L. A., Jager, P. L. D., Farnham, P. J., Fisher, S. J., Haussler, D., Jones, S. J. M., Li, W., Marra, M. A., McManus, M. T., Sunyaev, S., Thomson, J. A., Tlsty, T. D., Tsai, L.-H., Wang, W., Waterland, R. A., Zhang, M. Q., Chadwick, L. H., Bernstein, B. E., Costello, J. F., Ecker, J. R., Hirst, M., Meissner, A., Milosavljevic, A., Ren, B., Stamatoyannopoulos, J. A., Wang, T. and Kellis, M. (2015). Integrative analysis of 111 reference human epigenomes. Nature *518*, 317–330.

Lambrechts, D., Wauters, E., Boeckx, B., Aibar, S., Nittner, D., Burton, O., Bassez, A., Decaluwé, H., Pircher, A., Van den Eynde, K., Weynand, B., Verbeken, E., De Leyn, P., Liston, A., Vansteenkiste, J., Carmeliet, P., Aerts, S. and Thienpont, B. (2018). Phenotype molding of stromal cells in the lung tumor microenvironment. Nature Medicine *24*, 1277–1289.

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

Lara-Astiaso, D., Weiner, A., Lorenzo-Vivas, E., Zaretsky, I., Jaitin, D. A., David, E., Keren-Shaul, H., Mildner, A., Winter, D., Jung, S., Friedman, N. and Amit, I. (2014). Chromatin state dynamics during blood formation. Science *345*, 943–949.

Leng, N., Chu, L.-F., Barry, C., Li, Y., Choi, J., Li, X., Jiang, P., Stewart, R. M., Thomson, J. A. and Kendziorski, C. (2015). Oscope identifies oscillatory genes in unsynchronized single-cell RNA-seq experiments. Nature Methods *12*, 947–950.

Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., Amir, E.-A. D., Tadmor, M. D., Litvin, O., Fienberg, H. G., Jager, A., Zunder, E. R., Finck, R., Gedman, A. L., Radtke, I., Downing, J. R., Pe'er, D. and Nolan, G. P. (2015). Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. Cell *162*, 184–197.

Levings, P. P. and Bungert, J. (2002). The human β-globin locus control region. European Journal of Biochemistry *269*, 1589–1599.

Li, B., Carey, M. and Workman, J. L. (2007). The role of chromatin during transcription. Cell *128*, 707–719.

Li, W. V. and Li, J. J. (2018). An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nature Communications *9*.

Libbrecht, M. W., Rodriguez, O., Weng, Z., Hoffman, M., Bilmes, J. A. and Noble, W. S. (2018). A unified encyclopedia of human functional DNA elements through fully automated annotation of 164 human cell types. bioRxiv .

Lister, R., O'Malley, R. C., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H. and Ecker, J. R. (2008). Highly integrated single-base resolution maps of the epigenome in *Arabidopsis*. Cell *133*, 523–536.

Lorch, Y., LaPointe, J. W. and Kornberg, R. D. (1987). Nucleosomes inhibit the initiation of transcription but allow chain elongation with the displacement of histones. Cell *49*, 203–210.

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

Lowe, R., Shirley, N., Bleackley, M., Dolan, S. and Shafee, T. (2017). Transcriptomics technologies. PLOS Computational Biology *13*, 1–23.

MacQuarrie, K. L., Yao, Z., Fong, A. P., Diede, S. J., Rudzinski, E. R., Hawkins, D. S. and Tapscott, S. J. (2013). Comparison of genome-wide binding of MyoD in normal human myogenic cells and rhabdomyosarcomas identifies regional and local suppression of promyogenic transcription factors. Molecular and Cellular Biology. *33*, 773–784.

Mathelier, A., Fornes, O., Arenillas, D. J., Chen, C.-Y., Denay, G., Lee, J., Shi, W., Shyr, C., Tan, G., Worsley-Hunt, R., Zhang, A. W., Parcy, F., Lenhard, B., Sandelin, A. and Wasserman, W. W. (2016). JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. Nucleic Acids Research *44*, D110–D115.

Maurano, M. T., Humbert, R., Rynes, E., Thurman, R. E., Haugen, E., Wang, H., Reynolds, A. P., Sandstrom, R., Qu, H., Brody, J., Shafer, A., Neri, F., Lee, K., Kutyavin, T., Stehling-Sun, S., Johnson, A. K., Canfield, T. K., Giste, E., Diegel, M., Bates, D., Hansen, R. S., Neph, S., Sabo, P. J., Heimfeld, S., Raubitschek, A., Ziegler, S., Cotsapas, C., Sotoodehnia, N., Glass, I., Sunyaev, S. R., Kaul, R. and Stamatoyannopoulos, J. A. (2012). Systematic localization of common disease-associated variation in regulatory DNA. Science *337*, 1190–1195.

Maves, L., Waskiewicz, A. J., Paul, B., Cao, Y., Tyler, A., Moens, C. B. and Tapscott, S. J. (2007). Pbx homeodomain proteins direct Myod activity to promote fast-muscle differentiation. Development *134*, 3371–3382.

Merico, D., Isserlin, R., Stueker, O., Emili, A. and Bader, G. D. (2010). Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. PLoS One *5*, e13984.

Merkin, J., Russell, C., Chen, P. and Burge, C. B. (2012). Evolutionary dynamics of gene and isoform regulation in Mammalian tissues. Science *338*, 1593–1599.

Mifsud, B., Tavares-Cadete, F., Young, A. N., Sugar, R., Schoenfelder, S., Ferreira, L., Wingett, S. W., Andrews, S., Grey, W., Ewels, P. A., Herman, B., Happe, S., Higgs, A., LeProust, E., Follows, G. A., Fraser, P., Luscombe, N. M. and Osborne, C. S. (2015). Mapping long-range promoter contacts in human cells with high-resolution capture Hi-C. Nature Genetics *47*, 598–606.

Molkentin, J. D., Black, B. L., Martin, J. F. and Olson, E. N. (1995). Cooperative activation of muscle gene expression by MEF2 and myogenic bHLH proteins. Cell *83*, 1125–1136.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L. and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nature Methods *5*, 621–628.

Mosser, D. M. and Edwards, J. P. (2008). Exploring the full spectrum of macrophage activation. Nature Reviews Immunology *8*, 958–969.

Mulqueen, R. M., Pokholok, D., Norberg, S. J., Torkenczy, K. A., Fields, A. J., Sun, D., Sinnamon, J. R., Shendure, J., Trapnell, C., O'Roak, B. J., Xia, Z., Steemers, F. J. and Adey, A. C. (2018). Highly scalable generation of DNA methylation profiles in single cells. Nature Biotechnology *36*, 428–431.

Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M. and Snyder, M. (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. Science *320*, 1344–1349.

Noordermeer, D. and de Laat, W. (2008). Joining the loops: β-Globin gene regulation. IUBMB Life *60*, 824–833.

Olins, A., Zwerger, M., Herrmann, H., Zentgraf, H., Simon, A., Monestier, M. and Olins, D. (2008). The human granulocyte nucleus: Unusual nuclear envelope and heterochromatin composition. European Journal of Cell Biology *87*, 279–290.

Pagès, H., Carlson, M., Falcon, S. and Li, N. (2018). AnnotationDbi: Annotation Database Interface. R package version 1.44.0.

Philipot, O., Joliot, V., Ait-Mohamed, O., Pellentz, C., Robin, P., Fritsch, L. and Ait-Si-Ali, S. (2010). The core binding factor CBF negatively regulates skeletal muscle terminal differentiation. PLoS One *5*, e9425.

Pliner, H. A., Packer, J. S., McFaline-Figueroa, J. L., Cusanovich, D. A., Daza, R. M., Aghamirzaie, D., Srivatsan, S., Qiu, X., Jackson, D., Minkina, A., Adey, A. C., Steemers, F. J., Shendure, J. and Trapnell, C. (2018). Cicero predicts *cis*-regulatory DNA interactions from single-cell chromatin accessibility data. Molecular Cell *71*, 858 – 871.

Puri, P. L., Sartorelli, V., Yang, X. J., Hamamori, Y., Ogryzko, V. V., Howard, B. H., Kedes, L., Wang, J. Y., Graessmann, A., Nakatani, Y. and Levrero, M. (1997). Differential roles of p300 and PCAF acetyltransferases in muscle differentiation. Molecular Cell *1*, 35–45.

Qiu, X., Hill, A., Packer, J., Lin, D., Ma, Y.-A. and Trapnell, C. (2017a). Single-cell mRNA quantification and differential analysis with Census. Nature Methods *14*, 309–315.

Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H. A. and Trapnell, C. (2017b). Reversed graph embedding resolves complex single-cell trajectories. Nature Methods *14*, 979–982.

Ramani, V., Deng, X., Qiu, R., Gunderson, K. L., Steemers, F. J., Disteche, C. M., Noble, W. S., Duan, Z. and Shendure, J. (2017). Massively multiplex single-cell Hi-C. Nature Methods *14*, 263–266.

Rao, S. S. P., Huntley, M. H., Durand, N. C., Stamenova, E. K., Bochkov, I. D., Robinson, J. T., Sanborn, A. L., Machol, I., Omer, A. D., Lander, E. S. and Aiden, E. L. (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell *159*, 1665–1680.

Rosenberg, A. B., Roco, C. M., Muscat, R. A., Kuchina, A., Sample, P., Yao, Z., Graybuck, L. T., Peeler, D. J., Mukherjee, S., Chen, W., Pun, S. H., Sellers, D. L., Tasic, B. and Seelig, G. (2018). Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science *360*, 176–182.

Sanborn, A. L., Rao, S. S. P., Huang, S.-C., Durand, N. C., Huntley, M. H., Jewett, A. I., Bochkov, I. D., Chinnappan, D., Cutkosky, A., Li, J., Geeting, K. P., Gnirke, A., Melnikov, A., McKenna, D., Stamenova, E. K., Lander, E. S. and Aiden, E. L. (2015). Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. Proceedings of the National Academy of Sciences *112*, E6456–65.

Sartorelli, V., Huang, J., Hamamori, Y. and Kedes, L. (1997). Molecular mechanisms of myogenic coactivation by p300: direct interaction with the activation domain of MyoD and with the MADS box of MEF2C. Molecular and Cellular Biology *17*, 1010–1026.

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. and Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. Nature Biotechnology *33*, 495–502.

Schwartzentruber, J., Korshunov, A., Liu, X.-Y., Jones, D. T. W., Pfaff, E., Jacob, K., Sturm, D., Fontebasso, A. M., Quang, D.-A. K., Tönjes, M., Hovestadt, V., Albrecht, S., Kool, M., Nantel, A., Konermann, C., Lindroth, A., Jäger, N., Rausch, T., Ryzhova, M., Korbel, J. O., Hielscher, T.,

Hauser, P., Garami, M., Klekner, A., Bognar, L., Ebinger, M., Schuhmann, M. U., Scheurlen, W., Pekrun, A., Frühwald, M. C., Roggendorf, W., Kramm, C., Dürken, M., Atkinson, J., Lepage, P., Montpetit, A., Zakrzewska, M., Zakrzewski, K., Liberski, P. P., Dong, Z., Siegel, P., Kulozik, A. E., Zapatka, M., Guha, A., Malkin, D., Felsberg, J., Reifenberger, G., von Deimling, A., Ichimura, K., Collins, V. P., Witt, H., Milde, T., Witt, O., Zhang, C., Castelo-Branco, P., Lichter, P., Faury, D., Tabori, U., Plass, C., Majewski, J., Pfister, S. M. and Jabado, N. (2012). Driver mutations in histone H3.3 and chromatin remodelling genes in paediatric glioblastoma. Nature *482*, 226–231.

Scott, W. A. and Wigmore, D. J. (1978). Sites in simian virus 40 chromatin which are preferentially cleaved by endonucleases. Cell *15*, 1511–1518.

Scrucca, L., Fop, M., Murphy, T. B. and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. The R Journal *8*, 205–233.

Serra, C., Palacios, D., Mozzetta, C., Forcales, S. V., Morantte, I., Ripani, M., Jones, D. R., Du, K., Jhala, U. S., Simone, C. and Puri, P. L. (2007). Functional interdependence at the chromatin level between the MKK6/p38 and IGF1/PI3K/AKT pathways during muscle differentiation. Molecular Cell *28*, 200–213.

Simone, C., Forcales, S. V., Hill, D. A., Imbalzano, A. N., Latella, L. and Puri, P. L. (2004). p38 pathway targets SWI-SNF chromatin-remodeling complex to muscle-specific loci. Nature Genetics *36*, 738–743.

Simpson, E. H. (1951). The interpretation of interaction in contingency tables. Journal of the Royal Statistical Society. Series B (Methodological) *13*, 238–241.

Smith, B., , Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P. L. and Lewis, S. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology *25*, 1251–1255.

Snider, L., Geng, L. N., Lemmers, R. J. L. F., Kyba, M., Ware, C. B., Nelson, A. M., Tawil, R., Filippova, G. N., van der Maarel, S. M., Tapscott, S. J. and Miller, D. G. (2010). Facioscapulohumeral dystrophy: incomplete suppression of a retrotransposed gene. PLoS Genetics *6*, e1001181.

Stegle, O., Teichmann, S. A. and Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. Nature Reviews Genetics *16*, 133–145.

Svensson, V., Vento-Tormo, R. and Teichmann, S. A. (2018). Exponential scaling of single-cell RNA-seq in the past decade. Nature Protocols *13*, 599–604.

Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group and Principal investigators (2018). Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature *562*, 367–372.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K. and Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. Nature Methods *6*, 377–382.

Tang, Z., Luo, O. J., Li, X., Zheng, M., Zhu, J. J., Szalaj, P., Trzaskoma, P., Magalska, A., Wlodarczyk, J., Ruszczycki, B., Michalski, P., Piecuch, E., Wang, P., Wang, D., Tian, S. Z., Penrad-Mobayed, M., Sachs, L. M., Ruan, X., Wei, C.-L., Liu, E. T., Wilczynski, G. M., Plewczynski, D., Li, G. and Ruan, Y. (2015). CTCF-mediated human 3D genome architecture reveals chromatin topology for transcription. Cell *163*, 1611–1627.

The ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. Nature *489*, 57–74.

Thurman, R. E., Rynes, E., Humbert, R., Vierstra, J., Maurano, M. T., Haugen, E., Sheffield, N. C., Stergachis, A. B., Wang, H., Vernot, B., Garg, K., John, S., Sandstrom, R., Bates, D., Boatman, L., Canfield, T. K., Diegel, M., Dunn, D., Ebersol, A. K., Frum, T., Giste, E., Johnson, A. K., Johnson, E. M., Kutyavin, T., Lajoie, B., Lee, B.-K., Lee, K., London, D., Lotakis, D., Neph, S., Neri, F., Nguyen, E. D., Qu, H., Reynolds, A. P., Roach, V., Safi, A., Sanchez, M. E., Sanyal, A., Shafer, A., Simon, J. M., Song, L., Vong, S., Weaver, M., Yan, Y., Zhang, Z., Zhang, Z., Lenhard, B., Tewari, M., Dorschner, M. O., Hansen, R. S., Navas, P. A., Stamatoyannopoulos, G., Iyer, V. R., Lieb, J. D., Sunyaev, S. R., Akey, J. M., Sabo, P. J., Kaul, R., Furey, T. S., Dekker, J., Crawford, G. E. and Stamatoyannopoulos, J. A. (2012). The accessible chromatin landscape of the human genome. Nature *489*, 75–82.

Tolhuis, B., Palstra, R.-J., Splinter, E., Grosveld, F. and de Laat, W. (2002). Looping and interaction between hypersensitive sites in the active β-globin locus. Molecular Cell *10*, 1453–1465.

Trapnell, C. (2015). Defining cell types and states with single-cell genomics. Genome Research *25*, 1491–1498.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S. and Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nature Biotechnology. *32*, 381–386.

Vallejos, C. A., Marioni, J. C. and Richardson, S. (2015). BASiCS: Bayesian analysis of single-cell sequencing data. PLoS Computational Biology *11*, e1004333.

van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., Bierie, B., Mazutis, L., Wolf, G., Krishnaswamy, S. and Pe'er,

D. (2018). Recovering gene interactions from single-cell data using data diffusion. Cell *174*, 716–729.

Väremo, L., Nielsen, J. and Nookaew, I. (2013). Enriching the gene set analysis of genome-wide data by incorporating directionality of gene expression and combining statistical hypotheses and methods. Nucleic Acids Research *41*, 4378–4391.

Varshavsky, A., Sundin, O. and Bohn, M. (1978). SV40 viral minichromosome: preferential exposure of the origin of replication as probed by restriction endonucleases. Nucleic Acids Research *5*, 3469–3477.

Vitak, S. A., Torkenczy, K. A., Rosenkrantz, J. L., Fields, A. J., Christiansen, L., Wong, M. H., Carbone, L., Steemers, F. J. and Adey, A. (2017). Sequencing thousands of single-cell genomes with combinatorial indexing. Nature Methods *14*, 302–308.

Weber, R., Schek, H.-J. and Blott, S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In Proceedings of the 24rd International Conference on Very Large Data Bases VLDB '98 pp. 194–205, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Weintraub, H. and Groudine, M. (1976). Chromosomal subunits in active genes have an altered conformation. Science *193*, 848–856.

Yang, M., Büsche, G., Ganser, A. and Li, Z. (2013). Morphology and quantitative composition of hematopoietic cells in murine bone marrow and spleen of healthy subjects. Annals of Hematology *92*, 587–594.

Yoshida, N., Yoshida, S., Koishi, K., Masuda, K. and Nabeshima, Y. (1998). Cell heterogeneity upon myogenic differentiation: down-regulation of MyoD and Myf-5 generates 'reserve cells'. Journal of Cell Science *111*, 769–779.

Zaret, K. S. and Carroll, J. S. (2011). Pioneer transcription factors: establishing competence for gene expression. Genes & Development *25*, 2227–2241.

Zeisel, A., Hochgerner, H., Lönnerberg, P., Johnsson, A., Memic, F., van der Zwan, J., Häring, M., Braun, E., Borm, L. E., La Manno, G., Codeluppi, S., Furlan, A., Lee, K., Skene, N., Harris, K. D., Hjerling-Leffler, J., Arenas, E., Ernfors, P., Marklund, U. and Linnarsson, S. (2018). Molecular Architecture of the Mouse Nervous System. Cell *174*, 999–1014.e22.

Zhang, X., Lan, Y., Xu, J., Quan, F., Zhao, E., Deng, C., Luo, T., Xu, L., Liao, G., Yan, M., Ping, Y., Li, F., Shi, A., Bai, J., Zhao, T., Li, X. and Xiao, Y. (2019). CellMarker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Research *47*, D721–D728.

Zhang, Y., Liu, T., Meyer, C. A., Eeckhoute, J., Johnson, D. S., Bernstein, B. E., Nusbaum, C., Myers, R. M., Brown, M., Li, W. and Liu, X. S. (2008). Model-based analysis of ChIP-Seq (MACS). Genome Biology *9*, R137.

Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J. and Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. Nature Communications *8*, 14049.

Zhu, L., Lei, J., Devlin, B. and Roeder, K. (2018). A unified statistical framework for single cell and bulk RNA sequencing data. The Annals of Applied Statistics *12*, 609–632.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society *67*, 301–320.

Appendix A: CICERO PACKAGE SOURCE AND MANUAL

A.1 SOURCE CODE

Cicero has been released as part of the Bioconductor project. Cicero source code is available at https://github.com/cole-trapnell-lab/cicero-release and a website with full documentation is available at https://cole-trapnell-lab.github.io/cicero-release/.

A.2 MANUAL

Package ‘cicero’

December 6, 2018

Type Package

Title Predict cis-co-accessibility from single-cell chromatin accessibility data

Version 1.1.3

Description Cicero computes putative cis-regulatory maps from single-cell chromatin accessibility data. It also extends monocle 2 for use in chromatin accessibility data.

Depends R (>= 3.5.0),
monocle,
Gviz (>= 1.22.3)

License MIT + file LICENSE

Encoding UTF-8

Imports assertthat (>= 0.2.0),
Biobase (>= 2.37.2),
BiocGenerics (>= 0.23.0),
data.table (>= 1.10.4),
dplyr (>= 0.7.4),
FNN (>= 1.1),
GenomicRanges (>= 1.30.3),
ggplot2 (>= 2.2.1),
glasso (>= 1.8),
grDevices,
igraph (>= 1.1.0),
IRanges (>= 2.10.5),
Matrix (>= 1.2-12),
methods,
parallel,
plyr (>= 1.8.4),
reshape2 (>= 1.4.3),
S4Vectors (>= 0.14.7),
stats,
stringr (>= 1.2.0),
tibble (>= 1.4.2),
VGAM (>= 1.0-5),
utils

RoxygenNote 6.1.0

Suggests AnnotationDbi (>= 1.38.2),
knitr,
rmarkdown,
rtracklayer (>= 1.36.6),
testthat,
vdiffr (>= 0.2.3),
covr

VignetteBuilder knitr

biocViews Sequencing, Clustering, CellBasedAssays, ImmunoOncology
GeneRegulation, GeneTarget, Epigenetics, ATACSeq, SingleCell

`cicero-package` *cicero*

Description

Cicero computes putative cis-regulatory maps from single-cell chromatin accessibility data. It also extends monocle 2 for use in chromatin accessibility data.

Author(s)

Maintainer: Hannah Pliner <hpliner@uw.edu>

Authors:

- Cole Trapnell <coletrap@uw.edu>

`aggregate_by_cell_bin` *Aggregate count CDS by groups of cells*

Description

Aggregates a CDS based on an indicator column in the `pData` table

Usage

```
aggregate_by_cell_bin(cds, group_col)
```

Arguments

`cds`	A CDS object to be aggregated
`group_col`	The name of the column in the `pData` table that indicates the cells assignment to its aggregate bin.

Details

This function takes an input CDS object and collapses cells based on a column in the pData table by summing the values within the cell group.

Value

A count cds aggregated by group_col

Examples

```
data("cicero_data")
#input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
#pData(input_cds)$cell_subtype <- rep(1:10, times=20)
#binned_input_lin <-aggregate_by_cell_bin(input_cds, "cell_subtype")
```

---

aggregate_nearby_peaks

Make an aggregate count cds by collapsing nearby peaks

Description

Make an aggregate count cds by collapsing nearby peaks

Usage

```
aggregate_nearby_peaks(cds, distance = 1000)
```

Arguments

cds	A CellDataSet (CDS) object. For example, output of make_atac_cds
distance	The distance within which peaks should be collapsed

Details

This function takes an input CDS object and collapses features within a given distance by summing the values for the collapsed features. Ranges of features are determined by their feature name, so the feature names must be in the form "chr1:1039013-2309023".

Value

A CDS object with aggregated peaks.

Examples

```
data("cicero_data")
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
agg_cds <- aggregate_nearby_peaks(input_cds, distance = 10000)
```

annotate_cds_by_site *Add feature data columns to fData*

Description

Annotate the sites of your CDS with feature data based on coordinate overlap.

Usage

```
annotate_cds_by_site(cds, feature_data, verbose = FALSE, maxgap = 0,
  all = FALSE, header = FALSE)
```

Arguments

cds	A CDS object.
feature_data	Data frame, or a character path to a file of feature data. If a path, the file should be tab separated. Default assumes no header, if your file has a header, set header = FALSE. For either a data frame or a path, the file should be in bed-like format, with the first 3 columns containing chromosome, start and stop respectively. The remaining columns will be added to the fData table as feature data.
verbose	Logical, should progress messages be printed?
maxgap	The maximum number of base pairs allowed between the peak and the feature for the feature and peak to be considered overlapping. Default = 0 (overlapping). Details in findOverlaps-methods. If maxgap is set to "nearest" then the nearest feature will be assigned regardless of distance.
all	Logical, should all overlapping intervals be reported? If all is FALSE, the largest overlap is reported.
header	Logical, if reading a file, is there a header?

Details

annotate_cds_by_site will add columns to the fData table of a CDS object based on the overlap of peaks with features in a data frame or file. An "overlap" column will be added, along with any columns beyond the three required columns in the feature data. The "overlap" column is the number of base pairs overlapping the fData site. When maxgap is used, the true overlap is still calculated (overlap will be 0 if the two features only overlap because of maxgap) NA means that there was no overlapping feature. If a peak overlaps multiple data intervals and all is FALSE, the largest overlapping interval will be chosen (in a tie, the first entry is taken), otherwise all intervals will be chosen and annotations will be collapsed using a comma as a separator.

Value

A CDS object with updated fData table.

Examples

```
data("cicero_data")
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
feat <- data.frame(chr = c("chr18", "chr18", "chr18", "chr18"),
                   bp1 = c(10000, 10800, 50000, 100000),
                   bp2 = c(10700, 11000, 60000, 110000),
                   type = c("Acetylated", "Methylated", "Acetylated",
                   "Methylated"))
input_cds <- annotate_cds_by_site(input_cds, feat)
```

---

assemble_connections *Combine and reconcile cicero models*

Description

Function which takes the output of generate_cicero_models and assembles the connections into a data frame with cicero co-accessibility scores.

Usage

```
assemble_connections(cicero_model_list, silent = FALSE)
```

Arguments

cicero_model_list	A list of cicero output objects, generally, the output of generate_cicero_models.
silent	Logical, should the function run silently?

Details

This function combines glasso models computed on overlapping windows of the genome. Pairs of sites whose regularized correlation was calculated twice are first checked for qualitative concordance (both zero, positive or negative). If they not concordant, NA is returned. If they are concordant the mean is returned.

Value

A data frame of connections with their cicero co-accessibility scores.

See Also

generate_cicero_models

Examples

```
data("cicero_data")
data("human.hg19.genome")
sample_genome <- subset(human.hg19.genome, V1 == "chr18")
sample_genome$V2[1] <- 100000
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds, reduced_coordinates = tsne_coords)
model_output <- generate_cicero_models(cicero_cds,
                                       distance_parameter = 0.3,
                                       genomic_coords = sample_genome)
cicero_cons <- assemble_connections(model_output)
```

`build_gene_activity_matrix` *Calculate initial Cicero gene activity matrix*

Description

This function calculates the initial Cicero gene activity matrix. After this function, the activity matrix should be normalized with any comparison matrices using the function `normalize_gene_activities`.

Usage

```
build_gene_activity_matrix(input_cds, cicero_cons_info,
  site_weights = NULL, dist_thresh = 250000, coaccess_cutoff = 0.25)
```

Arguments

`input_cds`	Binary sci-ATAC-seq input CDS. The input CDS must have a column in the fData table called "gene" which is the gene name if the site is a promoter, and `NA` if the site is distal.
`cicero_cons_info`	Cicero connections table, generally the output of `run_cicero`. This table is a data frame with three required columns named "Peak1", "Peak2", and "coaccess". Peak1 and Peak2 contain coordinates for the two compared elements, and coaccess contains their Cicero co-accessibility score.
`site_weights`	NULL or an individual weight for each site in input_cds.
`dist_thresh`	The maximum distance in base pairs between pairs of sites to include in the gene activity calculation.
`coaccess_cutoff`	The minimum Cicero co-accessibility score that should be considered connected.

Value

Unnormalized gene activity matrix.

Examples

```
data("cicero_data")
data("human.hg19.genome")
sample_genome <- subset(human.hg19.genome, V1 == "chr18")
sample_genome$V2[1] <- 100000
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- detectGenes(input_cds)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds,
                              reduced_coordinates = tsne_coords)
cons <- run_cicero(cicero_cds, sample_genome, sample_num=2)

data(gene_annotation_sample)
gene_annotation_sub <- gene_annotation_sample[,c(1:3, 8)]
names(gene_annotation_sub)[4] <- "gene"
input_cds <- annotate_cds_by_site(input_cds, gene_annotation_sub)
num_genes <- pData(input_cds)$num_genes_expressed
names(num_genes) <- row.names(pData(input_cds))
unnorm_ga <- build_gene_activity_matrix(input_cds, cons)
```

---

cell_data *Metadata for example cells in cicero_data*

Description

Metadata information for cicero_data

Usage

```
cell_data
```

Format

A data frame with 200 rows and 2 variables:

timepoint Time at cell collection

cell Cell barcode

cicero_data *Example single-cell chromatin accessibility data*

Description

A dataset containing a subset of a single-cell ATAC-seq dataset collected on Human Skeletal Muscle Myoblasts. Only includes data from chromosome 18.

Usage

```
cicero_data
```

Format

A data frame with 35137 rows and 3 variables:

Peak Peak information

Cell Cell ID

Count Reads per cell per peak

compare_connections *Compare Cicero connections to other datasets*

Description

Compare two sets of connections and return a vector of logicals for whether connections in one are present in the other.

Usage

```
compare_connections(conns1, conns2, maxgap = 0)
```

Arguments

conns1	A data frame of Cicero connections, like those output from assemble_connections. The first two columns must be the coordinates of peaks that are connected.
conns2	A data frame of connections to be searched for overlap. The first two columns must be coordinates of genomic sites that are connected.
maxgap	The number of base pairs between peaks allowed to be called overlapping. See findOverlaps-methods in the IRanges package for further description.

Value

A vector of logicals of whether the Cicero pair is present in the alternate dataset.

Examples

```
## Not run:
cons$in_dataset <- compare_connections(conns, alt_data)

## End(Not run)
```

`df_for_coords` *Construct a data frame of coordinate info from coordinate strings*

Description

Construct a data frame of coordinate info from coordinate strings

Usage

```
df_for_coords(coord_strings)
```

Arguments

`coord_strings`	A list of coordinate strings (each like "chr1:500000-1000000")

Details

Coordinate strings consist of three pieces of information: chromosome, start, and stop. These pieces of information can be separated by the characters ":", "_", or "-". Commas will be removed, not used as separators (ex: "chr18:8,575,097-8,839,855" is ok).

Value

data.frame with three columns, chromosome, starting base pair and ending base pair

Examples

```
df_for_coords(c("chr1:2,039-30,239", "chrX:28884:101293"))
```

`estimate_distance_parameter` *Calculate distance penalty parameter*

Description

Function to calculate distance penalty parameter (`distance_parameter`) for random genomic windows. Used to choose `distance_parameter` to pass to `generate_cicero_models`.

Usage

```
estimate_distance_parameter(cds, window = 5e+05, maxit = 100,
  s = 0.75, sample_num = 100, distance_constraint = 250000,
  distance_parameter_convergence = 1e-22, max_elements = 200,
  genomic_coords = cicero::human.hg19.genome)
```

Arguments

`cds`	A cicero CDS object generated using `make_cicero_cds`.
`window`	Size of the genomic window to query, in base pairs.
`maxit`	Maximum number of iterations for distance_parameter estimation.
`s`	Power law value. See details for more information.
`sample_num`	Number of random windows to calculate `distance_parameter` for.
`distance_constraint`	Maximum distance of expected connections. Must be smaller than `window`.
`distance_parameter_convergence`	Convergence step size for `distance_parameter` calculation.
`max_elements`	Maximum number of elements per window allowed. Prevents very large models from slowing performance.
`genomic_coords`	Either a data frame or a path (character) to a file with chromosome lengths. The file should have two columns, the first is the chromosome name (ex. "chr1") and the second is the chromosome length in base pairs. See `data(human.hg19.genome)` for an example. If a file, should be tab-separated and without header.

Details

The purpose of this function is to calculate the distance scaling parameter used to adjust the distance-based penalty function used in Cicero's model calculation. The scaling parameter, in combination with the power law value `s` determines the distance-based penalty.

This function chooses random windows of the genome and calculates a `distance_parameter`. The function returns a vector of values calculated on these random windows. We recommend using the mean value of this vector moving forward with Cicero analysis.

The function works by finding the minimum distance scaling parameter such that no more than 5 distance_constraint have non-zero entries after graphical lasso regularization and such that fewer than 80 nonzero.

If the chosen random window has fewer than 2 or greater than max_elements sites, the window is skipped. In addition, the random window will be skipped if there are insufficient long-range comparisons (see below) to be made. The max_elements parameter exist to prevent very dense windows from slowing the calculation. If you expect that your data may regularly have this many sites in a window, you will need to raise this parameter.

Calculating the distance_parameter in a sample window requires peaks in that window that are at a distance greater than the distance_constraint parameter. If there are not enough examples at high distance, the function will return the warning "Warning: could not calculate sample_num distance_parameters -see documentation details" Generally, this means your window parameter needs to be larger or your distance_constraint parameter needs to be smaller. A less likely possibility is that your max_elements parameter needs to be larger. This would occur if your data is particularly dense.

The parameter s is a constant that captures the power-law distribution of contact frequencies between different locations in the genome as a function of their linear distance. For a complete discussion of the various polymer models of DNA packed into the nucleus and of justifiable values for s, we refer readers to (Dekker et al., 2013) for a discussion of justifiable values for s. We use a value of 0.75 by default in Cicero, which corresponds to the "tension globule" polymer model of DNA (Sanborn et al., 2015). This parameter must be the same as the s parameter for generate_cicero_models.

Further details are available in the publication that accompanies this package. Run citation("cicero") for publication details.

Value

A list of results of length sample_num. List members are numeric distance_parameter values.

References

- Dekker, J., Marti-Renom, M.A., and Mirny, L.A. (2013). Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nat. Rev. Genet. 14, 390–403.
- Sanborn, A.L., Rao, S.S.P., Huang, S.-C., Durand, N.C., Huntley, M.H., Jewett, A.I., Bochkov, I.D., Chinnappan, D., Cutkosky, A., Li, J., et al. (2015). Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. Proc. Natl. Acad. Sci. U. S. A. 112, E6456–E6465.

See Also

generate_cicero_models

Examples

```
data("cicero_data")
data("human.hg19.genome")
sample_genome <- subset(human.hg19.genome, V1 == "chr18")
```

```
sample_genome$V2[1] <- 100000
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds, reduced_coordinates = tsne_coords)
distance_parameters <- estimate_distance_parameter(cicero_cds,
                                                   sample_num=5,
                                                   genomic_coords = sample_genome)
```

`find_overlapping_ccans` *Find CCANs that overlap each other in genomic coordinates*

Description

Find CCANs that overlap each other in genomic coordinates

Usage

```
find_overlapping_ccans(ccan_assignments, min_overlap = 1)
```

Arguments

`ccan_assignments`	A data frame where the first column is the peak and the second is the CCAN assignment. For example, output of `generate_ccans`.
`min_overlap`	The minimum base pair overlap to count as overlapping.

Value

A data frame with two columns, CCAN1 and CCAN2. CCANs in this list are overlapping. The data frame is reciprocal (if CCAN 2 overlaps CCAN 1, there will be two rows, 1,2 and 2,1).

Examples

```
ccan_df <- data.frame(peak = c("chr18_1408345_1408845", "chr18_1779830_1780330",
                               "chr18_1929095_1929595", "chr18_1954501_1954727",
                               "chr18_2049865_2050884", "chr18_2083726_2084102",
                               "chr18_2087935_2088622", "chr18_2104705_2105551",
                               "chr18_2108641_2108907"),
                      CCAN = c(1,2,2,2,3,3,3,3,2))
olap_ccans <- find_overlapping_ccans(ccan_df)
```

find_overlapping_coordinates *Find peaks that overlap a specific genomic location*

Description

Find peaks that overlap a specific genomic location

Usage

```
find_overlapping_coordinates(coord_list, coord, maxgap = 0)
```

Arguments

coord_list	A list of coordinates to be searched for overlap in the form chr_100_2000.
coord	The coordinates that you want to find in the form chr1_100_2000.
maxgap	The maximum distance in base pairs between coord and the coord_list that should count as overlapping. Default is 0.

Value

A character vector of the peaks that overlap coord.

Examples

```
test_coords <- c("chr18_10025_10225", "chr18_10603_11103",
                 "chr18_11604_13986",
                 "chr18_157883_158536", "chr18_217477_218555",
                 "chr18_245734_246234")
find_overlapping_coordinates(test_coords, "chr18:10,100-1246234")
```

generate_ccans *Generate cis-co-accessibility networks (CCANs)*

Description

Post process cicero co-accessibility scores to extract modules of sites that are co-accessible.

Usage

```
generate_ccans(connections_df, coaccess_cutoff_override = NULL,
  tolerance_digits = 2)
```

Arguments

connections_df	Data frame of connections with columns: Peak1, Peak2, coaccess. Generally, the output of run_cicero or assemble_connections
coaccess_cutoff_override	Numeric, co-accessibility score threshold to impose. Overrides automatic calculation.
tolerance_digits	The number of digits to calculate cutoff to. Default is 2 (0.01 tolerance)

Details

CCANs are calculated by first specifying a minimum co-accessibility score and then using the Louvain community detection algorithm on the subgraph induced by excluding edges below this score. For this function, either the user can specify the minimum co-accessibility using coaccess_cutoff_override, or the cutoff can be calculated automatically by optimizing for CCAN number. The cutoff calculation can be slow, so users may wish to use the coaccess_cutoff_override after initially calculating the cutoff to speed future runs.

Value

Data frame with two columns - Peak and CCAN. CCAN column indicates CCAN assignment. Peaks not included in a CCAN are not returned.

Examples

```
## Not run:
  data("cicero_data")
  set.seed(18)
  data("human.hg19.genome")
  sample_genome <- subset(human.hg19.genome, V1 == "chr18")
  sample_genome$V2[1] <- 100000
  input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
  input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                               reduction_method = 'tSNE',
                               norm_method = "none")
  tsne_coords <- t(reducedDimA(input_cds))
  row.names(tsne_coords) <- row.names(pData(input_cds))
  cicero_cds <- make_cicero_cds(input_cds, reduced_coordinates = tsne_coords)
  cicero_cons <- run_cicero(cicero_cds, sample_genome, sample_num = 2)
  ccan_assigns <- generate_ccans(cicero_cons)

## End(Not run)
```

generate_cicero_models *Generate cicero models*

Description

Function to generate graphical lasso models on all sites in a CDS object within overlapping genomic windows.

Usage

```
generate_cicero_models(cds, distance_parameter, s = 0.75,
  window = 5e+05, max_elements = 200,
  genomic_coords = cicero::human.hg19.genome)
```

Arguments

cds	A cicero CDS object generated using make_cicero_cds.
distance_parameter	Distance based penalty parameter value. Generally, the mean of the calculated distance_parameter values from estimate_distance_parameter.
s	Power law value. See details.
window	Size of the genomic window to query, in base pairs.
max_elements	Maximum number of elements per window allowed. Prevents very large models from slowing performance.
genomic_coords	Either a data frame or a path (character) to a file with chromosome lengths. The file should have two columns, the first is the chromosome name (ex. "chr1") and the second is the chromosome length in base pairs. See data(human.hg19.genome) for an example. If a file, should be tab-separated and without header.

Details

The purpose of this function is to compute the raw covariances between each pair of sites within overlapping windows of the genome. Within each window, the function then estimates a regularized correlation matrix using the graphical LASSO (Friedman et al., 2008), penalizing pairs of distant sites more than proximal sites. The scaling parameter, distance_parameter, in combination with the power law value s determines the distance-based penalty.

The parameter s is a constant that captures the power-law distribution of contact frequencies between different locations in the genome as a function of their linear distance. For a complete discussion of the various polymer models of DNA packed into the nucleus and of justifiable values for s, we refer readers to (Dekker et al., 2013) for a discussion of justifiable values for s. We use a value of 0.75 by default in Cicero, which corresponds to the "tension globule" polymer model of DNA (Sanborn et al., 2015). This parameter must be the same as the s parameter for estimate_distance_parameter.

Further details are available in the publication that accompanies this package. Run citation("cicero") for publication details.

Value

A list of results for each window. Either a `glasso` object, or a character description of why the window was skipped. This list can be directly input into `assemble_connections` to create a reconciled list of cicero co-accessibility scores.

References

- Dekker, J., Marti-Renom, M.A., and Mirny, L.A. (2013). Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nat. Rev. Genet. 14, 390–403.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9, 432–441.
- Sanborn, A.L., Rao, S.S.P., Huang, S.-C., Durand, N.C., Huntley, M.H., Jewett, A.I., Bochkov, I.D., Chinnappan, D., Cutkosky, A., Li, J., et al. (2015). Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. Proc. Natl. Acad. Sci. U. S. A. 112, E6456–E6465.

See Also

`estimate_distance_parameter`

Examples

```
data("cicero_data")
data("human.hg19.genome")
sample_genome <- subset(human.hg19.genome, V1 == "chr18")
sample_genome$V2[1] <- 100000
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds, reduced_coordinates = tsne_coords)
model_output <- generate_cicero_models(cicero_cds,
                                       distance_parameter = 0.3,
                                       genomic_coords = sample_genome)
```

---

gene_annotation_sample

Example gene annotation information

Description

Gencode gene annotation data from chromosome 18 of the human genome (hg19).

Usage

```
gene_annotation_sample
```

Format

A data frame with 15129 rows and 8 variables:

chromosome Chromosome

start Exon starting base

end Exon ending base

strand Exon mapping direction

feature Feature type

gene Gene ID

transcript Transcript ID

symbol Gene symbol

---

human.hg19.genome *Chromosome lengths from human genome hg19*

Description

A list of the chromosomes in hg19 and their lengths in base pairs.

Usage

```
human.hg19.genome
```

Format

A data frame with 93 rows and 2 variables:

V1 Chromosome

V2 Chromosome length, base pairs

make_atac_cds *Make ATAC CDS object*

Description

This function takes as input a data frame or a path to a file in a sparse matrix format and returns a properly formatted `CellDataSet` (CDS) object.

Usage

```
make_atac_cds(input, binarize = FALSE)
```

Arguments

`input`	Either a data frame or a path to input data. If a file, it should be a tab-delimited text file with three columns and no header. For either a file or a data frame, the first column is the peak coordinates in the form "chr10_100013372_100013596", the second column is the cell name, and the third column is an integer that represents the number of reads from that cell overlapping that peak. Zero values do not need to be included (sparse matrix format).
`binarize`	Logical. Should the count matrix be converted to binary?

Value

A CDS object containing your ATAC data in proper format.

Examples

```
data("cicero_data")
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
```

make_cicero_cds *Create cicero input CDS*

Description

Function to generate an aggregated input CDS for cicero. `run_cicero` takes as input an aggregated cicero CDS object. This function will generate the CDS given an input CDS (perhaps generated by `make_atac_cds`) and a value for k, which is the number of cells to be aggregated per bin. The default value for k is 50.

Usage

```
make_cicero_cds(cds, reduced_coordinates, k = 50, summary_stats = NULL,
  size_factor_normalize = TRUE, silent = FALSE)
```

Arguments

cds	Input CDS object.
reduced_coordinates	A data frame with columns representing the coordinates of each cell in reduced dimension space (generally 2-3 dimensions). row.names(reduced_coordinates) should match the cell names in the CDS object. If dimension reduction was done using monocle, tSNE coordinates can be accessed by t(reducedDimA(cds)), and DDRTree coordinates can be accessed by t(reducedDimS(cds)).
k	Number of cells to aggregate per bin.
summary_stats	Which numeric pData(cds) columns you would like summarized (mean) by bin in the resulting CDS object.
size_factor_normalize	Logical, should accessibility values be normalized by size factor?
silent	Logical, should warning and info messages be printed?

Details

Aggregation of similar cells is done using a k-nearest-neighbors graph and a randomized "bagging" procedure. Details are available in the publication that accompanies this package. Run citation("cicero") for publication details. KNN is calculated using knn.index

Value

Aggregated CDS object.

Examples

```
data("cicero_data")

input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds, reduced_coordinates = tsne_coords)
```

---

make_sparse_matrix *Make a symmetric square sparse matrix from data frame*

Description

Convert a data frame into a square sparse matrix (all versus all)

Usage

```
make_sparse_matrix(data, i.name = "Peak1", j.name = "Peak2",
  x.name = "value")
```

Arguments

data	data frame
i.name	name of i column
j.name	name of j column
x.name	name of value column

Value

sparse matrix

---

normalize_gene_activities

Normalize gene activities

Description

Normalize the output of build_gene_activity_matrix. Input is either one or multiple gene activity matrices. Any gene activities to be compared amongst each other should be normalized together.

Usage

```
normalize_gene_activities(activity_matrices, cell_num_genes)
```

Arguments

activity_matrices	A gene activity matrix, output from build_gene_activity_matrix, or a list of gene activity matrices to be normalized together.
cell_num_genes	A named vector of the total number of accessible sites per cell. Names should correspond to the cell names in the activity matrices. These values can be found in the "num_genes_expressed" column of the pData table of the CDS used to calculate the gene activity matrix.

Value

Normalized activity matrix or matrices.

Examples

```
data("cicero_data")
data("human.hg19.genome")
sample_genome <- subset(human.hg19.genome, V1 == "chr18")
sample_genome$V2[1] <- 100000
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- detectGenes(input_cds)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds,
                              reduced_coordinates = tsne_coords)
cons <- run_cicero(cicero_cds, sample_genome, sample_num=2)

data(gene_annotation_sample)
gene_annotation_sub <- gene_annotation_sample[,c(1:3, 8)]
names(gene_annotation_sub)[4] <- "gene"
input_cds <- annotate_cds_by_site(input_cds, gene_annotation_sub)
num_genes <- pData(input_cds)$num_genes_expressed
names(num_genes) <- row.names(pData(input_cds))
unnorm_ga <- build_gene_activity_matrix(input_cds, cons)
cicero_gene_activities <- normalize_gene_activities(unnorm_ga, num_genes)
```

`plot_accessibility_in_pseudotime` *Plot accessibility by pseudotime*

Description

Make a barplot of chromatin accessibility across pseudotime

Usage

```
plot_accessibility_in_pseudotime(cds_subset, breaks = 10)
```

Arguments

`cds_subset`	Subset of the CDS object you want to plot. The CDS must have a column in the pData table called "Pseudotime".
`breaks`	Number of breaks along pseudotime. Controls the coarseness of the plot.

Details

This function plots each site in the CDS subset by cell pseudotime as a barplot. Cells are divided into bins by pseudotime (number determined by `breaks`) and the percent of cells in each bin that are accessible is represented by bar height. In addition, the black line represents the pseudotime-dependent average accessibility from a smoothed binomial regression.

Value

ggplot object

Examples

```
## Not run:
plot_accessibility_in_pseudotime(input_cds_lin[c("chr18_38156577_38158261",
                                                 "chr18_48373358_48374180",
                                                 "chr18_60457956_60459080")])

## End(Not run)
```

`plot_connections` *Plot connections*

Description

Plotting function for Cicero connections. Uses `plotTracks` as its basis

Usage

```
plot_connections(connection_df, chr, minbp, maxbp, coaccess_cutoff = 0,
  peak_color = "#B4656F", connection_color = "#7F7CAF",
  connection_color_legend = TRUE, alpha_by_coaccess = FALSE,
  connection_width = 2, connection_ymax = NULL, gene_model = NULL,
  gene_model_color = "#81D2C7", gene_model_shape = c("smallArrow",
  "box"), comparison_track = NULL, comparison_coaccess_cutoff = 0,
  comparison_peak_color = "#B4656F",
  comparison_connection_color = "#7F7CAF",
  comparison_connection_color_legend = TRUE,
  comparison_connection_width = 2, comparison_ymax = NULL,
  collapseTranscripts = FALSE, include_axis_track = TRUE,
  return_as_list = FALSE, viewpoint = NULL,
  comparison_viewpoint = TRUE, viewpoint_color = "#F0544F",
  viewpoint_fill = "#EFD8D7", viewpoint_alpha = 0.5)
```

Arguments

connection_df	Data frame of connections, which must include the columns 'Peak1', 'Peak2', and 'coaccess'. Generally, the output of run_cicero or assemble_connections.
chr	The chromosome of the region you would like to plot in the form 'chr10'.
minbp	The base pair coordinate of the start of the region to be plotted.
maxbp	The base pair coordinate of the end of the region to be plotted.
coaccess_cutoff	The minimum cicero co-accessibility score you would like to be plotted. Default is 0.
peak_color	Color for peak annotations - a single color, the name of a column containing color values that correspond to Peak1, or the name of column containing a character or factor to base peak colors on.
connection_color	Color for connection lines. A single color, the name of a column containing color values, or the name of a column containing a character or factor to base connection colors on.
connection_color_legend	Logical, should connection color legend be shown?
alpha_by_coaccess	Logical, should the transparency of connection lines be scaled based on co-accessibility score?
connection_width	Width of connection lines.
connection_ymax	Connection y-axis height. If NULL, chosen automatically.
gene_model	Either NULL or a data.frame. The data.frame should be in a form compatible with the Gviz function GeneRegionTrack-class (cannot have NA as column names).
gene_model_color	Color for gene annotations.
gene_model_shape	Shape for gene models, passed to GeneRegionTrack-class. Options described at GeneRegionTrack-class.
comparison_track	Either NULL or a data frame. If a data frame, a second track of connections will be plotted based on this data. This data frame has the same requirements as connection_df (Peak1, Peak2 and coaccess columns).
comparison_coaccess_cutoff	The minimum cicero co-accessibility score you would like to be plotted for the comparison dataset. Default = 0.
comparison_peak_color	Color for comparison peak annotations - a single color, the name of a column containing color values that correspond to Peak1, or the name of a column containing a character or factor to base peak colors on.

comparison_connection_color
: Color for comparison connection lines. A single color, the name of a column containing color values, or the name of a column containing a character or factor to base connection colors on.

comparison_connection_color_legend
: Logical, should comparison connection color legend be shown?

comparison_connection_width
: Width of comparison connection lines.

comparison_ymax
: Connection y-axis height for comparison track. If NULL, chosen automatically.

collapseTranscripts
: Logical or character scalar. Can be one in gene, longest, shortest or meta. Variable is passed to the GeneRegionTrack-class function of Gviz. Determines whether and how to collapse related transcripts. See Gviz documentation for details.

include_axis_track
: Logical, should a genomic axis be plotted?

return_as_list
: Logical, if TRUE, the function will not plot, but will return the plot components as a list. Allows user to add/customize Gviz components and plot them separately using plotTracks.

viewpoint
: NULL or Coordinates in form "chr1_10000_10020". Use viewpoint if you would like to plot cicero connections "4C-seq style". Only connections originating in the viewpoint will be shown. Ideal for comparisons with 4C-seq data. If comparison_viewpoint is TRUE, any comparison track will be subsetted as well.

comparison_viewpoint
: Logical, should viewpoint apply to comparison track as well?

viewpoint_color
: Color for the highlight border.

viewpoint_fill
: Color for the highlight fill.

viewpoint_alpha
: Alpha value for the highlight fill.

Value

A gene region plot, or list of components if return_as_list is TRUE.

Examples

```
cicero_cons <- data.frame(
           Peak1 = c("chr18_10034652_10034983", "chr18_10034652_10034983",
                     "chr18_10034652_10034983", "chr18_10034652_10034983",
                     "chr18_10087586_10087901", "chr18_10120685_10127115",
                     "chr18_10097718_10097934", "chr18_10087586_10087901",
                     "chr18_10154818_10155215", "chr18_10238762_10238983",
                     "chr18_10198959_10199183", "chr18_10250985_10251585"),
           Peak2 = c("chr18_10097718_10097934", "chr18_10087586_10087901",
                     "chr18_10154818_10155215", "chr18_10238762_10238983",
                     "chr18_10198959_10199183", "chr18_10250985_10251585",
```

```
                      "chr18_10034652_10034983", "chr18_10034652_10034983",
                      "chr18_10034652_10034983", "chr18_10034652_10034983",
                      "chr18_10087586_10087901", "chr18_10120685_10127115"),
            coaccess = c(0.0051121787, 0.0016698617, 0.0006570246,
                         0.0013466927, 0.0737935011, 0.3264019452,
                         0.0051121787, 0.0016698617, 0.0006570246,
                         0.0013466927, 0.0737935011, 0.3264019452))
plot_connections(cicero_cons, chr = "chr18",
                 minbp = 10034652,
                 maxbp = 10251585,
                 peak_color = "purple")
```

`ranges_for_coords` *Construct GRanges objects from coordinate strings*

Description

Construct GRanges objects from coordinate strings

Usage

```
ranges_for_coords(coord_strings, meta_data_df = NULL,
  with_names = FALSE)
```

Arguments

`coord_strings`	A list of coordinate strings (in the form "chr1:500000-1000000")
`meta_data_df`	A data frame with any meta data columns you want included with the ranges. Must be in the same order as coord_strings.
`with_names`	logical - should meta data include coordinate string (field coord_string)?

Details

Coordinate strings consist of three pieces of information: chromosome, start, and stop. These pieces of information can be separated by the characters ":", "_", or "-". Commas will be removed, not used as separators (ex: "chr18:8,575,097-8,839,855" is ok).

Value

GRanges object of the input strings

See Also

GRanges-class

Examples

```
ran1 <- ranges_for_coords("chr1:2039-30239", with_names = TRUE)
ran2 <- ranges_for_coords(c("chr1:2049-203902", "chrX:489249-1389389"),
                          meta_data_df = data.frame(dat = c("1", "X")))
ran3 <- ranges_for_coords(c("chr1:2049-203902", "chrX:489249-1389389"),
                          with_names = TRUE,
                          meta_data_df = data.frame(dat = c("1", "X"),
                                                    stringsAsFactors = FALSE))
```

`run_cicero` *Run Cicero*

Description

A wrapper function that runs the primary functions of the Cicero pipeline with default parameters. Runs `estimate_distance_parameter`, `generate_cicero_models` and `assemble_connections`. See the manual pages of these functions for details about their function and parameter options. Defaults in this function are designed for mammalian data, those with non-mammalian data should read about parameters in the above functions.

Usage

```
run_cicero(cds, genomic_coords, window = 5e+05, silent = FALSE,
  sample_num = 100)
```

Arguments

`cds`	Cicero CDS object, created using `make_cicero_cds`
`genomic_coords`	Either a data frame or a path (character) to a file with chromosome lengths. The file should have two columns, the first is the chromosome name (ex. "chr1") and the second is the chromosome length in base pairs. See `data(human.hg19.genome)` for an example. If a file, should be tab-separated and without header.
`window`	Size of the genomic window to query, in base pairs.
`silent`	Whether to print progress messages
`sample_num`	How many sample genomic windows to use to generate `distance_parameter` parameter. Default: 100.

Value

A table of co-accessibility scores

Examples

```
data("cicero_data")
data("human.hg19.genome")
sample_genome <- subset(human.hg19.genome, V1 == "chr18")
sample_genome$V2[1] <- 100000
input_cds <- make_atac_cds(cicero_data, binarize = TRUE)
input_cds <- reduceDimension(input_cds, max_components = 2, num_dim=6,
                             reduction_method = 'tSNE',
                             norm_method = "none")
tsne_coords <- t(reducedDimA(input_cds))
row.names(tsne_coords) <- row.names(pData(input_cds))
cicero_cds <- make_cicero_cds(input_cds, reduced_coordinates = tsne_coords)
cons <- run_cicero(cicero_cds, sample_genome, sample_num = 2)
```

Appendix B: GARNETT PACKAGE SOURCE AND MANUAL

B.1 SOURCE CODE

Garnett source code is available at https://github.com/cole-trapnell-lab/garnett-release and a website with full documentation is available at https://cole-trapnell-lab.github.io/garnett/.

B.2 MANUAL

Package 'garnett'

February 25, 2019

Type Package

Title Automated cell type classification

Version 0.1.4

Description Garnett facilitates automated cell type classification from single-cell expression (and other genomic) data. Garnett works by taking single-cell data, along with a cell type definition (marker) file, and training a regression-based classifier. Once a classifier is trained for a tissue/sample type, it can be applied to classify future datasets from similar tissues.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0),
monocle

Imports AnnotationDbi (>= 1.44.0),
assertthat (>= 0.2.0),
Biobase (>= 2.42.0),
DelayedArray (>= 0.8.0),
DelayedMatrixStats (>= 1.4.0),
doParallel (>= 1.0.14),
forcats (>= 0.3.0),
ggplot2 (>= 3.1.0),
ggrepel (>= 0.8.0),
glmnet (>= 2.0-16),
igraph (>= 1.2.2),
irlba (>= 2.3.2),
Matrix (>= 1.2-15),
methods,
org.Hs.eg.db (>= 3.7.0),
org.Mm.eg.db,
plyr (>= 1.8.4),
RANN (>= 2.6),
reshape2 (>= 1.4.3),
rly (>= 1.6.2),

stringr (>= 1.3.1),
viridis (>= 0.5.1)

Suggests testthat,
knitr,
rmarkdown,
vdiffr

RoxygenNote 6.1.1

VignetteBuilder knitr

`garnett-package` *garnett*

Description

Garnett facilitates automated cell type classification from single-cell expression (and other genomic) data. Garnett works by taking single-cell data, along with a cell type definition (marker) file, and training a regression-based classifier. Once a classifier is trained for a tissue/sample type, it can be applied to classify future datasets from similar tissues.

`cell_rules` *cell_rules class*

Description

Representation of cell type derived from marker file.

Slots

`name` character. Name of the cell type.

`gene_names` character. A list of all of the genes included in the definition.

`expressed` character. A list of genes defined as "expressed:".

`not_expressed` character. A list of genes defined as "not expressed:".

`gene_rules` vector of GeneRules-class. A list of genes defined under specific rules using "expressed below:", "expressed above:", or "expressed between:".

`meta` data.frame of meta data rules specified in marker file.

`parenttype` character. The name of the parent type - specified by "subtype of:".

`references` character. A list of references included in the definition.

check_markers *Check marker file*

Description

Check the markers chosen for the marker file and generate a table of useful statistics. The output of this function can be fed into plot_markers to generate a diagnostic plot.

Usage

```
check_markers(cds, marker_file, db, cds_gene_id_type = "SYMBOL",
marker_file_gene_id_type = "SYMBOL", propogate_markers = TRUE,
use_tf_idf = TRUE, classifier_gene_id_type = "ENSEMBL")
```

Arguments

cds	Input CDS object.
marker_file	A character path to the marker file to define cell types. See details and documentation for Parser by running ?Parser for more information.
db	Bioconductor AnnotationDb-class package for converting gene IDs. For example, for humans use org.Hs.eg.db. See available packages at Bioconductor. If your organism does not have an AnnotationDb-class database available, you can specify "none", however then Garnett will not check/convert gene IDs, so your CDS and marker file must have the same gene ID type.
cds_gene_id_type	The type of gene ID used in the CDS. Should be one of the values in columns(db). Default is "ENSEMBL". Ignored if db = "none".
marker_file_gene_id_type	The type of gene ID used in the marker file. Should be one of the values in columns(db). Default is "SYMBOL". Ignored if db = "none".
propogate_markers	Logical. Should markers from child nodes of a cell type be used in finding representatives of the parent type? Should generally be TRUE.
use_tf_idf	Logical. Should TF-IDF matrix be calculated during estimation? If TRUE, estimates will be more accurate, but calculation is slower with very large datasets.
classifier_gene_id_type	The type of gene ID that will be used in the classifier. If possible for your organism, this should be "ENSEMBL", which is the default. Ignored if db = "none".

Details

This function checks the chosen cell type markers in the marker file provided to ensure they are good candidates for use in classification. The function works by estimating which cells will be chosen given each marker gene and returning some statistics for each marker. Note that this function does not take into account meta data information when calculating statistics.

The output data.frame has several columns:

marker_gene Gene name as provided in the marker file

ENSEMBL The corresponding ensembl ID derived from db conversion

parent The parent cell type in the cell type hierarchy - 'root' if top level

cell_type The cell type the marker belongs to

in_cds Whether the marker is present in the CDS

nominates The number of cells the marker is estimated to nominate to the cell type

total_nominated The total number of cells nominated by all the markers for that cell type

exclusion_dismisses The number of cells no longer nominated to the cell type if this marker is excluded (i.e. not captured by other markers for the cell type)

inclusion_ambiguates How many cells become ambiguous (i.e. are nominated to multiple cell types) if this marker is included

most_overlap The cell type that most often shares this marker (i.e. is the other side of the ambiguity). If inclusion_ambiguates is 0, most_overlap is NA

ambiguity inclusion_ambiguates/nominates - if high, consider excluding this marker

marker_score (1/(ambiguity + .01)) * nominates/total_nominated - a general measure of the quality of a marker. Higher is better

summary A summary column that identifies potential problems with the provided markers

Value

Data.frame of marker check results.

Examples

```
library(org.Hs.eg.db)
data(test_cds)

# generate size factors for normalization later
test_cds <- estimateSizeFactors(test_cds)
marker_file_path <- system.file("extdata", "pbmc_bad_markers.txt",
package = "garnett")
marker_check <- check_markers(test_cds, marker_file_path,
db=org.Hs.eg.db,
cds_gene_id_type = "SYMBOL",
marker_file_gene_id_type = "SYMBOL")
```

`classify_cells` *Classify cells from trained garnett_classifier*

Description

This function uses a previously trained `garnett_classifier` (trained using `train_cell_classifier`) to classify cell types in a CDS object.

Usage

```
classify_cells(cds, classifier, db, cds_gene_id_type = "ENSEMBL",
rank_prob_ratio = 1.5, cluster_extend = FALSE, verbose = FALSE)
```

Arguments

cds	Input CDS object.
classifier	Trained garnett_classifier - output from train_cell_classifier.
db	Bioconductor AnnotationDb-class package for converting gene IDs. For example, for humans use org.Hs.eg.db. See available packages at Bioconductor. If your organism does not have an AnnotationDb-class database available, you can specify "none", however then Garnett will not check/convert gene IDs, so your CDS and marker file must have the same gene ID type.
cds_gene_id_type	The type of gene ID used in the CDS. Should be one of the values in columns(db). Default is "ENSEMBL". Ignored if db = "none".
rank_prob_ratio	Numeric value greater than 1. This is the minimum odds ratio between the probability of the most likely cell type to the second most likely cell type to allow assignment. Default is 1.5. Higher values are more conservative.
cluster_extend	Logical. When TRUE, the classifier provides a secondary cluster-extended classification, which assigns type for the entire cluster based on the assignments of the cluster members. If the pData table of the input CDS has a column called "garnett_cluster", this will be used for cluster-extended assignments. Otherwise, assignments are calculated using Louvain community detection in PCA space. This assignment is returned as a column in the output CDS pData table. For large datasets, if the "garnett_cluster" column is not provided and cluster_extend = TRUE, the function can be significantly slower the first time it is run. See details for more information.
verbose	Logical. Should progress messages be printed.

Details

This function applies a previously trained multinomial glmnet classifier at each node of a previously defined garnett_classifier tree. The output is a CDS object with cell type classifications added to the pData table.

When cluster_extend = TRUE, Louvain communities are calculated in PCA space. Any cluster where >90 >5 be assigned that cluster-extended type. Both cluster-extended type and originally calculated cell type are reported.

Value

CDS object with classifications in the pData table.

Examples

```
library(org.Hs.eg.db)
data(test_classifier)
data(test_cds)

# classify cells
test_cds <- classify_cells(test_cds, test_classifier,
db = org.Hs.eg.db,
rank_prob_ratio = 1.5,
cluster_extend = TRUE,
cds_gene_id_type = "SYMBOL")
```

---

garnett_classifier *The garnett_classifier class*

Description

Classifies cells according to a hierarchy of types.

Details

Classifies cells according to a hierarchy of types via user-defined gating functions.

Slots

`classification_tree`: Object of class `"igraph"`

`cell_totals`: Object of class `"numeric"`

`gene_id_type`: Object of class `"character"`

`references`: Object of class `"list"`

---

gene_rule-class *gene_rule class*

Description

Class definition for gene_rule, which is a container to hold a gene name along with upper and lower expression bounds.

Slots

`gene_name` character. The name of the gene

`lower` numeric. Lower bound of expression - same units as CDS object.

`upper` numeric. Upper bound of expression - same units as CDS object.

get_classifier_references *Retrieve marker references from garnett_classifier*

Description

Retrieve marker references from garnett_classifier

Usage

```
get_classifier_references(classifier, cell_type = NULL)
```

Arguments

classifier	garnett_classifier created using train_cell_classifier.
cell_type	Cell type name or NULL. References for which cell type should be printed? If NULL, all are printed.

Value

List of references included when garnett_classifier was trained.

Examples

```
data(test_classifier)
get_classifier_references(test_classifier)
```

get_feature_genes *Extract feature genes*

Description

Extract the genes chosen as features in cell type classification from a trained garnett_classifier

Usage

```
get_feature_genes(classifier, node = "root", convert_ids = FALSE,
db = NULL)
```

Arguments

classifier	Trained garnett_classifier - output from train_cell_classifier.
node	Character. The name of the parent node of the multinomial classifier you would like to view features for. If top level, use "root".
convert_ids	Logical. Should classifier IDs be converted to SYMBOL?
db	Bioconductor AnnotationDb-class package for converting gene IDs. For example, for humans use org.Hs.eg.db. See available packages at Bioconductor. If convert_ids = FALSE, db can be NULL.

Value

A data.frame of coefficient values for each gene with non-zero coefficients in the classifier.

Examples

```
library(org.Hs.eg.db)
data(test_classifier)
featuresdf <- get_feature_genes(test_classifier, db=org.Hs.eg.db)
featuresdf2 <- get_feature_genes(test_classifier,
convert_ids = FALSE,
node = "T cells")
```

Parser *Parsing the Garnett marker file*

Description

Garnett uses a marker file to allow users to specify cell type definitions. While the marker file is designed to be easy to construct and human-readable, it is parsed by Garnett automatically, and so it needs to follow certain formatting constraints.

Usage

```
Parser
```

Format

An object of class R6ClassGenerator of length 24.

Details

The following describes the constraints necessary in the input to the marker_file argument of train_cell_classifier and check_markers.

Elements of a cell type description

The basic structure of the Garnett marker file is a series of entries, each describing elements of a cell type. After the cell name, each additional line will be a descriptor, which begins with a keyword, followed by a colon (':'). After the colon, a series of specifications can be added, separated by commas (','). Descriptors may spill onto following lines so long as you do not split a specification across multiple lines (i.e. if breaking up a long descriptor across multiple lines, all but the last line should end with a comma). Each new descriptor should begin on a new line. A generic cell type entry looks like this:

"' > cell type name descriptor: spec1, spec2, spec3, spec4 descriptor2: spec1 "'

The following are the potential descriptors:

cell name Required Each cell type must have a unique name, and the name should head the cell type description. To indicate a new cell type, use the > symbol, followed by the cell name, followed by a new line. For example, `> T cell`.

expressed: Required After the cell name, the minimal requirement for each cell type is the name of a single marker gene. The line in the marker file will begin with `expressed:`, followed by one or more gene names separated by commas. The last gene name of the descriptor is not followed by a comma. Gene IDs can be of any type (ENSEMBL, SYMBOL, etc.) that is present in the Bioconductor `AnnotationDb-class` package for your species. (See available packages on the Bioconductor website). For example, for human, use `org.Hs.eg.db`. To see available gene ID types, you can run `columns(db)`. You will specify which gene ID type you used when calling `train_cell_classifier`.If your species does not have an annotation dataset of type `AnnotationDb-class`, you can set `db = 'none'`, however Garnett will then not convert gene ID types, so CDS and marker file gene ID types need to be the same.

not expressed: In addition to specifying genes that the cell type should express, you can also specify genes that your cell type should not express. Details on specifying genes are the same as for `expressed:`.

subtype of: When present, this descriptor specifies that a cell type is a subtype of another cell type that is also described in the marker file. A biological example would be a CD4 T cell being a subtype of a T cell. This descriptor causes the cell type to be classified on a separate sublevel of the classification hierarchy, after the classification of its parent type is done (i.e. first T cells are discriminated from other cell types, then the T cells are subclassified using any cell types with the descriptor `subtype of: T cell`). `subtype of:` can only include a single specification, and the specification must be the exact name of another cell type specified in this marker file.

references: This descriptor is not required, but is highly recommended. The specifications for this descriptor should be links/DOIs documenting how you chose your marker genes. While these specifications will not influence cell type classification, they will be packaged with the built classifier so that future users of the classifier can trace the origins of the markers/

***meta data:** This wildcard descriptor allows you to specify any other property of a cell type that you wish to specify. The keyword will be the name of the column in your `pData` (meta data) table that you wish to specify, and the specifications will be a list of acceptable values for that meta data. An example use of this would be `tissue: liver,kidney`, which would specify that training cells for this cell type must have "liver" or "kidney" as their entry in the "tissue" column of the `pData` table.

expressed below: While we recommend that you use `expressed:` and `not expressed:` to specify the cell type's marker genes, because these terms utilize the entirety of Garnett's built-in normalization and standardization, you can also specify expression using the following logical descriptors `expressed below:`,`expressed above:`,`expressed between:`. Note that no normalization occurs with these descriptors; they are used as logical gates only. To specify `expressed below:`, use the gene name, followed by a space, followed by a number. This will only allow training cells that have this gene expressed below the given value **in the units of the expression matrix provided**. For example, `expressed below: MYOD1 7,MYH3 2`.

expressed above: Similar to `expressed below:`, but will only allow training cells expressing the given gene above the value provided.

expressed between: Similar to `expressed below:`, but provide two values separated by spaces. For example `expressed between: ACT5 2 5.5,ACT2 1 2.7`. This descriptor will only allow training cells expressing the given gene between the two values provided.

Checking your marker file

Because only specific expressed markers are useful for Garnett classification, we recommend that you always check your marker file for ambiguity before proceeding with classification. To do this, we have provided the functions `check_markers` and `plot_markers` to facilitate marker checking. See that manual pages for those functions for details.

See Also

`train_cell_classifier`

`plot_markers` *Plot marker metrics*

Description

This function takes as input the output of the `check_markers` function and generates a plot to visualize the most important metrics.

Usage

```
plot_markers(marker_check_df, amb_marker_cutoff = 0.5, label_size = 2)
```

Arguments

`marker_check_df`	Marker check data.frame - output of check_markers.
`amb_marker_cutoff`	Numeric. Cutoff at which to label ambiguous markers. Default 0.5.
`label_size`	Numeric, size of the text labels for ambiguous markers and unplotted markers.

Value

A ggplot object of the marker plot.

Examples

```
library(org.Hs.eg.db)

marker_file_path <- system.file("extdata", "pbmc_test.txt",
package = "garnett")
data(test_cds)
marker_check <- check_markers(test_cds,
marker_file_path,
db=org.Hs.eg.db,
cds_gene_id_type = "SYMBOL",
marker_file_gene_id_type = "SYMBOL")

plot_markers(marker_check)
```

---

test_cds *Small test CDS object*

Description

A CDS object used for testing and demonstration. Derives from the 10x Genomics V1 PBMC single-cell RNA-seq dataset.

Usage

```
test_cds
```

Format

A CDS object with 32738 features and 800 samples

Source

https://support.10xgenomics.com/single-cell-gene-expression/datasets/

---

test_classifier *Small test garnett_classifier object*

Description

A small test garnett_classifier trained using the `test_cds` object.

Usage

```
test_classifier
```

Format

An object of class `garnett_classifier` of length 1.

`train_cell_classifier` *Train a cell type classifier*

Description

This function takes single-cell expression data in the form of a CDS object and a cell type definition file (marker file) and trains a multinomial classifier to assign cell types. The resulting `garnett_classifier` object can be used to classify the cells in the same dataset, or future datasets from similar tissues/samples.

Usage

```
train_cell_classifier(cds, marker_file, db, cds_gene_id_type = "ENSEMBL",
marker_file_gene_id_type = "SYMBOL", min_observations = 8,
max_training_samples = 500, num_unknown = 500,
propogate_markers = TRUE, cores = 1, lambdas = NULL,
classifier_gene_id_type = "ENSEMBL")
```

Arguments

`cds`	Input CDS object.
`marker_file`	A character path to the marker file to define cell types. See details and documentation for `Parser` by running `?Parser`for more information.
`db`	Bioconductor AnnotationDb-class package for converting gene IDs. For example, for humans use org.Hs.eg.db. See available packages at Bioconductor. If your organism does not have an AnnotationDb-class database available, you can specify "none", however then Garnett will not check/convert gene IDs, so your CDS and marker file must have the same gene ID type.
`cds_gene_id_type`	The type of gene ID used in the CDS. Should be one of the values in `columns(db)`. Default is "ENSEMBL". Ignored if db = "none".
`marker_file_gene_id_type`	The type of gene ID used in the marker file. Should be one of the values in `columns(db)`. Default is "SYMBOL". Ignored if db = "none".
`min_observations`	An integer. The minimum number of representative cells per cell type required to include the cell type in the predictive model. Default is 8.
`max_training_samples`	An integer. The maximum number of representative cells per cell type to be included in the model training. Decreasing this number increases speed, but may hurt performance of the model. Default is 500.

num_unknown — An integer. The number of unknown type cells to use as an outgroup during classification. Default is 500.

propogate_markers — Logical. Should markers from child nodes of a cell type be used in finding representatives of the parent type? Should generally be TRUE.

cores — An integer. The number of cores to use for computation.

lambdas — NULL or a numeric vector. Allows the user to pass their own lambda values to cv.glmnet. If NULL, preset lambda values are used.

classifier_gene_id_type — The type of gene ID that will be used in the classifier. If possible for your organism, this should be "ENSEMBL", which is the default. Ignored if db = "none".

Details

This function has three major parts: 1) parsing the marker file 2) choosing cell representatives and 3) training the classifier. Details on each of these steps is below:

Parsing the marker file: the first step of this function is to parse the provided marker file. The marker file is a representation of the cell types expected in the data and known characteristics about them. Information about marker file syntax is available in the documentation for the Parser function, and on the Garnett website.

Choosing cell representatives: after parsing the marker file, this function identifies cells that fit the parameters specified in the file for each cell type. Depending on how marker genes and other cell type definition information are specified, expression data is normalized and expression cutoffs are defined automatically. In addition to the cell types in the marker file, an outgroup of diverse cells is also chosen.

Training the classifier: lastly, this function trains a multinomial GLMnet classifier on the chosen representative cells.

Because cell types can be defined hierarchically (i.e. cell types can be subtypes of other cell types), steps 2 and 3 above are performed iteratively over all internal nodes in the tree representation of cell types.

See the Garnett website and the accompanying paper for further details.

Examples

```
library(org.Hs.eg.db)
data(test_cds)
set.seed(260)

marker_file_path <- system.file("extdata", "pbmc_bad_markers.txt",
package = "garnett")

test_classifier <- train_cell_classifier(cds = test_cds,
marker_file = marker_file_path,
db=org.Hs.eg.db,
min_observations = 10,
cds_gene_id_type = "SYMBOL",
num_unknown = 50,
```

```
marker_file_gene_id_type = "SYMBOL")
```

VITA

Hannah A. Pliner grew up in 'The Witch City' of Salem, Massachusetts. She graduated *cum laude* from Phillips Exeter Academy in Exeter, New Hampshire and *magna cum laude* from Georgetown University in Washington, D.C. with a B.S. in Biology and a minor in Psychology. While at Georgetown, she studied the evolutionary development of ectoderm in *Xenopus laevis* in the lab of Elena Casey. Before beginning graduate school, Hannah worked under Bryan Traynor in the Laboratory of Neurogenetics, National Institutes of Health in Bethesda, Maryland, studying the genetic etiology of neuromuscular disease, where she discovered an interest in computational biology. When not in front of a computer, Hannah tries to spend as much time as possible in the mountains, preferably with her God-dog Maizie.