FACULTY OF SCIENCES

A comparative review of dimensionality reduction methods for high-throughput single-cell transcriptomics

Sofie VEYS
Student number: 01207152

Promoter: Prof. Dr. Yvan Saeys
Scientific supervisor: Robrecht Cannoodt

Master's dissertation submitted to Ghent University to obtain the degree of Master of Science in Biochemistry and Biotechnology. Major Bioinformatics and Systems biology

Academic year: 2016 - 2017

UNIVERSITEIT GENT

Ghent University – Department of applied mathematics, computer science and statistics (W102)
VIB – Center for Inflammation Research
Research Group: Data Mining and modelling for biomedicine (DAMBI)

VIB-UGENT CENTER FOR INFLAMMATION RESEARCH

Confidentiality statement

Master thesis "Comparative review of dimensionality reduction methods for high-throughput single cell data"
by Sofie Veys
under the promotership of Prof. Dr. Yvan Saeys

This document and the information in it are provided in confidence, for the sole purpose of evaluation of the MASTER thesis of Sofie Veys, and may not be disclosed to any third party or used for any other purpose without the explicit written permission of Prof. Dr. Yvan Saeys."

Acknowledgements

At the end of my thesis I would like to thank all those who contributed in some way to my thesis and made this an unforgettable experience for me.

First, I would like to express my sincere gratitude to Promotor Professor Dr. Yvan Saeys for giving me the opportunity to do my thesis in his research lab. I thank him for the continuous advice and the great effort he put into the training of me and the other students throughout the course of our thesis. Additionally, I would like to thank the other members of the research group for their encouragement, insightful comments, and hard questions. It has been a period of intense learning for me. Their passion for bioinformatics was truly an inspiration for me.

I would particularly like to single out my supervisor R. Cannoodt for the wonderful guidance. He was there for me whenever I ran into a trouble spot or had a question about my research and writing. With his motivation, charisma, knowledge and sometimes extra ordinary ideas, he pointed me in the right direction. I could not imagine having a better advisor.

I also would like to thank my fellow thesis students. Besides sharing glasses, we share a mutual interest in bioinformatics. We were not only able to support each other by debating over our frustrations and findings, but also by talking about things other than our thesis.

To all my friends and family for supporting and helping me survive all the stress from this year and previous years. I am forever indebted to my parents who encouraged me to work hard and to never give up. This accomplishment would not have been possible without them.

And finally I am grateful to my boyfriend for the love and support. He was always there cheering me up and stood by me through the good and bad times.

Thank you very much, everyone!

Sofie Veys

Ghent, June 15, 2015

Table of contents

List of Abbreviations

NGS	Next generation sequencing
DNA	Deoxyribonucleic acid
cDNA	complementary DNA
RNA	Ribonucleic acid
RNA-seq	RNA sequencing
scRNA-seq	single cell RNA sequencing
GRN	gene regulatory networks
PCA	principal component analysis
t-SNE	t-distributed stochastic neighbour embedding
ZIFA	zero-inflated factor analysis
ICA	independent component analysis
MST	minimum spanning tree
MDS	multi-dimensional scaling
Bp	basepairs
LOOCV	leave one out cross validation
KNN	k-nearest neighbours
CVA	cross validation accuracy
LCMC	local continuity meta-criterion
HCA	Human Cell Atlas
GEO	Gene expression omnibus
UMI	Unique Molecular Identifier
ERCC	External RNA Control Consortium
LMDS	Landmark MDS
LLE	Local Linear Embedding

Nederlandse samenvatting

De transcriptome analyse van enkele cellen (scRNA-seq) is een nieuwe en snel evoluerende methode die met de tijd algemene RNA analyse zal vervangen. Door het transcriptoom op enkel cel niveau te bestuderen is scRNA-seq in staat nieuwe inzichten te geven over hoe biologische processen bijdragen tot de identiteit van een cel in levende organismen. Deze methode laat ons toe nieuwe biologische vragen zoals het ontrafelen van subpopulaties in heterogene cel culturen, het bestuderen van dynamische processen en het construeren van gen regulatieve netwerken verder te onderzoeken. Deze methode staat echter nog in zijn kinderschoenen. Ondanks de vele voordelen brengt deze techniek nieuwe statistische en computationele uitdagingen in de analyse met zich mee.

Een belangrijk onderdeel in deze analyses is de dimensie reductie van hoog dimensionele data naar een lagere, representatieve ruimte. De techniek laat de classificatie, visualisatie, geruis verwijdering en de compressie van grote datasets toe. Door de jaren heen werden klassieke lineaire dimensie reductie methoden zoals PCA en MDS, mits aanpassingen, toegepast en verscheidene niet-lineaire methoden zoals t-SNE ontwikkeld gezien single cell data meestal op of nabij een niet-lineaire ruimte ligt. Hoewel al deze methoden een gelijkaardig doel voor oog hebben, zijn hun werkingsmethoden verschillend waardoor de nood aan een algemeen overzicht stijgt. Bovendien is door de recent ontwikkelde single cell protocollen zoals inDrop en Drop-seq de omvang van de single cell datasets gestegen naar tien tot zelfs honderd duizend cellen per experiment, waardoor verwacht wordt dat de uitvoeringskost zal stijgen en uiteindelijk onhaalbaar zal worden, vooral voor nieuwe projecten zoals de menselijke cel atlas die de analyse van miljoenen cellen vereist.

Gedreven door het gebrek aan een systematische consensus en de verwachte stijging in tijd bij het uitvoeren van een dimensie reductie op de nieuwe single cell data, spitst dit onderzoek zich toe op het vergelijken van de meest gebruikte dimensie reductie methoden op een set van single cell data die in grootte variëren van 66 tot 70000 cellen op basis van drie evaluatie technieken. Een variatie van MDS werd toegevoegd die uitgebreid kan worden naar datasets van miljoenen cellen door gebruik te maken van een landmark methode die de hoge uitvoeringskost bij grote datasets omzeilt. Onze resultaten bevestigen dat de berekeningstijd van de huidige dimensie reductie methoden te groot wordt voor datasets die meer dan tienduizend cellen bevat. Daarentegen is landmark MDS in staat om een dimensie reductie op extreem grote datasets op een efficiënte manier uit te voeren binnen een aanvaardbare tijd. Verder zijn t-SNE en MDS de meest aangewezen methoden om een dimensie reductie uit te voeren afhankelijk van de grootte van de datasets en het aantal dimensies naar waar de data gereduceerd wordt. Gebaseerd op onze resultaten en verdere analyses is het mogelijk om bepaalde richtlijnen op te stellen die onderzoekers kunnen helpen bij het kiezen van de geschikte dimensie reductie techniek in hun onderzoek. Deze bevindingen zijn niet enkel toepasbaar in single cell genomics maar in elk onderzoeksveld waarin hoog-dimensionele data geproduceerd wordt, zoals gezichtsherkenning.

Abstract

Single cell RNA seq (scRNA-seq) is a new and rapidly upcoming method in the field of single cell genomics and will in time replace bulk RNAseq. By analysing the transcriptome at single cell level, scRNA-seq generates new insights into the complex biological systems that give rise to a cell's identity in living organisms. This allows us to address scientific questions that were evaded in the past years such as deconvolving the heterogeneity in cell populations, studying dynamic processes like cell state transitions and constructing gene regulatory networks. However, the new method is still in its infancy and with its advantages comes computational challenges that are just beginning to address.

An important tool in the analysis is dimensionality reduction, which transforms high-dimensional data into a meaningful reduced subspace. The technique allows classification, noise removal, visualisation and compression of high-dimensional data. Over the years, classical linear dimensionality reduction techniques such as PCA, MDS have been used and new non-linear methods such as t-SNE have been proposed since real-world data lies on or near a nonlinear manifold. While all of these have a similar goal, approaches to the problem are different and the lack of a general review arises. Moreover, as the magnitude of single cell data increases due to recent advancements in single cell protocols such as inDrop and Drop-seq allowing the sequencing of tens of thousand cells in a single experiment, the computational cost of dimensionality reduction methods to leverage out the information is expected to increase and become infeasible especially for projects such as Human cell atlas that requires the analysis of million of cells.

Motivated by the lack of a systematic consensus and the expected increase in computational complexity for new high-throughput transcriptomic data, this research generated a comparative review of the performances of the most frequently used dimensionality reduction techniques on a diverse set of scRNA-seq datasets using different evaluation metrics. A variation of MDS has been created that can scale to a number of tens to even hundred thousand cells using a landmark approach to circumvent the computational demand. Our results confirm that the computational time and memory of the current used dimensionality reduction methods become impervious for datasets larger than ten thousand cells. However Landmark MDS succeed to perform a dimension reduction on extreme large datasets in a feasible time compared to others without sacrificing in terms of performance. Further MDS is the recommended dimensionality reduction method depending on the size of the datasets and the number of dimensions in the low-dimensional space. Based on our results and further analysis, a concise set of guidelines regarding the throughput, strength and weakness of each technique can be put together to assist researchers in selecting the best fitting dimensionality reduction methods for the analysis of their single cell data. This will not only be applicable in the field of single cell genomics but also on other high-dimensional real-data such as text mining, face recognition.

I Introduction

1.1 The cell, the basic unit of life

To understand a cell – the basic unit of life- we must determine multiple factors that affects its identity (Wagner *et al*, 2016). These factors include the cell's spatial context, the type of a cell positioned in a hierarchical taxonomy of cell types and the state of a cell arising from multiple time-dependent processes that take place simultaneously. These processes can be either temporal progressions that are unidirectional e.g. during differentiation, or vacillating processes that can be oscillatory e.g. cell-cycle or circadian rhythms, or transitions between cell states with no predefined order. The spatial context of a cell refers to its absolute location in the tissue and the identity of its neighboring cells. Regarding the spatial context, the cell's response to signals from its local environment need to be considered as well (Wagner *et al*, 2016) (Figure 1).

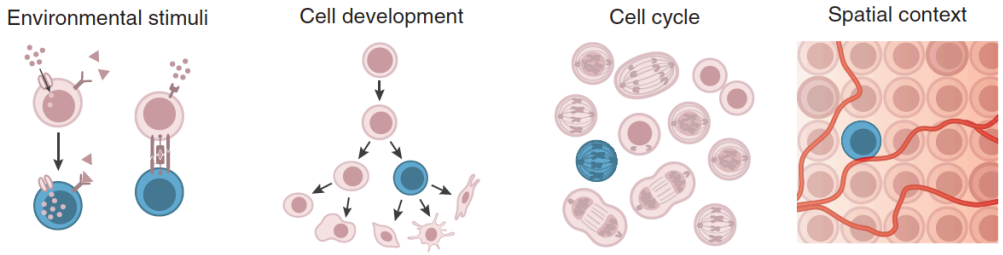

Figure 1: Factors that determine the cell's identity. A cell participates simultaneously in multiple biological contexts. The illustration depicts a particular cell (blue) as it experiences multiple contexts that shape its identity simultaneously (from left to right): environmental stimuli, such as nutrient availability or the binding of a signaling molecule to a receptor; a specific state on a developmental trajectory; the cell cycle; and a spatial context, which determines its physical environment (e.g., oxygen availability), cellular neighbors, and developmental cues (e.g., morphogen gradients). (Wagner *et al*, 2016)

1.2 Transcriptome analysis at the beginning

The cell's identity is manifested in its molecular contents (Wagner *et al*, 2016). In other words the development and its identity is driven and controlled by temporal and spatial changes in gene transcription, followed by translation of the resulting mRNAs into proteins (Tang *et al*, 2009). To infer information on the cells identity a transcriptome profile of the expressed genes must be composed. The transcriptome is broadly defined as the entire RNA component of an individual cell, or it is narrowly and practically defined as the polyadenylated products of RNA polymerase II activity (Wang *et al*, 2010; Tang *et al*, 2009).

1.2.1 Microarrays

During the last two decades various technologies have been developed to deduce and quantify the transcriptome, including hybridization-or sequence-based approaches.

Its origin can be traced back to pioneering experiments that allowed the detection of gene expression of pooled cells by microarrays (Park *et al*, 1995; L. *et al*, 2008; Tang *et al*, 2011). This hybridization-based approach typically involves incubating fluorescently labeled cDNA with custom-made microarrays or commercial high-density oligo-microarrays coated with probes (Wang *et al*, 2010) (Figure 2). Although this method is powerful and whole-genome gene expression patterns can be obtained in a high-throughput and inexpensive way, microarrays suffer from high background signals due to cross-hybridization (Okoniewski & Miller, 2006) and have a limited dynamic range of detection owing to both background and saturation of signals. Also the comparison of expression levels across different experiments is often difficult and can require complicated normalization methods. Moreover, prior knowledge of the genome sequence is required and therefore the method can only be used to detect known genes (Wang *et al*, 2010).

1.2.2 RNA-seq

With the emergence of "next-generation" DNA sequencing (NGS) methods such as RNA sequencing termed as RNA-seq, transcriptomics really took-off (Wang *et al*, 2010; Nagalakshmi *et al*, 2010). In general, RNA is extracted from a population of cells and total or fractioned RNA is converted to a library of cDNA fragments with adaptors attached to one or both ends (Figure 2). Each molecule, with or without amplification, is then sequenced in a high-throughput manner to obtain short sequences from one end (single end sequencing) or both ends (pair-end sequencing) (Figure 2). The reads are typically 30–400 bp, depending on the DNA-sequencing technology used. RNA-Seq uses recently developed deep-sequencing technologies. In principle, any high-throughput sequencing technology e.g. Illumina, SOLiD and Roche life sciences can be used for RNA-sequencing (Wang *et al*, 2010).

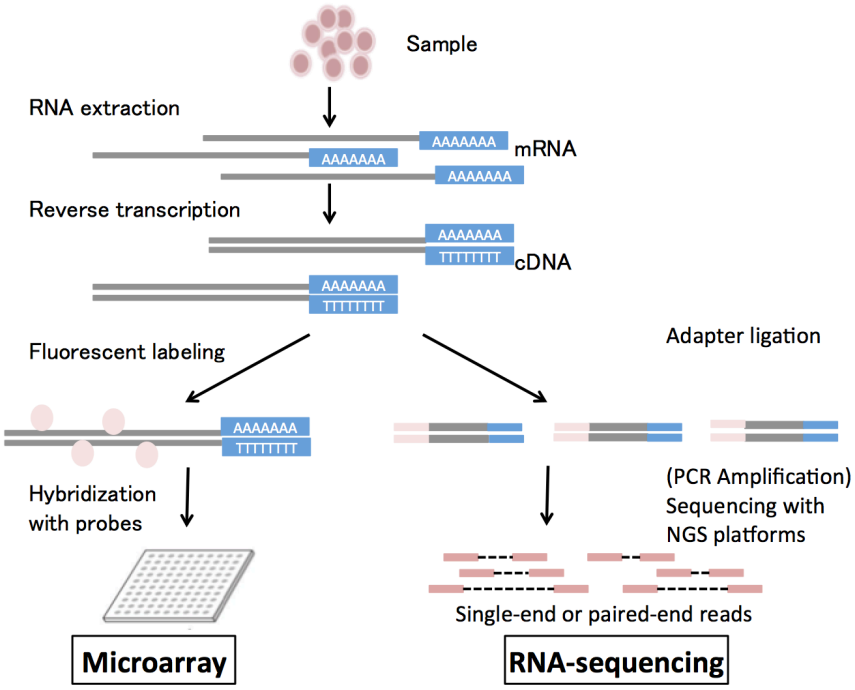

Figure 2: General workflow of traditional transcriptome analyzing methods: Microarray and RNA-sequencing.

1.2.3 Limitations of traditional transcriptome analysis

The ability to derive genome-wide mRNA expression data from a population of cells has proven to be useful in thousands of studies over the past three decades (Bacher & Kendziorski, 2016). However, traditional expression experiments such as microarrays and bulk RNA-seq are limited in providing gene expression measurements that are averaged over thousands of single cells from a sample (Bacher & Kendziorski, 2016). Distributions in single cells demonstrate that only a negligible portion of cells express mRNAs close to the estimated mean levels (Munsky *et al*, 2012). Depending on the skewness of the distributions, this could mean that cells that express certain mRNAs at outlier levels are functionally important, yet remain undetected by traditional experiments. In other words given the heterogeneity of a cell population, measurements of the mean values of signals can mask or misrepresent signals of interest. While tumor heterogeneity can be attributed to accumulated mutations, even genetically identical cells under the same conditions, display high variability of gene and protein expression levels (Munsky *et al*, 2012). This is usually referred to as biological noise or transcriptomic stochasticity (Hebenstreit, 2012). A number of studies have probed into the origins and mechanisms of that noise and found it to be mostly due to the stochastic effects associated with the low numbers of involved molecules and discovered a strong connection between noise and gene regulation mechanisms such as transcriptional bursts (Munsky *et al*, 2012; Hebenstreit, 2012; Maheshri & Shea, 2007). As a consequence, only partial information of the molecular state of biological systems is provided by these cell-averaging experiments (Shapiro *et al*, 2013; Kim *et al*, 2015). A better approach is to investigate the expression profiles at single cell level. Flow cytometry and imaging techniques have been instrumental tools in profiling and characterizing single cells in a high-throughput manner for the last decade. Flow sorting can separate heterogeneous cell populations into their constituent cell types, but this approach requires a priori knowledge of all cell types present, and good markers for these cell types. Often, neither of these are available (Junker & vanOudenaarden, 2015).

1.3 Single cell RNA seq, a revolutionary technology

To overcome these limitations, scientists have now moved forward to the emerging single cell RNA sequencing (scRNA-seq) technology that provides the expression profiles at the cell level and therefore circumventing the average artifact of traditional bulk RNA-seq experiments. Single cell transcriptome sequencing has initially been applied by the Surani laboratory in 2009 (Tang *et al*, 2009). Since then many single cell mRNA sequencing protocols and platforms have been developed. Currently published scRNA-seq protocols all follow the same general workflow: the isolation and lysis of single cells, reverse transcription of captured RNA into cDNA and the amplification of cDNA to generate libraries used for high-throughput sequencing and downstream analysis (Liu & Trapnell, 2016). Figure 3 out (Ziegenhain *et al*, 2017)provide a comprehensive review of individual scRNA-seq protocols and their relative strengths and weaknesses.

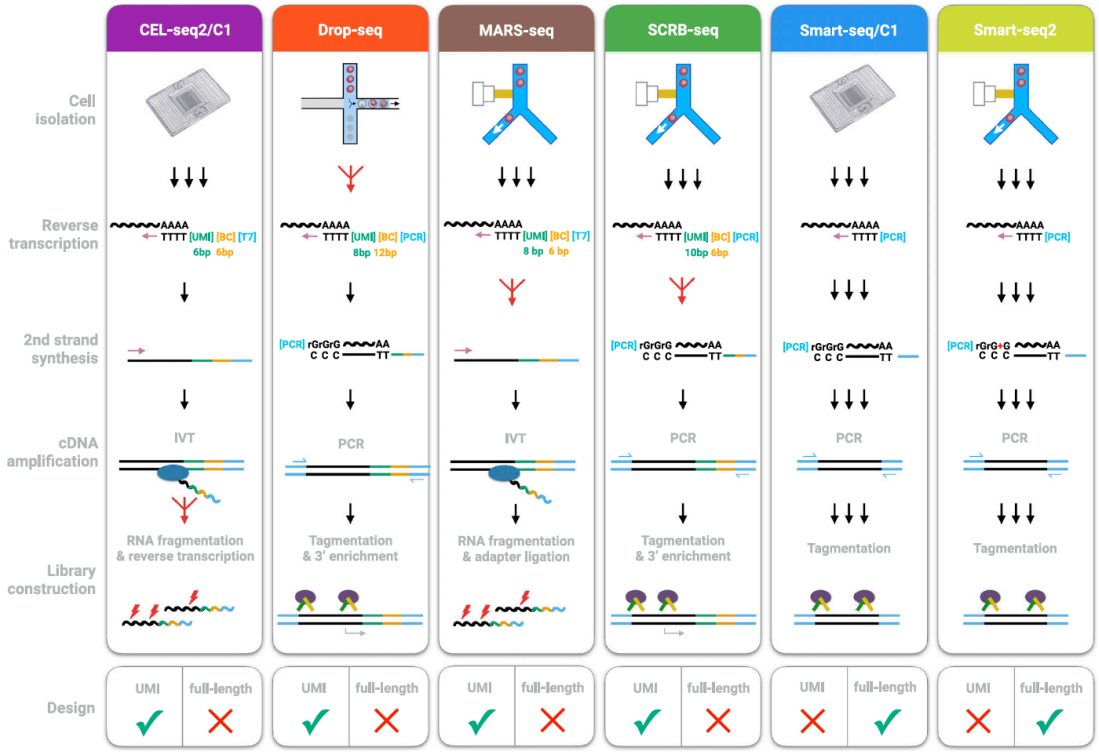

Figure 3: Schematic overview of library preparation steps of recent developed single cell RNA-sequencing platforms. (Ziegenhain *et al*, 2017)

1.3.1 Distinct features of scRNA-seq protocols

The data obtained from scRNA-seq are often structurally identical to those from a bulk expression experiment (some K million mRNA transcripts are sequenced from n samples or cells). However the relative small amount of starting material and increased resolution give rise to distinct features in scRNA-seq data, including an abundance of zeros (both biological and technical), increased variability, and complex expression distributions (Bacher & Kendziorski, 2016). Moreover during pre-amplification, necessary due to the low amount of RNA captured from each cell, substantial technical variation arises (Liu & Trapnell, 2016). Technical variability includes sensitivity (i.e., the probability to capture and convert a particular mRNA transcript present in a single cell into a cDNA molecule present in the library), accuracy (i.e., how well the read quantification corresponds to the actual concentration of mRNAs) and the precision (i.e., the technical variation of the quantification) of a scRNA-seq protocol (Ziegenhain *et al*, 2017). Methods that have a high sensitivity permits the detection of very weakly expressed genes, whereas high accuracy suggests that detected variations in expression reflect true biological differences in mRNA abundance across cells, rather than technical factors (Svensson *et al*, 2017). The sensitivity and accuracy of an scRNA-seq experiment depends on the choice of various protocols and platforms available for scRNA-seq. Each platform vary substantially with respect to the capacity, cost and time due to small deviations in their experimental design for the isolation and amplification of single cells (Bacher & Kendziorski, 2016) (Figure 3). Some protocols have sacrificed full-length coverage in order to sequence part of the primer used for cDNA generation. This enabled early

barcoding of libraries (i.e. the incorporation of cell-specific barcodes) and incorporation of unique molecular identifiers () (Ziegenhain *et al*, 2017) (Figure 3). UMIs are used to barcode individual RNA molecules during the reverse transcription step, allowing direct transcript counting, and many of the newer scRNA-seq protocols use UMIs to improve transcript quantification (Macosko *et al*, 2015; Klein *et al*, 2015; Jaitin *et al*, 2014). Alternatively, exogenous RNA standards such as those from the External RNA Control Consortium (ERCC) can be "spiked in" with cellular RNA to map between relative and absolute transcript counts (Jiang *et al*). Spike-ins are exogenous RNA sequences that are added in known quantities during library preparation and are assumed to be unaffected by the biological covariates. They thus constitute a well-defined set of negative controls for adjusting differences in total RNA content between cells, as well as for quality diagnostics of libraries and experiments (Wagner *et al*, 2016).

1.3.2 Improving scalability of single cell genomics

Besides advanced sensitivity and reduced technical noise, scRNA-seq protocols have also been improving in the recent years in terms of throughput and scalability. Sacrificing full-length coverage allowed multiplexing of cDNA amplification and thereby increasing the throughput of scRNA-seq library generation by one to three orders of magnitude (Ziegenhain *et al*, 2017). Whereas most earlier methods have been limited to measuring hundreds or thousands of cells at a time, recent advancements in micro well and droplet-based (Macosko *et al*, 2015; Klein *et al*, 2015) cell-barcoding strategies have enabled the analysis of tens of thousands of cells in a single experiment (Liu & Trapnell, 2016). Drop-Seq (Macosko *et al*, 2015) and inDrop (Klein *et al*, 2015) analyzes mRNA transcripts from tens of thousands of individual cells by encapsulating them in tiny droplets for parallel analysis. To retain a molecular memory of the cell identity from which each mRNA transcript was isolated, a molecular barcoding strategy was developed in both droplet-based approaches (Macosko *et al*, 2015).

1.3.3 General workflow of scRNA-seq analysis

In general, scRNA-seq experiments generate FASTQ files from the sequencing machine, which contain millions of reads composed of RNA sequences and optional add-on sequences (UMI tag and the cell tag etc.) (Garmire *et al*, 2016) (Figure 4). These reads need to be pre-processed before being aligned back to the reference genome. The principles of the following steps as preprocessing, quality control and alignment are similar to RNA-seq bulk experiments (Figure 4). Therefore the methods used in bulk experiments, perhaps slightly modified, can be applied directly. Before performing downstream analysis, the raw read counts should be normalized (Garmire *et al*, 2016). Normalization commonly refers to adjusting for differences in expression levels so that expression may be compared within or between samples (Bacher & Kendziorski, 2016). Normalization methods available for bulk RNA-seq experiments have been used in the majority of single cell studies. However when synthetic spike-ins and/or UMIs are available, further refinement is possible (Figure 4) (Bacher & Kendziorski, 2016). In contrast, novel statistical algorithms were required for further analysis depending on the goal of the single cell experiment (see further).

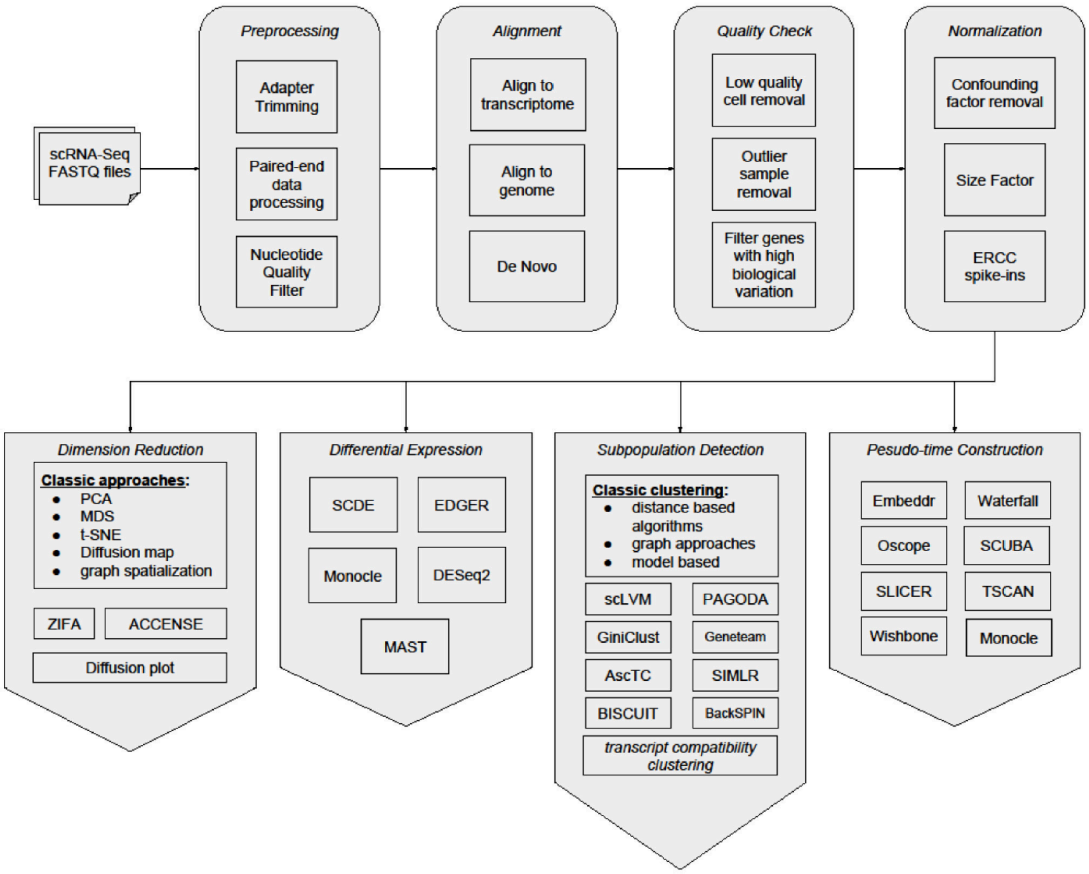

Figure 4: General workflow of the analysis of single cell RNA sequencing data. scRNA-seq experimental protocols produce FASTQ files from the sequencing machine, that contain million of reads of RNA sequences and add-on sequences (Garmire *et al*, 2016)

1.4 Applications of scRNA-seq

Large-scale single cell data allow us to address biological questions that were previously out of reach. The ultimate goal of single cell RNA-seq is to identify a cell's identity by jointly determining the factors such as cell type and cell state in time and space that define the cells identity. Following applications consider each such facet separately, however they are at least partly interdependent. We expect that, eventually, the measured genomic profiles of a cell derived from scRNA-seq experiments will be used to catalogue sources of biological variations and to characterize and quantify the relative contributions of each factor to the cell's identity all in once. This will make it possible to create a comprehensive atlas of human cells (Wagner *et al*, 2016).

1.4.1 Identification of a cell's type

Whereas cell types were traditionally defined based on criteria such as morphology, physiology, and marker protein expression, single cell analysis provide a mean of systematically detecting cellular subtypes that cannot be defined by a handful of markers, or for which markers are not yet known by clustering the high dimension in an unsupervised way (Wagner *et al*, 2016) (Figure 5). Identification of cell sub-populations in a complex mixture with scRNA-seq based on transcriptome similarities

has led to new insights in the field of microbiology (Marcy *et al*, 2007), neurobiology (McConnell *et al*, 2013), immunology and cancer research (Wang *et al*, 2012). By sequencing cells from dissociated tissues such as the spleen (Jaitin *et al*, 2014), the lung (Treutlein *et al*, 2014) and the brain (Zeisel *et al*, 2015), researchers have been able to group cells by their gene expression profiles in a completely unbiased way and to identify many novel cell types and markers along the way (Junker & vanOudenaarden, 2015).

1.4.2 Revealing dynamic processes

Another popular application of scRNA-seq is the study of dynamic transitions (Trapnell, 2015), including cell development and differentiation (Schlitzer *et al*, 2015; Ishizuka *et al*, 2016; Paul *et al*, 2015), short-term responses to environmental signals (Shalek *et al*, 2014), and cyclic/oscillatory processes such as the cell cycle (Kim *et al*, 2015). Each dynamic process is typically reflected in the cell's molecular profile, such that single cell analysis of RNA can position a cell along a temporal trajectory of the molecular profile. In the past, when studying a continuous dynamic process through bulk genomic assays, measurements at different time points were required to reconstruct the trajectory. Moreover populations of cells need to be synchronized at front. In contrast, single cell genomics provides a snapshot of the entire dynamic process. Since cells are unsynchronized, the set of single cells captured at any time point, represent different instantaneous time points along the temporal trajectory (Wagner *et al*, 2016) (Figure 5).

1.4.3 Spatial context of a cell

Single cell RNA-seq of cells from the same type of dissociated tissue can also be combined to infer the spatial location of the dissociated cells (Achim *et al*, 2015; Shalek *et al*, 2014) (Figure 5). The spatial context of a cell and its physical position to neighboring cells are critical to their function. As mentioned above, classical deep-sequencing methods as Hi-Seq and recent high-throughput droplet-based technologies (Macosko *et al*, 2015; Klein *et al*, 2015) allowing barcoding and sequencing of tens of thousands cells in a single experiment, has enabled the profiling of entire tissues and organs at a greater depth but also the reconstruction of complex 3D architecture of entire embryos or organs (Achim *et al*, 2015; Junker *et al*, 2014; Satija *et al*, 2015; Faridani & Sandberg, 2015). This application will become more important for projects as the human cell atlas projects that analyze millions of cells to infer the properties and location of all cell types in the human body (Regev *et al*, 2017).

1.4.4 Dissection of transcription mechanics

A fourth application of single cell RNA genomics is the dissection of transcription mechanics. Growing evidence suggests that genes are not transcribed continuously but rather undergo short bursts of transcription interspersed with silent intervals (Liu & Trapnell, 2016). ScRNA-seq can be used to explore transcriptional mechanism and to model the kinetics of stochastic gene transcription (Kim & Marioni, 2013; Featherstone *et al*, 2015; Daigle *et al*, 2015).

1.4.5 *Discovering gene regulatory networks*

Finally the inherent gene expression variability between cells in scRNA-seq data can be used to infer gene regulatory networks (GRN) (Liu & Trapnell, 2016). Elucidating the structure and function of transcriptional regulatory networks is a central goal of numerous studies. The GRN problem can be viewed as a graph in which genes or transcripts represent nodes and edges represent interactions among nodes (Bacher & Kendziorski, 2016). Most commonly, genes that interact with each other are grouped into co-regulated "modules" on the basis of expression profile similarity (Liu & Trapnell, 2016). In contrast to scRNA-seq, bulk studies do not provide information about regulatory relationships among nodes. To do so, temporal or perturbation experiments are typically required. As explained above, it is possible to derive a pseudo-temporal ordering from snapshot scRNA-seq experiment, which can be combined with traditional methods for regulatory network reconstruction to infer regulatory relationships among genes (Moignard *et al*, 2015; Ocone *et al*, 2015).

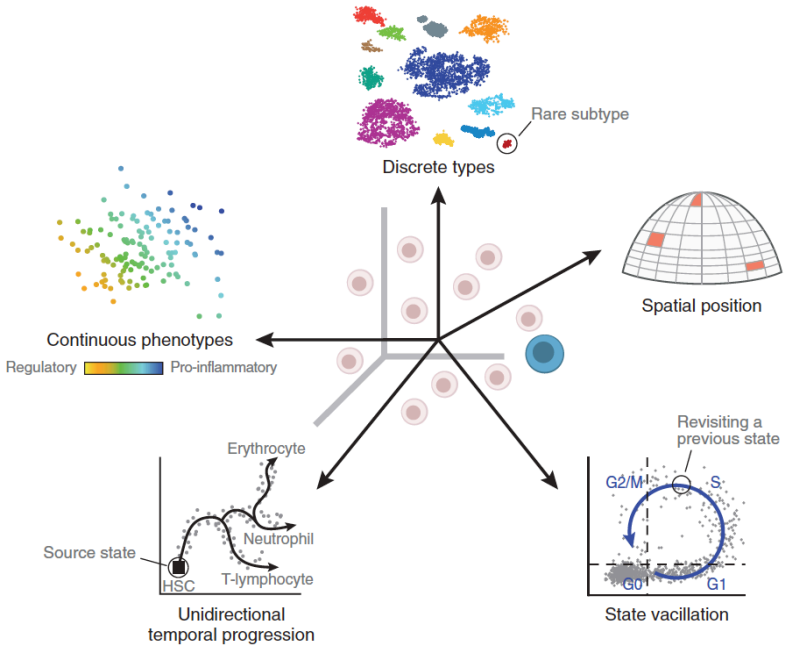

Figure 5: Applications of single cell RNAseq. The biological factors affecting the cell combine to create its unique, instantaneous identity, which is captured in the cell's molecular profile. Computational methods dissect the molecular profile and tease apart facets of the cell's identity. Key examples include (counterclockwise from top): (1) discrete cell types (e.g., cell populations in the retina; cell type frequency can vary by multiple orders of magnitude from the most abundant to the rarest subtype; (2) continuous phenotypes (e.g., the pro-inflammatory potential of each individual T cell, quantified through a gene expression signature derived from bulk pathogenic T cell profiles; (3) unidirectional temporal progression (e.g., normal differentiation, such as hematopoiesis); (4) temporal vacillation between cellular states (e.g., oscillation through cell cycle; (5) physical location (e.g., a cell's location during embryo development determines its exposure to morphogen gradients. Dividing an organ into discrete spatial bins, combined with independent data on landmark genes, allows inference of spatial bins (highlighted) from which single cells had likely originated. The scatterplots represent single cells (dots) projected onto two dimensions (e.g., first two principal components or using t-SNE). Wagner *et al.* 2016

1.5 Computational methods for scRNA-seq analysis

Innovative, efficient, robust and scalable computational analysis methods were essential to deliver information about the cell type and cell state of dynamic processes out single cell data (Wagner *et al*, 2016). Pioneering work over the past few years has provided an initial toolbox of computational methods and algorithms to infer these facets of cellular identity from scRNA-seq datasets. Several of these analytical methods are viewed in Figure 4.

1.5.1 Clustering high-dimension to identify subtypes

Classification of cells into discrete cell types from single cell profiles corresponds to an unsupervised clustering problem in which datapoints are grouped into clusters reflecting subsets of data points that are more similar to each other then to the remaining datapoints (Junker & vanOudenaarden, 2015). Clustering in high-dimensional space is obstructed by the instability of distance metrics in high dimension, partial due to the curse of dimensionality. As a result, dimensionality reduction with linear or nonlinear approaches has been used extensively as an initial step (Wagner *et al*, 2016) (Figure 4). Among linear approaches, principal component analysis (PCA) and its variants (e.g. Kernel PCA) are commonly applied in different studies (Garmire *et al*, 2016; Treutlein *et al*, 2014; Satija *et al*, 2015; Trapnell *et al*, 2014a). Also the non-linear technique t-SNE has beautifully visualized heterogeneity within the retina (Macosko *et al*, 2015) or the hippocampus (Zeisel *et al*, 2015). Among the dimension reduction methods, Zero-inflated factor analysis (ZIFA) (Pierson & Yau, 2015) is a new method that reduces the dimension of scRNA-seq datasets and allows the probability of each gene expression to be zero (Figure 4). Experiments in the original study suggest that ZIFA is a more robust alternative to PCA (Garmire *et al*, 2016). Also SIMLR (Wang *et al*, 2017) is a new clustering method that learns a cell-to-cell similarity matrix that best fits the structure of the data (Figure 4). The distance function is a linear combination of several distance metrics. The learned distance matrix is then provided to t-SNE for dimensionality reduction, followed by clustering and visualization (Wagner *et al*, 2016; Garmire *et al*, 2016). Importantly, the reduced dimensionality data are less noisy than the high-dimensional data bust lose some of the biological variance.

1.5.2 Trajectory inference for dynamic processes

Another application of scRNA-seq that required new computational analysis was the trajectory inference of dynamic processes as previous explained. Pioneering computational methods recovered the temporal ordering by creating a graph that connects cells by their profiles' similarity and finding an optimal path on this graph. This path introduces the notion of 'pseudo-time', meaning a scalar measure of a cell's progress along the temporal trajectory (Wagner *et al*, 2016). Monocle (Trapnell *et al*, 2014b) was one of the first bioinformatics tool to infer the temporal ordering of single cells (Figure 4). It reconstructs a tree describing the biological process and assigns each cell a pseudo-time. It first uses Independent Component Analysis (ICA) to reduce the dimension, then computes a Minimum Spanning Tree (MST) on the graph constructed by Euclidean distance between cell pairs and assumes that the path

through the MST corresponds to the main temporal trajectory (Garmire *et al*, 2016; Wagner *et al*, 2016). Another similar method, Waterfall (Shin *et al*, 2015) (Figure 4), uses PCA coupled with k-means to produce clusters, and then connects the cluster centroids with MST. A recent in house developed method SCORPIUS (Cannoodt *et al*, 2016), uses first multidimensional scaling (MDS) to reduce the dimensionality, followed by k-mean clustering to construct the initial trajectory. The initial trajectory is subsequently refined in an iterative way using the principal curves algorithm. (Cannoodt *et al*) provides a comprehensive overview of the current applied trajectory inference algorithms.

1.6 Dimensionality reduction

Most machine learning and data mining techniques may not be effective for high-dimensional data, partly due to the *curse of dimensionality* (Yu *et al*). High-dimensional data appear in numerous disciplines of science, from signal processing to bioinformatics. Real-world data, such as speech signals, digital photographs, face recognition or fMRI scans, usually has a high-dimensionality (Van Der Maaten *et al*, 2009). Moreover omics studies on the NGS platforms have generated petabytes of 'Big data', especially the recent high-throughput scRNA-seq datasets (Yu & Lin, 2016).

1.6.1 The curse of dimensionality

The term *curse of dimensionality* was first defined by (Bellman, 1961) as the sample size needed to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables (Carreira-Perpinan, 1996). In other words when the dimensionality increases, the volume of space augments so fast that the available data become sparse. (Scott & Thompson) call this phenomenon the *empty space phenomenon*. The curse of the dimensionality has consequences for the density and distance between points estimations: since most density estimation methods are based on some local average of the neighboring observations (Silverman, 1986), in order to find enough neighbors in high-dimensional spaces, the

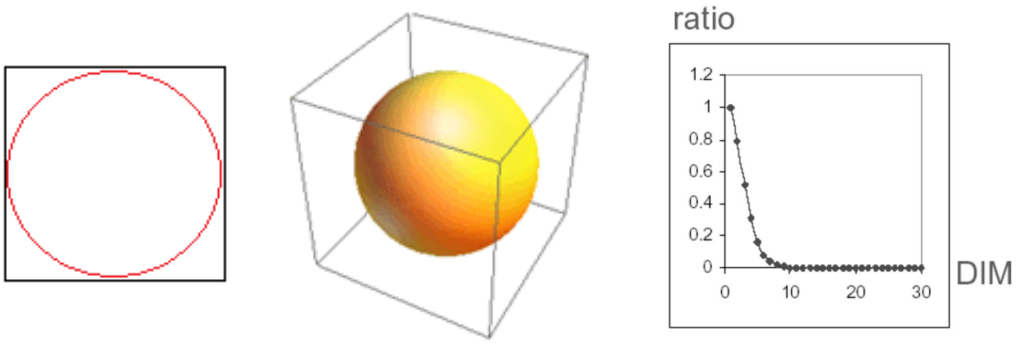

Figure 6: Empty space phenomena. From left to right: **Left** red circle represents the biggest circle possible in a 2-dimensional black Square. **Middle.** Yellow sphere representing biggest sphere possible in 3D black cube **Right.** graph where the ratio of covered volume by the biggest radius is plotted against the number of dimensions of the volume. The volume covered drops exponentially from 78% in 2D, to 52% in 3D to even 0.24 % in 10D, illustrating that high-dimensional space is sparse and the corners are more important. (Saeys, 2016)

neighborhood has to reach out farther and the locality is lost. As illustration, in a 2-dimension, we can imagine that two points are near if one falls within a certain radius of another. Consider the left image of figure 6, we notice that 78% of the black square is covered by the biggest circle. Yet the biggest sphere possible inside a 3D-cube covers only 52% (Figure 6.B). The volume reduces even exponentially to 0.24% for just 10-dimensions, meaning counter intuitively that the data points lie near the corners in a high-dimensional space then in the center. This has large consequences for distance calculations in cluster algorithms. To avoid this problem, the amount of data or samples needed to support the result is required to grow exponentially with the number of variables, which is often impossible. Therefore an important step in downstream analysis of high-dimensional data such as scRNA-seq experiments is dimensionality reduction.

1.6.2 The principle of a dimension reduction

Dimensionality reduction transforms high-dimensional data into a meaningful representation of reduced dimensionality (Van Der Maaten *et al*, 2009). Ideally, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data. The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data. In other words the intrinsic dimensionality is the minimum number of dimensions that represent a manifold on which the original data is embedded. Dimensionality reduction reduces the amount of memory and time required by data mining algorithms and it allows the data to be more easily visualized (Yu *et al*). It may also help to eliminate irrelevant features and noise out the data. Dimensionality reduction methods can be subdivided in two subgroups: feature selection when a subset of the original features set is selected or feature extraction when a new set of features is built based on the old feature set. This last is also called feature transformation.

1.6.3 Linear and non-linear techniques

Traditionally feature transformation was performed using linear techniques such as Principal component analysis (PCA) and classical scaling (Van Der Maaten *et al*, 2009). PCA is the dimensionality reduction technique most widely used in practice, due to its conceptual simplicity and that relatively efficient algorithms exist for its computation (Carreira-Perpiñán, 1996). PCA converts a set of correlated variables into a set of orthogonal uncorrelated variables, termed principal components. These principal components are ordered by the fraction of the total variance they explain, and usually only the first two or three principal components are analyzed (Grün & van Oudenaarden, 2015). For distance metrics employed by these linear dimensionality reduction methods, Euclidean distance, Pearson and Spearman correlation coefficients have been popular choices (Garmire *et al*, 2016). A main issue of scRNA-seq analysis is that gene expression data cannot be expressed as a linear combination of the relationship between two cells in general. Also classical similarities (such as cosine and Euclidean distances) are less meaningful as the dimensionality increases and may not be appropriate for scRNA-seq (Garmire *et al*, 2016). Therefore several non-linear dimensionality reduction methods have been developed (Van Der Maaten & Hinton, 2008). Many of these methods were reviewed by (Lee & Verleysen,

2007). Previous studies have shown that nonlinear techniques outperform their linear counterparts on complex artificial tasks, but on natural datasets the results are less convincing (Van Der Maaten *et al*, 2009). An frequently used nonlinear technique at the moment is t-SNE (t distributed Stochastic neighborhood embedding)(Van Der Maaten & Hinton, 2008) which is capable of capturing much of the local structure of the high-dimensional data, while also revealing global structure. However unlike PCA, t-SNE does not learn an explicit mapping between the high- and low- dimensional spaces; meaning points that are close in the high-dimensional space will be close in the low-dimensionality embedding, but more global relations are not directly interpretable (Wagner *et al*, 2016).

1.6.4 Landmarks as an alternative

The recent concern in dimensionality reduction techniques and manifold learning is due, in part, to the multiplication of very large datasets of high-dimensional data from various fields of research (Silva & Isr, 1950), with focus on the increasing interest in a human cell atlas that would infer datasets that contain tens of thousands cells to even million of cells (Regev *et al*, 2017). As the existing algorithms have quadratic complexity in number of observations, the computational demand is expected to become infeasible. Selecting a subset of the datapoints termed as landmarks to perform the embedding on has been proposed to circumvent the computational burden. Several landmark selection methods as naïve and minmax have been proposed and have been implemented with success in Isomap and LLE (Chi & Melba, 2012; Shi *et al*, 2015; Silva & Tenenbaum, 2004).

1.7 Challenging problems of scRNA-seq

Based on the annual growth of scRNA-seq datasets uploaded to the NCBI Gene Expression Omnibus (GEO) database (Edgar *et al*, 2002) and the increasing number of new articles in PubMed over the past seven years that involve scRNA-seq and big-data (Figure 7), an explosive growth of scRNA-seq data is expected (Yu & Lin, 2016). While studies convincingly demonstrates that single cell mRNA sequencing is a powerful tool and that they harbor a wealth of information, they also pose specific analytical and technical challenges (Wagner *et al*, 2016). Computational methods to leverage the full complexity within single cell transcriptome data are just beginning to emerge. Without a doubt, the detailed and extremely valuable information that single cell technology provides is at a significant cost due to sophisticated data acquisition, large data-storage requirements, as well as challenging data processing and management (Yu & Lin, 2016). First of all NGS data has become one of the largest big-data domains in terms of data acquisition, storage, and distribution. Just like bulk-cell RNA-seq and other NGS-based studies, scRNA-seq generates a high volume of raw sequencing data and high-dimensional expression data (Yu & Lin, 2016). Moreover using the latest protocols (Macosko *et al*, 2015; Klein *et al*, 2015) for massively parallel scRNA-seq, a single laboratory can now readily collect tens of thousands to hundreds of thousand single cell RNA-seq profiles. The magnitude of single cell data is expected to increase in the near future to million of cells in new projects as the Human Cell Atlas (HCA) (Regev *et al*, 2017). Data at this scale present additional difficulties in basic processing

such as calculating a covariance matrix and in assessing statistical significance and robustness of results. It is also a challenge to effectively visualize such magnitudes of data. As previously mentioned, the data volume of scRNA-seq is higher than that of bulk-cell RNA-seq. Consequently, high data-transfer bandwidth, parallel algorithms, and high-performance computers are required to generate and process data. Despite the progress in experimental and computational methods to leverage the characteristics of single cell data, the field is still in its infancy. An important task for the future is to define a set of best practices through comparisons of different statistical methods and experimental platforms. We anticipate substantial development of computational methods to tackle challenges such as the growing scale of the data and the need for effective visualizations (Wagner *et al*, 2016). With the increasing popularity of single cell assays and ever increasing number of computational methods developed, these methods need to be more accessible to research groups without bioinformatics expertise.

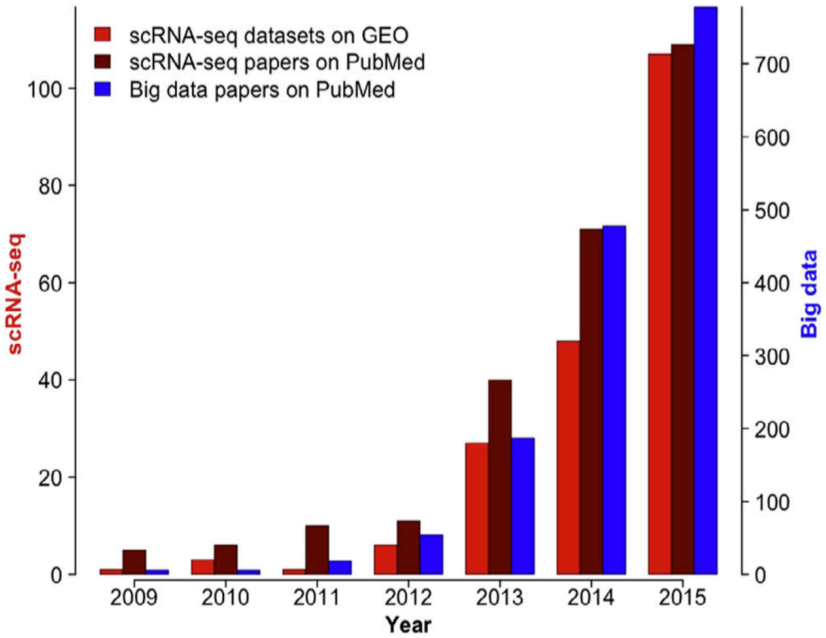

Figure 7: Number of papers/datasets addressing single cell data and big data. Searches were performed on January 04, 2016 on nlm.nih.gov/gds for datasets and http://www.ncbi-.nlm.nih.gov/pubmed for papers. Data were obtained according to the search criteria as follows filtered by year: (1) for scRNA-seq datasets on GEO: ‘‘single cell” AND ‘‘Expression profiling by high throughput sequencing; (2) for scRNA-seq papers on PubMed: ‘‘single cell” AND (‘‘rna-seq” OR ‘‘rna sequencing “OR (‘‘sequencing” AND ‘‘transcriptome”)); and (3) for big-data papers on PubMed: ‘‘big data” OR ‘‘hadoop”. (Yu & Lin, 2016)

II Aim of Research Project

2.1 The two sides of scRNA-seq

The revolutionary technology scRNA-seq provides the transcriptome profiles at single cell level and therefore circumvents the averaging artifact of traditional bulk experiments. ScRNA-seq allows us to address biological questions that were recently out of reach. Thus far, single cell RNA-sequencing has already shown great effectiveness in unrevealing complex cell populations, reconstructing developmental trajectories, and modeling transcriptional dynamics. Though single cell experimental technologies have been improving in accuracy and scalability for the past years, computational and statistical methods to leverage the full complexity of single cell data are just beginning to emerge and substantial development to tackle challenges such as data acquisition, data processing and data storage will be required (Yu & Lin, 2016). An important technique during the analyses of single cell data is the dimensionality reduction. Just like bulk-cell RNA-seq, scRNA-seq generates high-dimensional data. To perform downstream analysis a dimensionality reduction of the single cell data to a representable low-dimensional space is required in order to remove biological and technical noise, to compress high-dimensional data and to visualize the data in an efficient way.

2.2 Challenging problems for dimensionality reduction

2.1.1 The cost for the increasing data size

Whereas earlier techniques to isolate and analyze single cells have been limited to hundreds of cells at a time, recent advancements in droplet-based and cell-barcoding strategies enables high-throughput sequencing of thousand to tens of thousands of cells in a single experiment (Liu & Trapnell, 2016). New projects as the human cell atlas project will even generate datasets of million cells (Regev *et al*, 2017). With the advent of these technologies, data at this scale impose computational difficulties in basic calculations of a covariance or distance matrix during a dimensionality reduction and will lead to an infeasible computational demand. As a consequence some of the dimension reduction techniques will require extensions or approximations to handle the computational burden (Bacher & Kendziorski, 2016).

2.1.2 The lack of a general review

Moreover with the increasing popularity of single cell assays and ever increasing number of computational methods developed, a general computational review is lacking until today. An important task is to define a set of best practices through comparison of different dimensionality reduction methods. Recently, this was not possible due to the modest size of the experiments and the little data that was available forcing bioinformatics to use their data in the inference model and the confirmatory data analysis (Norel *et al*, 2011). Today, as a result of the increasing popularity of single cell research and the new advantages in single cell sequencing and barcoding techniques, the number of single cell datasets and its sample size are

increasing rapidly. This enables to perform a comparative review of the state-of-art dimension reduction methods.

2.3 Providing a review of dimension reduction methods

In this research, we aim to review and evaluate a variety of frequently used dimensionality reduction techniques to define a set of best practices. The performance of a number of state-of-the-art dimensionality reduction methods will be compared between a set of single cell datasets using different metrics as an evaluation. Also the computational time will be taken into consideration when comparing the different methods.

The analysis will be divided into the results of the dimensionality reduction on datasets smaller then two thousand cells and datasets with a magnitude of tens of thousand cells. By comparing the performance and the computation time of the dimensionality reduction techniques between small and large datasets, we anticipate to find methods that can handle 'big' datasets in an efficient way since the computational complexity is expected to become infeasible for the new high-throughput single cell data of ten of thousand cells. Due to this expected increase in computation time, we also added a variation of MDS that use landmark points instead of the entire dataset to calculate the lower embedding. By using landmarks, landmark MDS may be able to circumvent the computational burden while performing a dimension reduction on the recent high-throughput single cell data and make it possible to perform a dimension reduction on datasets with a scale of tens to even hundred thousand cells. During the analyses we will also perform a parameter optimization. As a good performance of a method generally depends on the correct choice of parameters, parameter tuning is a necessary but often overlooked task when reviewing a set of methods. Instead of using the default parameters that are determined on a specific set of datasets leading to overfitting of the data, we perform a grid search to reduce the bias. Further we will also focus on the performances of the methods when the data is embedded into a two or-three dimensional space since a dimension reduction is often performed to visualize high-dimensional data into a two or three-dimensional space. Also the capacity of a dimension reduction to remove noise out high-dimensional data will be examined by comparing the accuracy scores of the different techniques before and after the dimension reduction.

At the end we aim to define general guidelines that are based on our results of the throughput, strength and weakness of each technique. As mentioned earlier, a dimensionality reduction method is used for different goals such as the visualization of high-dimensional data into a two or three-dimensional space or to remove biological and technical noise out data. Depending on the purpose of the research and the size of the datasets, our research can assist the growing number of researchers in the selection of dimensionality reduction methods to leverage the maximum out of their single cell sequencing data. These guidelines will not only be of use for researchers who are currently working on single cell data as our research group but also in other fields where they are dealing with highly dimensional data such as text mining, image retrieval, face recognition, protein classification…(Yu *et al*).

III Results

3.1 Evaluation workflow

Our evaluation procedure was structured as follows (Figure 8). We applied a set of 10 dimensionality reduction methods (Supplementary List S2) on 20 publicly available single cell RNA-seq datasets (Figure 8.B). The majority of these datasets were retrieved from the Gene Expression Omnibus (GEO) database and 10X genomics website (Figure 8.A.). The magnitude of the single cell datasets (Supplementary list S1) scope an order of hunderd to tens of thousands cells necessary to compare the performance of a dimensionality reduction method between small and large single cell data. In order to evaluate the dimensionality reduction methods, we scored the methods using three different quality metrics: KNN accuracy, cluster accuracy and coRanking (Figure 8.C). KNN accuracy determines the accuracy of class label predictions based on the class labels of the k-nearest neighbors, whereas cluster accuracy infer the labels on the class labels of the cells within a cluster. CoRanking metric estimates how well the distances between the cells are preserved in the low dimensional space after transformation without prior knowledge about the class labels in contrast to KNN and cluster accuracy. The computing time is also taken into consideration when comparing the different methods.

First we performed subsequent analysis on a subset of the datasets that contain at most 2000 cells followed by the analyses on the entire set of datasets in order to compare the performance and the computational time of the different dimensionality reduction methods between small and large datasets (Figure 8.D). As a good performance of a method generally depends on the correct choice of parameters, we applied a grid search to optimize the parameters for every method (Figure 8.E)(Supplementary list S3) and used leave one out cross validation (LOOCV) approach to obtain the best parameter settings for each dimensionality reduction technique. We also focused on the performances of the methods when the data is mapped into a two or three-dimensional space and how well the techniques remove noise from the high-throughput single cell data (Figure 8 F-G). Based on our results, we define a set of best practices to guide researchers in the selection of the appropriate dimensionality reduction technique during their single cell analysis (Figure 8 H).

3.2 Dimensionality Reduction methods

After a profound literature study, we made a collection of the most popular and widely used dimensionality reduction methods used for the analysis of single cell data. An overview of the ten dimensionality reduction methods including their characteristics can be found in Supplementary (Supplementary List 3). Principal component analysis PCA is an old-fashioned but still widely used method due to its simple algorithm. Through the years, different variations of PCA such as kernel PCA and other linear techniques Multi Dimensional Scaling (MDS) and Sammon has been developed. Most of the time, single cell data is not linear but lies on or near a non-

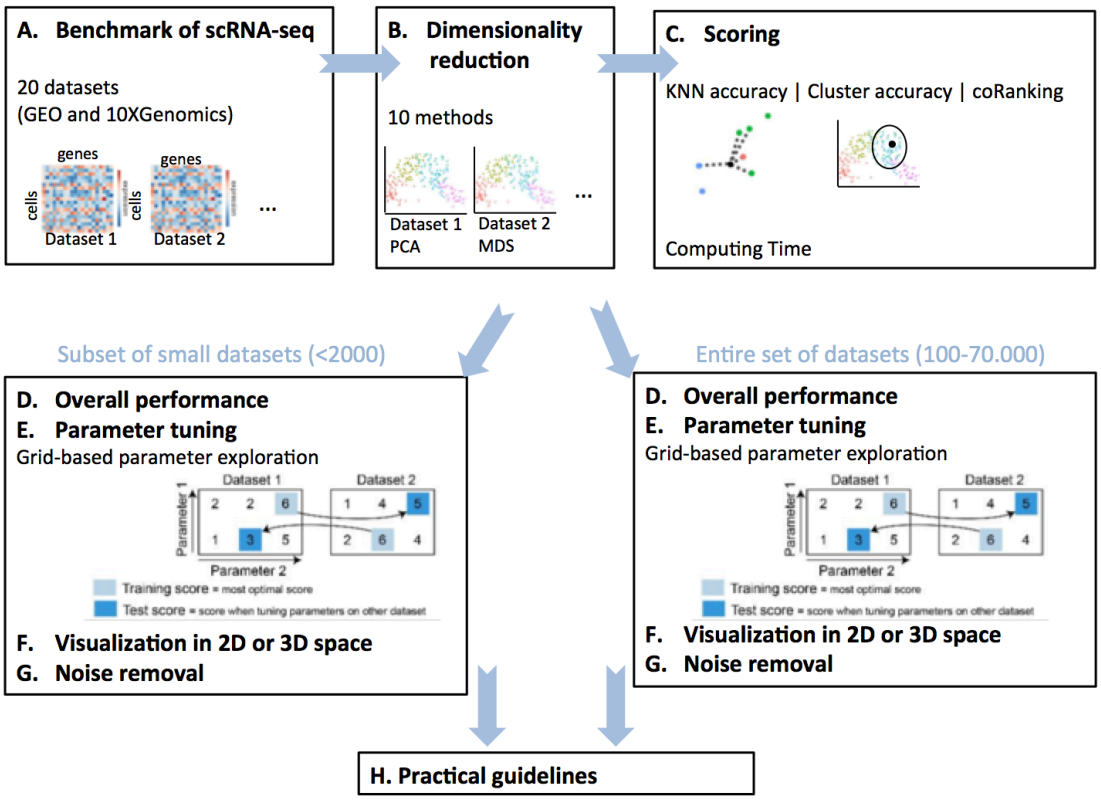

Figure 8: Overview of evaluation workflow. 10 dimensionality reduction methods were applied on 20 single cell datasets (Top left). The performance of a method is scored with KNN accuracy, cluster accuracy, coRanking metric and computing time (Top right). To investigate the strength and weakness of each method, we compare the performance of the methods on small (100 to 2000 cells) and large (2000-70000 cells) (middle). For each method a parameter optimization is executed using a grid-based approach. At the end the performance is compared between the different dimensionality reduction methods and practical guidelines are constructed (Bottom).

linear manifold. As a consequence, a wealth of non-linear methods has been developed during the last 20 years. Local Linear Embedding (LLE), isoMDS, Isomap and Diffusion Maps are some examples of these non-linear methods. A popular non-linear technique at the moment is t-Stochastic neighborhood Embedding (t-SNE) which focus on preserving local distances.

4.4.1 Landmark MDS

Despite the large amount of dimensionality reduction methods that has been and is still being developed, we expect that the computational time will become infeasible when performing previous methods on new high-throughput single cell data of tens of thousands cells produced by recently improved scRNA-seq protocols. Therefore we adapted multidimensional scaling (MDS) with landmark selection to overcome the computational burden. Instead of training the mapping on the entire set of datapoints, the manifold is developed on a small subset of the data referred as 'landmarks'. We implemented three landmark selection methods: naive, minmax and degree centrality. Naive simply selects random points as landmarks, whereas minmax select points to minimize the maximum distance between the landmark and non-landmark. We constructed and implemented degree centrality ourselves. In degree centrality, data points with the highest node degree are picked as landmarks.

3.3 A set of practical guidelines

Based on our results of the performance and computation time of the state-of-the-art dimension reduction on single cell datasets ranging from 66 to 70000 cells, we construct a set of general guidelines that can assist researchers in selecting the appropriate dimensionality reduction technique depending on the data size and purpose of their research as seen in Table 1 and Figure 9. In this section we pinpoint the most important and interesting practices of the different dimensionality reduction methods, as we will explain these results and findings more in detail in the sections 3.4 and 3.5.

3.3.1 A set of worst practices

For a start, PCA with kernel Anovadot and t-SNE require a significant larger computing time for every dataset independent on the size compared to all other methods and are therefore not recommended (Table 1 and Figure 9). Regarding the quality performance, LLE, Isomap and especially PCA with kernel rbfdot perform a dimension reduction significant inferior to all other methods independent on the size and the number of dimensions in the low-dimensional space. Also PCA kernel rbfdot, Isomap and LLE are not able to remove noise out high-dimensional but rather loose biological information when mapping the data into a low-dimensional space (Table 1 and Figure 9). Therefore are these methods never the best option to perform a dimension reduction.

3.3.2 The relation between data size and performance

Further we observed a distinct relation between the computation time of a dimensionality reduction method and the data size. The computational cost of a dimensionality reduction method is often quadratic to the number of data points, as a consequence the time general increase as the data size increase with 500 and 5000 cells as critical points. While the data contain less then 500 cells, all techniques execute a dimension reduction within a short time window independent of the data magnitude except for the methods t-SNE and Anovadot. However if the magnitude exceeds 500 cells, the computational demand of several methods such as diffusion map, isoMDS and Sammon expand significantly and continue to increase as the data size further augments (Table 1). On the contrary, methods such PCA kernel polydot, PCA kernel rbfdot, LLE and MDS perform a dimension reduction without increasing in computing time once the data size pass 500 cells. When the data size reaches 5000 cells, we observe nearly an exponential increase in the computation time of all methods with the expectation of landmark MDS. At the end, all methods eventually fail to complete a dimension reduction on datasets that contain more than 15000 cells within a reasonable time window.

3.3.3 The goals of a dimension reduction

As mentioned before, researchers use a dimension reduction to compress high-dimensional data as well as to visualize high-dimensional data into a two or three-dimensional space or to eradicate biological and technical noise from the high-dimensional data. Our results indicated that the ability to remove noise out data does

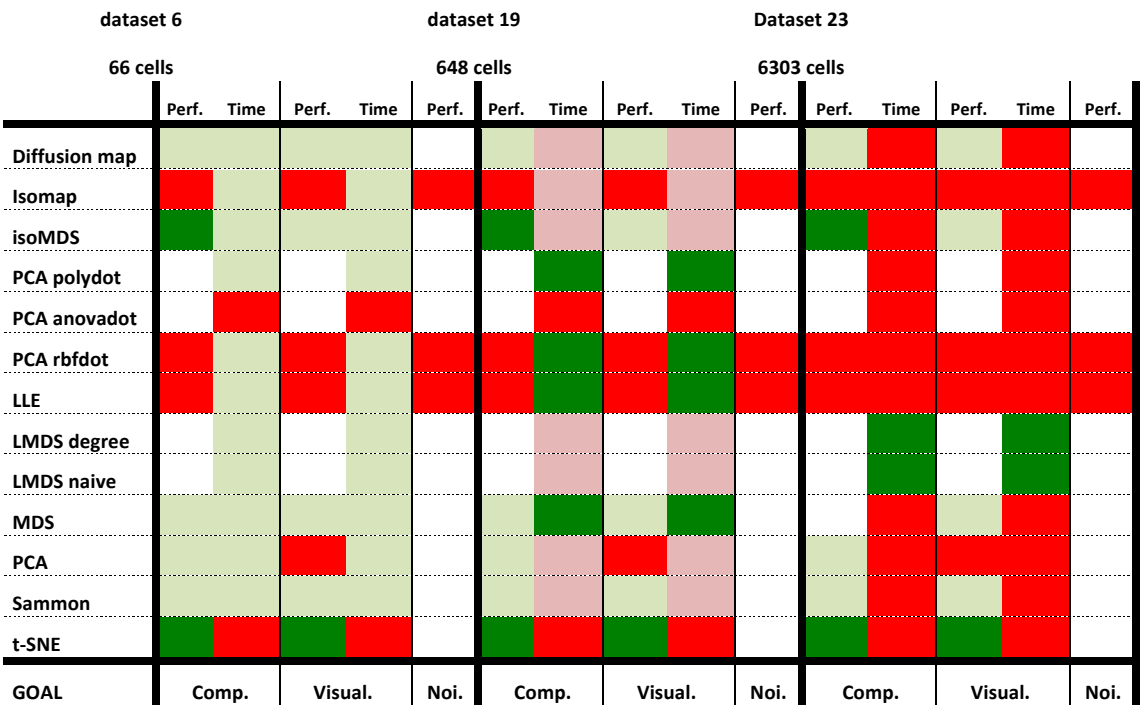

Table 1: General overview of the results from the comparative review of the state-of-the-art dimensionality reduction techniques. The performance of 10 dimensionality reduction techniques were evaluated on 20 single cell datasets based on three different quality metrics and the computational time. The scores of the three metrics are combined and scaled as seen in the columns Perf. And the computation times are shown in the columns time; three goals of a dimension reduction are studied and shown Comp.: compression of high-dimensional data, Visual.: Visualization of high-dimensional data into a 2 or 3-dimensional space, Noi.: removing biological and technical noise out high-dimensional data; the performance and computation time is strong correlated with the datasize: in general the computation increases as the data size exceed 500 cells and 5000 cells ;color scale performance: red low performance, white neutral, light green good performance, dark green very good performance; color scale time: red extreme long, light red long, light green efficient time.

not depend on the method but rather on the noise level present in the data. At front of a dimension reduction a scientist need to make a trade off between noise reduction and loss of biological information. However performing a dimension reduction with the help of PCA rbfdot, Isomap or LLE cause in general the loss of biological information (Figure 9).

Further we noticed that mapping of high-dimensional data into a two-dimensional, three-dimensional or five-dimensional space has no significant effect on the performance of the majority of dimensionality reduction methods. T-SNE is the best option to visualize the 'big' data into a two-dimensional space with as trade-off a tremendous longer computing time. In this way isoMDS, diffusion map and Sammon are able to perform a dimension reduction with a relative high performance within reasonable time as long as the datasets contain less then 500 cells. Once the magnitude exceeds 500 cells, MDS is the appropriate technique to visualize high dimensional data into a two-dimensional space, but also for the embedding in higher dimensions. Finally we noticed that PCA is able to embed the data with high accuracy

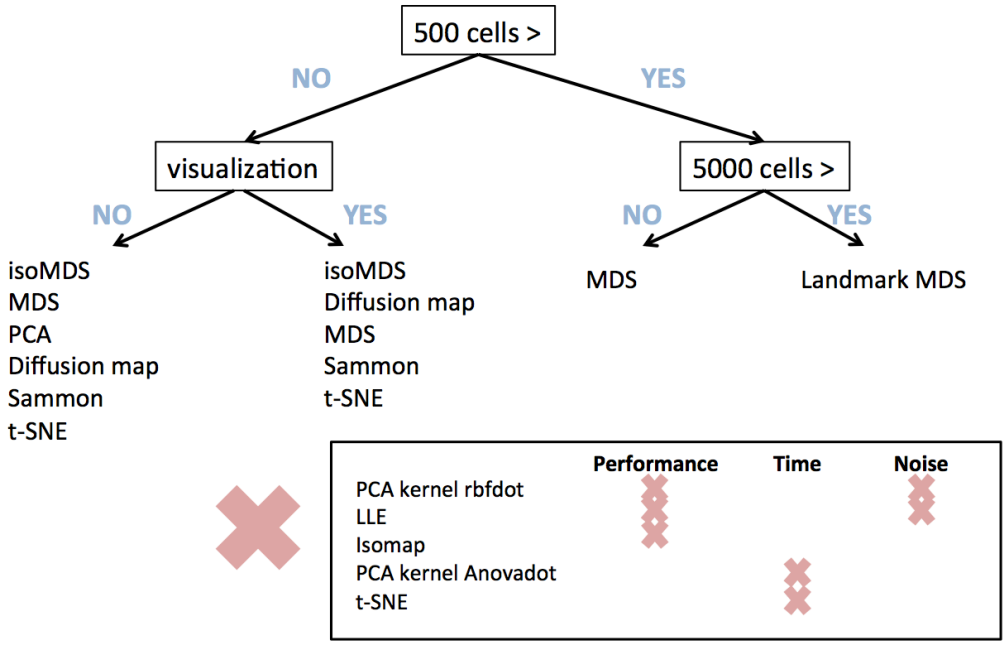

Figure 9: Practical guidelines for selecting the appropriate dimensionality reduction method based on the data size and purpose of the dimensionality reduction. The findings are based on the results from comparative review wherein 10 dimensionality reduction were evaluated on 20 single cell datasets using KNN accuracy, cluster accuracy and coRanking as evaluation.

in a five dimensional space but loose accuracy when mapped into a two-dimensional space (Figure 9).

3.3.4 Landmark MDS

Our results confirmed that the current used dimensionality reduction techniques are not competent to map high-dimensional data within an efficient time window. The execution of a dimension reduction on datasets with a magnitude of 15000 cells demanded an enormous amount of computational memory and time. As singular technique, Landmark MDS succeeded to perform a dimension reduction on datasets that contain more then 5000 cells without increasing in computation time. Despite the use of only a part of the data to train the embedding on, a minor loss in accuracy is seen. However the landmark approach is only beneficial for large datasets. The supplementary steps to select the appropriate landmarks require some computation time and are redundant if the data size is small. We implemented three different landmark selection techniques: naive, minmax and degree centrality. We noticed that minmax demanded a significant longer time to perform a dimension reduction and is therefore not recommended. Further there is no large contrast between naive and degree centrality. Both implementations have a similar performance and computational demand. Out the parameter optimization we concluded that at least 200 landmarks are necessary for an optimal performance of Landmark MDS with naive as landmark selection and at least 100 for degree centrality.

3.4 Performance on small single cell data

In this section the results are shown of the different dimensionality reduction methods with varying parameters on small datasets containing less then 2000 cells before and after parameter tuning. At front the scores of the three different quality metrics are normalized according to each dataset resulting in relative scores with a scale of 0 to 1, since the performance of a technique not only depends on the method itself but also on the quality of the data (Supplementary Figure S1-2).

3.4.1 Overall performance

In Figure 10 the normalized scores of the three quality metrics are shown for each method before parameter tuning. We took the average score over all datasets for each parameter setting of the methods. Overall, our results indicate that none of the dimensionality reduction methods outperforms the rest, thought several methods fall behind compared to the rest. PCA with kernel rbfdot has significantly the lowest scoring (Anova and Tukey HDS $p<0.05$) according to each metric (Figure 10). Also Isomap, LLE and kernel PCA polydot perform significantly less against the others (Anova and Tukey HDS $p<0.05$) (Figure 10). LLE has even a more inferior score by coRanking metric and also PCA with kernel Anovadot has a significantly lower score compared to the other methods according to the coRanking metric (Figure 10 C-F). Based on the coRanking scores diffusionMap, isoMDS, MDS and t-SNE seem to have a slightly advantage over the other methods, but this is not seen in the results of KNN and cluster accuracy.

Comparing the absolute computing time over all parameter settings between the different methods for every dataset separately, we noticed that the computation time of LMDS minmax is significantly larger for every dataset (Anova and Tukey HSD $p<0.05$) (Supplementary Figure S3). As mentioned earlier, LMDS has three different landmark selection implementations: naïve, minmax and degree centrality. Out the data, we concluded that it takes a significantly longer time executing LMDS with Minmax as landmark selection implementation (Supplementary Figure S4). Considering the purpose of LMDS to avoid the computational burden when computing the distance matrix of high-Throughput single cell data, we decided to leave out Minmax for further considerations.

In Figure 11 the absolute computing time of the different dimensionality reduction methods except LMDS minmax with their varying parameters before parameter tuning is represented for each dataset separately. The datasets are ordered from small to large. Based on statistical significance (Anova and Tukey HSD $p<0.05$) between the computing time of the different dimensionality reduction methods, we can divide the dimensionality reduction methods in four groups (Figure 11). Kernel PCA Anovadot and t-SNE require a significantly (Anova and Tukey $p<0.05$) larger computing time for every dataset compared to the other methods. We expect that the computation time of these two methods will increase drastically and eventually become infeasible when executed on new high-throughput single cell datasets of thousands to even ten of thousands cells.

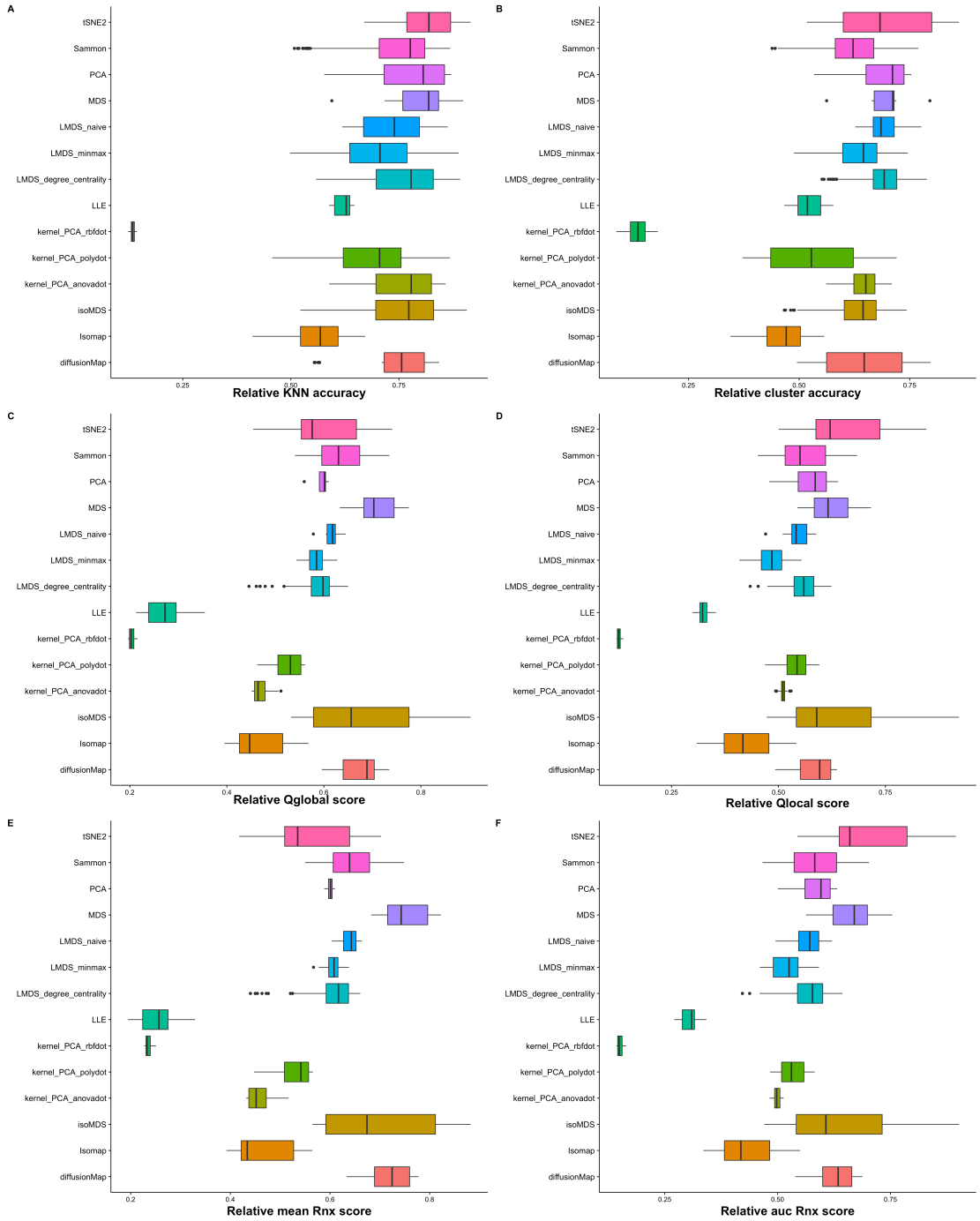

Figure 10: Overall relative performance of different dimensionality reduction methods with varying parameters before parameter tuning according to three quality metrics on small single cell datasets (<2000 cells). At front the scores are normalized at a scale of 0 to 1. The parameters of each method are varied according to a grid search. For each method the average score over all datasets of each parameter setting is represented. **A.** Relative KNN accuracy **B**. relative cluster accuracy **C-D** relative coRanking metric **C.** Relative Qglobal **D**. Relative Qlocal **E**. Relative mean Rnx **F.** Relative auc Rnx

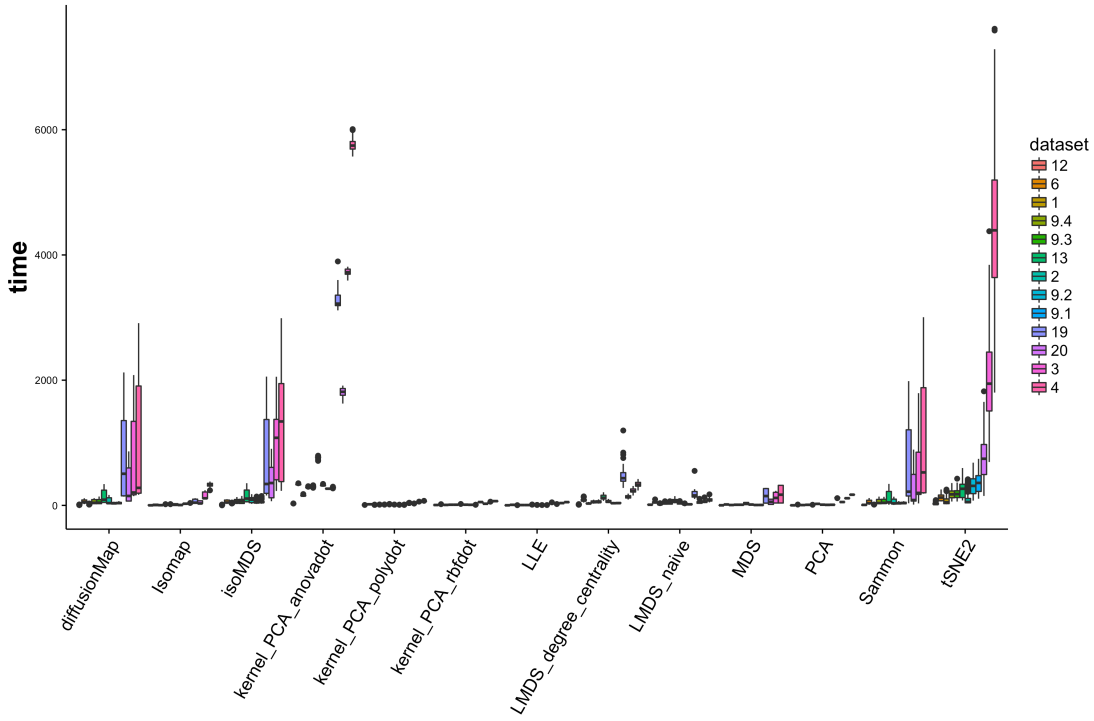

Figure 11: Absolute computing of different dimensionality reduction methods before parameter tuning on small single cell datasets (<2000 cells). The absolute computing for every parameter setting of each dimensionality reduction method is shown for every dataset separately. The datasets are ordered in size from small (dataset 12: 66 cells) to large (dataset 4: 1790 cells).

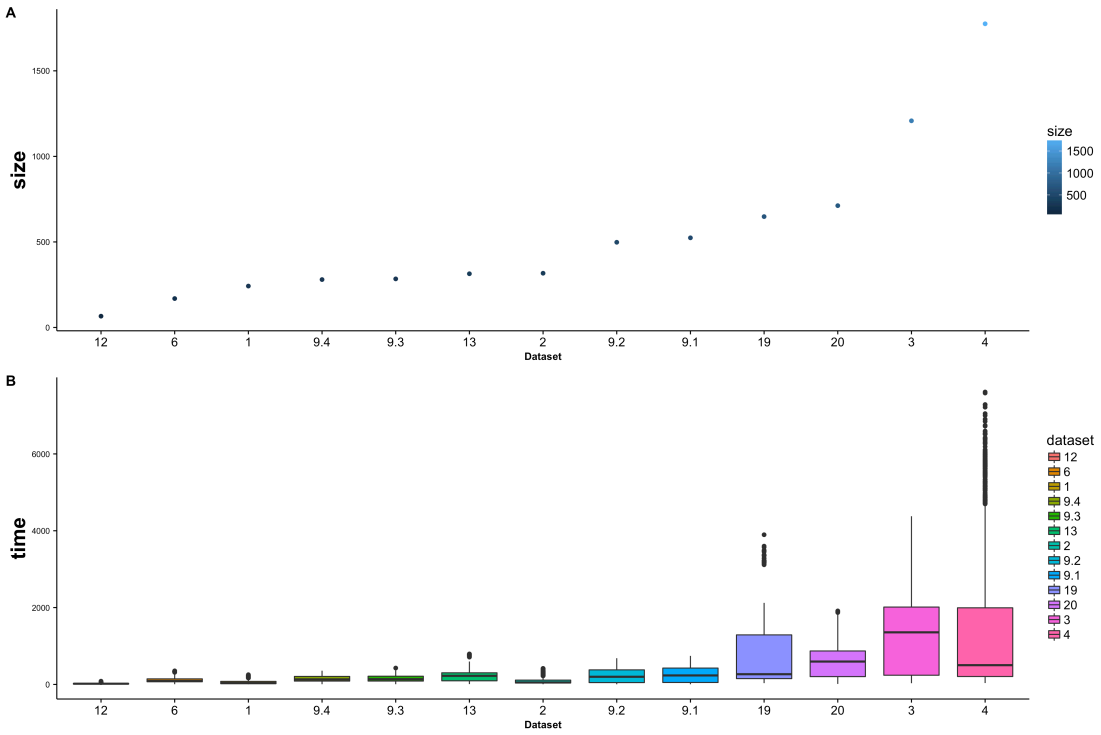

Figure 12: Relation between the size of the single cell datasets and the computational cost to execute dimensionality reduction on these datasets. A. The number of cells in each single cell dataset ordered from small to large **B.** Absolute time of all dimension reduction taken together represented for every dataset separately.

Further we observe that the computational cost increases significantly (Anova and Tukey HSD $p<0.05$) when the size of the single cell data exceeds 500 cells (dataset 19648 cells) for the methods diffusion Map, isoMDS and Sammon (Figure 11). This indicate that these methods can handle small datasets in a sufficient time but demand a larger computational time for datasets with a size larger then 500 cells. The third group includes MDS, LMDS degree centrality, LMDS naïve and Isomap. They perform the dimensionality reduction still in a sufficient time but the computational cost slightly increases as the size of the datasets exceeds 500 cells. Finally classical PCA, PCA with kernel polydot and rbfdot and LLE execute the dimension reduction in very efficient time independent on the size of the small single cell datasets. As a remark the computation time of t-SNE, diffusion Map, isoMDS and Sammon not only increase for datasets larger then 500 cells but also fluctuate more, indicating that there may be space for parameter optimization. Since t-SNE has a lot more parameters to vary, the computation time deviate significantly larger. After parameter optimization we expect that these fluctuations will decrease.

When the overall computation time of all dimensionality reduction methods for every dataset is set against the magnitude of each dataset, a distinct relation between the data size and computation time can be found (Figure 12). As long as the single cell data contains less then 500 cells, the computation time stays stable. However the computational cost increases significantly if the size exceeds 500 cells. Dataset 3, 4, 19 and 20 that have a range from 650 to 1800 cells show a significantly larger computing time, especially dataset 4 that has a size of 1775 cells (Anova and Tukey HDS $p<0.05$) (Figure 12). We expect that the computation time will further increase as the size of the datasets augments in magnitude.

3.4.2 Parameter tuning

As mentioned above we want to perform a parameter optimization to fine-tune the parameters of each dimensionality reduction method, since the default parameters are frequently determined on a specific set of datasets leading to overfitting of the parameters towards these datasets. In Supplementary List S3 an overview of the different parameter settings for each method is shown. Parameter k representing the number of dimensions in the low-dimensional space, on which the datapoints will be embedded, is a common feature of all dimensionality reduction methods and is varied from two to five. Also distance measure is a regular parameter of the techniques as the bulk of the dimensionality reduction methods are based on preserving local and or global distances between datapoints between the high and low-dimensional space. We included two popular approaches Euclidean distance and Spearman distance.

The best parameter settings for every dimensionality reduction are listed in Supplementary List S4. When comparing the best parameter settings between the different techniques, we notice that all methods perform a dimensionality reduction optimally when embedded in a 5-dimensional space and that Spearman distance above Euclidean distance lead to a better dimensionality reduction.

In figure 13 the relative quality scores of the best parameter settings for all datasets are presented for each dimensionality reduction method. The performances of the

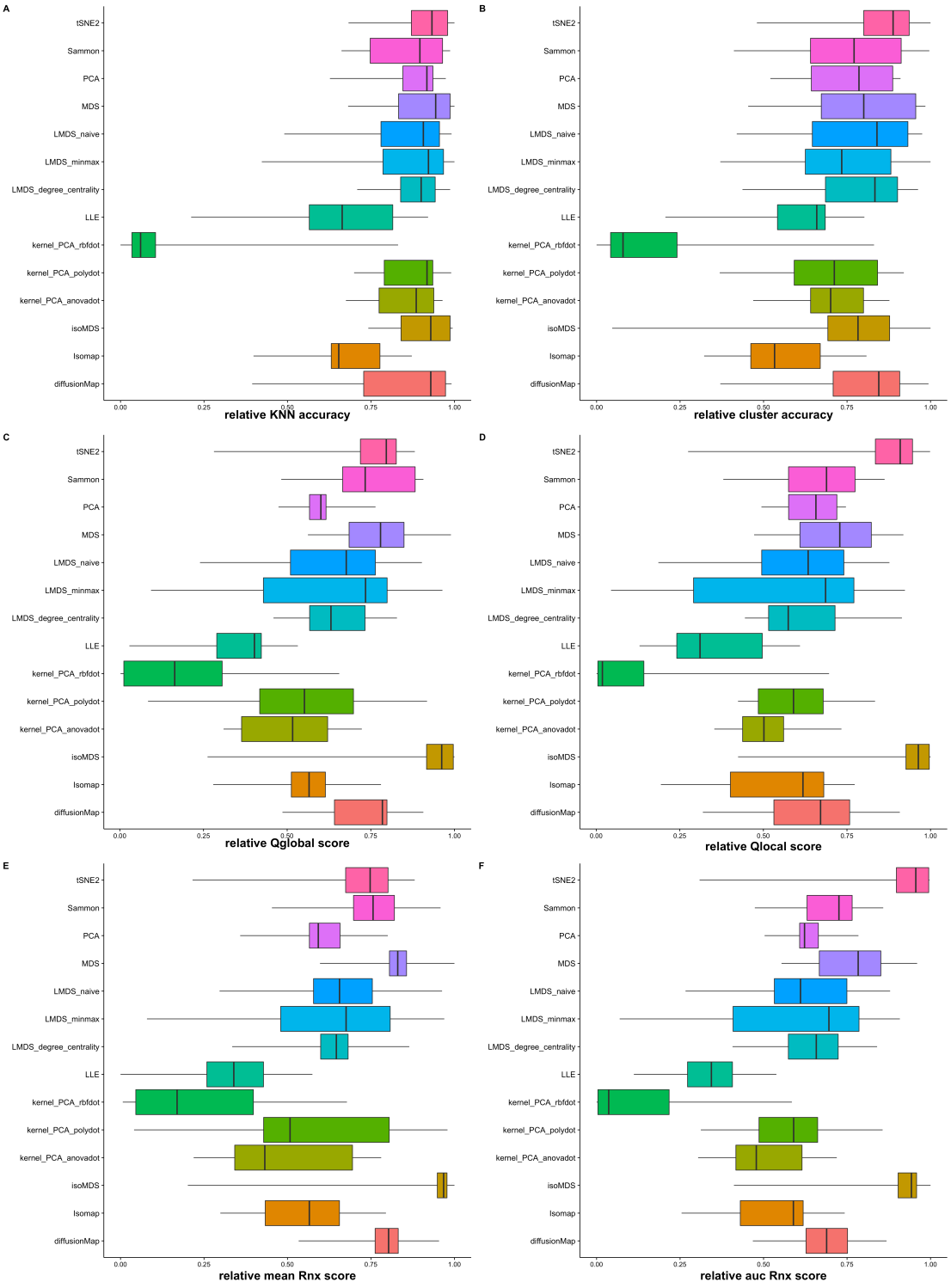

Figure 13: Relative performance of different dimensionality reduction methods after parameter optimization based on three quality metric scorings on small single cell datasets (<2000 cells). At front the scores are normalized at a scale of 0 to 1. For each method the scores of the best parameter setting of every dataset is represented. **A.** Relative KNN accuracy **B.** relative Cluster accuracy**. D.** Relative Qglobal **C.** Relative Qlocal **E.** Relative mean R nx **F.** Relative auc R nx

dimensionality reduction methods after parameter tuning are almost equivalent to the results before (Figure 10). LLE and especially PCA with kernel rbfdot achieve again the lowest scores for all quality metrics (Anova and Tukey HSD p<0.05). Also Isomap perform a dimension reduction significantly inferior compared to the others according to the KNN and cluster accuracy (Figure 13 A-B). Further we notice that isoMDS and t-SNE have a significantly (Anova and Tukey HSD p<0.05) higher coRanking score in contrast to the others (Figure 13 C-D). However this difference is not notable in the scorings of the KNN and cluster accuracy. Also diffusion map, MDS and Sammon perform a dimension reduction slightly better according to the coRanking score (Figure 13 C-D), but this is not seen in the KNN and cluster accuracy score. Even when the optimal parameters are chosen for each dimensionality reduction method, there is not one technique that explicit outperforms the rest.

Despite the minimum effect of the parameter optimization on the performance, we notice that parameter tuning had a positive influence on the computational cost for some methods, as hoped (Figure 14). The computational time of the methods diffusion Map, isoMDS and Sammon made a significantly drop (Anova and Tukey HSD p<0.05) for the datasets exceeding 500 cells. The computational time of t-SNE also improved significantly for every dataset after parameter optimization. PCA with kernel Anovadot is still the most unfavorable method concerning the computational time. The time required by PCA Anovadot to execute even extreme small datasets, as dataset 12 (66 cells) and 6 (169 cells) is significantly larger compared to the others

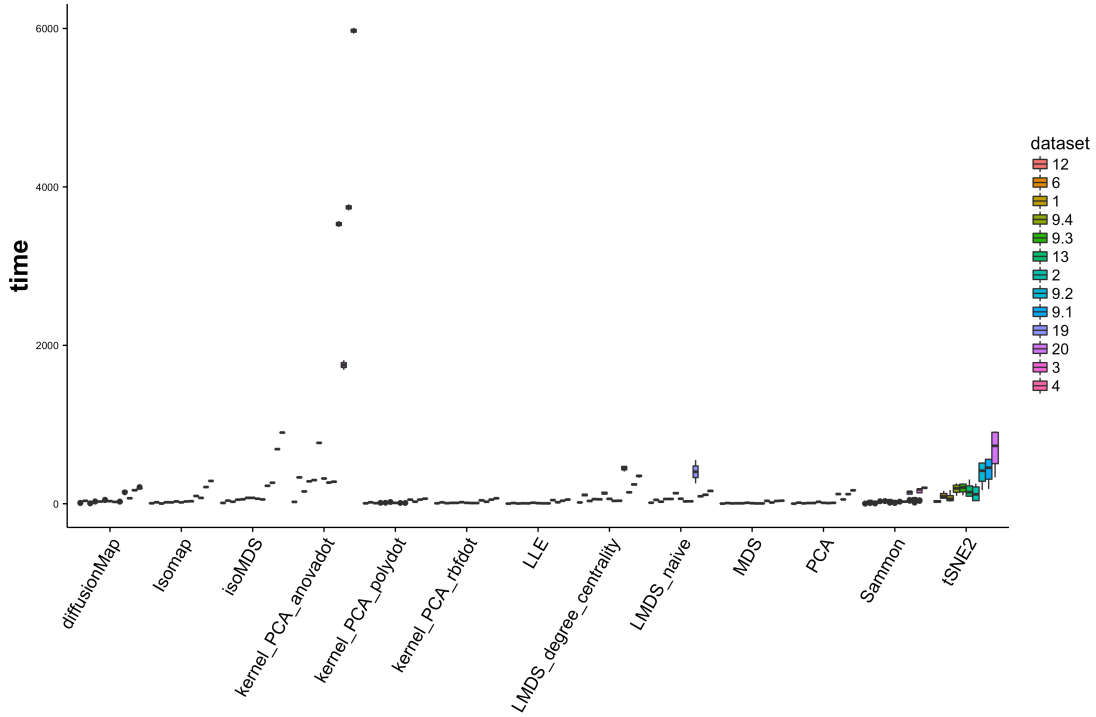

Figure 14: Absolute computing of different dimensionality reduction after parameter optimization on small single cell dataset (<2000 cells). The absolute computational time for the best parameter settings of each dimensionality reduction method is shown for each dataset separately. The datasets are ordered from small (dataset 12 = 66 cells) to large (dataset 4 = 1790 cells)

(Anova and Tukey HSD $p<0.05$)(Figure 14). We notice that also t-SNE despite the improvements requires a significantly larger computing time for every dataset compared to the rest (Anova and Tukey HSD $p<0.05$). Further we see that the computational demand slightly increase as the data size cross 500 cells for the methods diffusion Map, isoMDS, Isomap, Sammon and PCA. On the contrary, the computation time of MDS, LLE, PCA polydot and rbfdot is stable and independent of the data size as long as the datasets are below 2000 cells.

As mentioned above we added a variation of MDS to circumvent the computational burden of the current dimensionality reduction methods on large datasets by using a subset of the datapoints, termed as landmarks, to train the embedding on. When we compare the quality scores of the best parameter settings for landmark MDS with classical MDS, we notice that choosing a landmarking approach has no significant influence on the performance of the method itself (Figure 13). The three different landmark selection methods have a slightly lower coRanking score compared to MDS but this is not seen for the KNN and cluster accuracy score. As a consequence of the supplementary steps such as selecting landmarks compared to MDS, landmark MDS require a longer computation time to perform a dimension reduction on relative small datasets (Figure 14). From Supplementary Figure S3-4 we already concluded that LMDS with minmax as landmark selection implementation demand a significantly larger computing time compared to classical MDS, but also to all other methods. Landmarking has not an advantage as long as the magnitude of the datasets remains small.

3.4.3 Dimensionality reduction into a 2 or 3-dimensional space

After parameter tuning, we identified that almost all dimensionality reduction methods prefer the embedding of the datapoints in a low dimensional space of five ($k= 5$). We suspect that even better scorings could be obtained when the parameter k would be increased because less data will be lost during the embedding of the high-dimensional data in a low-dimensional space of five or more instead of only two. However computational algorithms used to deconvolve heterogeneous cell populations or to reconstruct lineage trajectory perform at front a dimensional reduction to a two or three-dimensional space due to the simple fact that it is better for visualization and afterwards for clustering or trajectory inference respectively. Therefore we set a restriction on the number of dimensions to two or three in the low dimensional space and performed subsequent analysis as above (3.4.1-2 overall performance before and after parameter tuning).

Figure 15 A-B represents the averaged relative KNN and cluster accuracy score for every dimensionality reduction method over the small single cell datasets with parameter k set on two before parameter tuning. We conclude that t-SNE significantly (Anova and HSD Tukey test $p<0.05$) outperforms the others. Also diffusion Map, isoMDS, MDS, LMDS with degree centrality and naive achieve a relative better scoring compared to the others (Figure 15 A-B). The almost equal performance of MDS and LMDS with degree centrality and naïve is logical since they use the same principle to reduce the dimensionality namely MDS. Despite that LMDS naïve and degree

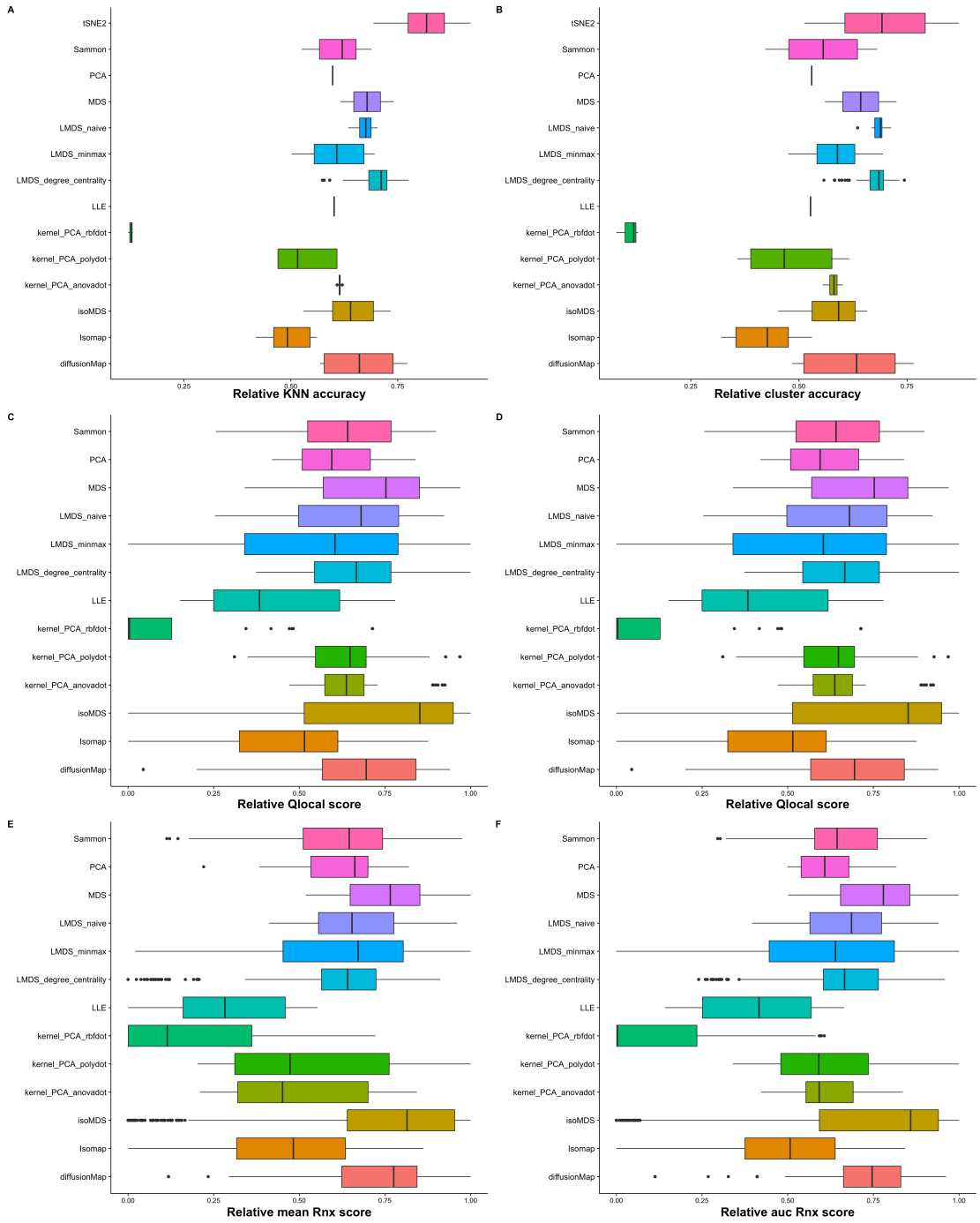

Figure 15: Relative performance of different dimensionality reduction methods into a two-dimensional space before parameter tuning based on three quality metrics for small single cell datasets (<2000 cells). At front the scores are normalized at a scale of 0 to 1.The dimension of the low dimensional space on which the datapoints are embedded is fixed on two (k = 2)**.** The parameters of each method are varied according to a grid search. For each method the average score over all datasets of each parameter setting is represented. **A.** Relative KNN accuracy **B**. relative cluster accuracy **C-D** relative coRanking metric **C.** Qglobal **D**. Qlocal **E**. mean Rnx **F.** auc Rnx

centrality use only a part of the data –landmarks- chosen either randomly or based on highest link centrality use only a part of the data –landmarks- chosen either randomly or based on highest link to calculate the embedding, this has no large influence on the KNN and cluster accuracy scoring compared to classical MDS. On the contrary LMDS with minmax as implementation has a poor performance compared to the other landmark selection implementations. Further PCA with kernel rbfdot is again the worst performing method and also Isomap and PCA with kernel polydot are inferior against to the others (Anova and Tukey HSD p>0.05). However, considering the results of the coRanking metric (Figure 15 C-D), t-SNE has not an explicit better performance as seen in the KNN and cluster accuracy. DiffusionMap, isoMDS and MDS together with t-SNE perform significantly better compared to the rest (Figure 15 C-D).

When the absolute computing time of the different methods except LMDS minmax (see Results section overall performance 3.4.1) for each dataset separately is juxtaposed (Figure 16), we examine almost the same results as in the overall performance before parameter optimization. PCA with kernel function Anovadot and t-SNE have again a significantly larger computing time for every dataset against the rest (Anova and HSD Tukey p<0.05) and the computation time for datasets ranging from 500 to 2000 cells increases significantly (Anova and Tukey HSD p<0.05) for the methods diffusion Map, isoMDS and Sammon.

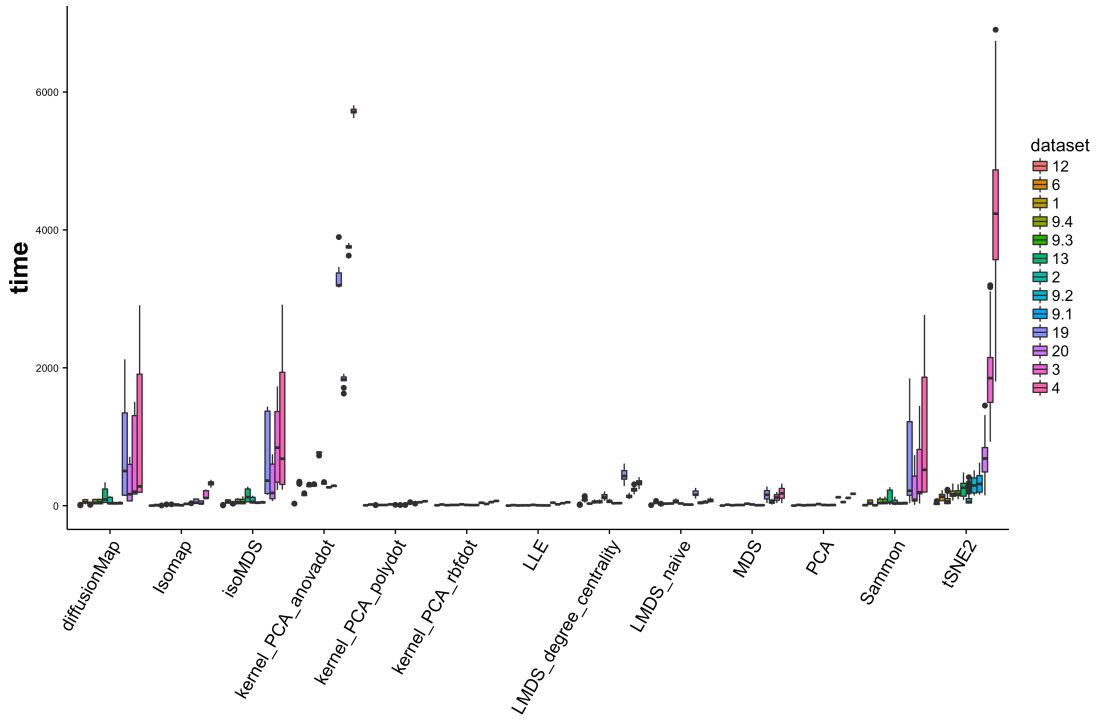

Figure 16: Absolute computing time of a dimensionality reduction into a two-dimensional space for a set of dimensionality reduction methods before parameter tuning on small single cell datasets (<2000 cells). The absolute computing for every parameter setting with parameter k fixed on two is shown for every dataset separately for a set of different dimensionality reduction methods. The datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4: 1799 cells).

After parameter tuning using LOOCV principle, we see that the scores of the different methods are more balanced against each other then before the parameter tuning (Figure 17). T-SNE works out to be the best performing method for a dimensionality reduction to a two-dimensional space, confirmed by every scoring metric except Qglobal and mean Rnx (Anova and HSD Tukey test $p<0.05$) (Figure 17). IsoMDS execute a mapping to a two-dimensional space significantly better according to coRanking metric, but this is not confirmed by KNN and cluster accuracy. PCA with kernel rbfdot remains the worst method according to all scoring metrics (Anova and Tukey HSD test $p<0.05$). Also the scores of Isomap, PCA kernel polydot and LLE are still inferior to the others according to all quality metrics. Further we see that PCA is less suitable for mapping the data into a two-dimensional space then in a five dimensional space. Also the difference in coRanking score between LMDS and MDS has been disappeared after parameter optimization, indicating that choosing landmarks has no large influence on the accuracy if the appropriate parameters are chosen. However here we performed landmark MDS and the other methods on relative small single cell datasets, performing LMDS on larger datasets may substitute in accuracy. Further we noticed that diffusion Map, MDS and LMDS with degree centrality and naïve execute the mapping to a two-dimensional space in a more efficient way compared to the rest.

As in the previous section, we performed again a parameter optimization but with parameter k fixed on two. The parameter optimization has again fine-tuned the computing time of diffusion Map, isoMDS, Sammon and t-SNE, proving that parameter optimization is profitable (Figure 18). PCA with kernel Anovadot still has a significantly longer computing time against the others and also t-SNE, isoMDS and LMDS with degree centrality perform the dimension reduction on datasets larger then 500 cells in an insufficient time opposed to the rest (Figure 18). We notice again that landmark MDS has larger time complexity fro datasets exceeding 500 cells compared to classical MDS. Therefore landmark MDS don't have any additive value compared to classical MDS on datasets smaller then thousand cells.

Considering the results of the different scoring metrics and the computational time before and after parameter tuning on the small datasets (Figure 15-16-17-18), we can conclude that t-SNE performs a dimension reduction into a two-dimensional space with the highest accuracy but however demands a larger computing time compared to the others even if the dimension reduction is performed on 'relative' small datasets. Therefore MDS could be a better choice as it performs a dimension reduction in an effect way within a reasonable time.

Previous analyses were repeated but with the low-dimensional embedding fixed on three, which can be applicable for example in the reconstruction of the spatial context of cells out single cell data. The results are shown in Supplementary Figure S5-8. In short, both the performances of the dimensionality reduction methods before and after parameter optimization indicate that Isomap, LLE and especially PCA with kernel rbfdot execute the dimension reduction significantly lower compared to the others (Anova and Tukey HSD $p<0.05$). Further we notice that t-SNE that outperforms the others and this difference augmented after parameter optimization.

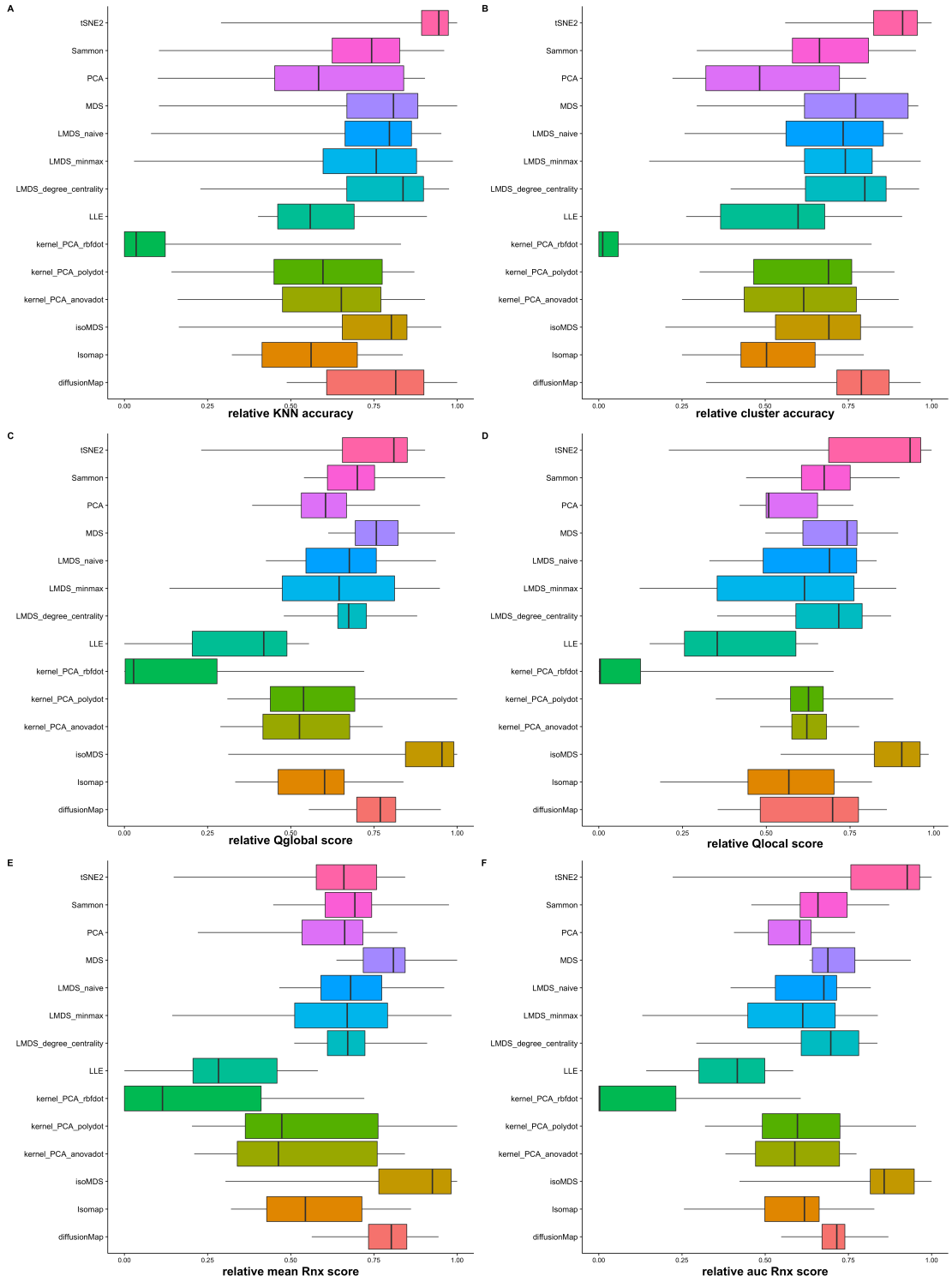

Figure 17: Relative performance of different dimensionality reduction methods into a two-dimensional space based on three quality scores after parameter tuning on small single cell datasets (<2000 cells). At front the scores are normalized at a scale of 0 to 1. Parameter k presenting the number of dimensions in the low-dimensional space is fixed on two. For each method the score of the best parameter setting of every dataset is represented. **A.** Relative KNN accuracy **B.** relative Cluster accuracy. **C.** Relative Qglobal **D.** Relative Qlocal **E.** Relative mean Rnx **F.** Relative auc Rnx

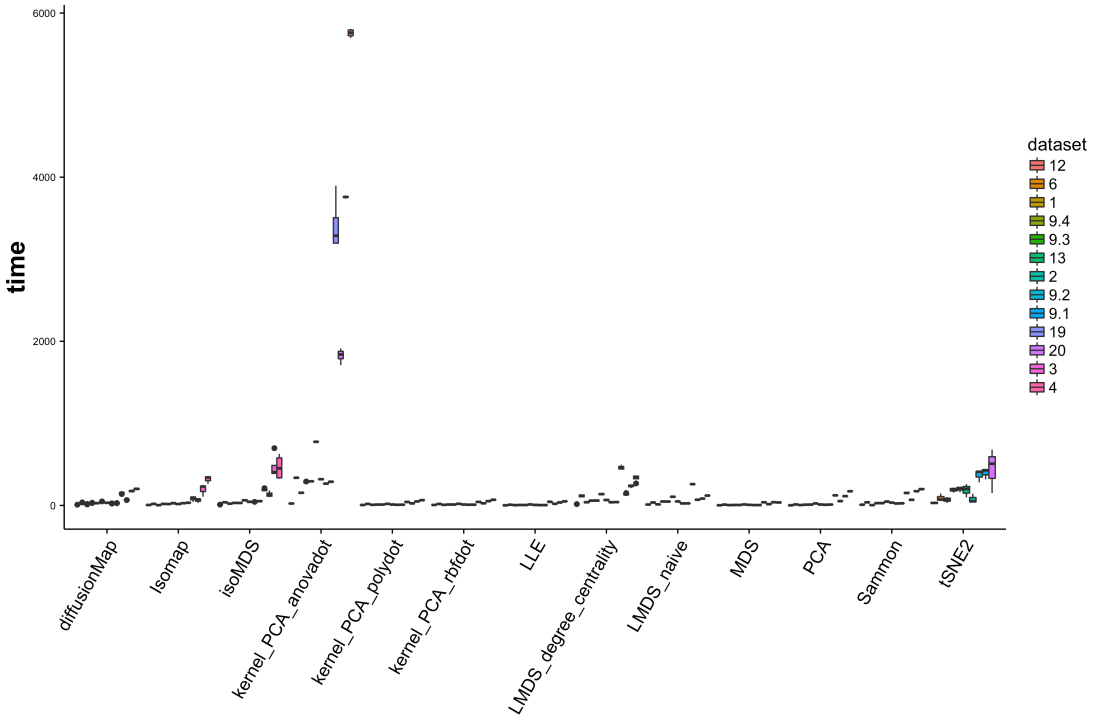

Figure 18: Absolute computing time of a dimensionality reduction into a two-dimensional space for a set of dimensionality reduction methods after parameter optimization on small single cell datasets (<2000 cells). At front the best parameter settings were selected with parameter grid search and LOOCV. The absolute computing for the best parameter setting with parameter k fixed on two is shown for every dataset separately for a set of different dimensionality reduction methods. The datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4: 1799 cells).

Also isoMDS has a better performance as seen in the coRanking score. As a remark we noticed that PCA has a significantly better performance when the dimensionality reduction is performed in a three-dimensional space then a two-dimensional space. Looking at the computation time, PCA with kernel Anovadot has moreover a larger computational cost compared to the others. Additionally the parameter tuning resulted in better timings for Diffusion Map, isoMDS, Sammon and t-SNE. However these methods are still demanding a larger computational time for the larger datasets compared to the others. Their better performance has the consequence of demanding a larger computing time.

3.4.4 Noise removal due to a dimension reduction

Besides the visualisation of high-dimensional data into a two or three-dimensional space, dimensionality reduction is also used to reduce irrelevant features and biological and/or technical noise in complex data, by mapping the data in a five to ten dimensional space. However, executing a dimension reduction on big data can have as trade-off the loss in valuable biological information. To examine which methods remove noise in the most efficient way, we compare the KNN and cluster accuracy score before and after dimensionality reduction with the best parameter settings out the parameter grid search.

Figure 19: Noise removal by different dimensionality reduction methods based on cluster accuracy. The difference in cluster accuracy score before and after a dimension reduction is shown for each dataset separately. The scores of the dimensionality reduction methods are the average scores of the best parameter settings after parameter tuning. The datasets are ordered from small to large.

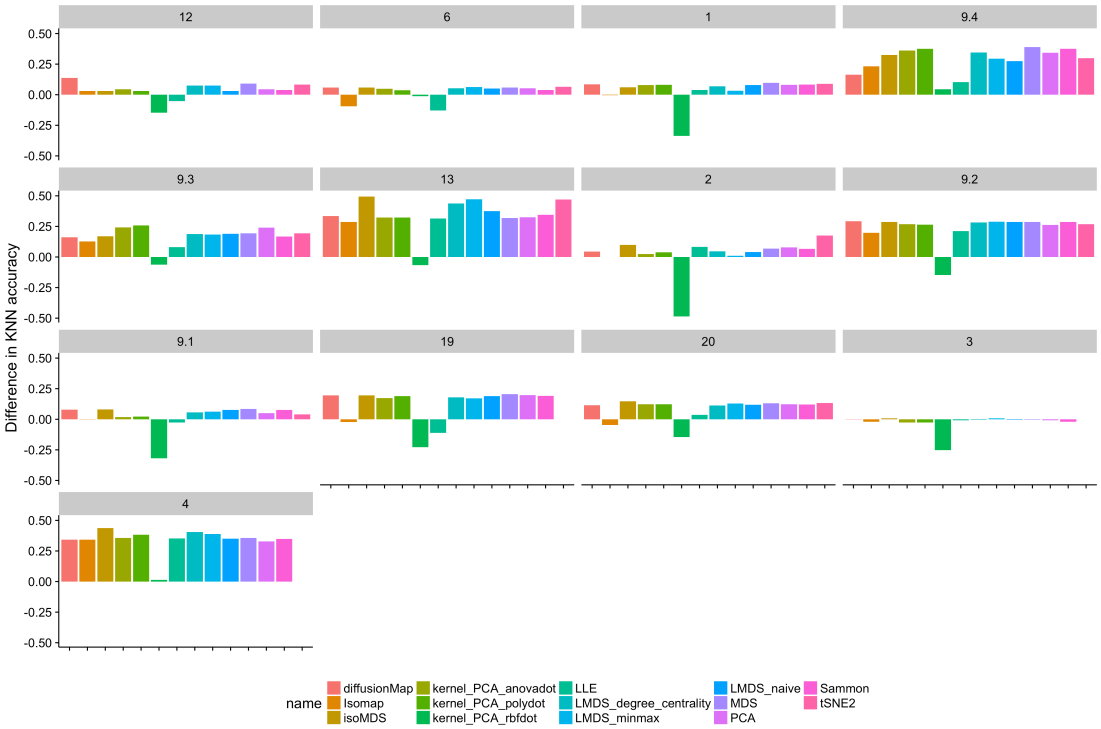

Figure 20: Noise removal by different dimensionality reduction methods based on KNN accuracy. The difference in KNN accuracy score before and after a dimension reduction is shown for each dataset separately. The scores of the dimensionality reduction methods are the average scores of the best parameter settings after parameter tuning. The datasets are ordered from small to large.

Figure 19 and 20 show the differences between the KNN and cluster accuracy scores before and after dimensionality reduction for each dataset separately. The datasets are ordered from small to large. The amount of noise present in data alters extensively between the scRNA-seq datasets and does not depend on the size of the data (Figure 19-20). Also the results from KNN and cluster accuracy are often not similar, in the disadvantage of cluster accuracy. Further we can see that PCA with kernel rbfdot, Isomap and LLE cause a loss in accuracy for almost every dataset (Figure 19-20). Previous results have already shown that kernel PCA rbfdot, Isomap and LLE perform inferior compared to the others. Considering the other methods, there is not one that explicit overthrow the other methods.

3.5 Performance on the entire set of single cell data

The amount of high-throughput data with a magnitude of tens to even hundred thousand have been increasing exponentially due to recent advancements in scRNA-seq protocols and new projects as HCA. Therefore it is critical when reviewing different methods to investigate which methods are capable to handle these large datasets. Following section has the same structure as the previous section (3.4) but the results of large high-throughput single cell datasets ranging from 2000 to 70000 cells are included. We will particularly focus on the computational time of the different dimensionality reduction methods since we expect that the performance of the methods will not depend on the size of the data but rather on the algorithm of the method itself. As a final remark we fixed the memory on 25GB when executing the different dimensionality reduction methods with their varying parameter setting on datasets exceeding 2000 cells. Any method that requires more memory then 25GB is an indication that the method doesn't perform the dimensionality reduction in an efficient way. After running isoMDS, LLE, MDS, PCA and Sammon we experienced that some methods demanded a running time of more then two days which is insufficient. Hence we set a limitation of five hours on the running time for the remaining methods.

3.5.1 Overall performance

At first none of the methods were capable to execute a dimension reduction on datasets 8 (44808 cells), 22 (66991 cells) and 25 (22003 cells) due to a lack of computational memory. Performing a dimensionality reduction on datasets that exceeds 20000 cells demands a higher amount of memory then 25GB and is therefore insufficient. This is a first indication that the current used dimensionality reduction methods lack the necessary complexity to leverage the information out of the new high-throughput single cell transcriptomics data within an efficient time.

The analyses of the computing time of all dimensionality reduction methods on small datasets with a size below 2000 cells (see section 3.4) showed us that generally the computational cost significantly increases once the data size crosses 500 cells (Anova and Tukey HSD $p<0.05$) (Figure 11-12). Figure 21 represents the computing time of all dimensionality reduction methods for each dataset separately in a logarithmic scale against the size of the datasets ranging from 66 to 67.000 cells. Once more we see that the computing time stays stable as long as the datasets are smaller then 500 cells, but significantly increases exceeding 500 cells (dataset 19: 621 cells) and keep on augmenting as the size expends (Anova and Tukey HSD $p<0.05$).

However, when examining the computational time for every dimensionality reduction technique before parameter optimization more in detail, we notice a lot of alterations between the different methods (Supplementary Figure S9) (Figure 22). As a remark, in order to compare the computing time on large datasets for the different methods in an efficient way, we used the size of 500 cells as cut-off and only represented the datasets that have a size larger then 500 in Supplementary Figure S9. However, due to the large differences in the computing time between datasets with a magnitude

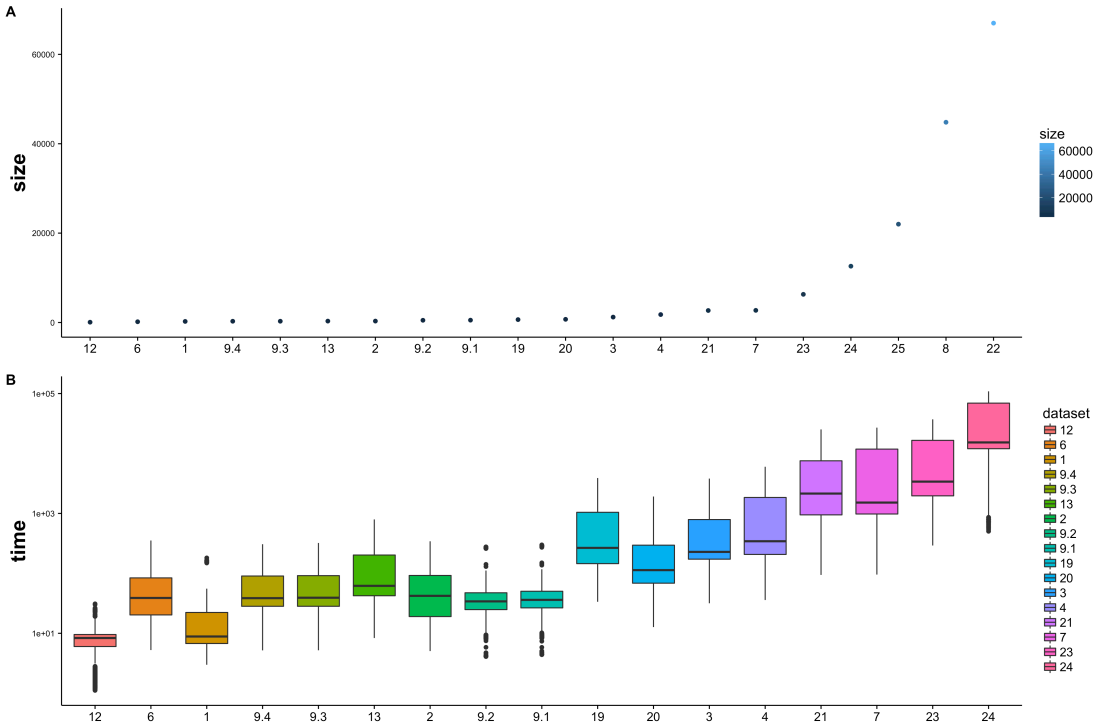

Figure 21: Relation between the size of the single cell datasets and the computational time to execute dimensionality reduction on these datasets. A. The number of cells in each single cell dataset, the datasets are ordered from small to large **B.** Absolute time of all dimension reduction is represented for every dataset separately. Y-axis was transformed to log10 scale.

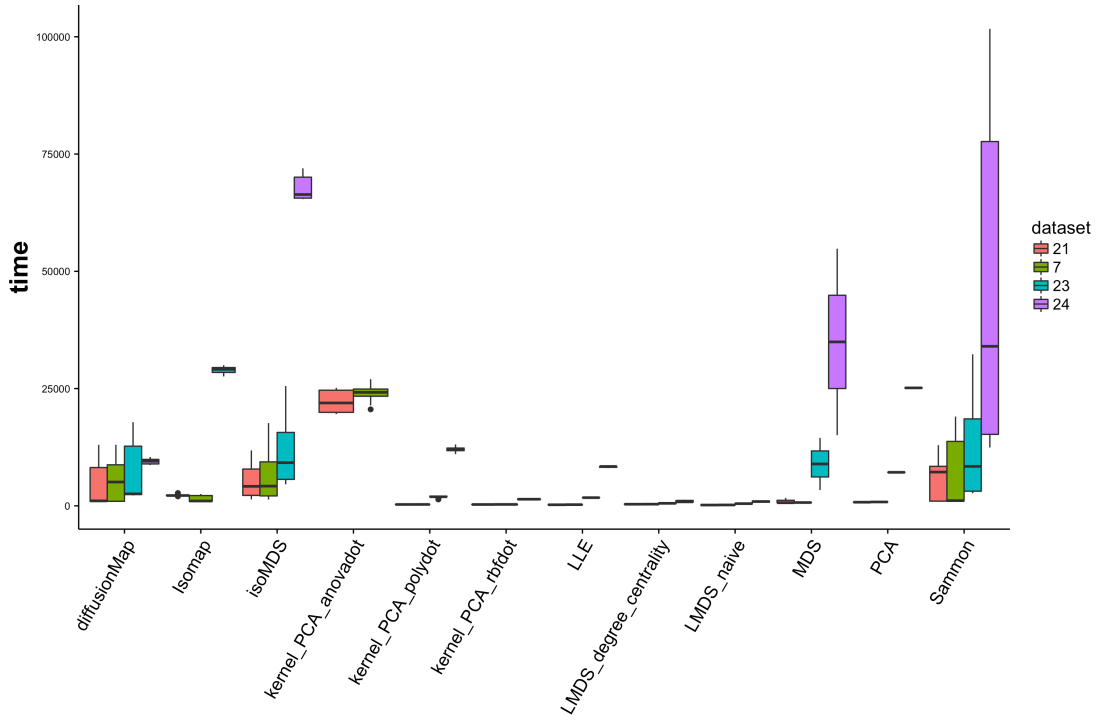

Figure 22: Absolute computational time of dimensionality reduction methods. The absolute computing time for every parameter setting of each dimensionality reduction method is shown for every dataset separately. The datasets are ordered from small (dataset 21: 2684 cells) to large (dataset 24: 66991 cells).

of thousand and tens of thousand cells, we only represent in Figure 22 the datasets larger then 2500 cells (dataset 7, 21, 23, 24). The overall trend is that the computational time for every dimensionality reduction method extremely expands when the scale of 5000 cells is crossed except for LMDS (Anova and Tukey HSD $p<0.05$).

Because we fixed the computing time on five hours for kernel PCA, Isomap, diffusion Map and t-SNE, dataset 23 and 24 did not always run for these methods. PCA with kernel Anovadot is again the worst performing method regarding the computational complexity. The computation time of PCA Anovadot is for every dataset significantly larger compared to the rest (Anova and Tukey HSD $p<0.05$) and due to the limitation in time, PCA Anovadot couldn't even perform a dimension reduction on datasets 23 and 24. As seen in the results for the small datasets, diffusion Map, isoMDS and Sammon require a significantly larger computing time when the magnitude exceed 500 cells against the rest (Anova and Tukey HSD $p<0.05$) (Figure 11). As expected, the computation time of these methods kept increasing as the data size augmented and take a significant longer time to perform the embedding of datasets with a size that exceeds 500 cells compared to the other methods (Figure 22). Further we see that the computational time of Isomap significantly increase for dataset 23, and again due to the time limitations Isomap failed to execute a dimension reduction on dataset 24. MDS and PCA show also a significant increase in the computing time for datasets larger then 10000 cells (Anova and Tukey HSD $p<0.05$), as do PCA polydot and LLE but less explicit. On the contrary LMDS, both degree centrality and naïve, don't display a significant increase in the computing time for the extreme large datasets, which was the initial purpose of landmark MDS. Choosing a subset of the datapoints as landmarks to train the embedding has a significantly positive influence on the computing time in contrast with classical MDS that use all datapoints to calculate the embedding. The computing time is independent of the choice of landmark selection method.

Figure 23 represents the averaged performance of every parameter setting over the datasets for each dimensionality reduction technique before parameter tuning. The results are similar to the results on small datasets (figure 10). PCA rbfdot has again the lowest accuracy and Isomap together with LLE are inferior compared to the others. Though diffusion map, isoMDS and MDS have a significantly higher coRanking score, this is not seen in the KNN and cluster accuracy score. So far none of the methods outperform the rest. Further when we focus on the performances of the dimensionality reduction methods on datasets that contain more then 500 cells since the computation time increase in general for all techniques once the data size exceeds 500 cells, we notice similar results (Supplementary Figure S11).

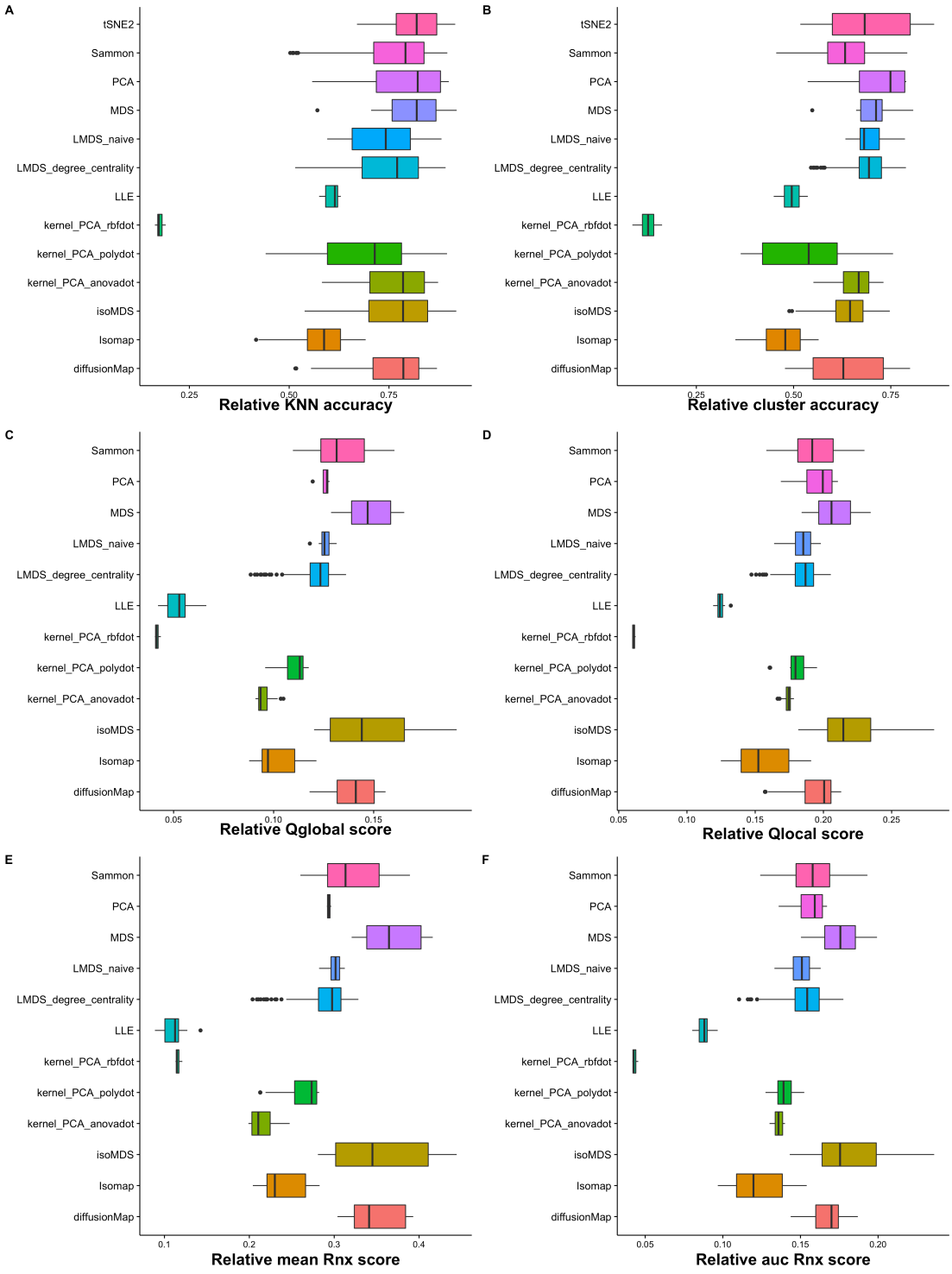

Figure 23: Overall relative performance of different dimensionality reduction methods with varying parameters before parameter tuning according to three quality metrics on small and large single cell datasets (66 – 70000 cells). At front the scores are normalized at a scale of 0 to 1. The parameters of each method are varied according to a grid search. For each method the average score over all datasets of each parameter setting is represented. **A.** Relative KNN accuracy **B**. relative cluster accuracy **C-D** relative coRanking metric **C.** Relative Qglobal **D**. Relative Qlocal **E**. Relative mean Rnx **F.** Relative auc Rnx

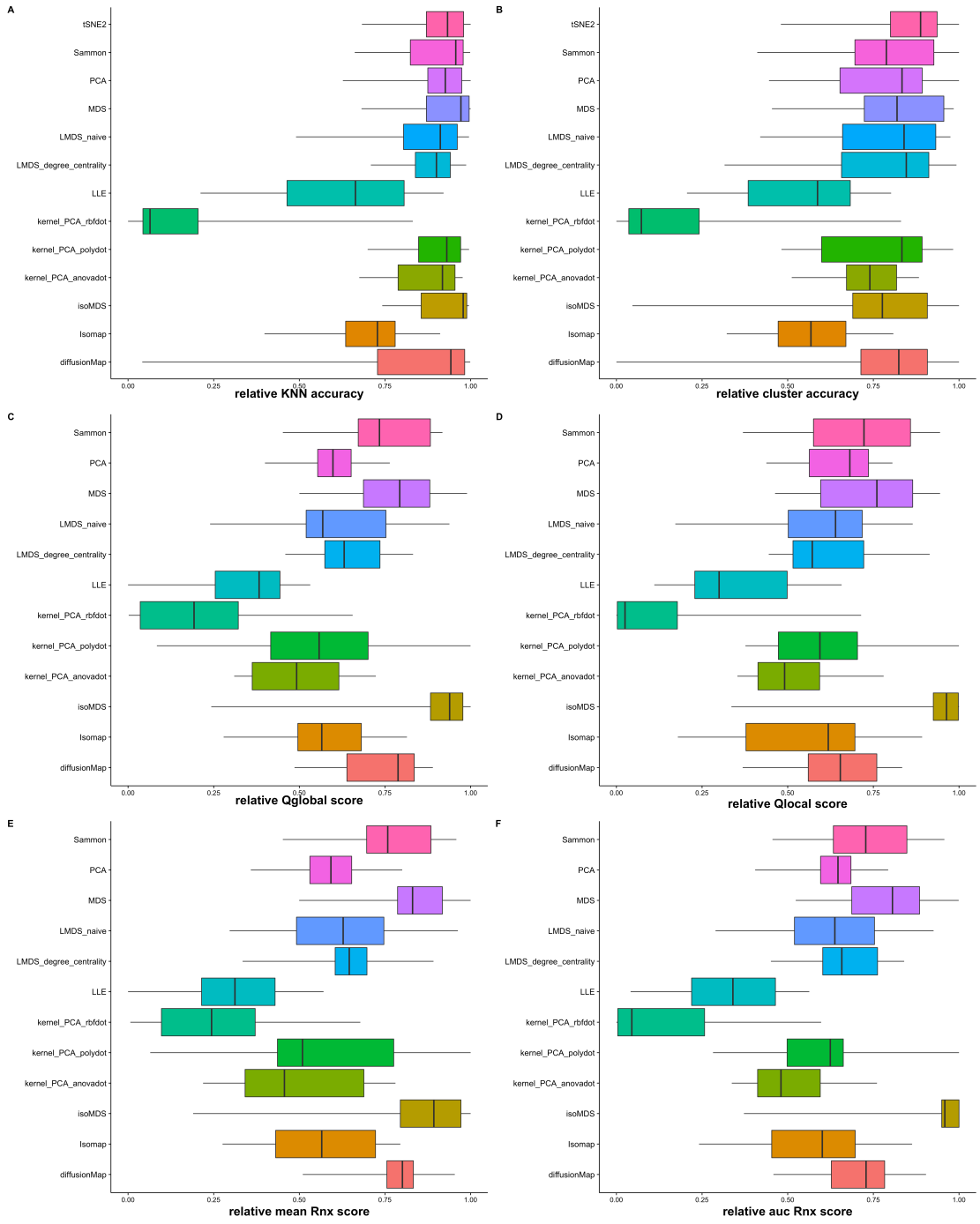

Figure 24: Relative performance of different dimensionality reduction methods after parameter optimization based on three quality metric scorings on small and large single cell datasets (66-7000 cells). For each method the scores of the best parameter setting of every dataset is represented. A. Relative KNN accuracy **B.** relative Cluster accuracy. **C.** Relative Qglobal **D.** Relative Qlocal **E.** Relative mean R nx **F.** Relative auc R nx

3.5.2 *Parameter tuning*

As in section 3.4.2, we executed a parameter grid search to optimize the parameters for every dimensionality reduction method. We used again LOOCV to train the parameters of the different techniques on the single cell datasets.

The best parameter settings according to the KNN accuracy are listed in Supplementary List S4. The parameter values trained on all datasets with a range of 66 cells to 70000 cells don't show significant alterations with the parameter values determined on small datasets below 2000 cells. All methods are most optimal when the data is mapped into a five-dimensional space and when Spearman is used as distance measure.

When examining the performances of the fine-tuned methods on small and large datasets (Figure 24), we once more see that PCA kernel rbfdot and LLE have a significantly lower scoring for the three evaluation metrics compared to the rest (Anova and Tukey HSD $p<0.05$). Isomap perform a dimension reduction with a lower KNN and cluster accuracy but this disadvantage is not seen in the coRanking score. Further we see that isoMDS has a significantly higher coRanking score compared to the rest (Anova and Tukey HSD $p<0.05$) as do diffusion map, MDS and Sammon (Figure 24C-D). However based on the KNN and cluster accuracy all methods have a similar performance except PCA rbfdot and LLE. Therefore we conclude that none of the methods outperform the rest.

Further parameter optimization fine-tuned once again the computational time of all dimensionality reduction method, though the general trends remain the same. The

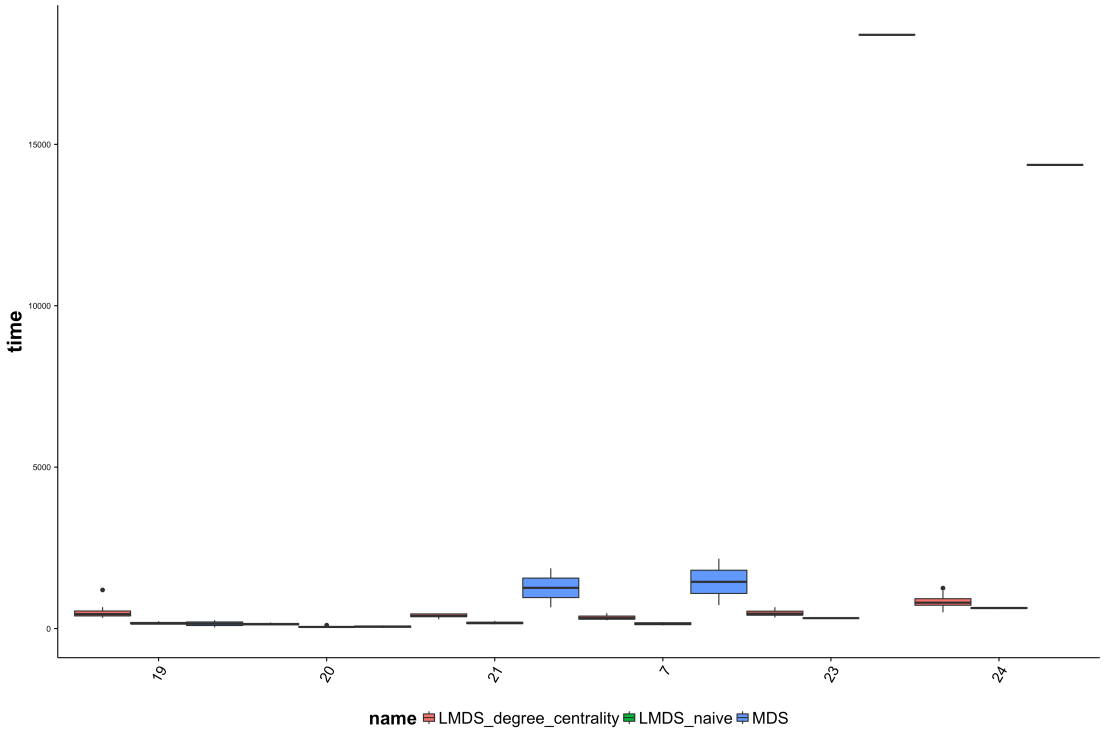

Figure 25: Absolute computational time of MDS and Landmark MDS with fine-tuned parameters on datasets with a size of 648 to 66991 cells.

computational time expand nearly exponential as the size exceeds 500 cells and further augments except for Landmark MDS. An overview of the computational time to execute a dimension reduction on datasets larger then 500 cells is represented for each optimized dimensionality reduction methods in Supplementary Figure S10.

In Figure 25 we focus on the computational demand of classical MDS versus Landmark MDS with naive and degree centrality as landmark selection implementations. The computation time of MDS is stable as long as the size of the data remain below 2000 cells (dataset 21 2648 cells) and increase significantly once the magnitude exceeds 5000 cells (dataset 23 6303 cells). On the contrary Landmark MDS perform a dimension reduction on large datasets without increasing in computational time. Lanmark MDS with degree centrality result in a slightly higher computation time compared to naïve since it requires supplementary steps in the selection of landmarks. When considering the performance of MDS and lanmark MDS, we notice a minus difference between them.

3.5.3 Dimensionality reduction into a two dimensional space

We restricted again the number of dimensions in the low-dimensional space on two to find methods that perform better or worse compared to a dimension reduction into five dimensions, since most of the time high-dimensional data is embedded into a two or three dimensional space for further analysis.

Supplementary Figures S12-15 show the performances and computational time of the different dimensionality reduction methods before and after parameter optimization on the large single cell datasets with a size of 648 to 70000 cells. There are no large differences between the performances of a dimension reduction into a two or five dimensional space on large datasets. PCA kernel rbfdot and LLE are once again the worst performing methods and none of the methods outperform the others. Further we see that PCA Anovadot demand a larger computing time compared to the rest, as do diffusion map, isoMDS and Sammon. The computational time expand significantly once the magnitude of the datasets exceeds 5000 cells except Landmark MDS.

3.5.4 Noise removal

As mentioned earlier, high-dimensional data often contain irrelevant features and biological and/or technical noise. The mapping of high-dimensional data into a low-dimensional space can remove this noise.

As seen in section 3.4.3, the amount of noise alters substantial between the different datasets independent on the size. In Figure 26 and 27 the difference in KNN and cluster accuracy score before and after parameter optimization is shown. Several methods couldn't perform a dimension reduction on datasets with a size larger then 5000 cells due to the restriction on time and memory leading to missing values in Figure 26 and 27. However we see that once more PCA with rbfdot, LLE and Isomap in general cause a loss in biological information.

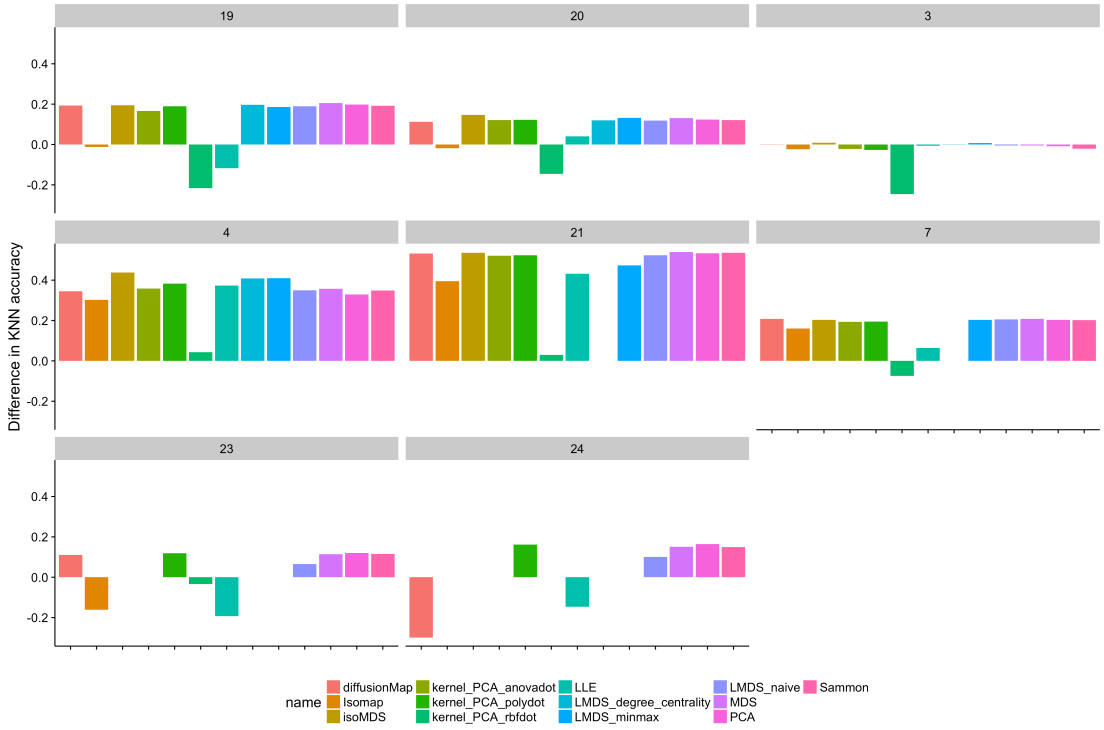

Figure 26: Noise removal by different dimensionality reduction methods based on KNN accuracy. The difference in KNN accuracy score before and after a dimension reduction is shown for each dataset separately. The scores of the dimensionality reduction methods are the average scores of the best parameter settings after parameter tuning. The datasets are ordered from small to large.

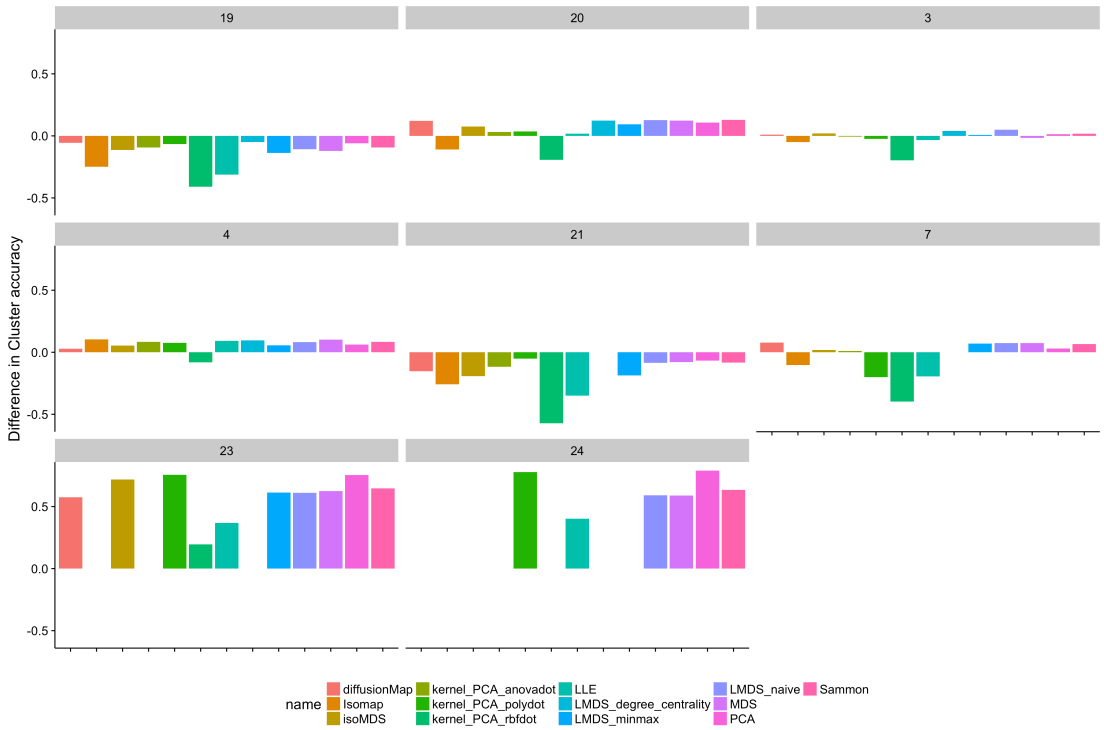

Figure 27: Noise removal by different dimensionality reduction methods based on cluster accuracy. The difference in cluster accuracy score before and after a dimension reduction is shown for each dataset separately. The scores of the dimensionality reduction methods are the average scores of the best parameter settings after parameter tuning. The datasets are ordered from small to large.

IV Discussion

Despite the improvements in single cell experimental protocols and the possible new insights of a cell's identity that single cell data can give us, our results prove that the currently used computational algorithms fail to leverage this information out high-throughput single cell data. The state-of-the-art dimensionality reduction methods lack the necessary computational complexity to perform a dimension reduction on single cell data with a magnitude of 10000 cells within an efficient time and memory. However our method, landmark MDS, succeed to perform a dimension reduction on 'big' single cell data with a magnitude of 10000 cells and may make it possible to scale the throughput of any dimension reduction to hundred thousand even millions of cells.

4.1 Practical guidelines

In general PCA with kernel Anovadot and t-SNE require a significant larger computing time for every dataset independent of the size compared to all other methods and are therefore not recommended. Regarding the quality performance, LLE, Isomap and especially PCA with kernel rbfdot perform a dimension reduction significant inferior to all other methods independent on the size and the number of dimensions in the low-dimensional space. PCA kernel rbfdot, Isomap and LLE are also not able to remove noise out high-dimensional instead they cause a loss in biological information when mapping the data into a low-dimensional space. Therefore are these methods never the best option to perform a dimension reduction. For datasets less then 500 cells, several methods are suitable regarding the desired accuracy, computational time and purpose of the dimension reduction such as MDS, isoMDS, diffusion map, Sammon and PCA. All though t-SNE performs a dimension reduction with a high performance compared to the others independent of the number of dimensions in the low dimensional space, the technqiue requires a high computational demand independent on the data magnitude. As a consequence, MDS is the preferred method If the data size lies between 500 and 5000 cells. Once the magnitude of a single-cell dataset exceeds 10000 cells, none of the methods has the capacity to perform a dimension reduction within an efficient time window except landmark MDS.

4.2 Landmarking approach

While all dimensionality reduction methods fail to perform a dimension reduction on datasets with a size that exceeds 10000 cells within a reasonable time, landmark MDS succeed to execute it without a substantial increase in time. The use of landmark points instead of all datapoints to train the embedding on circumvents the computational burden of MDS when performed on large datasets since the computational cost of these methods is quadratic in the number of data points. As long as the datasets are smaller then 5000 cells, the computation time of MDS remain stable and the use of landmark doesn't pay of since it requires more calculations steps in the algorithm, leading to a longer calculation time even longer then the original dimensionality reduction method. Moreover the trade-off of using landmarks is of

course the accuracy in the final embedding. Landmarking approach will always lead to a loss in accuracy, therefore is this approach only suitable on large datasets. As we have proven that landmarking works successful on MDS, we can implement this approach on other techniques such as t-SNE. T-SNE has a high performance when mapping in a two-dimensional space as in a higher space. However the method demands a significantly longer computation time independent of the data size. The combination of landmarks with t-SNE can improve the technique and may even outperform the others.

4.3 Critical points in single cell data size

There is an explicit relation between the magnitude of the single cell data and the computing time of a dimensionality reduction method. As long as the data contain less then 500 cells, the computation time of all dimensionality reduction methods remain steady and independent of the size of the data. However if the size of the data reaches 500 cells, the computational cost increase significantly as the magnitude of the single cell data augments for every dimensionality reduction method except MDS, LLE and PCA with kernel polydot. Furthermore the computational demand increases nearly exponentially if the magnitude of the datasets exceeds 5000 cells except for Landmark MDS.

4.4 Workflow and alternatives

4.4.1 Single cell datasets and dimensionality reduction methods

During our analysis we compared ten different dimensionality reduction methods on twenty publicly available single cell data. It is always possible to add other dimensionality reduction methods such as Destiny, LaplicianEigenmaps, but also to include recent high-throughput single data or even synthetic data to study in detail the features of each dimensionality reduction technique. Due to KNN and cluster accuracy, we were obliged to choose single cell data for which class labels were available. The coRanking metric doesn't depend on prior knowledge, therefore it is possible to select datasets without prior determined class labels, however other quality metrics besides coRanking are then required.

4.4.2 Quality metrics

To evaluate a dimension reduction we used three different quality metrics: KNN accuracy, cluster accuracy and coRanking. Further we used also the computation time to evaluate a dimensionality reduction technique. KNN and cluster accuracy infer the accuracy of correct predictions of a datapoints' neighbourhood. We used five as the number of k-nearest-neighbours and ten as the number of clusters to determine respectively the KNN and cluster accuracy. We fixed these default parameter values by comparing the quality scores of different parameter values on three random single cell datasets. In addition a parameter tuning can be performed to optimize these parameters and quality metric. Out the analyses we concluded that the KNN and cluster accuracy scores were correlated since the two quality methods are based on the same golden standard namely LOOCV. However, results from the coRanking

metric altered frequently with those from the KNN and cluster accuracy since the coRanking metric evaluate a method on how well the distances are preserved between high and low dimensional space. Depending on the approach of a quality metric, each metric will favour other dimensionality reduction techniques depending on its algorithm and features. As an example, coRanking metric prefer isoMDS, thought isoMDS doesn't have always a prominent performance according to KNN and cluster accuracy. Therefore it is important to use several quality metrics that evaluate different aspects of a method in order to compare and evaluate a set of methods in a correct way. However the appropriate quality metrics to evaluate a dimensionality reduction method are sparse and often based on how well a dimension reduction preserves the local distances between datapoints in the low-dimensional space. Nonetheless dimensionality reduction methods are not only used to purely compress data but also to visualize clusters, trajectories and network inferences out the data. As a consequence, the development of new metrics is required. As an alternative, instead of scoring the preservation of distances between the data points after a dimension reduction, we could score a dimension reduction on its capacity to visualize the information in single cell or other high-dimensional data such as trajectories, clusters or network inference. A set of dimensionality reduction methods can be evaluated on synthetic dataset from which the information a priori is known.

4.4.3 Parameter tuning

As mentioned before a good performance doesn't stand along with the algorithm itself but also with the correct choice of parameters since these parameters usually control (in some way) which relations should be preserved best. Hence, depending on the parameters of the method, even a single dimensionality reduction technique can lead to qualitatively very diverse results (Lueks *et al*, 2011). Moreover the default parameters of a method are determined on a specific set of datasets, which result in the overfitting of the parameters towards these specific datasets. As a consequence, by choosing the default parameters a method can have a high performance for a specific data that resemble the datasets used to determine the default parameters but score extremely poor for other datasets. Therefore we applied a parameter grid search with LOOCV to select the best parameter settings for each method. We checked that during the parameter optimization the training and the test score remained similar to avoid overfitting of the parameters. Our results confirmed that depending on the parameter setting the performances of a method alter considerable. The parameter tuning had also a positive influence on the computational time of several methods especially isoMDS, diffusionMap, Sammon and t-SNE proving that parameter optimization is a necessary but often overlooked task. As an alternative for parameter grid search, Iterated Racing procedure can be performed to automatically configure optimization algorithms by finding the most appropriate settings given a set of tuning instances of an optimization problem. The irace package implements the iterated racing procedure, which is an extension of Iterated F-race (I/F-Race) (Manuel, 2013).

4.4.4 Different features of dimension reduction

During the last years, a dimension reduction has not only been conducted solely to compress high-dimensional data into a lower dimensional space. The visualisation of big data into a two-dimensional space or the extraction of biological and technical noise out high-dimensional datasets is some other features of a dimensionality reduction technique. Hence we also focused and compared in our analysis the capability of a dimensionality reduction to visualize data into a two or three-dimensional space by restricting the number of dimensions in the low-dimensional space to two or three, but also the ability to eliminate noise out high dimensional data by comparing the accuracy scores of a datasets before and after a dimensionality reduction. Another characteristic that we didn't investigate in our analysis is the robustness of a dimensionality reduction method. The robustness of an algorithm is a criterion for the stability of the performance when small deviations are added in the training and test samples (Xu & Mannor; Musselman & Sanchez, 2007). Gradually adding noise to the data and analysing the deviations in the performances of a dimension reduction on the original data compared to noisy datasets (x% noise level), gives an indication of the robustness-to-noise of an algorithm (Zhu & Wu, 2004; Sáez *et al*, 2016). Down sampling the datasets is another way of levelling the robustness of an algorithm.

4.5 The future of single cell analysis

As the revolutionary technology, single cell RNA-sequencing has proven to be effective in characterizing the state of a cell though sophisticated computational and statistical analyzing methods to leverage the information out single cell data are just beginning to emerge. An important tool in the analysis of single cell data is the dimension reduction. Here we provided a comparative review of the state-of-the-art dimension reduction techniques and some subsequent guidelines to assist the researchers in the analysis of their data. However, the current used dimensionality reduction techniques lack the necessary computational complexity to perform a dimension reduction within an efficient time on datasets that have a magnitude of tens of thousand cells. Today, as a result of the increasing popularity of single cell research and the new advantages in single cell barcoding techniques, the number of single cell datasets and its sample size increase nearly exponentially. As the computational cost is quadratic to the sample size, the adaptation of the current dimension reduction methods or the development of new techniques to avoid the computational burden in an inevitable, yet challenging problem. This will become even more imperious as the interest increase in single cell projects as the Human Cell Atlas, requiring the analyses of million of cells.

Moreover, the dimensionality reduction is just the beginning of a long process of analyzing. In the last seven years, researchers have been investigating and developing new algorithms such as trajectory inference models, clustering methods and gene regulatory inference models to infer the biological information of single cell data. An important task for the future is to define a set of best practices through comparisons of these statistical methods, as we did in our research.

V Online methods

5.1 Code availability

Code to reproduce and further expand the evaluation of dimensionality reduction methods is available at github.com/sofieve/dimensionalityReduction_review.

5.2 Benchmark of single cell RNA seq datasets

A collection of scRNA-seq datasets is made representing several types of single cell data used for classification, subpopulation detection and pseudo temporal ordering of dynamic processes. An scRNA-seq dataset had to contain labels for each cell that were experimentally determined beforehand. These class labels are required to calculate the KNN and cluster accuracy. Most of the single cell data were selected from papers or by simply browsing through the GEO (Gene expression omnibus) database (Edgar *et al*, 2002) with the exception of datasets 22 to 25. We composed these datasets based on public available datasets on the 10X genomics website (The chromium single cell 3' Solution : datasets). These datasets originated from (Zheng *et al*, 2016) where they studied subpopulation in peripheral blood monocellulars (PBMC). The benchmark contains datasets that dates back to seven years when these datasets were modest in size and contained a lot of noisy data. Also high-throughput single cell data that were recently generated by new scRNA-sequencing protocols are included: dataset 7 was generated for the new protocol inDrop by (Klein *et al*, 2015) and his colleagues and also dataset 8 originate from the paper of (Macosko *et al*, 2015) wherein Drop-seq was represented. For each of the datasets, the expression data together with the label info and metadata is downloaded from the respective accession codes listed in Supplementary List 1. A detailed list of the different features of the selected datasets is represented in Supplementary List 1. Aside from log-transforming the expression values, no further preprocessing of the expression data was performed.

5.3 Dimensionality reduction methods

Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality (Van Der Maaten *et al*, 2009). The fundamental assumption that justifies the dimensionality reduction is that the data is lying on or near a manifold of a smaller dimension than the original data space. The goal of dimensionality reduction is to find a representation of that manifold and project the data vectors on it without modifying its topological properties (Carreira-Perpinan, 1996; Van Der Maaten *et al*, 2009; Ã & Verleysen, 2009). Suppose we have a dataset X $[n\ x\ D]$ consisting of n datavectors x_i $(i \in \{1,2,\dots,n)$ embedded in a data space of dimensionality D, dimensionality reduction techniques transform dataset X into a new dataset Y with dimensionality d $(d \ll D)$, while retaining the intrinsic dimensionality of dataset X.

$$\mathrm{X} = [\mathrm{x_i}]_{1 \le \mathrm{i} \le \mathrm{n}} \in \mathrm{R^D} \rightarrow \mathrm{Y} = [\mathrm{y_i}]_{1 \le \mathrm{i} \le \mathrm{n}} \in \mathrm{R^d}$$

In practice, however, neither the intrinsic nor the topological properties are known. Therefore, the goal of dimensionality reduction methods is most often to preserve the structure of the data set, which is indicated for instance by some sort of neighborhood relationships, such as proximities or similarities (Ã & Verleysen, 2009). These features can be obtained by measuring pairwise distances/similarities e.g. Euclidian distance and covariance matrix in PCA and MDS, geodesic distance matrix in Isomap and probabilistic similarities matrix in t-SNE. Traditionally, dimensionality reduction was performed using linear techniques such as Principal Component analysis (PCA) and classical metric multidimensional scaling (MDS). In the last decade nonlinear dimensionality reduction methods such as t-SNE, LLE, Isomap emerged due to their ability to deal with complex nonlinear data in contrast to the traditional linear methods. In Supplementary List 2 an overview can be found of several dimensionality reduction methods along with their specific features that are taken up in the comparative review. Also the packages that were used for each dimensionality reduction method is presented in the Supplementary List 2.

5.4 Landmarks in dimension reduction

The recent interest in manifold learning and dimensionality reduction is due, in part, to the multiplication of high-dimensional data from numerous disciplines of science, from signal processing to single cell sequencing (Belabbas & Wolfe, 2009). The complexity of most existing dimensionality reduction algorithms is, in general, dependent not only on the dimensionality itself but also on the number of observations. Therefore the computational cost of these methods is quadratic in the number of data points (Silva & Isr, 1950). Due to the ever-increasing magnitude of the data, researchers have been investigating approaches to reduce the computational cost by choosing a small subset of the data referred to as “landmarks” to train the embedding on, e.g. L-Isomap (Shi *et al*, 2015) and Landmark LLE (Chi & Melba, 2012). Landmarks are selected either randomly or intelligently, to develop the manifold and then applying their mapping to the other non-landmark points in such as to minimize the error between the normal embedding and landmark embedding (Doster *et al*, 2011; Shi *et al*, 2015). The tradeoff is of course the accuracy in the final embedding. By using a set of n landmarks in Isomap, the computational demands decrease from $\mathrm{O}(N^2\mathrm{logN})$ associated with constructing the shortest path matrix and solving the eigenvalue problem, to $\mathrm{O}(nNlogN)$ (Chi & Melba, 2012). Similar for LLE, the computational complexity for eigenvalue decomposition is $\mathrm{O}(dN^2)$, but it could be decreased to $\mathrm{O}(dn^2 + d(N-n))$ when a landmark set is exploited (Chi & Melba, 2012).

5.4.1 Landmark MDS

Also Classical multidimensional scaling (MDS) suffers from the same computational problem. The complexity is approximately $\mathrm{O}(kN^2)$, where N is the number of data points and k is the dimension of the embedding. When the number of datapoints is very large, this algorithm may be too expensive in practice. The bottleneck in classical MDS is the calculation of the top k eigenvalues and eigenvectors of an N x N matrix derived from the input distance matrix D (Silva & Tenenbaum, 2004). By using a

subset of n distinguished points referred to as "landmarks", the dimensionality reduction method can be decomposed into two-stages. First classical MDS is applied to the landmark points. The second step is a distance-based triangulation procedure, which uses distances to the already-embedded landmark points to determine where the remaining non-landmark points should go (Silva & Tenenbaum, 2004). The landmark MDS algorithm was introduced by the authors in (Silva & Tenenbaum, 2003) and adapted by (Silva & Tenenbaum, 2004) to make it more accessible and appealing to a broad audience. Pseudocode for the algorithm can be found in Table 1.

Table 2: Pseudocode of landmark MDS based on (Silva & Tenenbaum, 2004)

Landmark MDS	
Step 1	Select a set of n landmark points (naïve, minmax, degree centrality)
Step 2	Apply classical MDS to find a $k\ x\ n$ matrix L representing an embedding of the n landmark points. As input, use the $n\ x\ n$ matrix D_n of the distances between pairs of landmark points
Step 3	Apply distance-based triangulation to find a $k\ x\ N$ matrix X representing an embedding of the N data points. As input, use the $n\ x\ N$ matrix $D_{n,N}$ of distances between landmark points and data points.

5.4.2 Landmark selection methods

In order to minimize the loss of accuracy in the final embedding due to using landmarks, datapoints that summarize the most relevant information in the data has to be chosen as landmarks. Hence landmark selection is a crucial step and determines the success of the algorithm. There are many methods to select sets of landmark points but we have chosen to implement three methods: two methods taken from (Silva & Tenenbaum, 2004) and a third method based on degree centrality. The first method, which we call naive, takes a random selection of data points as a subset, irrespective of their topological features. Random selection of landmarks is easy to implement, and generally performs well for smoothly varying manifolds, but does no always represent manifolds with complex geometry properly (Doster *et al*, 2011; Rafailidis *et al*, 2017; Shi *et al*, 2015). The second, and computational more demanding method is minmax (Table 2) that seeks to create an optimal set of landmark points where the addition of each landmark point maximizes the minimum distance from the set of landmarks to all other non-landmark points. Despite the additional complexity of $O((l-s)*n)$ over the random method, the greedy optimization has the advantage to require a much smaller set of landmarks compared to random selection to obtain approximately the same results as random selection and it creates a repeatable set of points each time (Doster *et al*, 2011). However for some datasets, minmax has the tendency to select outliers that could lead to embeddings that do not preserve the important characteristics of the data.

Table 3: Pseudocode of landmark selection method Minmax based on (Doster *et al*, 2011)

Minmax (landmark selection)	
Step 1	Choose $1 \leq s \leq l$ seed points at random, adding them to the set of landmark points S and removing them from the dataset X.
Step 2	For $X_i \in X$ and $S_j \in S$, let $d_i = min_{j=1:\|\|S\|\|}\{X_i, S_j\}$.
Step 3	Let $d_k$be the maximum of $\{d_i\}$. Add X_k to the set of landmark points S and remove it from the set of data points X. If $\|\|S\|\| = l$ then done, otherwise got to step 2.

Table 4: Pseudocode of landmark selection method Degree centrality

Degree centrality algorithm	
Step 1	If $N < 2000$: Calculate distance matrix D_N of dataset X Else if $N \geq 2000$ If 'simple': random subsample L of 1000 points x_i calculate distance matrix $D_L$of subsample L If 'advanced': For point $x_i \in X$ Select 1000 other datapoints randomly Calculate distances between them Combine to one distance matrix
Step 2	Calculate knn graph of distance matrix
Step 3	Calculate degree centrality
Step 4	Select s point with highest degree centrality and add them to set of landmarks S

Degree centrality is the third and final method. Degree centrality is defined as the number of links incident upon a node. Nodes with the highest degree centrality are selected as landmarks (Rafailidis *et al*, 2017). The degree centrality is calculated from knn-graph representing the k-nearest neighbors for each cell. A throwback is the additional computational cost for the calculation of the distance matrix of the data, preceding the knn- graph. As a solution to this problem, we propose a distance matrix adjusted on a part of the actually data when the dataset contains more then 2000 cells. We implemented two different ways to do the subsampling for the calculations of the distance matrix. A first method, called the 'simple' method, determines the

distance matrix from a random subsample of 1000 datapoints. Advanced is the second approach, in which we calculate for every point the distances to 1000 others points randomly chosen. This is repeated for every data point. In the end we combine all the distances of every point into a large distance matrix. The second method is computational more demanding due to the for-loop compared to the naïve subsampling method. The Pseudocode of degree centrality can be viewed in Table 3.

5.5 Parameter tuning

All dimensionality reduction techniques have various parameters to control the embedding in the low-dimensional space. These parameters usually control (in some way) which relations should be preserved best. Hence, depending on the parameters of the method, even a single dimensionality reduction technique can lead to qualitatively very diverse results (Lueks *et al*, 2011). Parameter tuning is therefore a necessary but often overlooked challenge. In publications of new dimensionality reduction methods or in evaluation studies researchers use often the default parameters, which were optimized for some specific datasets by the authors (Saelens). Therefore, to make sure an evaluation is unbiased as possible, some parameter optimization is required. However, one should be careful of overfitting parameters on specific characteristics of one dataset, as such parameters will lead to suboptimal results when generalizing the parameter settings to other datasets. (Saelens) This could again introduce a bias in the analysis, where methods with a lot of parameters would better adapt on particular datasets, but would not generalize well on other datasets. In this study we tried to address both problems as follows. We first used a grid search approach to explore the parameter space of every dimensionality reduction method (Saelens). We refer to Supplementary List 3 for an overview of the parameters that were optimized for every method. In a process akin to leave-one-out cross validation (LOOCV) we determined the most optimal parameters setting for every dimensionality reduction method. For start, we left one dataset out and averaged the performance score for every parameter setting over all single cell datasets except for left out dataset. The parameter setting with the maximum score for each dimensionality reduction method is called the training score and is selected as the optimal parameter setting. Next, we used the optimal parameter setting to calculate the performance on the left out dataset, which resulted in a test score. The test and training score should be similar, if not some overfitting took place. This is repeated n times as there are single cell datasets. At the end the average is taken of the training score and test score of every dataset.

5.6 Evaluation metrics

We used three different metrics to measure the quality of the dimensionality reduction on the benchmark of single cell datasets. These measures should evaluate, in an automated and objective way, in how far the structure of the original data corresponds to the structure observed in the low-dimensional representation (Lueks *et al*, 2011). Most quality measures that have been proposed recently, are based on

neighborhood relations of the data. Here we used k-nearest neighbors (KNN) accuracy, clustering accuracy and co-Ranking metric respectively.

5.6.1 KNN accuracy

A first approach is the cross validation accuracy (CVA) using k-nearest neighbor KNN leave-one out cross validation (LOOCV) principle. Cross validation is a common model to perform evaluations. Instead of using the entire dataset as training for the learner model that can lead to over fitting, a part of the data is removed before the learning phase. The withheld data can be used afterwards to test the performance of the learned model. Leave-one-out cross validation is the K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the dataset (Schneider, 1997). That means that N separate times, the function approximator is trained on all the data except for one point, for which a prediction is made based the approximator (Schneider, 1997). Here the label of each data point in the low dimensional subspace is left out and subsequently predicted based on the labels of its k-nearest neighboring data points (Cannoodt *et al*, 2016). The predicted labels are then compared to the original label. Based on the accuracies of the predictions (percentage of correct predictions) the performance of the DR techniques is evaluated. We choose the default value of five for the parameter k, representing the number of nearest neighbors.

Define E_I as the experimentally observed class label of a cell $I \in \mathbb{C}$,

$$KNN\ CVA := mean_{x \in \mathbb{C}}(E_x \in modes_{Y \in 5NN(x)} E_Y),$$

with modes the modes of the 5-nearest-neighbor. For example, if $5NN(X) = \{1,2,2,3,3\}$, then the modes would be $\{2,3\}$ (Cannoodt *et al*, 2016).

5.6.2 Cluster accuracy

Based on the same golden standard as KNN CVA, we developed the metric cluster accuracy. However, instead of looking to the class labels of the k-nearest neighbors, cluster accuracy predicts the class labels of a specific point based on the class labels of the points in a cluster, to which that specific point belongs. The data in the reduced subspace is first clustered into k clusters by k-mean algorithm. Next the label of each data point is left out, similar to KNN accuracy, but predicted based on the labels of the points belonging to the same cluster as the left out point. At the end the predicted class labels are set against the original labels in the high dimension to calculate the accuracies. If the accuracies of the predictions are high, a good performance is allocated to that dimensionality reduction method. The number of clusters in which the reduced data should be divided is set default on ten. While testing the metric with different parameter settings for the number of clusters (varying from one to fifteen) on three independent single cell datasets differing in size, we concluded that the number of clusters is independent of size and the metric performance didn't increase further once the number of clusters passed ten.

Define E_I as the experimentally observed class label of a cell $I \in \mathbb{C}$ and $cluster(x)$ as the cluster, to which datapoint x belongs after k-mean clustering,

$$cluster\ CVA := mean_{x \in \mathbb{C}}\big(E_x \in modes_{Y \in cluster(x)} E_Y\big),$$

with modes the modes of the cluster. For example, if $cluster(X) = \{1,2,2,2,2,2,3,3\}$, then the modes would be $\{2\}$.

5.6.3 Co-Ranking criteria

The major disadvantage of previous explained metrics is that they depend on priori knowledge of class labels, which is most of the time not available. The class labels are often allocated after a dimensionality reduction and clustering had been performed on the data. Using class labels that were determined afterwards can lead to biases in the analysis. Therefore we included a Rank-based criteria based on co-Ranking matrix and K-ary neighborhoods (Ã & Verleysen, 2009; Lee *et al*, 2013; Lueks *et al*, 2011). Ranks are allocated to the distances between every pair of point in the high and low dimensional space. Based on these ranks a coRanking matrix is reconstructed. The quality criteria measure in how far the ranks are preserved after the reduction to a low-dimensional space (Lueks *et al*, 2011). The co-ranking matrix Q (Ã & Verleysen, 2009) is defined by

$$Q_{kl} = \ |\{i,j\}|\ \rho_{ij} = k\ and\ r_{ij} = l\}|,$$

where ρ_{ij} is the rank of ξ_j with respect ξ_i to in the high-dimensional space and analogously, r_{ij} is the rank of x_i to x_j in the low dimensional space. Errors made by dimensionality reduction methods correspond to off-diagonal entries in the co-ranking matrix, hence, rank errors for large ranks are not as critical as rank errors of close points (Lueks *et al*, 2011). Therefore a cut-off K is introduced that partition the co-ranking matrix into four sub matrices divided by the K-th row and K-th column. A quality measure can than be defined by the number of points that remain inside the K-neighborhood:

$$Q_{NX}(K) = \ \frac{1}{NK}\sum_{K}^{k=1}\sum_{K}^{l=1} Q_{kl}\, Q_{kl}$$

The normalization ensures that criterion varies between zero and one, perfect mapping. A variant of Q_{NX} that was used by (Lee *et al*, 2013) is R_{NX}, which shows the neighborhood preservation of K-th nearest neighbors but additionally corrects for random point distribution.

$$R_{NX}(K) = \ \frac{(N-1)Q_{NX}(K)-K}{N-1-K}$$

Another included co-Rank criteria quality measure is local continuity meta-criterion (LCMC) (Lee *et al*, 2013; Lueks *et al*, 2011), which is very similar to Q_{NX} except it coincides up to a linear term that accounts for the quality of random mapping.

$$LCMC(K) = Q_{NX}(K) - \frac{K}{N-1}$$

These measures have the disadvantage that they depend on K and generate a graph of the quality values over all possible K, and thus do not give a single decisive number that determines the quality of the mapping. Therefore we also added $mean\ R_{NX}$ and $AUC\ R_{NX}$ that represents respectively the overall mean and the total area under the R_{NX} graph (Ã & Verleysen, 2009). For LCMC we make a local vs. global evaluation of the quality graph. To estimate which values of K should be considered local, the following splitting point was used (Lueks *et al*, 2011):

$$K_{max} = \arg max_k LCMC(K)$$

The local and global quality measures are obtained by averaging the respective parts of the quality graphs (Lueks *et al*, 2011).

$$Q_{local} = \frac{1}{K_{max}} \Sigma_{K_{max}}^{K=A} Q_{NX}(K),$$

$$Q_{global} = \frac{1}{N-K_{max}} \sum_{K_{max}}^{N-1} Q_{NX}(K).$$

5.7 Statistical analysis

We used the one-way analysis of variance, also called ANOVA test to determine whether there are any statistically significant differences between the means of different (unrelated) dimensionality reduction methods. To determine which specific groups differed from each other, a post-hoc test is executed. As post-hoc test we optioned for Tukey honest significant difference test (Tukey HSD) based on studentized range distribution. Tukey's test compares the mean of every group to the means of every other group and identifies any difference between two means that is greater than the expected standard error (Haynes, 2013).

5.8 Performance comparison

At front an important step in the analysis is the normalization of the scores over the benchmark of datasets in order to compare the performances of the different methods between the datasets in a statistical correct way. As seen in Supplementary Figure S1-2 the performance of the dimensionality reduction depends not only on the method itself but also on the dataset since this can contain a lot of (technical and biological) noise, which makes dimensionality reduction challenging. The scores are normalized to each dataset according to,

$$z_i = \frac{x_i - \min(x)}{\max(x) - man(x)}$$

resulting in relative scores with a scale of 0 to 1. The performance of a dimensionality reduction method is evaluated with three different quality metrics: KNN accuracy, Cluster accuracy and coRanking metric. KNN and cluster accuracy infer the accuracy on the correct predictions of the neighborhood of a data point. On the contrary coRanking metric evaluate the performance of a dimensionality reduction method on the preservation of distances between the datapoints in high and low dimensional space. Further we use the computational time to compare the performances between dimensionality reduction methods.

VI References

Ã JAL & Verleysen M (2009) Neurocomputing Quality assessment of dimensionality reduction : Rank-based criteria. **72:** 1431–1443

Achim K, Pettit J-B, Saraiva LR, Gavriouchkina D, Larsson T, Arendt D & Marioni JC (2015) High-throughput spatial mapping of single-cell RNA-seq data to tissue of origin. *Nat. Biotechnol.* **33:** 503–509 Available at: http://www.ncbi.nlm.nih.gov/pubmed/25867922 [Accessed December 10, 2016]

Bacher R & Kendziorski C (2016) Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biol.* **17:** 63 Available at: http://www.ncbi.nlm.nih.gov/pubmed/27052890 [Accessed December 5, 2016]

Belabbas M-A & Wolfe PJ (2009) On landmark selection and sampling in high-dimensional data analysis. *Philos. Trans. R. Soc.***:** 4295–4312

Bellman R (1961) Adaptive Control Processes: A Guided Tour Princeton university press

Cannoodt R, Saelens W & Saeys Y Computational methods for trajectory inference from single-cell transcriptomics.

Cannoodt R, Saelens W, Sichien D, Tavernier S, Janssens S, Guilliams M, Lambrecht BN, De Preter K & Saeys Y (2016) SCORPIUS improves trajectory inference and identifies novel modules in dendritic cell development. *bioRxiv*

Carreira-Perpinan (1996) Dimensionality reduction.

Carreira-Perpiñán MA (1996) Dimensionality reduction. **10:** 617–634 Available at: http://faculty.ucmerced.edu/mcarreira-perpinan/papers.html

Chi J & Melba MC (2012) Landmark selection using homogeneity on nonlinear manifolds for unmixing hyperspectral data. *IGARSS***:** 1373–1376

Cox TF & Cox MAA (2000) Multidimensional scaling Second edi. Chapman and Hall/CRC

Daigle BJ, Soltani M, Petzold LR & Singh A (2015) Gene expression Inferring single-cell gene expression mechanisms using stochastic simulation. **31:** 1428–1435

Doster T, Benedetto J & Czaja W (2011) Nonlinear Dimensionality Reduction for Hyperspectral Image Classification Final Report. **:** 1–32

Edgar R, Domrachev M & Lash AE (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30:** 207–210 Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/30.1.207 [Accessed December 10, 2016]

Faridani OR & Sandberg R (2015) Putting cells in their place. *Nat. Biotechnol.* **33:** 490–491 Available at: http://www.nature.com/doifinder/10.1038/nbt.3219 [Accessed December 10, 2016]

Featherstone K, Davis JRE, White MRH & Rand DA (2015) A stochastic transcriptional switch model for single cell. **:** 655–669

Garmire L, Poirion OB, Zhu X & Ching T (2016) Single-Cell Transcriptomics Bioinformatics and Computational Challenges. *Front. Genet.* **7:** 1–11

Grün D & van Oudenaarden A (2015) Design and Analysis of Single-Cell Sequencing Experiments. *Cell* **163:** 799–810

Haghverdi L, Buettner F & Theis FJ (2015) Gene expression Diffusion maps for high-dimensional single-cell analysis of differentiation data. **31:** 2989–2998

Haynes W (2013) Tukey's test. In *encyclopedia of systems biology* pp 2303–2304.

Hebenstreit D (2012) Methods, Challenges and Potentials of Single Cell RNA-seq. *Biology (Basel).* **1:** 658–667 Available at: www.mdpi.com/journal/biology

Ishizuka IE, Chea S, Gudjonson H, Constantinides MG, Dinner AR, Bendelac A & Golub R (2016) Single-cell

analysis defines the divergence between the innate lymphoid cell lineage and lymphoid tissue–inducer cell lineage. *Nat. Immunol.* **17:** 269–276 Available at: http://www.nature.com/doifinder/10.1038/ni.3344 [Accessed December 10, 2016]

Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, Mildner A, Cohen N, Jung S, Tanay A & Amit I (2014) Massively Parallel Single-Cell RNA-Seq for Marker-Free Decomposition of Tissues into Cell Types. *Science (80-.).* **343:**

Jiang L, Schlesinger F, Davis CA, Zhang Y, Li R, Salit M, Gingeras TR & Oliver B Synthetic spike-in standards for RNA-seq experiments. **:** 1543–1551

Junker JP, Noël ES, Guryev V, Peterson KA, Shah G, Huisken J, McMahon AP, Berezikov E, Bakkers J & van Oudenaarden A (2014) Genome-wide RNA Tomography in the Zebrafish Embryo. *Cell* **159:** 662–675 Available at: http://www.ncbi.nlm.nih.gov/pubmed/25417113 [Accessed December 10, 2016]

Junker JP & vanOudenaarden A (2015) Single-Cell Transcriptomics Enters the Age of Mass Production. *Mol. Cell* **58:** 563–564 Available at: http://dx.doi.org/10.1016/j.molcel.2015.05.019

Kim JK, Kolodziejczyk AA, Illicic T, Teichmann SA, Marioni JC, Yan L, Tang F, Shalek AK, Shalek AK, Marinov GK, Jaitin DA, Treutlein B, Patel AP, Kolodziejczyk AA, Kim JK, Svensson V, Marioni JC, Teichmann SA, Grün D, Kester L, et al (2015) Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nat. Commun.* **6:** 8687 Available at: http://www.nature.com/doifinder/10.1038/ncomms9687 [Accessed December 10, 2016]

Kim JK & Marioni JC (2013) Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. **:** 1–12

Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, Peshkin L, Weitz DA & Kirschner MW (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161:** 1187–1201 Available at: http://dx.doi.org/10.1016/j.cell.2015.04.044

L. TA, Roberto R & Sorin D (2008) Analysis of microarray experiments of gene expression profiling. **195:** 373–388

Lee JA, Renard E, Bernard G, Dupont P & Verleysen M (2013) Neurocomputing Type 1 and 2 mixtures of Kullback – Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing* **112:** 92–108 Available at: http://dx.doi.org/10.1016/j.neucom.2012.12.036

Lee JA & Verleysen M (2007) Nonlinear dimensionality reduction Springer, New York

Linnarsson S, Teichmann SA, Tang F, Lao K, Surani M, Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, Islam S, Kjallquist U, Moliner A, Zajac P, Fan J, Lonnerberg P, Ramskold D, Luo S, Wang Y, et al (2016) Single-cell genomics: coming of age. *Genome Biol.* **17:** 97 Available at: http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0960-x [Accessed December 10, 2016]

Liu S & Trapnell C (2016) Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research* **5:** 182 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4758375&tool=pmcentrez&rendertype=abstract

Lueks W, Mokbel B, Biehl M & Hammer B (2011) How to Evaluate Dimensionality Reduction ? – Improving the Co-ranking Matrix. **:** 1–12

Van Der Maaten L & Hinton G (2008) Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9:** 2579–2605

Van Der Maaten L, Postma E & Van Den Herik J (2009) Tilburg centre for Creative Computing Dimensionality Reduction: A Comparative Review Dimensionality Reduction: A Comparative Review. Available at: http://www.uvt.nl/ticc [Accessed December 1, 2016]

Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, Tirosh I, Bialas AR, Kamitaki N, Martersteck EM, Trombetta JJ, Weitz DA, Sanes JR, Shalek AK, Regev A & McCarroll SA (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161:** 1202–1214 Available at: http://dx.doi.org/10.1016/j.cell.2015.05.002

Maheshri N & Shea EKO (2007) Living with Noisy Genes : How Cells Function Reliably with Inherent Variability in Gene Expression.

Manuel L (2013) The irace Package : Iterated Racing for Automatic Algorithm Configuration ´ n er ´ Thomas St u IRIDIA – Technical Report Series Technical Report No .

Marcy Y, Ouverney C, Bik EM, Losekann T, Ivanova N, Martin HG, Szeto E, Platt D, Hugenholtz P, Relman DA & Quake SR (2007) Dissecting biological "dark matter" with single-cell genetic analysis of rare and uncultivated TM7 microbes from the human mouth. *Proc. Natl. Acad. Sci.* **104:** 11889–11894 Available at: http://www.ncbi.nlm.nih.gov/pubmed/17620602 [Accessed December 10, 2016]

McConnell MJ, Lindberg MR, Brennand KJ, Piper JC, Voet T, Cowing-Zitron C, Shumilina S, Lasken RS, Vermeesch JR, Hall IM & Gage FH (2013) Mosaic Copy Number Variation in Human Neurons. *Science (80- .).* **342:**

Moignard V, Woodhouse S, Haghverdi L, Lilly AJ, Tanaka Y, Wilkinson AC, Buettner F, Macaulay IC, Jawaid W, Diamanti E, Nishikawa S, Piterman N, Kouskoff V, Theis FJ, Fisher J & Göttgens B (2015) Articles Decoding the regulatory network of early blood development from single-cell gene expression measurements. **33:**

Munsky B, Neuert G & van Oudenaarden A (2012) Using gene expression noise to understand gene regulation. *Science* **336:** 183–7 Available at: http://www.ncbi.nlm.nih.gov/pubmed/22499939 [Accessed December 10, 2016]

Musselman RD & Sanchez SM (2007) ROBUSTNESS: A BETTER MEASURE OF ALGORITHM PERFORMANCE.

Nagalakshmi U, Waern K & Snyder M (2010) RNA-Seq : A Method for Comprehensive Transcriptome Analysis. **:** 1–13

Norel R, Rice JJ & Stolovitzky G (2011) The self-assessment trap: can we all be better than average? *Mol. Syst. Biol.* **7:** 537 Available at: http://www.ncbi.nlm.nih.gov/pubmed/21988833 [Accessed December 1, 2016]

Ocone A, Haghverdi L, Mueller NS & Theis FJ (2015) Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data.

Okoniewski MJ & Miller CJ (2006) Hybridization interactions between probesets in short oligo microarrays lead to spurious correlations. *BMC Bioinformatics* **7:** 276

Park H, Chant J, Schena M, Shalon D, Davis RW & Brownt P (1995) Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray.

Paul F, Arkin Y, Giladi A, Jaitin DA, Kenigsberg E, Keren-Shaul H, Winter D, Lara-Astiaso D, Gury M, Weiner A, David E, Cohen N, Lauridsen FKB, Haas S, Schlitzer A, Mildner A, Ginhoux F, Jung S, Trumpp A, Porse BT, et al (2015) Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell* **163:** 1663–77 Available at: http://www.ncbi.nlm.nih.gov/pubmed/26627738 [Accessed December 10, 2016]

Pierson E & Yau C (2015) Open Access ZIFA : Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.***:** 1–10 Available at: http://dx.doi.org/10.1186/s13059-015-0805-z

Rafailidis D, Constantinou E & Manolopoulos Y (2017) Landmark selection for spectral clustering based on Weighted PageRank. *Futur. Gener. Comput. Syst.* **68:** 465–472 Available at: http://dx.doi.org/10.1016/j.future.2016.03.006

Regev A, Teichmann SA, Lander ES & Amit I (2017) The Human Cell Atlas.

Roweis ST & Saul LK (2000) Nonlinear Dimensionality Reduction by Locally Linear Embedding. **290:** 2323–2326

Saelens W A comprehensive evaluation of module detection methods for gene expression data.

Saeys Y (2016) Data mining 2. Data preprocessing. In

Sáez JA, Luengo J & Herrera F (2016) Evaluating the classifier behavior with noisy data considering performance and robustness: The Equalized Loss of Accuracy measure. *Neurocomputing* **176:** 26–35 Available at: http://linkinghub.elsevier.com/retrieve/pii/S0925231215005500 [Accessed November 30, 2016]

Satija R, Farrell JA, Gennert D, Schier AF & Regev A (2015) Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33:** 495–502 Available at: http://dx.doi.org/10.1038/nbt.3192

Schlitzer A, Sivakamasundari V, Chen J, Sumatoh HR Bin, Schreuder J, Lum J, Malleret B, Zhang S, Larbi A, Zolezzi F, Renia L, Poidinger M, Naik S, Newell EW, Robson P & Ginhoux F (2015) Identification of cDC1- and cDC2-committed DC progenitors reveals early lineage priming at the common DC progenitor stage in the bone marrow. *Nat Immunol* **16:** 718–728 Available at: http://dx.doi.org/10.1038/ni.3200%5Cn10.1038/ni.3200%5Cnhttp://www.nature.com/ni/journal/v16/n7/abs/ni.3200.html#supplementary-information

Schneider J (1997) Cross validation. Available at: https://www.cs.cmu.edu/~schneide/tut5/node42.html

Scott DW & Thompson JR we can do a good job of estimating the density function and its derivatives if only the sam- ple size is suf fi ciently large. Unfortunately, almost all of the supporting numerical investigations have been done in.

Shalek AK, Satija R, Shuga J, Trombetta JJ, Gennert D, Lu D, Chen P, Gertner RS, Gaublomme JT, Yosef N, Schwartz S, Fowler B, Weaver S, Wang J, Wang X, Ding R, Raychowdhury R, Friedman N, Hacohen N, Park H, et al (2014) Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* **510:** 363–9 Available at: http://www.nature.com.docelec.univ-lyon1.fr/nature/journal/v510/n7505/full/nature13437.html#f1

Shapiro E, Biezuner T & Linnarsson S (2013) Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat. Rev. Genet.* **14:** 618–630 Available at: http://www.ncbi.nlm.nih.gov/pubmed/23897237 [Accessed December 10, 2016]

Shi H, Yin B, Zhang X, Kang Y & Lei Y (2015) A Landmark Selection Method for L-Isomap Based on Greedy. **:** 7371–7376

Shin J, Berg DA, Christian KM, Shin J, Berg DA, Zhu Y, Shin JY, Song J & Bonaguidi MA (2015) Single-Cell RNA-Seq with Waterfall Reveals Molecular Cascades underlying Adult Neurogenesis Resource Single-Cell RNA-Seq with Waterfall Reveals Molecular Cascades underlying Adult Neurogenesis. *Stem Cell* **17:** 360–372 Available at: http://dx.doi.org/10.1016/j.stem.2015.07.013

Silva JG & Isr I (1950) Selecting Landmark Points for Sparse Manifold Learning.

Silva V De & Tenenbaum JB (2003) Global versus local methods in nonlinear dimensionality reduction # DCFE. *MIT Press***:** 705–712

Silva V De & Tenenbaum JB (2004) Sparse multidimensional scaling using landmark points. **:** 1–41

Svensson V, Natarajan KN, Ly L, Miragaia RJ, Labalette C, Macaulay IC, Cvejic A & Teichmann SA (2017) Power analysis of single-cell RNA-sequencing experiments. *Nat. Publ. Gr.* Available at: http://dx.doi.org/10.1038/nmeth.4220

Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, Wang X, Bodeau J, Tuch BB, Siddiqui A, Lao K & Surani MA (2009) mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* **6:** 377–382 Available at: http://www.nature.com/doifinder/10.1038/nmeth.1315 [Accessed December 10, 2016]

Tang F, Lao K & Surani MA (2011) Development and applications of single-cell transcriptome analysis. **8:**

Tenenbaum JB, Silva V De & Langford JC (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction. **290:** 2319–2323

The chromium single-cell 3' Solution : datasets Available at: https://www.10xgenomics.com/single-cell/

Torgerson WS (1952) Multidimensional scaling: I. Theory and method. *Psychometrika* **17:** 401–419 Available at: http://link.springer.com/10.1007/BF02288916 [Accessed December 5, 2016]

Trapnell C (2015) Defining cell types and states with single-cell genomics. *Genome Res.* **25:** 1491–8 Available at: http://www.ncbi.nlm.nih.gov/pubmed/26430159 [Accessed December 10, 2016]

Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, Lennon NJ, Livak KJ, Mikkelsen TS & Rinn JL (2014a) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32:** 381–6 Available at: http://dx.doi.org/10.1038/nbt.2859%5Cnhttp://www.nature.com/doifinder/10.1038/nbt.2859

Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Morse M, Lennon NJ, Livak KJ, Mikkelsen TS, Rinn L & Biology R (2014b) Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nat Biotechnol* **32:** 381–386

Treutlein B, Brownfield DG, Wu AR, Neff NF, Mantalas GL, Espinoza FH, Desai TJ, Krasnow MA, Quake SR, B T, DG B, AR W, NF N, GL M, FH E, TJ D, MA K & SR. Q (2014) Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature* **509:** 371–5 Available at: http://www.ncbi.nlm.nih.gov/pubmed/24739965 [Accessed December 5, 2016]

Wagner A, Regev A & Yosef N (2016) Revealing the vectors of cellular identity with single-cell genomics. *Nat. Publ. Gr.* **34:** 1145–1160 Available at: http://dx.doi.org/10.1038/nbt.3711

Wang B, Zhu J, Pierson E, Ramazzotti D & Batzoglou S (2017) Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Publ. Gr.* **14:** 414–416 Available at: http://dx.doi.org/10.1038/nmeth.4207

Wang W, Hu Y, Sun D, Staehelin C, Xin D & Xie J (2012) Identification and evaluation of two diagnostic markers linked to Fusarium wilt resistance (race 4) in banana (Musa spp.). *Mol. Biol. Rep.* **39:** 451–459

Wang Z, Gerstein M & Snyder M (2010) RNA-Seq : a revolutionary tool for transcriptomics. **10:** 57–63

Xu H & Mannor S Robustness and Generalization.

Yu L, Ye J & Liu H Dimensionality Reduction for Data Mining -Techniques, Applications and Trends.

Yu P & Lin W (2016) Single-cell Transcriptome Study as Big Data. *Genomics, Proteomics Bioinforma.* **14:** 21–30 Available at: http://dx.doi.org/10.1016/j.gpb.2016.01.005

Zeisel A, Manchado ABM, Codeluppi S, Lonnerberg P, La Manno G, Jureus A, Marques S, Munguba H, He L, Betsholtz C, Rolny C, Castelo-Branco G, Hjerling-Leffler J & Linnarsson S (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science (80-.).* **347:** 1138–42 Available at: http://science.sciencemag.org.docelec.univ-lyon1.fr/content/347/6226/1138.abstract

Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Ziraldo SB, Wheeler TD, Mcdermott GP, Zhu J, Mark T, Shuga J, Montesclaros L, Masquelier DA, Nishimura SY, Schnall-levin M, Wyatt PW, Hindson CM, Bharadwaj R, Ness KD, Beppu LW, et al (2016) Massively parallel digital transcriptional profiling of single cells.

Zhu X & Wu X (2004) Class Noise vs. Attribute Noise: A Quantitative Study. *Artif. Intell. Rev.* **22:** 177–210 Available at: http://link.springer.com/10.1007/s10462-004-0751-8 [Accessed November 30, 2016]

Ziegenhain C, Parekh S, Heyn H, Hellmann I, Enard W, Guillaumet-adkins A & Smets M (2017) Comparative Analysis of Single-Cell RNA Sequencing Methods Comparative Analysis of Single-Cell RNA Sequencing Methods. **:** 631–643

VII Supplementary

#	Ref.	Accession	Species	# Cells	Description research
1	Schlitzer et al. 2015	GSE60783	Mouse	242	DC development: MDP > CDP > PreDC
2	Deng et al. 2014	GSE45719	Mouse	317	Embryo development: zygote, early 2 cell, mid 2 cell, late 2 cell, 4 cell, 8 cell, 16 cell, early blastocyst, mid, late
3	Gokce et al. 2016	GSE82187	Mouse	1208	Subpopulations in stratial cells
4	Shalek et al. 2015	GSE48968	Mouse	1775	Response: DCs untreated/ treated with 3 pathogenic components
6	Molinaro et al. 2016	GSE79866	Schmidtae mediterranae	169	Lineage tracing neural stem cells X1/X2
7	Klein et al. 2015	GSE65525	Mouse	2717	Droplet-based high-throughput barcoding and sequencing
8	Macosko et al. 2015	GSE63472	Mouse/ Homo sapiens	44808	Droplet-based high-throughput barcoding and sequencing
9.1	Kowalczyk et al. 2015	GSE59114	Mouse	524	HSCs hematopoietic stem cells development: LT-HSC > ST-HSC > MPP
9.2	Kowalczyk et al. 2015	GSE59114	Mouse	498	HSCs hematopoietic stem cells development: LT-HSC > ST-HSC > MPP
9.3	Kowalczyk et al. 2015	GSE59114	Mouse	284	HSCs hematopoietic stem cells development: LT-HSC > ST-HSC > MPP
9.4	Kowalczyk et al. 2015	GSE59114	Mouse	280	HSCs hematopoietic stem cells development: LT-HSC > ST-HSC > MPP
12	Treutelein et al. 2014	GSE52583	Mouse	66	Lineage reconstruction of lung epithelium
13	Trapnell et al. 2015	GSE52529	Homo sapiens	314	Pseudotemporal ordering of differentiation process to myoblast
19	Bjorklund et al 2016	GSE70580	Homo sapiens	648	Heterogeneity in ILC (innate lymphoid cells) and NK
20	Dunkel et al 2016	see github	Mouse	712	Heterogeneity and Transcriptional Dynamics in the Adult Neural Stem Cell Lineage
21	Yao et al 2016	GSE86977	Homo sapiens	2684	Lineage Bifurcation in Human ESC of brain development
22	Zheng et al.2016	10X genomics	Homo sapiens	66991	Subpopulation detection of PBMC (Peripheral blood monocellulars)
23	Zheng et al.2016	10X genomics	Homo sapiens	6303	Subpopulation detection of PBMC (Peripheral blood monocellulars)
24	Zheng et al.2016	10X genomics	Homo sapiens	12600	Subpopulation detection of PBMC (Peripheral blood monocellulars)
25	Zheng et al.2016	10X genomics	Homo sapiens	22003	Subpopulation detection of PBMC (Peripheral blood monocellulars)

List S1: Overview of the benchmark of single cell datasets used in this study. Datasets had to contain class labels for the cells that were experimentally beforehand. We used 20 datasets from 14 different studies. The datasets cover different field of single cell research such as subpopulation detection and dynamic processes like differentiation and responses to stimuli

List S2: Overview of the dimensionality reduction method and their characteristics that were reviewed in this study

Overview Dimensionality reduction methods and their features		
PCA Principal Component Analysis	INPUT	Dataset X
	Complexity	$O(p^2n + p^3)$
	Focus	Retain variance of data
	Method	1) Calculate covariance matrix M 2) Eigendecomposition of covariance matrix
	Package	prcomp()
	Feature	Linear
Classical MDS Multidimensional scaling (Torgerson, 1952)	INPUT	Pairwise (Euclidean) distance matrix D of dataset X
	Complexity	$O(kN^2 + N^3)$
	Focus	Preserving pairwise Euclidean distances dij
	Method	1) Calculate Gram matrix (NxN matrix derived from D, double-centered) 2) Eigendecomposition of Gram matrix
	Package	cmdscale()
	Feature	Linear
Isomap (Tenenbaum *et al*, 2000)	INPUT	Pairwise distance matrix D of dataset X
	Complexity	O(N2logN)
	Focus	Preserving pairwise Geodesic distance dij
	Method	1) Construct neighborhood graph of k nearest neighbors 2) Compute shortest path 3) Eigendecomposition using MDS
	Package	RDRToolbox::Isomap()
	Feature	Non Linear
Kernel PCA	INPUT	Dataset X
	Complexity	$O(p^2n + p^3)$
	Focus	Preserving pairwise distance
	Method	1) Kernel function computes Kernel matrix K 2) K is double-centered 3) Eigendecomposition of double-centered K Note: If kernel function is linear, procedure identical to classical scaling

	Package	Kernlab::kpca()
	Feature	Linear and non-linear depending on kernel function
Sammon	Input	Dataset X
	Complexity	$O(p^2n + p^3)$
	Focus	Preserving local pairwise distances
	Method	Similar to MDS but cost function is adapted by weighting the contribution of each pair (i,j) to the cost function by the inverse of their pairwise distance in the high-dimensional space dij Cost function equation
	Package	MASS::sammon()
	Feature	Linear and nonconvex
isoMDS Kruskal's Non-metric Multidimensional Scaling (Cox & Cox, 2000)	INPUT	Dataset X
	Complexity	O()
	Focus	Preserving pairwise distances
	Method	Scaling rank order of dissimilarities (relative) 1) Choose initial random configuration 2) Calculate distances dij from configuration 3) Find optimal non-parametric monotonic transformation of the proximities f(x) 4) Minimize Kruskal's stress function between the optimally scaled data and the distances by finding a new configuration of points. Stress function equation 5) If the stress is small enough then exit the algorithm else return to 2. Similar to MDS but MDS restricted to rigid relationship between dissimilarities and distances and Euclidean geometry
	Package	MASS::isoMDS()
	Feature	Non Linear
LMDS Landmark MDS (Silva & Tenenbaum, 2004)	INPUT	Dataset X
	Complexity	O(nN)
	Focus	Preserving global pairwise distances
	Method	Similar to MDS but embedding is done on a subset of the data, the landmark points

		1) Landmark selection 2) Classical MDS on landmarks 3) Embedding other points based on embedding landmarks Reduce complexity when number of datapoints N is large
	package	Mds_withlandmarks (self made)
	Feature	Linear
LLE Local Linear Embedding (Roweis & Saul, 2000)	INPUT	Dataset X
	Complexity	O(dN2) eigenvalue decomposition
	Focus	Preserving local properties of the data
	Method	Similar to Isomap in that it constructs a graph representation 1) Compute the neighbors of each data point, . 2) Compute the weights that best reconstruct each data point from its neighbors, minimizing the cost in eq. (1) by constrained linear fits. 3) Compute the vectors best reconstructed by the weights, minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.
	package	RDRToolbox::LLE
	Feature	Non linear and sparse
Diffusion Map Principal Component Analysis (Haghverdi *et al*, 2015)	INPUT	Dataset X
	Complexity	
	Focus	Preserving pairwise diffusion distances
	Method	1) Graph matrix W is constructed, the weights of the edges are computed using the Gaussian Kernel function 2) N x N Markovian transition probability matrix P 3) Eigendecomposition of matrix P
	package	diffusionMap::diffuse
	Feature	Non Linear
tSNE t-Distributed stochastic neighborhood embedding	INPUT	Dataset X
	Complexity	
	Focus	Preserving local properties of the data
	Method	1) converting high-dimensional Euclidean distances between datapoints into

(Van Der Maaten & Hinton, 2008)		conditional probabilities representing similarities 2) minimize Kullback-Leibler divergences
	Package	Tsne::tsne
	Feature	Non-linear

List S3: Parameter space grid search of dimensionality reduction methods

Grid search of the parameter space of a set of dimensionality reduction methods			
PCA	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
Classical MDS	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	Distance.measure	{Euclidean, spearman}	Function to calculate pairwise distance matrix
Isomap	dims	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	k	{4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}	Number of neighbours to create neighbourhood graph
Kernel PCA	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	kernel	{polydot, rbfdot, anovadot}	Kernel functions
	Kernel = polydot	Degree = {1,2,3} Scale = {1,2,3} Offset = {1,2,3}	Polynomial kernel
	Kernel = rbfdot	Sigma = {1,2,3}	Gaussian RBF kernel
	Kernel = anovadot	sigma = {1,2,3} degree = {1,2,3}	Hyperbolic tangent kernel
Sammon	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	Distance.measure	{Euclidean, spearman}	Function to calculate pairwise distance matrix
	Niter	{60, 80, 100, 120, 140}	
	Magic	{0.1, 0.2, 0.3, 0.4}	

	tol	{1e-2, 1e-3, 1e-4}	Convergence tolerance.
isoMDS	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	Distance.measure	{Euclidean, spearman}	Function to calculate pairwise distance matrix
	maxit	{25, 50, 150}	The maximum number of iterations.
	P	{1, 2, 3, 4}	Power for Minkowski distance in the configuration space.
	tol	{1e-2, 1e-3, 1e-4}	Convergence tolerance.
LMDS	k	{5}	Dimensions d in low-dimensional space
	Dist.fun	Correlation. distance	Function to calculate pairwise distance matrix
	Landmark.method	{naïve, minmax, degree centrality}	Method for landmark selection
	Num.landmarks	{25, 50, 75, 100, 150, 200, 250}	Number of landmark points
	Pca.normalisation	FALSE	
	Landmark.method = naive		
	Landmark.method = Minmax	Num.seed.landmarks = {10, 15, 20}	
	Landmark.method = Degree centrality	Subsampling = {simple, advanced} Nknn = {5, 10, 15, 20}	
LLE	k	{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}	Dimensions d in low-dimensional space
diffusionMap	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	Distance.measure	{Euclidean, spearman}	Function to calculate pairwise distance matrix
	Maxdim	{25, 50, 75, 100}	the maximum number of diffusion map dimensions returned if 95% drop-off is not attained.
	T	{-1, 0, 1}	optional time-scale parameter in the diffusion map. The

			(recommended) default uses multiscale geometry.
	delta	{10^-3, 10^-4, 10^-5, 10^-6, 10^-7}	Sparsity cut-off for the symmetric graph Laplacian. Default of 10^-5 is used. Higher value induces more sparsity in Laplacian (and faster computations)
tSNE	k	{2, 3, 4, 5}	Dimensions d in low-dimensional space
	Distance.measure	{Euclidean, spearman}	Function to calculate pairwise distance matrix
	Initial dim.	{50, 100, 150}	The number of dimensions to use in reduction method.
	Perplexity	{20, 30, 40}	Perplexity parameter. (Optimal number of neighbours)
	Maximum iterations	{500, 1000, 1500}	Maximum number of iterations to perform.
	Minimum cost	{0,0.5, 1}	The minimum cost value (error) to halt iteration.
	Whiten	{TRUE, FALSE}	A Boolean value indicating whether the matrix data should be whitened.
	Epoch	{50, 100, 150}	The number of iterations in between update messages.

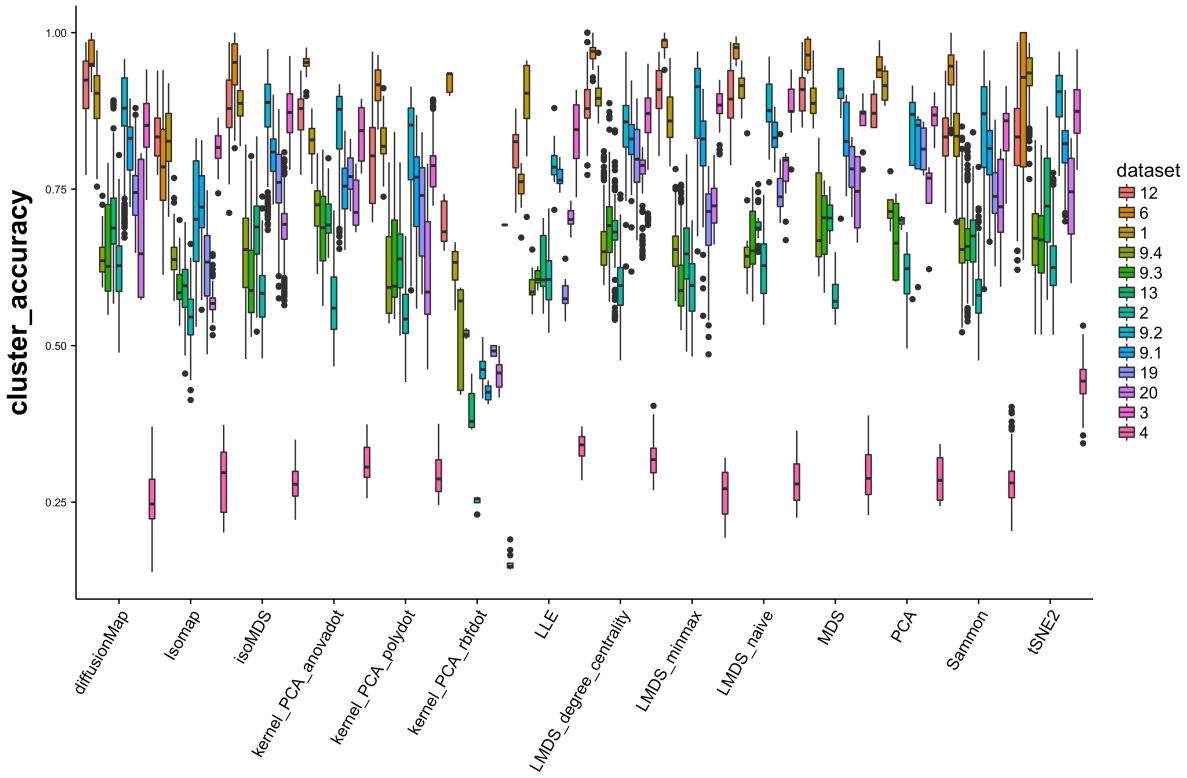

Figure S1: Overall absolute cluster accuracy for dimensionality reduction methods. The score for every parameter setting of each dimensionality reduction is represented for every dataset separately. Datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4 = 1790 cells)

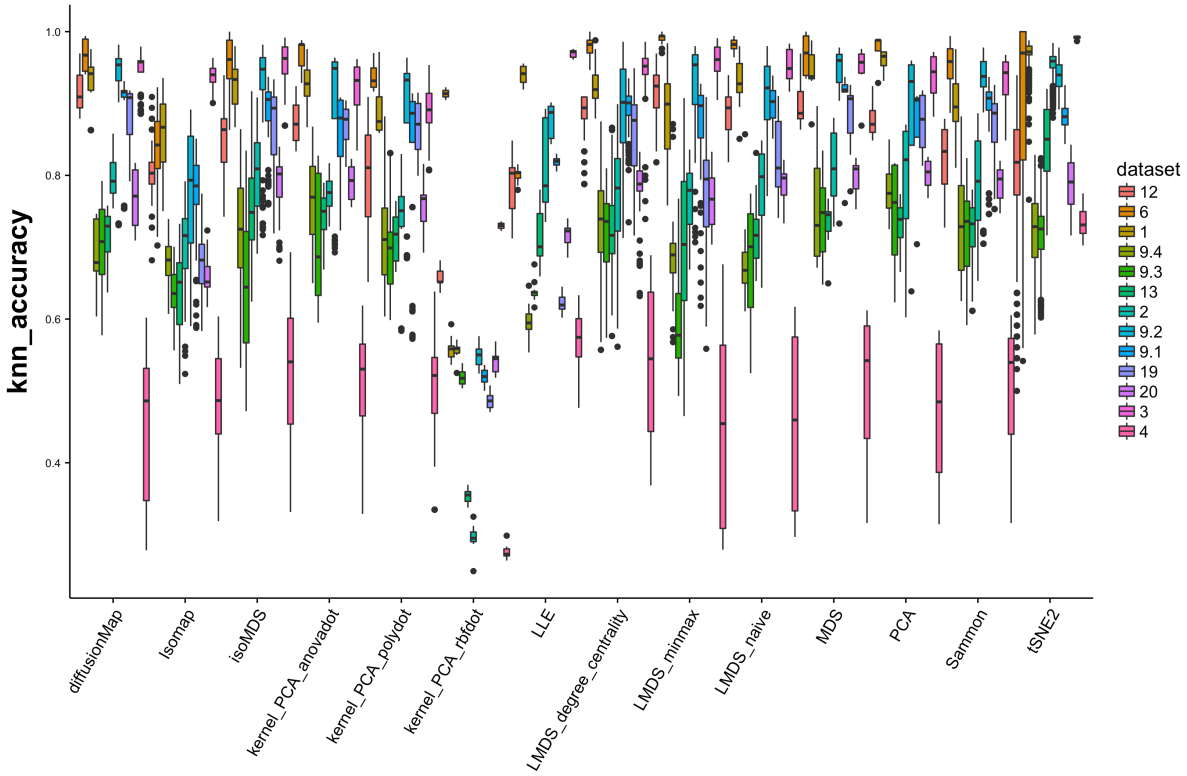

Figure S2: Overall absolute KNN accuracy score for dimensionality reduction methods. The score for every parameter setting of each dimensionality reduction is represented for every dataset separately. Datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4 = 1790 cells)

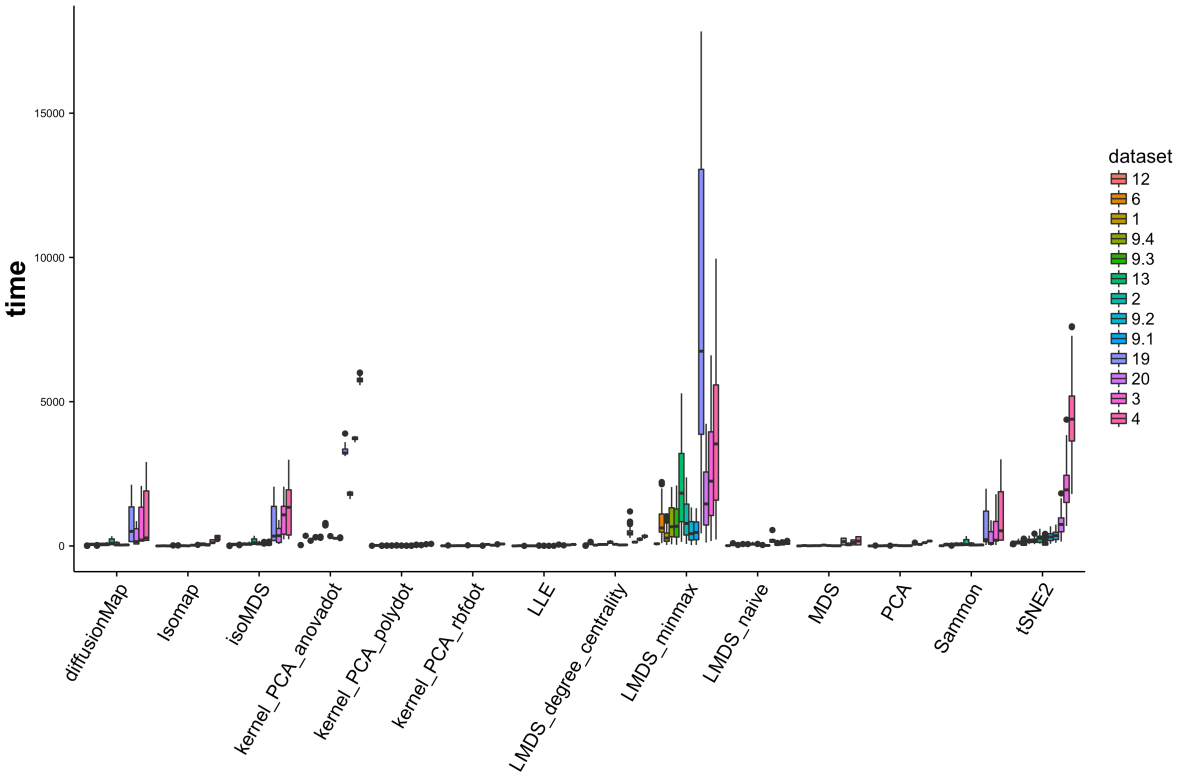

Figure S3: Absolute computing time for a set of dimensionality reduction methods. The absolute computing for every parameter setting is shown for every dataset separately for a set of dimensionality reduction methods. The datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4: 1799 cells).

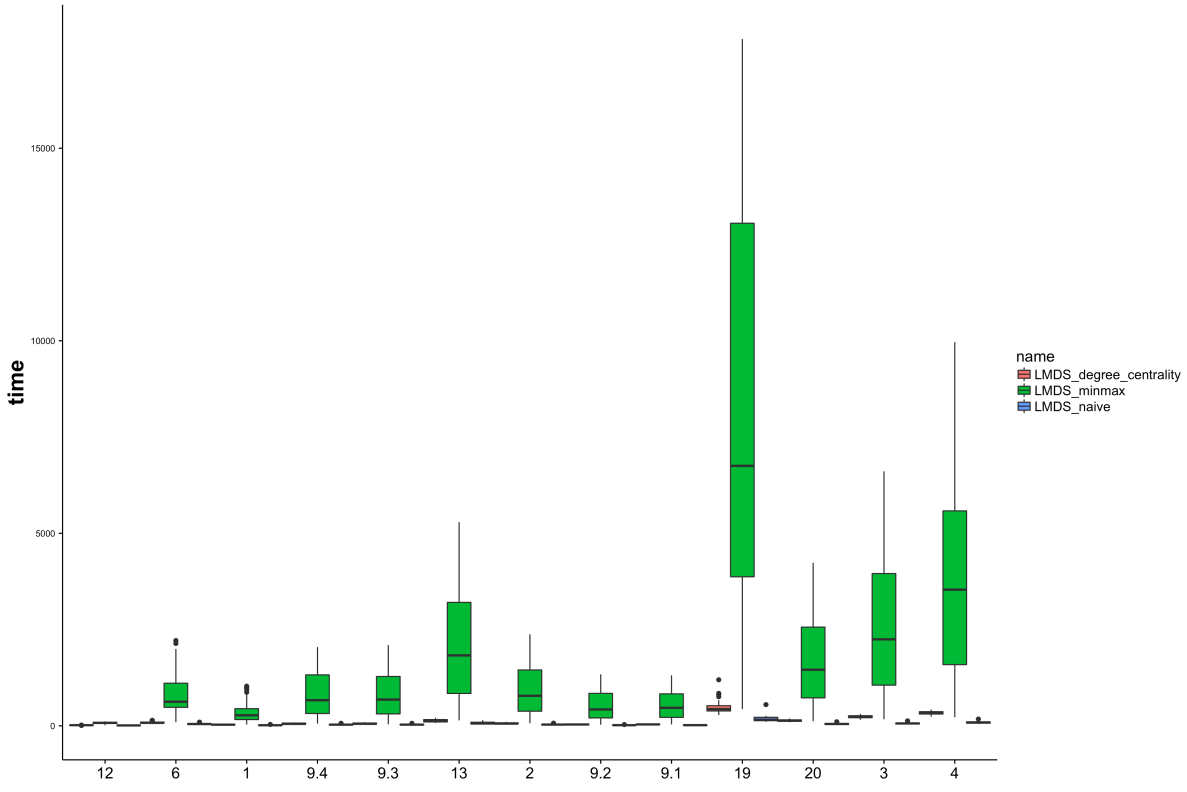

Figure S4: Absolute computing time of LMDS. The absolute computing time of the three different landmark selection methods of LMDS is shown for the subset of small single cell datasets

List S4: Best parameter settings for each dimensionality reduction method based on a parameter grid search

Grid search of the parameter space of a set of dimensionality reduction methods		
	On small single cell data	**On small and large single cell data**
PCA	K = 5	K = 5
Classical MDS	K = 5, Distance.measure = spearman	K = 5, Distance.measure = spearman
Isomap	Dims = 5, k = 15	Dims = 5, k = 15
Kernel PCA Polydot	K = 5 , Degree = 1, Scale = {1,2,3}, Offset = {1,2,3}	K = 5 , Degree = 1, Scale = {1,2,3}, Offset = {1,2,3}
Rbfdot	K = 5, sigma = 3 K = {2,3}, sigma = 2	K = 5, sigma = {1,3}
Anovadot	K = 5, sigma = {2,3}, degree = 1	K = 5, sigma = {2,3}, degree = 1
Sammon	K = 5, Distance.measure = spearman, magic = {60,80,100,120}, magic = 0.4, tol = {1e-3,1e-4} K = 5, Distance.measure = spearman, magic = {60,80,100,120}, magic = 0.2, tol = {1e-2,1e-4}	K = 5, Distance.measure = spearman, magic = {60,80,100,120}, magic = {0.2,0.4} , tol = {1e-3,1e-4}
isoMDS	K = 5, Distance.measure = spearman, maxit = 150, p = 2, tol = 1e-4	K = 5, Distance.measure = spearman, maxit = 150, p = 2, tol = 1e-4
LMDS Naïve	K = 5, Distance.measure = correlation distance	K = 5, Distance.measure = correlation distance
	Num.landmarks = {200,250}	Num.landmarks = {200,250}
Degree centrality	Num.landmarks = {100,150}, Subsampling = {simple, advanced}, Nknn = {20}	Num.landmarks = {100,150}, Subsampling = {simple, advanced}, Nknn = {20}
LLE	K = {11,14}	K = {11,14}
diffusionMap	K = 5, Distance.measure = spearman, delta = 1, maxdim = {25, 50, 75, 100}, T = {10^-3, 10^-4, 10^-5, 10^-6, 10^-7}	K = 5, Distance.measure = spearman, delta = 1, maxdim = {25, 50, 75, 100}, T = {10^-3, 10^-4, 10^-5, 10^-6, 10^-7}
tSNE	K = 4, Distance.measure =	

spearman, 100, 40, 1500, 0, TRUE, 50 K = 3, Distance.measure = spearman, 100, 40, 500, 0.5, TRUE, 100 K = 3, Distance.measure = spearman, 40, 1000, 0, FALSE, 100 K = 4, Distance.measure = spearman, 50, 40, 1500, 0.5, FALSE, 100

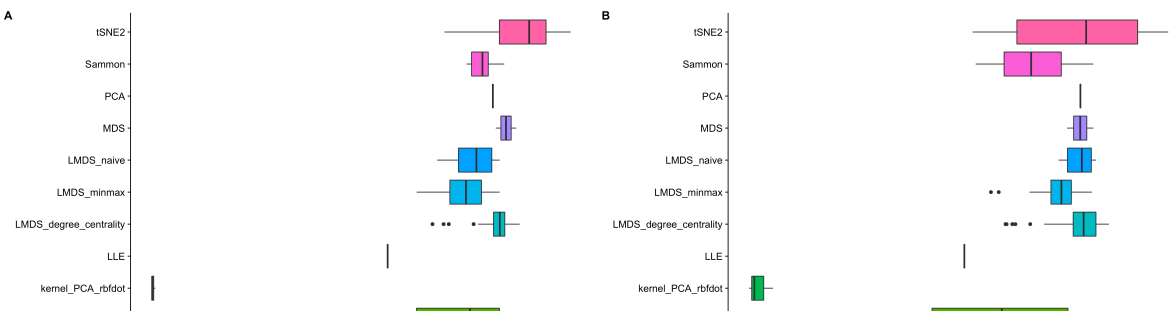

Figure S5: Relative performance of different dimensionality reduction methods into a three-dimensional space before parameter tuning based on three quality metrics for small single cell datasets (<2000 cells). The dimension of the low dimensional space on which the datapoints are embedded is fixed on two (k = 3)**.** The parameters of each method are varied according to a grid search. For each method the average score over all datasets of each parameter setting is represented. **A.** Relative KNN accuracy **B**. relative cluster accuracy **C-D** relative coRanking metric **C.** Relative Qglobal **D**. Relative Qlocal **E**. Relative mean Rnx **F.** Relative auc Rnx

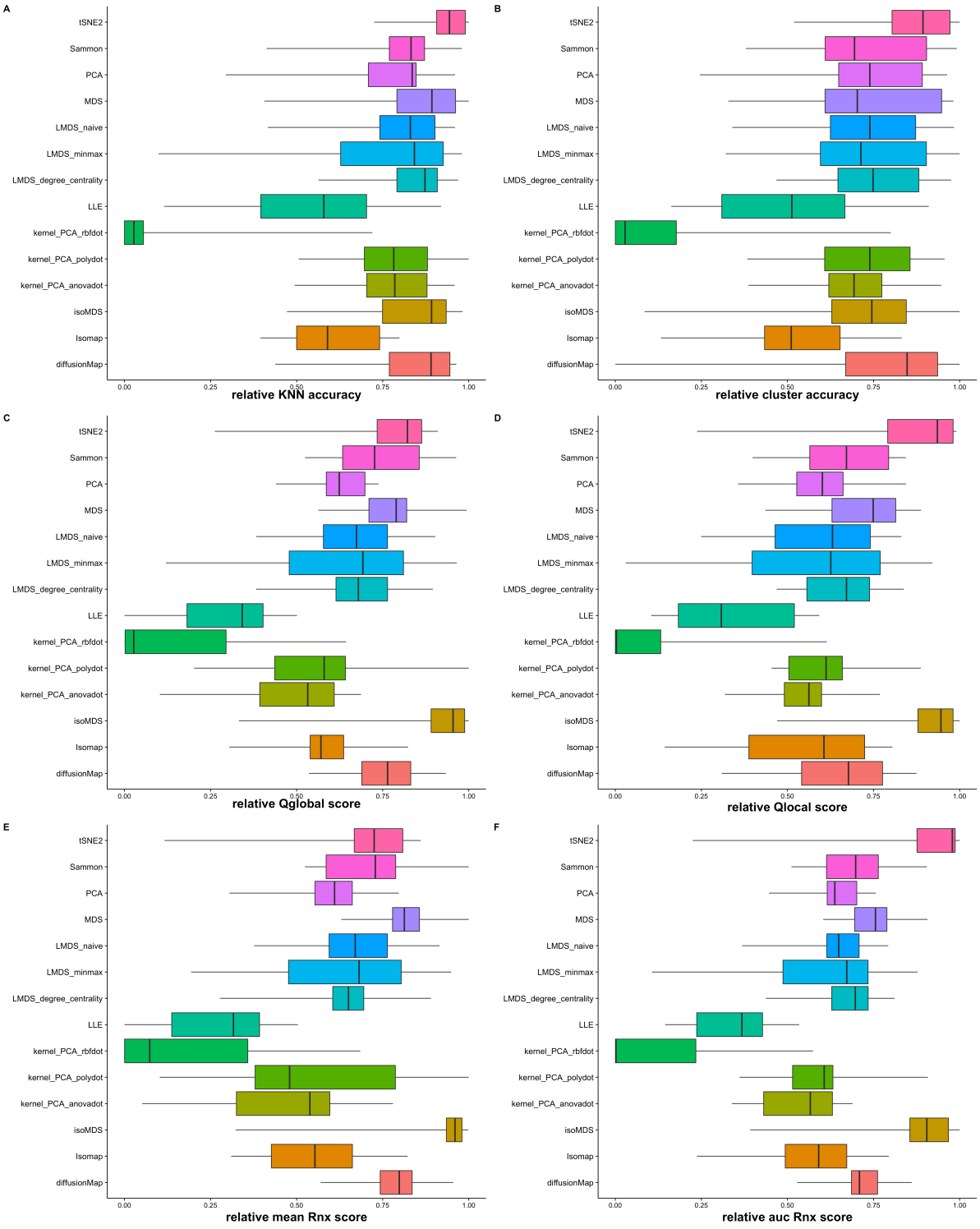

Figure S6: Relative performance of different dimensionality reduction methods into a three-dimensional space based on three quality scores after parameter tuning on small single cell datasets (<2000 cells). Parameter k presenting the number of dimensions in the low-dimensional space is fixed on three. For each method the score of the best parameter setting of every dataset is represented. A. Relative KNN accuracy B. relative Cluster accuracy. C. Relative Qglobal D. Relative Qlocal E. Relative mean Rnx F. Relative auc Rnx

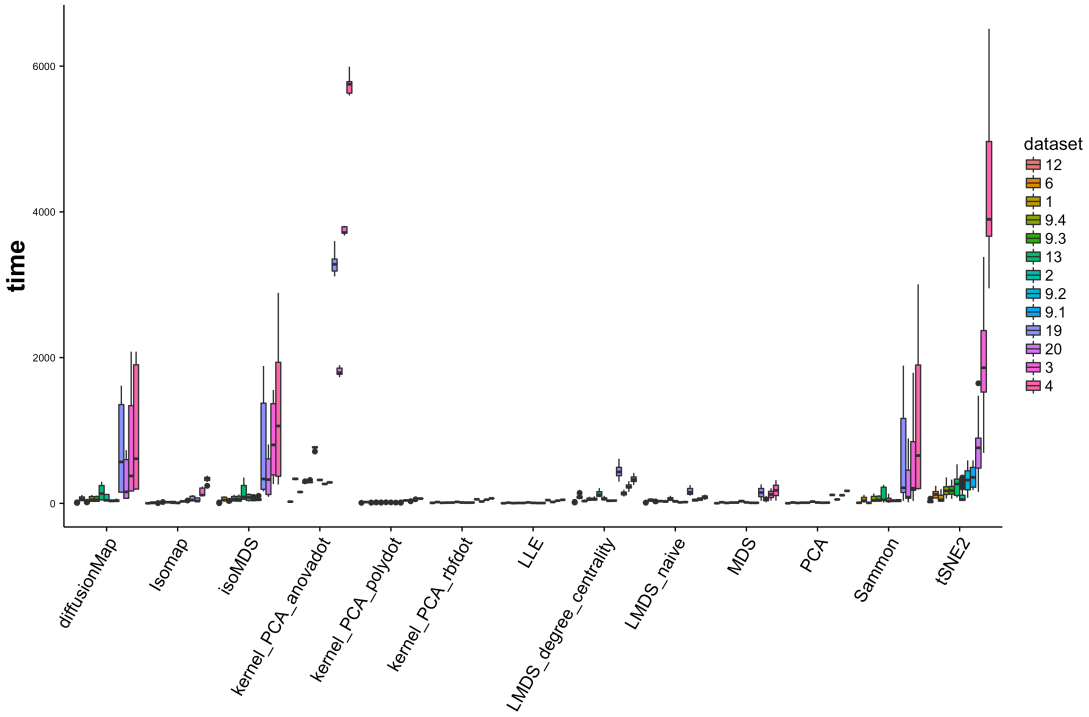

Figure S7: Absolute computing time of a dimensionality reduction into a three-dimensional space for a set of dimensionality reduction methods on small single cell datasets (<2000 cells). The absolute computing for every parameter setting with parameter k fixed on three is shown for every dataset separately for a set of different dimensionality reduction methods. The datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4: 1799 cells).

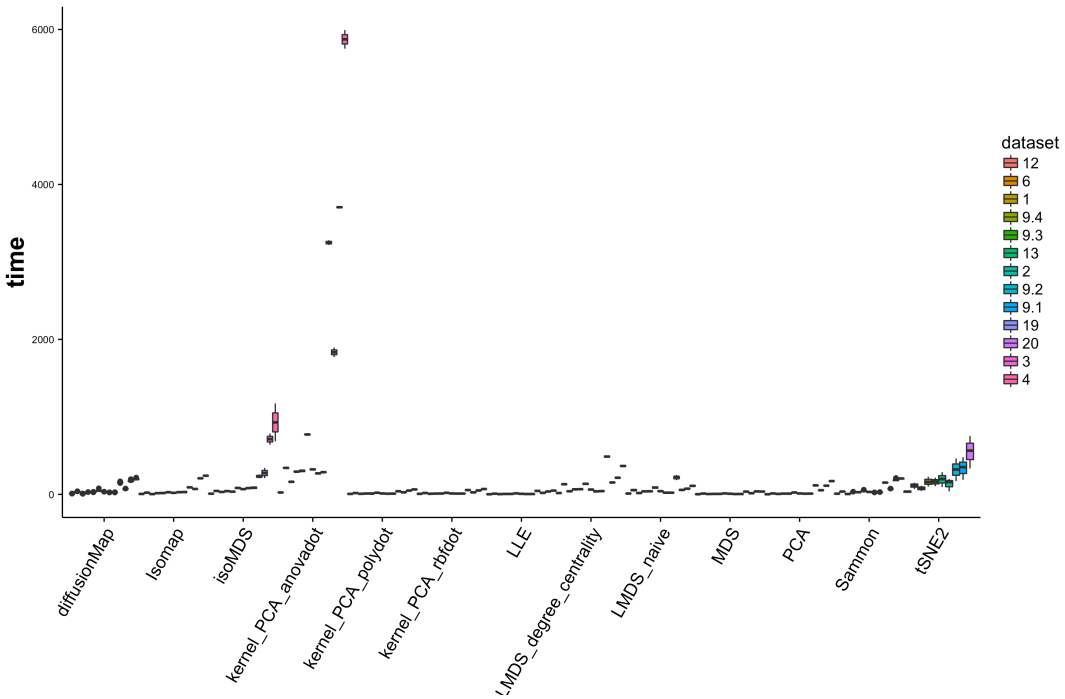

Figure S8: Absolute computing time of a dimensionality reduction into a three-dimensional space for a set of dimensionality reduction methods after parameter optimization on small single cell dataset (<2000 cells). At front the best parameter settings were selected with parameter grid search and LOOCV. The absolute computing for the best parameter setting with parameter k fixed on three is shown for every dataset separately for a set of different dimensionality reduction methods. The datasets are ordered from small (dataset 12: 66 cells) to large (dataset 4: 1799 cells).

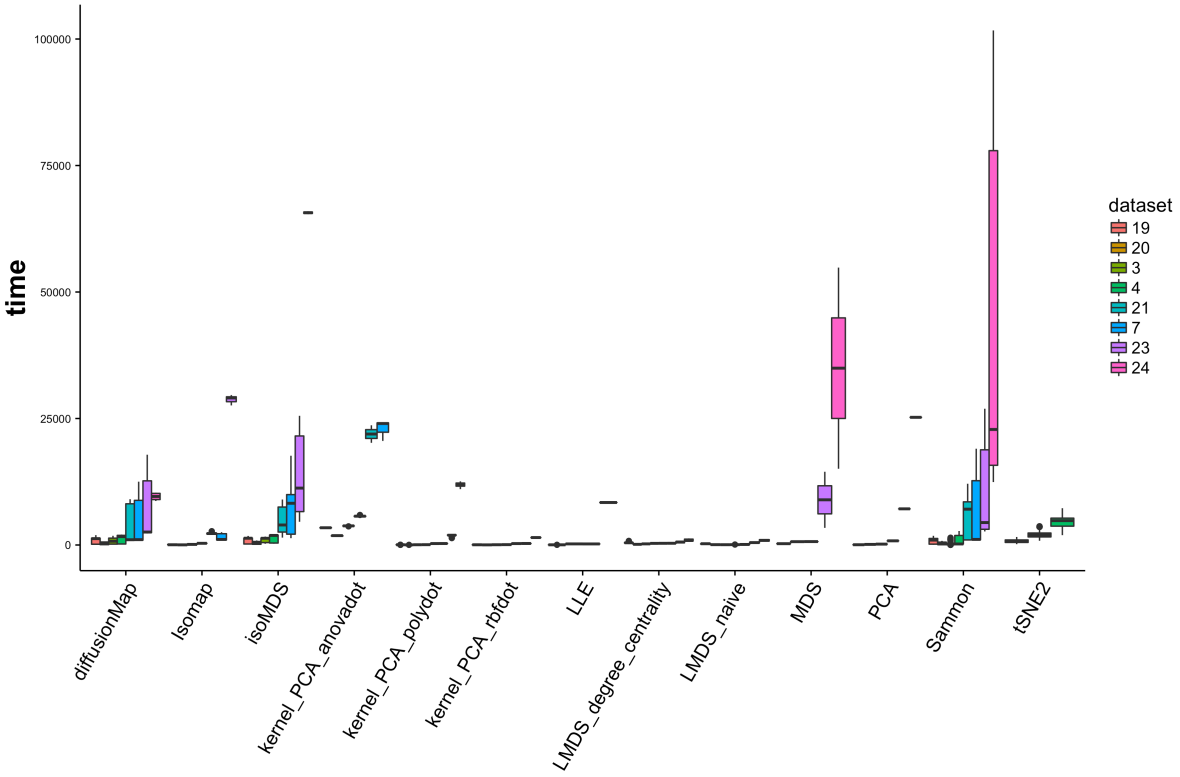

Figure S9: Absolute computational time of dimensionality reduction methods before parameter tuning on large datasets (>500 cells). The absolute computing time for every parameter setting of each dimensionality reduction method is shown for every dataset separately. The datasets are ordered from small (dataset 119: 648 cells) to large (dataset 24: 70000 cells).

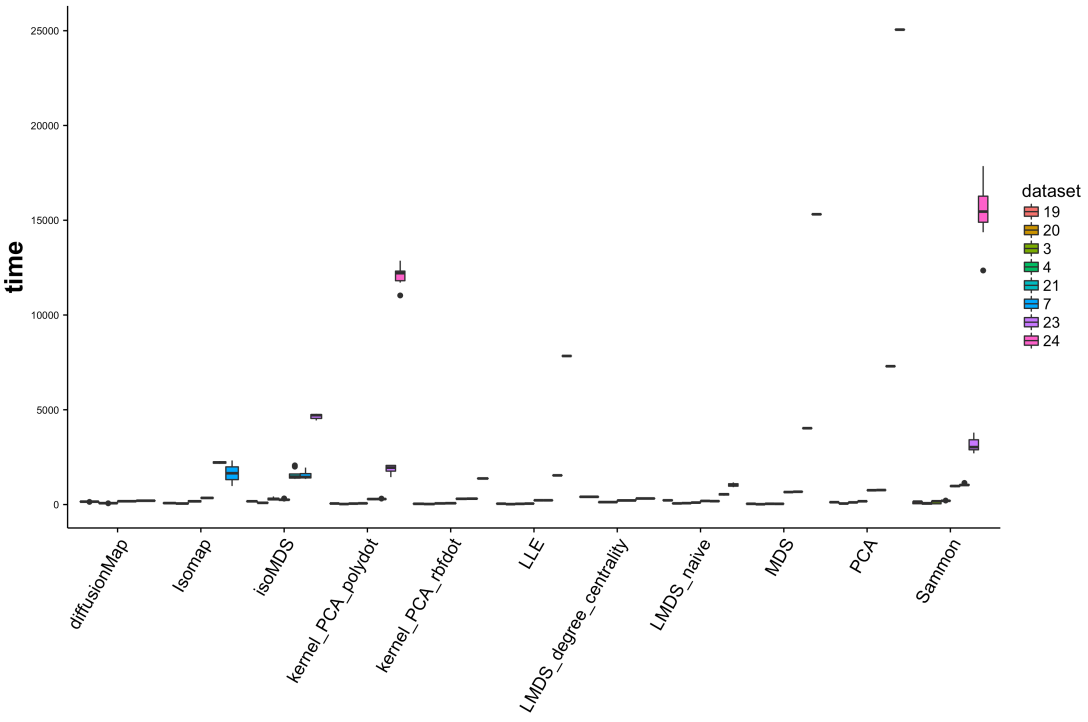

Figure S10: Absolute computing of different dimensionality reduction after parameter optimization on large single cell dataset (>500 cells). The absolute computational time for the best parameter settings of each dimensionality reduction method is shown for each dataset separately. The datasets are ordered from small (dataset 19 = 648 cells) to large (dataset 24 = 70000 cells)

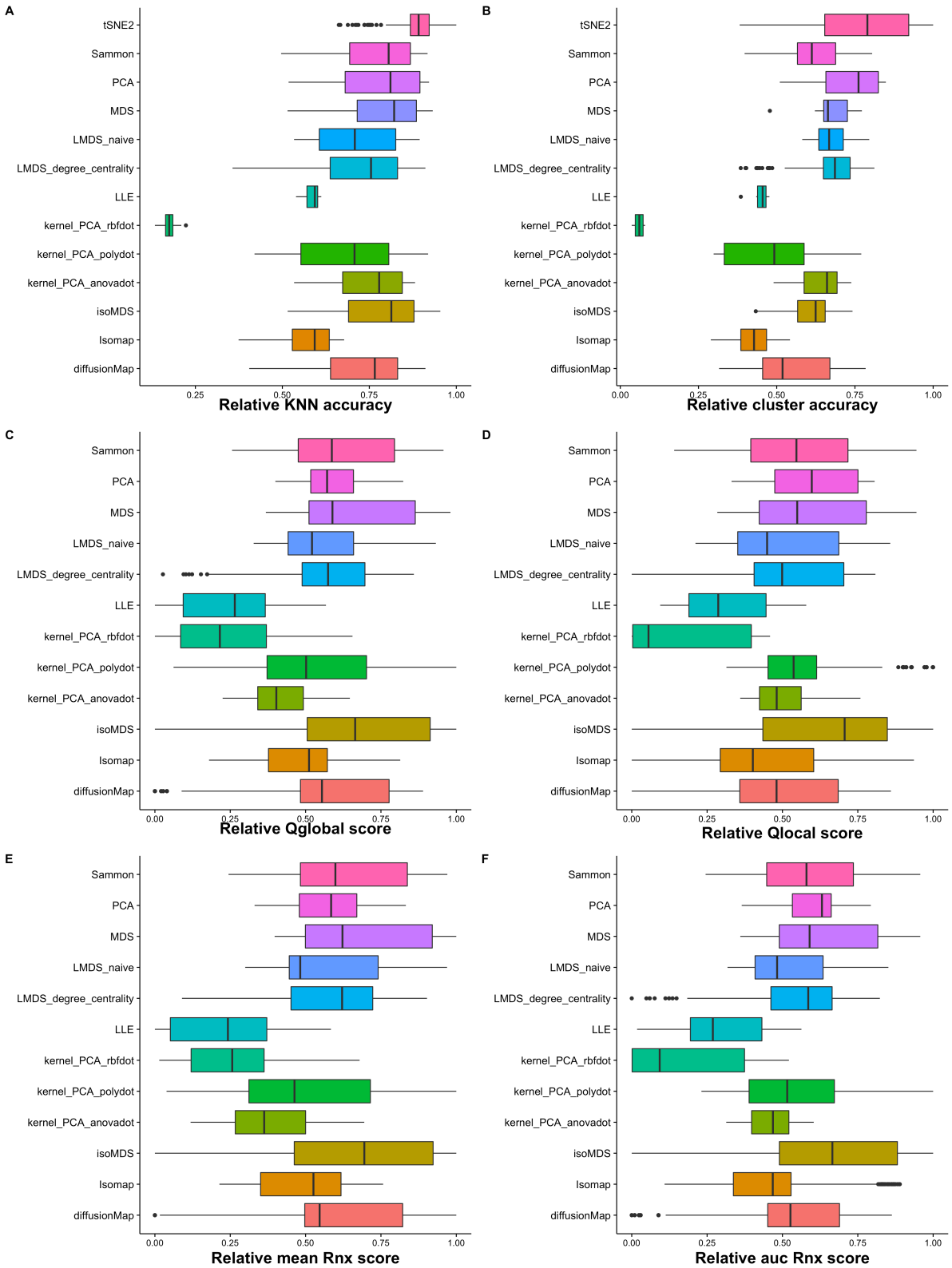

Figure S11: Overall relative performance of different dimensionality reduction methods with varying parameters before parameter tuning according to three quality metrics on large single cell datasets (500 – 70000 cells). At front the scores are normalized at a scale of 0 to 1. The parameters of each method are varied according to a grid search. For each method the average score over all datasets of each parameter setting is represented. **A.** Relative KNN accuracy **B**. relative cluster accuracy **C-D** relative coRanking metric **C.** Relative Qglobal **D**. Relative Qlocal **E**. Relative mean Rnx **F.** Relative auc Rnx

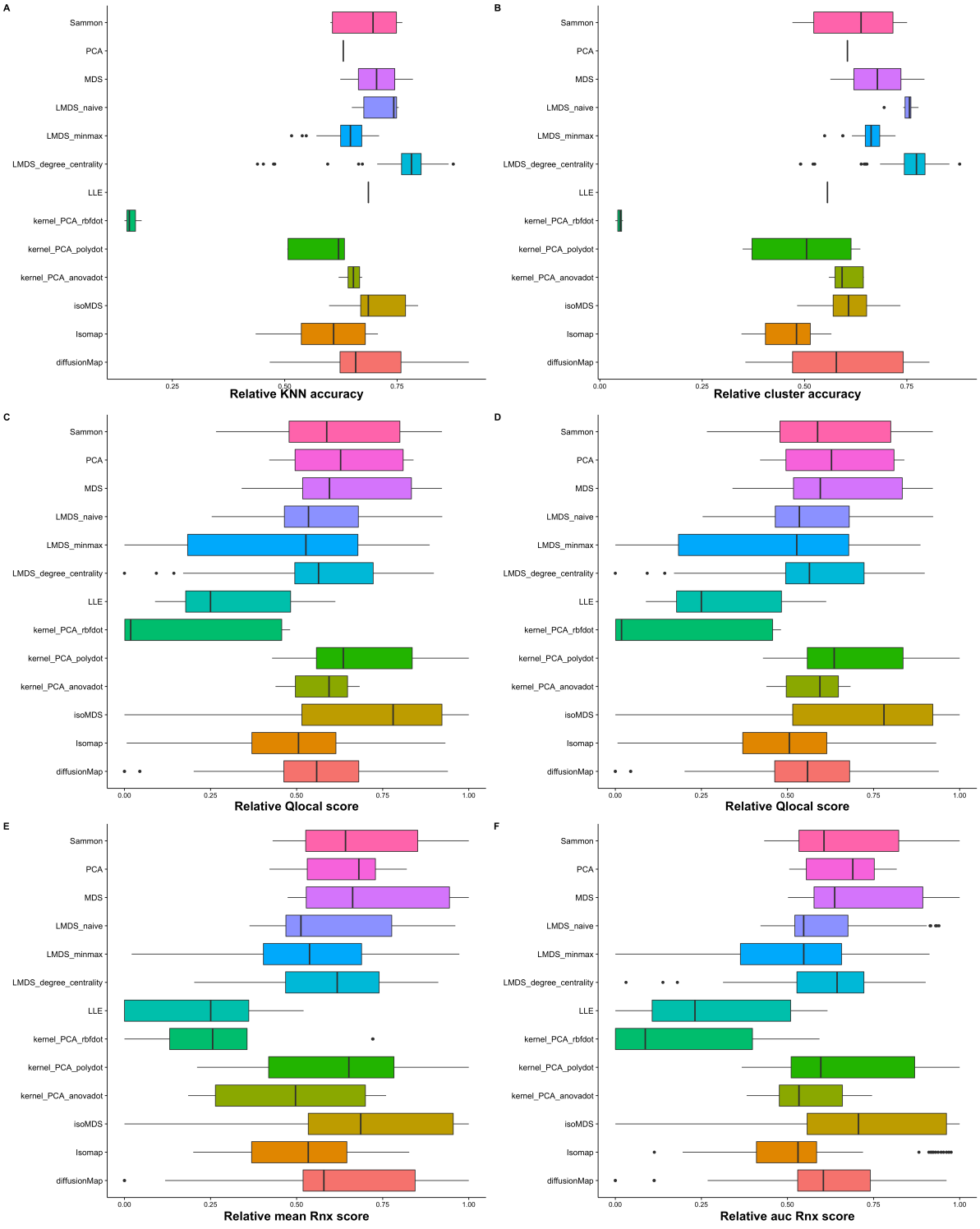

Figure S12: Relative performance of different dimensionality reduction methods into a two-dimensional space before parameter tuning based on three quality metrics for large single cell datasets (648 – 69000 cells). The dimension of the low dimensional space on which the datapoints are embedded is fixed on two (k = 2)**.** The parameters of each method are varied according to a grid search. For each method the average score over all datasets of each parameter setting is represented. **A.** Relative KNN accuracy **B**. relative cluster accuracy **C-D** relative coRanking metric **C.** Qglobal **D**. Qlocal **E**. mean Rnx **F.** auc Rnx

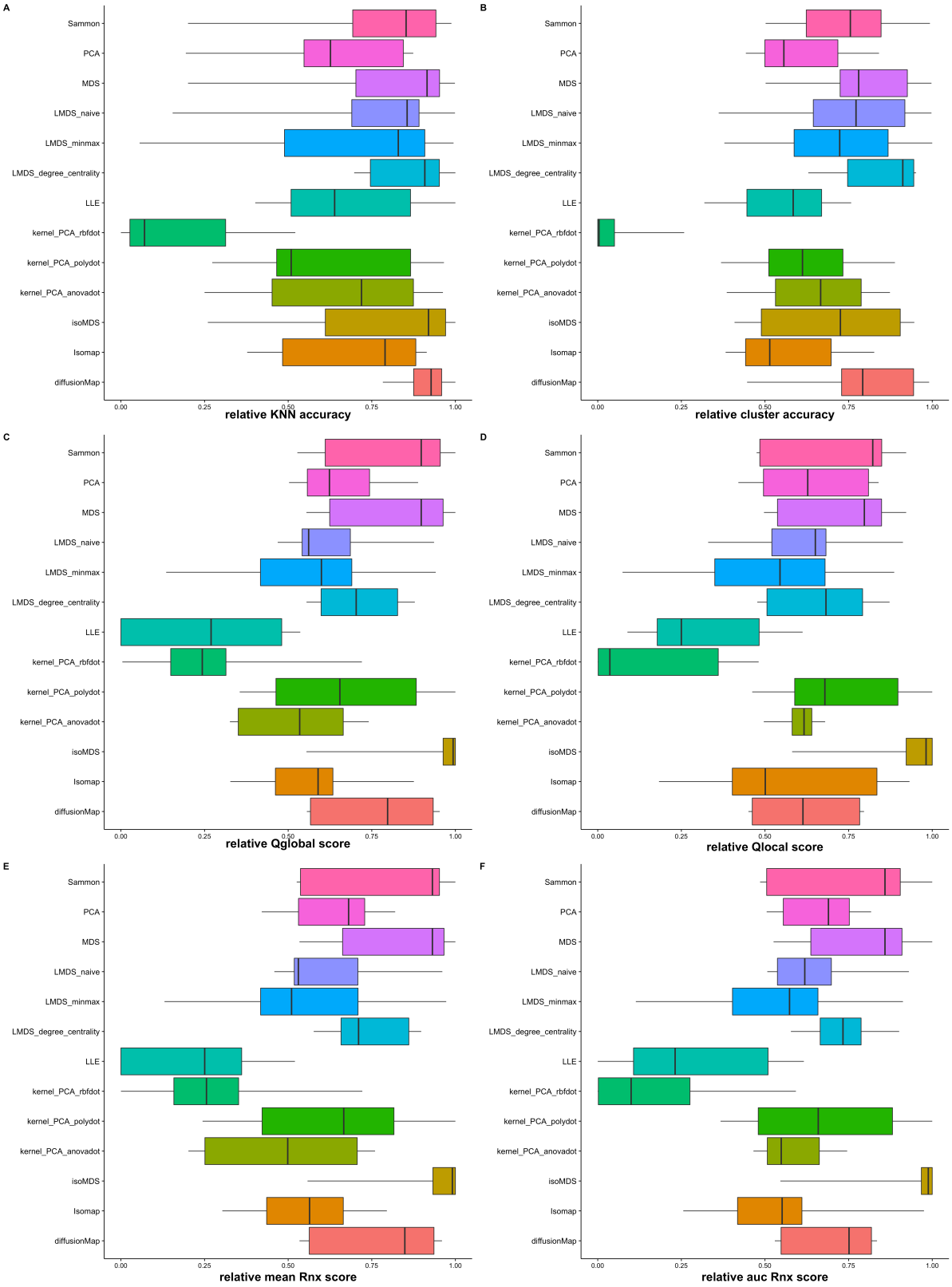

Figure S13: Relative performance of different dimensionality reduction methods into a two-dimensional space based on three quality scores after parameter tuning on small single cell datasets (<2000 cells). At front the scores are normalized at a scale of 0 to 1. Parameter k presenting the number of dimensions in the low-dimensional space is fixed on two. For each method the score of the best parameter setting of every dataset is represented. **A.** Relative KNN accuracy **B.** relative Cluster accuracy. **C.** Relative Qglobal **D.** Relative Qlocal **E.** Relative mean Rnx **F.** Relative auc Rnx

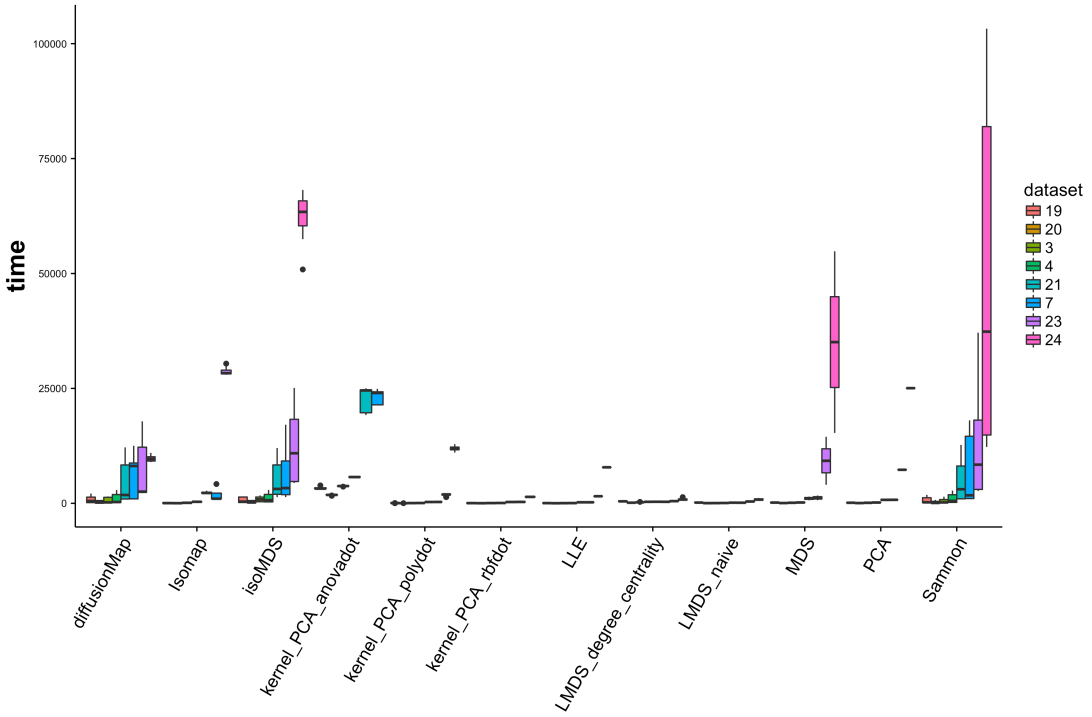

Figure S14: Absolute computing time of a dimensionality reduction into a two-dimensional space for a set of dimensionality reduction methods before parameter tuning on large single cell datasets (648 cells - 67000 cells). The absolute computing for every parameter setting with parameter k fixed on two is shown for every dataset separately for a set of different dimensionality reduction methods. The datasets are ordered from small (dataset 19: 648 cells) to large (dataset 24: 69991 cells).

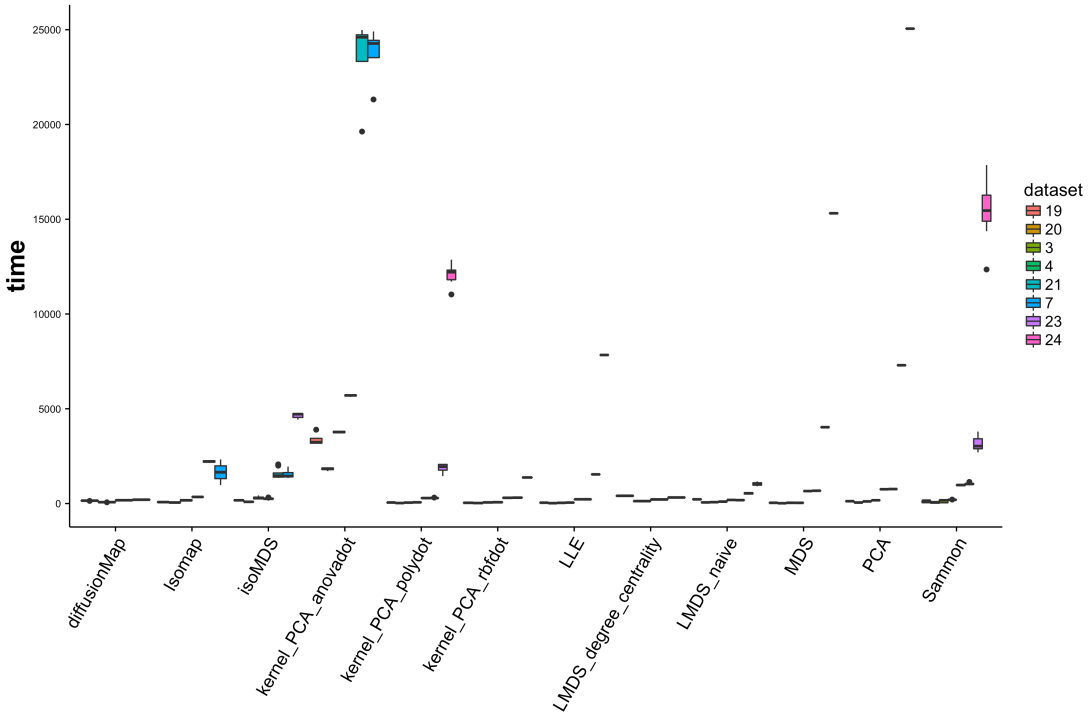

Figure S15: Absolute computing time of a dimensionality reduction into a two-dimensional space for a set of dimensionality reduction methods after parameter optimization on small single cell datasets (<2000 cells). At front the best parameter settings were selected with parameter grid search and LOOCV. The absolute computing for the best parameter setting with parameter k fixed on two is shown for every dataset separately for a set of different dimensionality reduction methods. The datasets are ordered from small (dataset 19: 6648cells) to large (dataset 24: 69991 cells).