DOCTORAL SCHOOL
UNIVERSITY OF MILANO-BICOCCA

Department of Informatics, Systems and Communication

PhD program in Computer Science

Cycle XXXV

Computational strategies for single-cell multi-omics data analysis and integration

Lucrezia PATRUNO

Registration number: 795291

Supervisor: Dr. Alex GRAUDENZI

Co-supervisor: Prof. Giulio CARAVAGNA

Co-supervisor: Prof. Marco ANTONIOTTI

Tutor: Prof. Enza MESSINA

PhD Program Director: Prof. Leonardo MARIANI

ACADEMIC YEAR 2021/22

Abstract

The ever-increasing availability of data generated from sequencing experiments of biological samples brings the need for efficient, scalable and reproducible algorithmic strategies for their analysis and integration. This is particularly true in cancer research, in which large amounts of high-dimensional data at the single-cell resolution can be now generated from patient biopsies and patient-derived models.

In this work, I will present the achievements obtained in three main areas, namely: (A) the development of methods for the analyses of single omics data types (DNA, RNA and ATAC), (B) the design of strategies for their integration, (C) the implementation of a reproducible, scalable and flexible pipeline for the comprehensive analysis of single-cell data, which includes both the new methods developed in tasks (A) and (B), and additional state of the art techniques. All these tasks were carried out within the Bioinformatics programme of the "Single-cell Cancer Evolution in the Clinic"(SCCEiC) CRUK/AIRC Accelerator project, which aims at integrating the efforts of wet- and dry-lab scientists to deliver a fine characterization of cancer evolution, with expected repercussions in clinical settings. However, the achievements of this work will have a more general impact, especially on the broad field of computational science and, in particular, in that of cancer data science and biomedical Artificial Intelligence.

In regard to task (A), I will show the most extensive to-date benchmarking of denosing and imputation methods for single-cell RNA-sequencing data, which might guide researchers both in the application of existing methods to real-world problems, and in the design of new algorithmic strategies. I will then present a new algorithmic framework applied to the inference of clonal trees from single-cell mutational profiles, which provides both the first method to characterize and visualise the solution space explored during MCMC search, and a new method to reconstruct a consensus optimal tree summarising the explored solutions.

In regard to task (B), I will illustrate two methods for the diagonal integration of multimodal data, which aim at integrating DNA-RNA and DNA-RNA-ATAC, respectively. Both frameworks exploit a sound Bayesian framework and learn the parameters through stochastic variational inference, and have the goal of mapping multiple omics on the latent space of genomic alterations. Not only the performance of both methods proved robust on simulations, but the application to real-world datasets showed the

effectiveness in producing usable knowledge with translational relevance.

Last, in task (C) the efforts were directed toward the definition of a comprehensive pipeline, with the general goal of enhancing the reproducibility and standardization of data analysis workflow. The resulting pipeline includes multiple building blocks tailored to the distinct omic data types, and - in its current form - includes either state-of-the-art methods or techniques developed during tasks (A) and (B).

All in all, starting from the specific questions of the the SCCEiC project, the achievements of this work produced theoretical frameworks and tools that proved effective in extracting knowledge from complex experimental settings and in generating of data-driven experimental hypotheses, confirming the current necessity of multi-disciplinary efforts in real-world scenarios.

Contents

List of Abbreviations

ASV	Amplicon Sequence Variant.
BAF	B-Allelic Frequency.
CMS	Consensus Molecular Subtype.
CNA	Copy Number Alteration.
COB-tree	Consensus Optimum Branching Tree.
DEG	Differentially Expressed Gene.
DNN	Deep Neural Network.
k-NN	k-Nearest Neighbor.
LFC	Log-Fold Change.
MCMC	Markov chain Monte Carlo.
ML	Machine Learning.
NB	Negative Binomial.
NGS	Next Generation Sequencing.
PCA	Principal Component Analysis.
PDO	Patient Derived Organoid.
scATAC-seq	single-cell ATAC sequencing.
scDNA-seq	single-cell DNA sequencing.
scRNA-seq	single-cell RNA sequencing.
SNP	Single Nucleotide Polymorphism.
SNV	Single Nucleotide Variation.
SOTA	State-Of-The-Art.
SV	Structural Variant.
SVD	Singular Value Decomposition.
SVI	Stochastic Variational Inference.
ZINB	Zero-Inflated Negative Binomial.

1 Introduction

Computer science has for a long time provided fundamental tools at the service of biological analysis. In fact, biological sciences have the goal of deriving explanations about real-world observations and build models to make predictions [87]. Over the last decade we have experienced a refinement of sequencing technologies, with an increase in both the volume and the heterogeneity of the data. Indeed, a decreasing cost in sequencing results in the increase of the volume and dimensionality of data, and the development of new technologies results in more data types and in the generation of multimodal measurements that extract multiple features from the same samples. At the same time, technical limitations in sequencing technologies generate noisy measurements, and may introduce bias in the data. For this reason, biological data analysis is tightly connected to Machine Learning and statistical inference. In fact, there is a growing need for sound computational approaches that enable to extract meaningful knowledge from the data and to create reproducible pipelines for the analysis and interpretation of biological data. By applying robust methods that model noise in the observations and enable the construction of comprehensive workflows, it is possible to investigate multiple biological questions and formulate new experimental hypotheses in an automated fashion (Figure 1.1).

The combination of advancements in both computer science and the design of new sequencing technologies has a direct application in cancer research. In fact, cancer is a complex evolutionary process that, starting from a single cell that acquires alterations of its (epi)genetic makeup, evolves into different heterogeneous populations that coexist,

evolve and compete over time [2] exhibiting different functional properties often referred to as a hallmarks [191]. Thus, measuring the genomic and epigenetic traits of cancer cells is fundamental to start unravelling such complex interplay. However, this is not sufficient to fully explain how malignant cell populations evolve and transform, as there are countless factors that determine cell behaviour and cell-to-cell interactions that need to be studied [169]. All these aspects are responsible for the high degree of heterogeneity found in tumors, both between tumors of different patients and within the same tumor in a single patient. The latter phenomena is the so-called Intra Tumor Heterogeneity (ITH), and is the main cause of disease relapse and of the acquisition of drug resistance [32]. In order to shed a light into such an heterogeneous system taking into consideration multiple aspects determining the behaviour of tumor cells, it is then fundamental to study it from different points of view, performing multiple experiments and extracting multiple features from the data.

1.1 Motivation and computational challenges

The advancements in sequencing technologies, coupled with the complexity of the data generated, brings the need for the development of sound computational approaches to extract knowledge and integrate different data types.

In figure Figure 1.2 we show a cartoon reporting the heterogeneity and complexity of an experimental setting involving multiple data types: the fundamental unit is a single patient affected by a tumor. From each patient, multiple samples can be collected through biopsies, that can be used to grow replicas of the corresponding tumor called Patient Derived Organoids (PDOs) [105, 103]. There can be multiple PDOs for each patient, that may correspond to different sections of the same tumor (e.g., primary tumor and metastatic sites [61]), and each of them can be treated with multiple drugs in different time points (longitudinal experiments), to assess the outcome of the therapy on the corresponding tumor [103]. Then, sequencing technologies are exploited to extract different data types from each sample, and we refer to these data types as *omics* [131]. The omics involved in this work are DNA [78, 34], RNA [80] and ATAC [45, 46], and they can be defined as multiple layers that measure different features from the data [131]: RNA-sequencing measures gene expression levels, DNA-sequencing enables to detect variations in the DNA sequence and ATAC-sequencing measures the amount of open chromatin detected in each genomic region. These measurements together provide a comprehensive description of the cancer system under different perspectives.

The experiments to extract these omics layers can be performed on multiple batches, which can be identical replicates used to measure the same information on the same sample multiple times [18]. Finally, each experiment can be performed at two resolutions: single-cell or bulk [117]. On the one hand, in the former case the result is a n cells $\times$

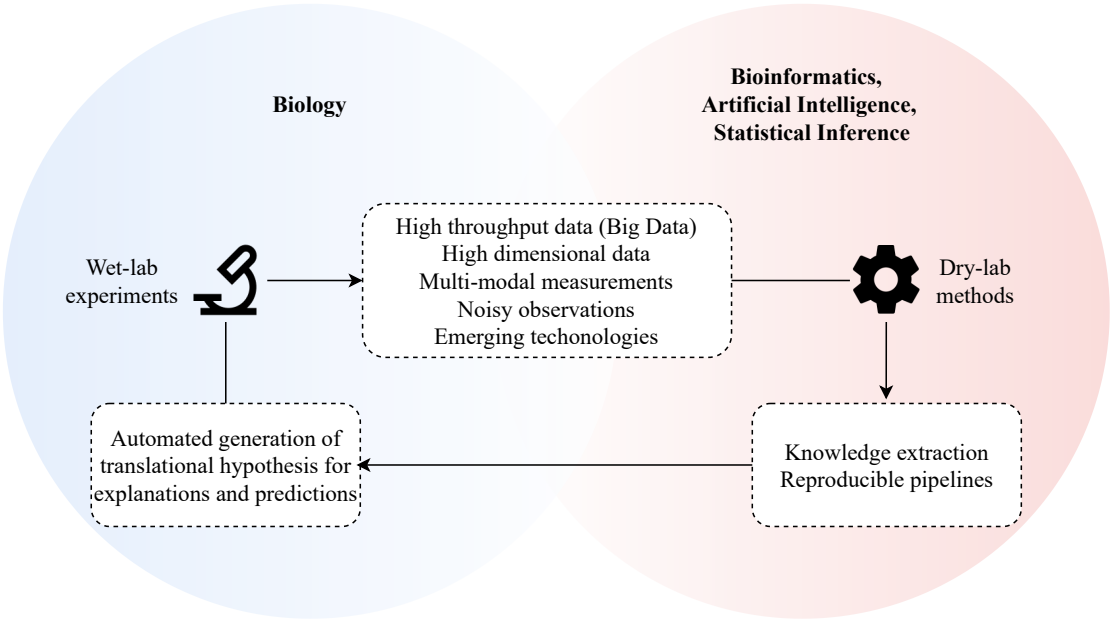


Figure 1.1: Biological and computational science are highly interconnected. In fact, the goal of achieving a deep understanding of different biological phenomena (e.g., mechanisms of tumor evolution), brings the need for the development of new technologies that generate high volume and high dimensional data to describe a biological system. However, such data are highly heterogeneous, they are contain different data types that need to be integrated and they are characterised by high dimensionality and high levels of noise. Thus, it is fundamental to use sound computational strategies to process these data from the fields of bioinformatics, machine learning and statistical inference.

m features matrix containing the signal for each feature measured for each cell in the sample. On the other hand, bulk resolution pools together the signal from an intermixed mixture of cells (the bulk) and provides as output a vector encoding the total signal extracted from each feature in the sample.

Given the multiple combinations of PDOs, drugs, omics layers and timepoints, and the obvious limitation of (experimental and financial) resources in real-world scenarios, sequencing is typically performed only on a subset of the samples, based on the research question under investigation. To give a broad overview of the experimental settings complexity and how different the sequencing experiments can be performed, in Figure 1.2 we use a non-transparent color to indicate the samples subsets that might actually be sequenced.

In order to extract usable and reproducible knowledge from these complicated and typically inhomogeneous experimental settings, computer science and two of its recent declinations, bioinformatics and computational biology, are vital. In particular, we need two classes of computational methods: (A) those that analyse each sample considering one single single omic layer, and (B) those that aim at integrating all the information extracted from distinct layers and samples to gain a snapshot of the system [159].

With respect to (B), three types of data integration tasks can be listed: (i) *horizontal*, where the information extracted from the same omic layer is integrated across multiple samples that contain different cells, (ii) *vertical*, where different omics layers are measured on the same cells and (iii) *diagonal*, where the goal is that of integrating different omics layers that are extracted from different sets of cells [159]. The types of integration are represented in Figure 1.3.

Both in the case of tasks (A) and (B), the application of sound computational approaches to perform data analysis has multiple goals. For example, features extracted from the data can be exploited to build classification models or to perform unsupervised clustering to assess the heterogeneous composition of each sample [155]. It is possible to build models that can describe and explain observed phenomena, for example inferring interaction networks between multiple cellular sub-populations [170, 141, 130], defining trajectories of differentiating cells using diffusion maps [52], or understanding which features are distributed differently across different conditions [49, 97], to detect the ones that are associated for example with therapy resistance. It is also possible to build models for predicting the outcome of a therapy [197] or to predict which set of mutations accumulated during cancer progression might have critical impact in disease progression [139].

The work of the PhD project has been mainly focused on (A) methods for analysing single-cell RNA and DNA datasets, (B) methods to perform diagonal integration of the three omics and (C) merging the newly developed methods into one analysis pipeline, that is composed of multiple building blocks combining both the new algorithmic frame-

works presented in this work and additional analyses performed using state-of-the-art methods. The efforts led to advancements in the fields of data processing, analysis, and integration, as they can be exploited by other researchers in the field of computational biology to perform knowledge extraction and to formulate and investigate experimental hypotheses.

1.2 The Single-Cell Cancer Evolution in the Clinic (SCCEiC) project.

The PhD project was funded by the CRUCK/AIRC Accelerator Award #22790 "Single cell cancer evolution in the clinic" led by Giovannni Tonon MD (Fondazione Centro San Raffaele, Milan, Italy), and Prof. Andrea Sottoriva (Human Technopole, Milan). The multidiscplinary project relies on an intermixed team including cancer biologists, clinicians, engineers and computational scientists: while people from wet-labs generate samples and heterogeneous data from Patient Derived Organoids (PDOs) of colorectal cancer samples in a variety of experimental settings, dry-lab researchers design and apply computational methods to analyse and integrate the data, so to extract usable knowledge that can be exploited to answer biological question, and formulate new experimental hypotheses in a possibly automated fashion.

This project is divided in 4 programs, and I worked in Programme 3: "Bioinformatics analysis, single-cell data integration and evolutionary methods" under the supervision of Dr. Alex Graudenzi (University of Milan-Bicocca), Prof. Giulio Caravagna (University of Trieste) and Prof. Marco Antoniotti (University of Milan Bicocca, PI of the Programme). The program has the overarching goal of delivering the bioinformatics analysis framework to integrate the single-cell genomic, epigenomic and transcriptomic data, which are generated by the institutes responsible for culturing and sequencing tumor cells.

Within the SCCEiC project, specific questions have emerged over the years. Namely, how can computational methods allow to exploit single-cell data generated from patient-derived organoids to explain and predict cancer evolution? Given the multiple layers extracted from each sample, what computational methods do we need to perform an in-depth analysis of each data type, and how can we integrate them to map the information across multiple samples? Finally, how can we merge the different methods into a comprehensive analysis pipeline? Most of the results of the methods and pipelines presented in this work were conceived to address these questions, and were applied on data from PDOs generated within the project, but given the generality and relevance of the topics, our approaches constitute also a contribution to the field of computational sciences.

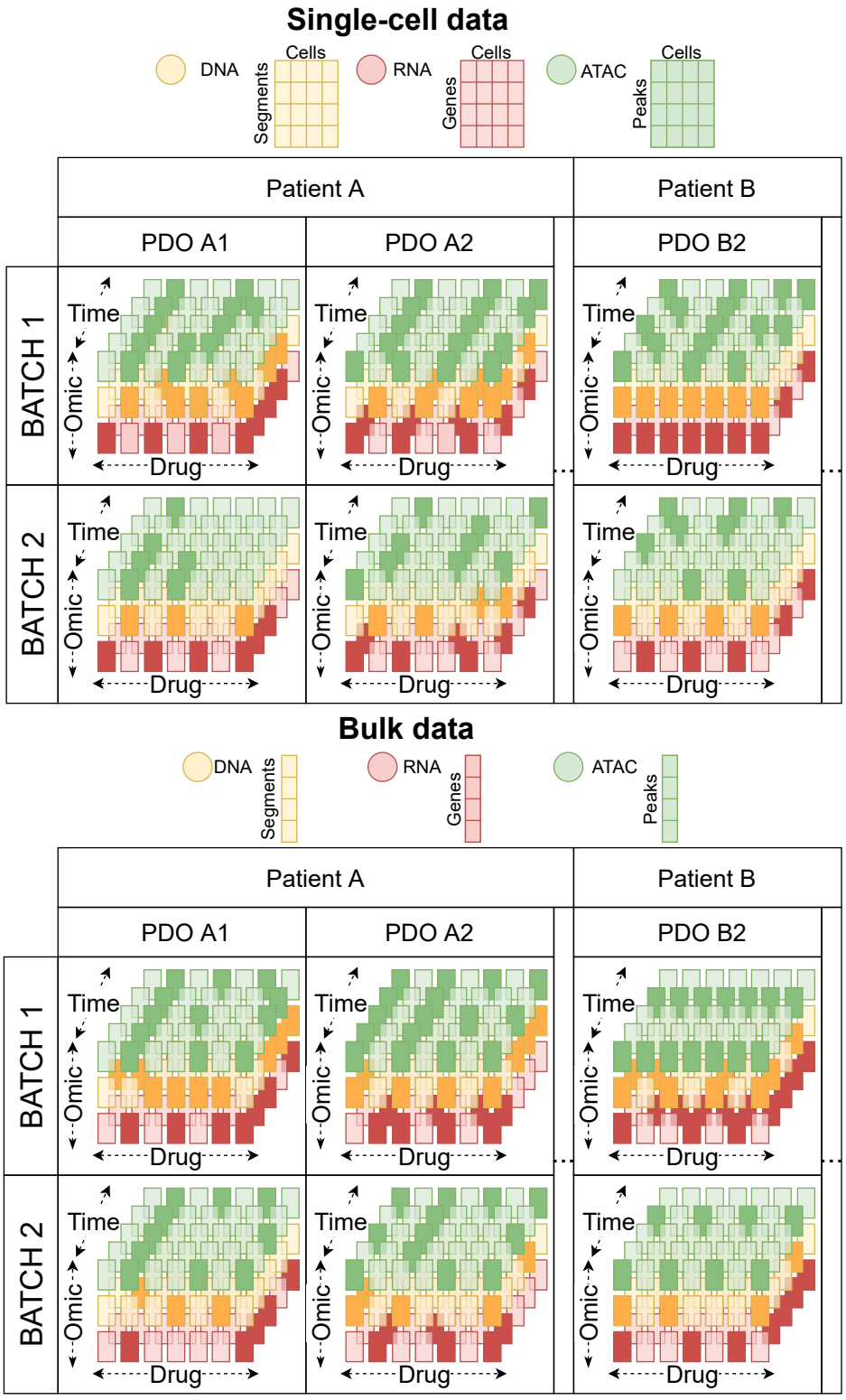


Figure 1.2: Typical complexity of a setting with single-cell sequencing experiments in cancer research. Multiple PDOs can be created from samples extracted from different patients. Each PDO can be treated with different drugs for multiple timepoints, and different sequencing technologies are exploited to extract multiple omic data types from the samples. In each experimental setting, not all combinations are measured, and we use non-transparent shapes to indicate those samples that undergo sequencing and need to be analysed.

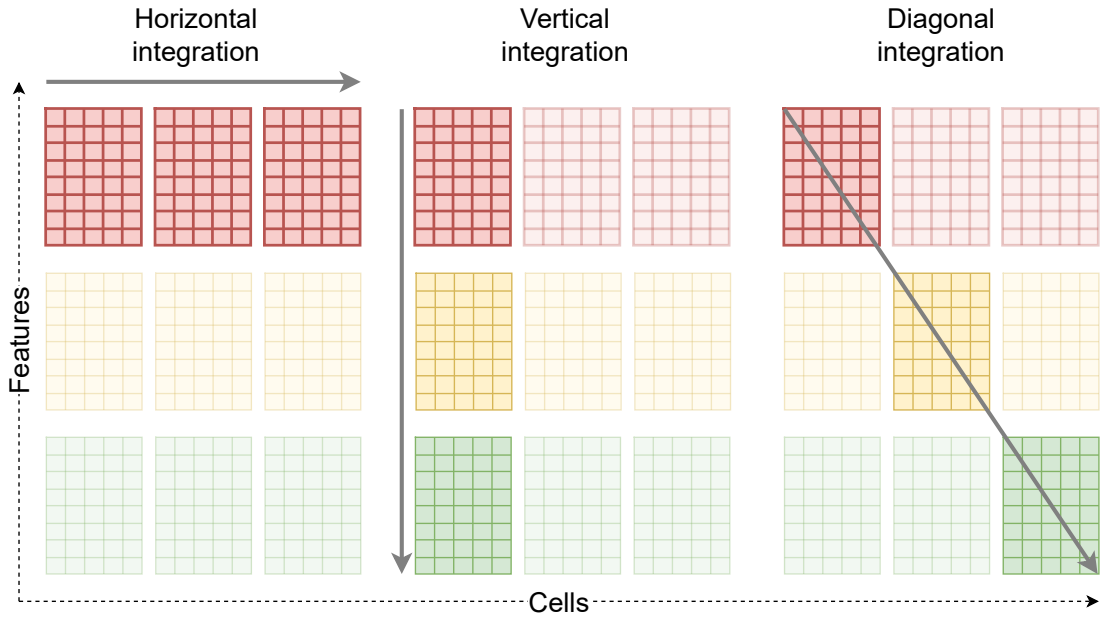


Figure 1.3: Three types of data integration tasks as presented in [159]. Horizontal: the same omic is measured on different samples. Vertical: multiple omics are measured on the same set of cells, yielding multiple signals for each single cell. Diagonal: multiple omics are measured on different sets of cells, and we need a model to perform mapping of the measurements.

1.3 Main achievements

This work is focused on the development and application of computational methods for the analysis (task A) and integration (task B) of single-cell multi omics data, and the design of a pipeline used merge the multiple building blocks to deliver a comprehensive analysis (task C). The main achievements, grouped according to the corresponding task are presented in the following sections.

1.3.1 Task A: Algorithmic methods for single omics data types

Benchmarking of methods to perform denoising and imputation of single-cell RNA-sequencing data. With the growing availability of single-cell RNA sequencing data that contain high levels of noise and missing data, multiple methods have been proposed in the literature to perform denoising or impute missing observations. We present a bechmarking of multiple state-of-the-art methods, were we compared their performance using a number of metrics computed on both synthetic and real datatsets.

- This work is presented in Section 3.1.
- **Collaborators**: Data and Computational Biology Lab (DCB Lab). University of Milano Bicocca, Milan, Italy.
- **Article**: *Patruno, L., *Maspero, D., Craighero, F., Angaroni, F., Antoniotti, M., and Graudenzi, A. *A review of computational strategies for denoising and imputation of single-cell transcriptomic.* **Briefings in Bioinformatics** 22.4 (Oct. 2020). doi:10.1093/bib/bbaa222. *equal contribution.
- **Oral presentation**: June 2020, monthly meeting of the Accelerator Award Project #22790.
- **Code**: `https://github.com/BIMIB-DISCo/review-scRNA-seq-DENOISING`

New algorithm for improving the reconstruction of clonal trees. We present a new algorithm that aims at improving the solution returned by methods for reconstructing clonal trees from single-cell mutational profiles. Our method summarises the solutions explored during inference and returns one consensus tree that summarises such solutions.

- This work is presented in Section 3.2.1.
- **Article**: Maspero, D., Angaroni, F., Patruno, L., Ramazzotti, D., Posada, D., Antoniotti, M., and Graudenzi, A. *Exploring the solution space of cancer evolution inference frameworks for single-cell sequencing data.* Accepted for publication in

the conference proceedings of the 16th International Workshop on Artificial Life and Evolutionary Computation (WIVACE 2022).

- **Oral presentation**: 16th International Workshop on Artificial Life and Evolutionary Computation (WIVACE 2022), 14-16 September 2022, Gaeta, Italy.
- **Collaborators**: DCB Lab, University of Milan-Bicocca, Milan, Italy. Department of Medicine and Surgery, University of Milan-Bicocca, Monza, Italy. Posada Lab, University of Vigo in Galicia, Spain.

1.3.2 Task B: Computational methods for omics data integration

CONGAS We present a method to detect clones defined based on Copy Number Alterations (CNAs), using single-cell RNA sequencing (scRNA-seq) data and bulk DNA sequencing to obtain a reliable estimation of copy number segments.

- This work is presented in Section 4.1.
- **Article**: Milite, S., Bergamin, R., Patruno, L., Calonaci, N., and Caravagna, G. *A Bayesian method to cluster single-cell RNA sequencing data using copy number alterations*. **Bioinformatics** 38.9 (May 2022). doi:10.1093/bioinformatics/btac143.
- **Collaborators**: Cancer Data Science Lab (CDS Lab), University of Trieste, Trieste, Italy. DCB Lab, University of Milan-Bicocca, Milan, Italy.
- **Code**: `https://github.com/caravagnalab/congas` and `https://github.com/caravagnalab/rcongas`.

CONGAS+ We present an extension of CONGAS, namely CONGAS+, that is a method to detect clones defined based on Copy Number Alterations (CNAs), from scRNA-seq data and single-cell ATAC sequencing (scATAC-seq), using bulk DNA sequencing to obtain a reliable estimation of copy number segments and set priors for the inference.

- This work is presented in Section 4.2.
- **Article**: *Patruno, L., *Milite, S., Bergamin, R., Antoniotti, M., Graudenzi, A., and Caravagna, G. *A Bayesian method to detect aneuploidy from single-cell RNA and ATAC sequencing*. In preparation.
- **Collaborators**: CDS Lab, University of Trieste, Trieste, Italy. DCB Lab, University of Milan-Bicocca, Milan, Italy.
- **Code**: `https://github.com/caravagnalab/CONGASp`. Interface currently under development, available at `https://github.com/caravagnalab/rcongas` [branch `categorical`].

1.3.3 Task C: A comprehensive pipeline for reproducible single-cell analyses

The SIgMOIDAL pipeline We present the design of SIgMOIDAL, a pipeline that combines state-of-the-art tools and new methods presented in this work to provide a comprehensive analysis of single-cell datasets. We also present the application of this pipeline to two case studies.

- This work is presented in Chapter 5.
- **Collaborators**: CDS Lab, University of Trieste, Trieste, Italy. DCB Lab, University of Milan-Bicocca, Milan, Italy.
- **Oral presentation**: this work was presented in during the annual External Advisory Board meetings for the Accelerator Award project #22790.
- **Article**: in preparation.
- **Collaborators**: DCB Lab, University of Milan-Bicocca, Milan, Italy. CDS Lab, Trieste, Italy. Sottoriva Lab, Institute of Cancer Research (ICR), London and Human Technopole (HT), Milan, Italy. Gastrointestinal Cancer Biology and Genomics Team, ICR, London. Functional Genomics of Cancer Unit, San Raffaele Hospital, Milan, Italy.
- **Code**: currently developing a `Nextflow` pipeline for preprocessing and an `R` ShinyApp for integration and downstream analyses.

1.3.4 Additional works

During these three years I collaborated on three additional projects in the fields of computational biology and biomedical AI, that are listed in this section and will be presented in the appendix. These projects were carried out with both the DCB Lab at University of Milano Bicocca, and external collaborators.

Implementation of a Machine Learning model to predict flux variations We present a Machine Learning model to predict flux variations in metabolic networks using variations in metabolite abundances as input.

- This work is presented in the Appendix.
- **Collaborators**: DCB Lab, University of Trieste, Trieste, Italy. Department of Biotechnology and Biosciences (BtBs), University of Milan-Bicocca, Milan, Italy.

- **Article**: *Patruno, L., *Craighero, F., Maspero, D., Angaroni, F., Graudenzi, A., and Damiani, C. *Combining multi-target regression deep neural networks and kinetic modeling to predict relative fluxes in reaction systems.* **Information and Computation** 281 (Dec. 2021). doi:10.1016/j.ic.2021.104798. **Code**: `https://github.com/BIMIB-DISCo/FLUX-PREDICT`.

EvoTraceR We present `EvoTraceR`, an `R` package that detects and analyses a set of Amplicon Sequence Variants from the the result of a new CRISPR/Cas9 barcode experimental kit.

- This work is presented in the Appendix.
- **Collaborators**: Nowak Lab, Weill Cornell Medicine, New York, USA. DCB Lab, University of Milan-Bicocca. Department of Medicine and Surgery, University of Milan-Bicocca, Monza, Italy.
- **Article**: manuscript in preparation.
- **Visiting** at Nowak Lab, Weill Cornell Medicine, New York, USA. From Dec. 2021 to May 2022.
- **Code**: `https://github.com/Nowak-Lab/EvoTraceR_pipeline`.

Implementation of a closed-loop optimization framework for personalised cancer therapy The growing availability of clinical data from cancer patients makes it possible to extract features that can be used to build methods for optimized drug protocols. Thus, in this context we present a framework for the optimization of the Imatinib scheduling in Chronic Myeloid Leukemia patients.

- This work is presented in the Appendix.
- **Collaborators**: DCB Lab, University of Milan Italy, Trieste, Italy. Department of Biotechnology and Biosciences (BtBs), University of Milan-Bicocca, Milan, Italy.
- **Article**: *Angaroni, F., *Pennati, M., *Patruno, L., *Maspero, D., Antoniotti, M., and Graudenzi, A. *A closed-loop optimization framework for personalized cancer therapy design.* **2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)** IEEE, (Dec. 2020). doi:10.1109/CIBCB48159.2020.9277647.
- **Code**: `https://github.com/BIMIB-DISCo/closedLoop-CT4TD`.

1.4 Structure of the thesis

Considering the structure presented in the previous sections, task A and B are related to the development of new methods that need then to be combined into a comprehensive analysis framework. This, consistent with this distinction,in this thesis, the work is presented with the following structure: first, we present methods developed for single-omics (task A), then we describe two new methods for data integration (task B) and finally, we present a comprehensive pipeline (task C) that combines both the methods presented above and additional building blocks implemented using state-of-the-art methods, which is presented in Figure 1.4. Notice that color red, yellow and green, respectively will be consistently associated to RNA, DNA and ATAC data types throughout the text.
The structure of this work is the following:

- Chapter 2 contains the background, where we present the biological data that is exploited in this work and the current state-of-the-art approaches to analyse and extract knowledge from the different biological omics-layers.

- Chapter 3 presents methods developed for single omics layers, divided according to the data type employed. In detail, in Section 3.1 is focused on single-cell RNA sequencing (scRNA-seq)data, and we present a benchmarking of methods to perform denoising and imputation of scRNA-seq expression profiles, and Section 3.2.1 is focused on single-cell DNA sequencing (scDNA-seq), and we present a method for improving the reconstruction of clonal trees, that returns one consensus tree that summarises multiple solutions.

- Chapter 4 is focused on methods developed to integrate multi-omics data. In detail, Section 4.1 presents CONGAS, a method that integrates bulk DNA sequencing data and scRNA-seq data to detect copy number events from gene expression profiles. Section 4.2 presents CONGAS+, a method designed to detect Copy Number events integrating single-cell ATAC sequencing (scATAC-seq), scRNA-seq and bulk bulk DNA sequencing data.

- Chapter 5 presents the SIgMOIDAL pipeline to perform analysis and integration of single-cell multi-omics data, and presents its application to two real-world case studies.

- Finally, the Section 6.2 contains three additional works that were carried out over the past three years. We present (i) a work on the development of a Machine Learning method to predict flux variations in metabolic networks, (ii) a package to detect Amplicon Sequence Variants in a CRISPR/Cas9 barcode kit and (iii) a closed-loop optimization framework for personalised cancer therapy design.

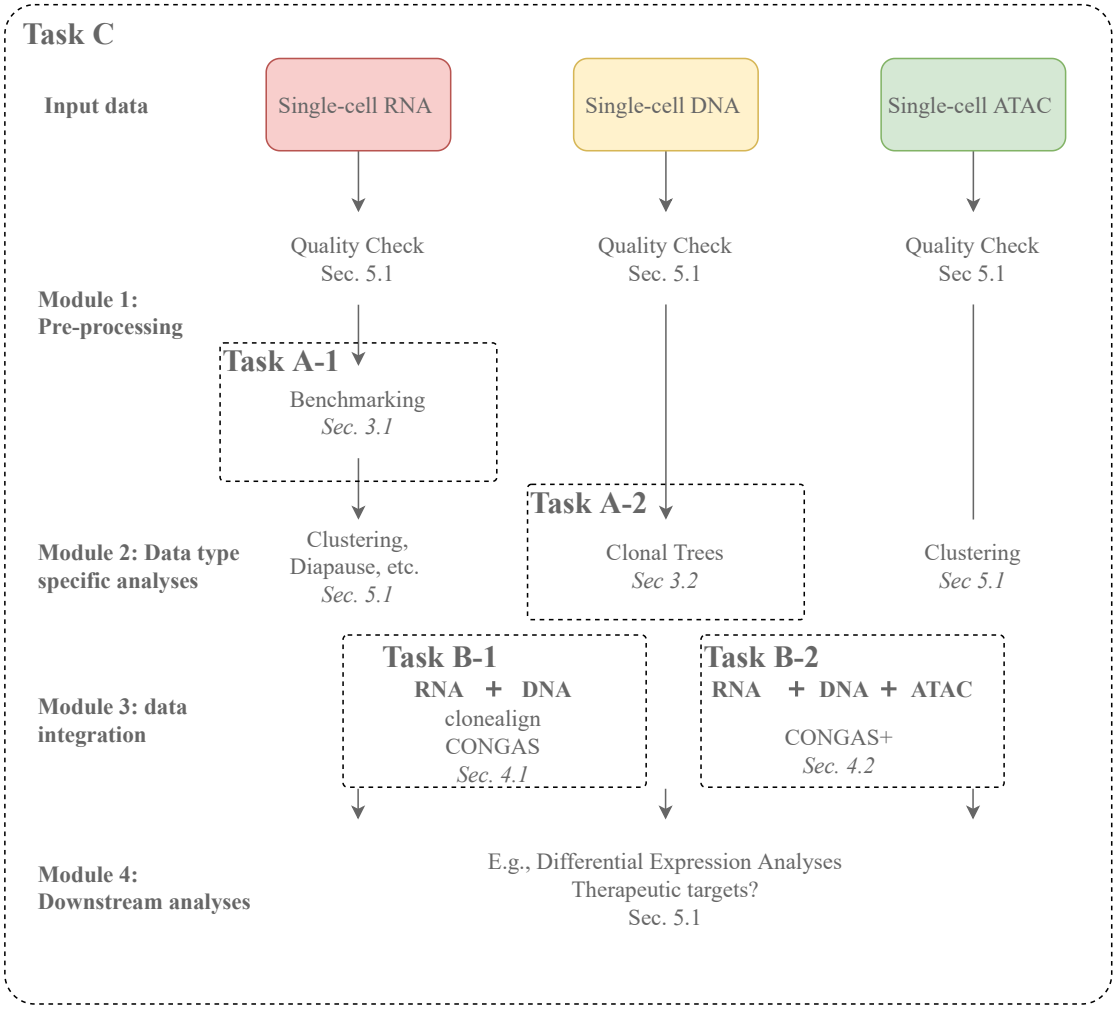


Figure 1.4: Complete overview on the pipeline SIgMOIDAL presented in this work. Three omics layers are exploited: RNA, DNA and ATAC. Each omic needs to be pre-processed to remove low quality observations. Then, scRNA-seq can optionally undergo a denoising step, and then specific analyses are applied to extract knowledge to characterise each sample. DNA sequencing data can be exploited to reconstruct clonal trees that describe tumor evolution. When RNA and DNA sequencing are performed on two independent subsets of cells from the same sample, they can be integrated with both CONGAS or clonealign [111]. Next, in case RNA, DNA and ATAC are sequenced, CONGAS+ can be exploited to perform data integration. Finally, downstream analyses such as differential expression can be applied, with the goal of identifying putative therapeutic targets.

Given that in this thesis I will discuss works that are either part of an article, and therefore a contribution of multiple authors, or the result of projects involving also other researchers, throughout this thesis I will use “we”, instead of “I” to discuss the efforts and contributions.

2 Background

The work presented in this thesis involves multiple data types at the resolution of both bulk and single-cell. Thus, in this Chapter we give an overview about the data involved in this work: we present the data generation process, we highlight differences between bulk and single-cell resolution and we provide the biological background describing the multiple data types. Finally, we give an overview of state-of-the-art methods used to perform analyses and knowledge extraction from each data type.

2.1 Biological background

2.1.1 Next Generation Sequencing data

The growth in volume and resolution of the data has been made possible by the advent of Next Generation Sequencing (NGS) technologies [99], that revolutionised the data generation process, by offering dramatic improvements with respect to the traditional Sanger sequencing technologies [43]. In fact, while the first human genome required over a decade to be sequenced with extremely high costs [30], today it is possible to sequence the genome of an individual in only a few hours, with limited costs. Since the advent of NGS in the beginning of the 2000s, there has been a continuous improvement of costs over the years, that has brought many advantages, such as the increase in sequencing resolution to the single-cell level and the increase in the type of information that can be measured from a sample. In fact, NGS technologies can be applied to measure data from different biological layers, such as genomic, epigenomic (ChIP-seq or ATAC-seq [20,

45]) and transcriptomic (RNA-seq [22]). Such technologies offer a great opportunity to perform in depth investigation of complex phenomena such as cancer, and thus they are extensively employed to extract data from biological samples.

2.1.2 Data generation: samples

Sequencing can be performed over a number of different biological samples, that can correspond to either sections of tissues extracted with a biopsy from a patient, or more sophisticated Patient Derived Models. The latter indicate models employed in the wet lab to grow tumor cells extracted from patients and to exploit the obtained sampled to test multiple drugs and investigate complex experimental hypotheses. There are three types of PDMs: Patient Derived Organoids, Patient Derived Xenografts and Patient Derived Cells. A review of such models is outside the scope of this work, and readers can refer to [192] for a thorough discussion.

2.1.3 Data resolution: bulk vs single-cell

Sequencing experiments can be carried out at two resolutions: bulk and single-cell. On the one hand, the former takes a set of cells and pools together their information, producing a signal which indicates the average behaviour of cells. On the other hand, the latter isolates each single cell, giving as output a specific signal for every cell. Thus, single-cell technologies can be exploited to study heterogeneous populations, enabling the identification of small sub-populations whose signal would be hidden in the output of a bulk experiment. However, when performing an experiment to analyse a biological sample such as a tissue, there is a trade-off between bulk and single-cell that needs to be taken into account. The former are older and thus they have been employed for a longer period of time and they are more extensively used, also due to the lower cost needed for a round of sequencing. Since they take a tissue and pool together all the cells, the amount of material that gets sequenced and is used to compute the output signal is greater than that employed by singe-cell technologies, where the starting material consists of the content of each isolated single-cell. Thus, bulk experiments are characterized by lower amounts of noise and their output is more reliable with respect to single-cell measurements where, due to technical problems, the output signal is affected by higher levels of noise and missing observations. However, if one is interested in studying the heterogeneity of a biological sample, single-cell technologies provide a signal which has higher resolution. Such experiments are relatively new and are evolving very fast, in fact while in 2011 it was possible to analyse around 100 cells per experiment [26], it is now possible to isolate up to 1 million cells. In addition to increasing the amount of sequenced cells, new protocols are currently being developed to increase the type of information we can extract from cells: while until approximately 2018, most of the efforts

in single-cell sequencing had been devoted to RNA-sequencing for the analysis of gene expression [112], in 2018 a new protocol that enabled scaling open chromatin sequencing (ATAC-seq) to thousands of cells was published [88], and in 2021 10x Genomics released a new technique that enables measuring both gene expression and chromatin accessibility signals from the same single cells [160]. The next paragraph will describe the most common data types we can measure, both with bulk and with single-cell technologies.

2.1.4 Multi-omics data types

The heterogeneous data that can be measured from a sample are known as omics-layers, and correspond to the multiple factors that determine the behaviour of a set of cells. In brief, the relationship between these layers can be described through the central dogma of molecular biology, which states that the information flows from DNA (genome) to RNA (transcriptome) to protein [1]. The DNA is a sequence of nucleic acids, and it can be thought as an ensemble of genes, that are portions of the genome that encode for specific products of the cell. In a healthy organism, the same sequence of DNA is present in each cell, which constitutes the genotype of the organism. The specific behaviour of different cell-types is then determined by additional regulatory elements that govern the synthesis of proteins. We can describe the protein synthesis as a process that, starting from a portion of the DNA corresponding to a specific gene, translates it into mRNA molecules (transcriptome) that are then used to produce the corresponding protein. There is thus a hierarchical structure, which is described in detail in [12], and in the next paragraphs we will give a simplified explanation of the omics-layers that play a role in this process, to provide an overview of the heterogeneous type of information that can be extracted from a sample.

2.1.4.1 Data type I: DNA-sequencing data

The genome can be represented as a sequence of 6 billion characters $\{x_i\}$, where $i = 1, \dots, 6^9$, over an alphabet $\Sigma = \{A, T, G, C\}$. This sequence is grouped into 23 pairs of chromosomes (for a total of 46 chromosomes) that constitute the genotype of an individual. In 2001 the Human Genome Project produced a draft of the human DNA reference sequence [11], which has been improved over the years [74, 195].

Point mutations Given the reference sequence, every individual is characterized by a set of mutations that affect single positions of the DNA sequence. These alterations are known as single-nucleotide polymorphism (SNPs) or germline variants and they are found in the genomic sequence of all cells. We recall that the human species has 23 chromosomes, and the genomic sequence in each cell has two copies for each of these chromosome. Thus, each SNP can either be homozygous or heterozygous. The former

refers to the case where both copies of the chromosome are characterised by an SNP with respect to the reference, while the latter refers to a state in which one copy has the same nucleotide as the reference sequence, and the second copy presents an SNP.
On top of this variations, tumor cell populations are characterized by additional single-nucleotide mutations that accumulate during disease progression, known as Single Nucleotide Variation (SNV). These mutations are somatic, as opposed to the germline SNPs, they are not inherited and emerge after conception, due to random errors or mutational processes.
Both SNPs and SNVs are the simplest type of alteration, as they affect only one nucleotide in the genomic sequence. There are also more complex types of alterations, such as insertions and deletions of a short sequence of nucleotides, whose length ranges between 1 to 10 000 bp [24] (also known as *indels*), and more complex Structural Variant (SV), that correspond to complex rearrangements in the genomic sequence, such as inversions and translocation of genomic regions, as well as deletions and duplications. The latter two are also defined as Copy Number Alterations (CNAs), and will be discussed further in the next paragraph [47].

Cancer can be seen as an evolutionary process that starts from one cell harboring a mutation and then grows into different subpopulations [56] showing different patterns of mutations. A subset of the mutations accumulated during disease progression confer selective advantages to cells, and thus in order to disentangle cancer complexity it is of great interest to understand which mutations accumulated in an individual.

Copy Number Alterations In addition to SNVs, another source of variation in cancer cells are Copy Number Alterations (CNAs): while healthy cells have two copies of each chromosome (that we can call alleles), cancer cells may be affected by gains or losses of one or more copies of specific DNA segments. These CNAs are known to play a key role in disease progression, as they are responsible for the activation of genes that promote tumor progression and the suppression of genes that would inhibit the growth of cancer cells [33]. Since these alterations affect a large portion of the genome and vary across different cancer types [33, 168], it is of great importance to analyse their presence, and to understand whether there are sub-populations of the same tumor sample that are characterized by different CNA events.

DNA sequencing technologies Two technologies that enable measuring the DNA sequence of cells found in a biological samples and enable the study of SNPs, SNVs and CNAs are Whole Genome Sequencing (WGS) and Whole Exome Sequencing (WES). While the former gives information about the full genomic sequence, the latter sequences only the exons, that are portions of the genome that are actually used to produce proteins. The most widely used methods measure the genome at the bulk level, and more

recently DNA sequencing has been extended to the single-cell level [163, 78, 119]. A detailed overview of single-cell methods is presented in [59].

2.1.4.2 Data type II: ATAC-sequencing data

The DNA in each cell is packaged in a structure called chromatin, whose organization determines which genes are active in each cell and can be transcribed. In fact, each portion of the chromatin can be either highly condensed (heterochromatin) or more accessible (euchromatin) [12]: genes found in euchromatin are accessible to transcriptional machinery and thus are active in the cell, while biologically inactive genes are isolated in heterochromatin [45]. This presence of both open and compact chromatin determines a pattern of heterochromatin and euchromatin, which is also heritable as it is maintained after cell division [12] and has a direct impact on gene expression. Thus, when studying heterogeneous populations of cells and complex diseases like cancer, it is of key importance to study which are the biologically active portions of the DNA. ATAC sequencing is a technique for measuring signal in the euchromatin, in 2015 [45] presented a method to perform ATAC sequencing at the bulk resolution, and then a new version was presented to perform the same assay at the single-cell level [46].
The output of a sequencing experiment are FASTQ files containing reads, i.e., fragments corresponding to the genomic sequence of open chromatin that were read during sequencing. Thus, higher reads that map to a specific genomic region correspond to more open chromatin detected. The obtained reads are aligned to a reference genome, in order to map each of them to their corresponding coordinates, and the final output is an integer count matrix $cells \times regions$, (or a vector in vase of a bulk experiment) where the counts in the matrix correspond to the amount of open chromatin observed in that region. Based on the pipeline used to process the output of a scATAC-seq experiment, features can correspond to either peaks [73], that are accessible regions of the genome [156], or fixed length genomic bins [164].

2.1.4.3 Data type III: RNA-sequencing data

So far we have described the DNA as an ensemble of genes that can be transcribed when they are located in the open portion of the chromatin. When the chromatin is open, the next step in the hierarchy is the actual production of the proteins that are active in the cell and thus determine its behaviour. This process can be divided into two main steps: (i) transcription, where the gene sequence is copied from the DNA into a so-called mRNA transcript and (ii) translation of the transcript into the corresponding protein (translation). The quantification of each mRNA transcript reflects how much the corresponding gene is active in the population of cells under study, and thus it plays an important role in the study of heterogeneous populations and a large branch of research

is devoted to studying differences between different cell types in term of gene expression profiles [68]. Bulk RNA-sequencing, that is the assay for quantifying the presence of each transcript, has been widely and extensively used [93]. The first single-cell counterpart was published in 2009 [21], when the cell isolation needed to be performed by hand, it scaled from hundreds of cells in 2011 [26] and since then the power of sequencing has increased to more than thousands of cells in each experiment [80].
The output of an RNA-seq experiments are FATSQ files containing reads, that are fragments of mRNA transcript read through sequencing. Once reads are mapped to their corresponding genomic coordinates, the final output of an RNA sequencing experiments are read count matrices cells $\times$ genes (or a vector in case of a bulk experiment), where a higher number of reads corresponds to higher gene expression levels. In case of single-cell sequencing, the output matrices are high dimensional, with a number of feature of the order of 10^4. However such matrices are highly sparse due to technical sequencing errors, and a discussion about noise sources is presented in Section 3.1.

2.2 Computational background

2.2.1 Methods for DNA-sequencing data

Genomic mutations for tumor evolution In order to reconstruct evolution models describing the history of accumulation of SNVs during disease progression, it is necessary to first identify which clones (genetically distinct cancer cells subpopulations) are present in a tumor sample. There are multiple methods in the state of the art that use bulk DNA data to estimate the frequency and genotype (set of mutations that characterize each clone) of the different clones. In fact, from such data it is possible to extract the frequency of each mutation in the sample, also known Variant Allele Frequency (VAF), and there are different methods that use this signal to identify clones [41, 138]. Once clones are detected, it is possible to reconstruct the tree that describes the evolutionary history of the tumor [138], inferring ancestral relationships between clones. Methods that reconstruct the evolutionary history use a binary matrix as input, where rows correspond to clones and columns represent mutations, and a 1 indicates the presence of the mutation in the corresponding clone, and they reconstruct trees that describe the order of accumulation of mutations.

In addition to using bulk data, given the advancements in single-cell sequencing, there are different computational strategies in the state-of-the-art that leverage single-cell data for reconstructing models of cancer evolution. In principle, single-cell DNA sequencing is a powerful technology that allows to correctly assign mutations to each single-cell. However, there are still many technological limitations that introduce high rates of false positives, false negatives and missing values in the observed data [59]. For this reason, there are methods in the literature that are designed to model noise in

single-cell data to derive robust and consistent models of cancer evolution [63, 123, 161, 196]. For instance, LACE [196] aims at finding the maximum likelihood longitudinal evolution model using single-cell mutation profiles.

Copy Number Alterations The second type of DNA alteration that can be measured from DNA sequencing experiments are CNAs. We can define three types of CNAs: *amplifications*, where a portion of a chromosome is duplicated one or more times, yielding to a copy number value of 3 or higher, *loss of heterozygosity* (LOH), where a portion of a chromosome loses one of the two copies and *copy-neutral LOH*, where a chromosome loses one of the two copies, and the copy that is left is affected by an amplification. The third case is referred to as copy-neutral LOH because the affected portion of the genome is still characterized by a diploid copy number state, but it actually underwent two CNAs. In order to measure CNAs from the output of a DNA sequencing experiments, two quantities are used: the depth ratio and the B-Allelic Frequency (BAF). The former is the ratio between the number of reads that map to a given genomic position in the tumor sample and the reads that map to the same position in a normal sample taken from the same patient, which is a proxy for CNAs, as deviations from 1 may be due to CNAs. BAF is calculated in the following way: first, the heterozygous SNPs are identified from a normal sample. Then, for each genomic position corresponding to an SNP, the BAF is computed as the fraction of reads carrying the mutated allele. In fact, if we define as A the nucleotide found on the reference sequence, and B the nucleotide corresponding to the SNP, in normal (diploid) cells, the heterozygous SNPs have a configuration AB since one copy of the corresponding genomic position carries the reference nucleotide *(A)* the second copies carries the SNP (B), and thus by definition BAF $= 1/2 = 0.5$. When a portion of the genome undergoes a CNA event, this reflects on the BAF. For example, a copy number of 3 indicates that one of the two alleles, i.e., either A or B, have been duplicated, and thus two configurations are possible: AAB or BBA. The former configuration has BAF $= 1/3$, while the latter has BAF $= 2/3$. Thus, this quantity is also a proxy for the copy number state.

Finally, the copy number state of each region in the genome can be computed at two resolutions: (i) total copy number and (ii) allele-specific copy number. The former gives the total number of copies without distinguishing between which of the two alleles have undergone the CNA, and the latter gives also the information about which one of the two copies was affected by the CNA, giving the copy number value for each of the two alleles. Thus, a total copy number $= 3$ in a genomic region can correspond to the following allele-specific configurations $(1, 2)$ and $(2, 1)$. Over the years, different methods have been proposed to compute CNAs from tumor samples using the two signals from bulk DNA sequencing data [48, 25]. Different methods have been developed for single-cell data [78, 119], that use the number of reads mapped to each genomic bin to infer

total CNAs. For the purpose of inferring allele-specific copy number values from single-cell data, the only method in the literature is CHISEL [184] using both read counts and BAF signal. A detailed review for CNA detection in single-cell data is presented in [148]. In general, CNA detection methods are used to infer a segmentation of the genome, i.e., the contiguous genomic regions that are affected by the same CNA, and to infer the actual copy number value (either total or allele-specific) of each segment.

2.2.2 Methods for ATAC-sequencing data

Over the last decade, much effort has been dedicated to the advancements of technologies that enable measuring the regions of open chromatin, both at the single-cell and bulk level [45, 88].
With respect to scRNA-seq, where there are multiple copies of mRNA transcripts that are captured and sequenced, single-cell ATAC sequencing (scATAC-seq) measures the open portions of the DNA, which is present in 2 copies in healthy cells. Thus, the starting material is lower and the resulting matrices are highly sparse [112], and different methods have been proposed in the literature to analyse this data and handle the sparsity. For example, some methods binarize the count matrix to reduce bias generated by technical sequencing problems [164], while other methods compute the z-score of each feature [86]. Once the data is processed, downstream analyses such as clustering are applied to detect different subpopulations. As for scRNA-seq data, one of the most extensively applied methods which has been shown to perform better than other clustering approaches like k-means and hierarchical clustering[112], is `Louvain` [127].
A detailed review of computational pipelines for scATAC-seq analysis is presented in [112].

2.2.3 Methods for RNA-sequencing data

The increasing availability of single-cell RNA sequencing data has led to the development of a plethora of computational methods that analyse the gene expression profiles to extract meaningful knowledge from the samples. The need for ad-hoc methods is driven by the high dimensionality of the data and the high levels of noise that are present in the expression profiles. In this regard, the first steps in the analysis of scRNA-seq data are quality control and pre-processing, that are needed to remove low-quality observations that may bias the conclusions driven from single-cell analyses. A detail overview over the best practices for the analyses of samples is presented in [120].
Over the last years different suites have been developed in order to make analyses reproducible and provide a standard for the single-cell community. Two of the most popular toolkits for `R` and `Python` are `Seurat` [167], and `SCANPY` [104] respectively, which can be used for quality control and downstream analyses.

Among the different analyses that can be performed, clustering and Differential Gene Expression enable respectively the detection and characterization of different subpopulations of cells present in one or multiple biological samples, that can correspond for example to a patient derived organoid, a cell culture or a biopsy taken from a tissue in a patient. Two of the most frequently applied methods for clustering of single-cell data are Louvain [14] and Leiden [127], that are community detection algorithms designed for graph embedded data that have have been shown to outperform other clustering methods on single-cell data in terms of scalability, robustness and accuracy [90] and have been extensively applied [115]. Each population identified through clustering can be characterized in terms of genes that exhibit a significant change in their expression levels, with the goal of associating a cell type to each cluster exploiting specific expression signatures [173, 108], or to identify which cellular processes are deregulated when studying a disease. There are multiple computational methods in the state-of-the-art that can be exploited to perform differential expression analysis, which can be divided in three groups: there are standard statistical tests such as the t-test or Wilcoxon test, methods designed for bulk sequencing data that can also be applied to single-cell datasets by computing pseudo-bulk samples and methods designed specifically for single-cell data [100]. It has been shown that methods designed for single-cell data provide poor performance compared to standard tests and bulk methods [100, 180], and for this reason the two latter classes of tools are widely applied on single-cell RNA sequencing datasets and they are implemented in both `Seurat` and `Scanpy`.

In addition to clustering and differential gene expression, new computational methods have been developed to extract a temporal ordering of the differentiation state of the cells [125], and to predict the future state of individual cells by exploiting the ratio of abundance between spliced and unspliced mRNA in each cell [94, 137]. Despite the great advancements achieved in the last years, a high number of challenges are still open and there is the need for new computational strategies to improve the reliability of existing analyses and pave the way for new ones [145].

2.2.4 Methods for data integration

In addition to perform single-omic analysis, in order to obtain a comprehensive description of the system under investigation it is important to integrate the information extracted from multiple omics layers [159]. As described in Section 1.1, and depicted in Figure 1.3, there are two types of data integration tasks involving multiple omics layers. On the one hand, vertical integration consists in exploiting multi-omics technologies such as G&T-seq [54] or GoT , and 10x multiome [160] which extract the genome-transcriptome and transcriptome-chromatin accessibility respectively from the same set of cells. On the other hand, diagonal integration refers to a setting where the multi-

ple omics layers are extracted from distinct subsets of cells, and therefore there is no straightforward method to match the two measurements. Thus, in order to characterize the same population of cells using multiple modalities, there is the need to develop computational methods to integrate the signals extracted from the different omics. Some methods like MAESTRO [155], integrate scRNA and scATAC datasets by first detecting subpopulations independently on the two assays, and then using the detected populations to match clusters of cells, and others aim at reconstructing an integrated latent space [77].

In the context of cancer research, performing data integration is fundamental in order to dissect the heterogeneity of this disease and formulate hypothesis regarding complex mechanisms responsible for drug resistance. In fact, as it is briefly described in Section 2.1.4.1, one of the main goals of performing DNA sequencing on a tumor sample is that of identifying which clones are present in a samples, where for simplicity we define a clone as a set of genetically distinct cells (please refer to Section 3.2.1 for a more detailed discussion on the definition of a clone). However, once genetic clones have been identified from the output of DNA-sequencing, it is not possible to characterize them by means of their gene expression, chromatin accessibility patterns or additional biological information. For this purpose, over the years different computational methods have been proposed with the goal of detecting CNAs from scRNA or scATAC sequencing data [165, 89, 38, 150, 174] and to integrate single-cell DNA with gene expression or chromatin accessibility measurements [183, 111].

3

Task A: Algorithmic methods for single omics data types

In this chapter we describe our contributions for task A, that is the design of computational strategies to analyse single omics data types.
We first present an extensive comparative assessment of multiple state-of-the-art methods for denoising and imputation methods of single-cell RNA sequencing, which enabled us to identify which methods can be applied to improve data quality and the quality of the results derived from downstream analyses.
Second, we introduce our contribution in the field of clonal evolution from single-cell mutation profiles, by proposing a strategy to visualise the solutions explored during a Markov chain Monte Carlo (MCMC) based method for the reconstruction of clonal tres, and by proposing a new algorithm, the COB-tree algorithm, to summarise multiple solutions explored during the search and return one consensus solution.

3.1 [RNA] Benchmarking of denoising methods for scRNA-seq data

3.1.1 Motivation

RNA-sequencing was first introduced in 2008 [93, 23], and has been vastly employed in the study of biological samples. In fact, by measuring the expression level of every gene in the sample, it also enables the identification of differentially expressed genes across

different samples. Over the last decade, this protocol has been refined in order to reduce the minimum amount of input material required for sequencing. This refinement, together with the advancements in microfluidics technologies that enable the automatic isolation of cells, made it possible to increase the RNA-seq resolution to the single-cell level: in 2009, the first single-cell RNA-seq experiment was published, where cells required to be manually isolated. Over the following years, new microfluidics technologies were developed that enabled the automatic isolation of cells, and this allowed scaling to thousands of single-cells sequenced in one run. Both the sensitivity and precision in the quantification have been improved over the last decade [134], and this has allowed for powerful analysis to be performed over biological samples. In fact, scRNA-seq experiments output a matrix, that reports the expression level of each gene in each single cell. This type of data allows to study heterogeneous samples, making it possible for example to characterize different cell-types present in each sample [37, 98], and it is paving the way towards the reconstruction of a Human Cell Atlas (HCA) [72] that contain information such as the function and the biological characteristics of multiple cell types in the human body. Such data play an important role in cancer research, where the detection of deregulated genes in tumor cells has been studied for a long time [7]. For example, by identifying gene expression differences between normal and tumor samples it is possible to detect which are the cellular processes that exhibit an anomalous behaviour and confer cancer cells the ability to proliferate more and expand over time. Exploiting single-cell data in cancer research has provided great insights into multiple aspects, such as tumor heterogeneity [39, 76] and the composition of the tumor microenvironment [85]. Also, through single-cell data, if one is able to separate single-cell expression profiles of cells sensitive to a therapy and cells that exhibit drug resistance, through differential gene expression analysis it is possible to identify putative genes or pathways that are deregulated in resistant cells. Single-cell gene expression is thus a powerful technology, able to provide insights into heterogeneous populations. However, given the low amount of data that is sequenced (mRNA transcripts from one single-cell versus transcripts extracted a pool of thousands of cells in bulk sequencing), they suffer from high levels of technical noise, due to problems during sequencing, which will be discussed further in the next paragraphs.

Data generation There are two main categories in which Single-cell RNA sequencing protocols can be divided: (i) full-length (e.g., Smarter and Smartseq2) [40], and (ii) UMI-based (e.g., Drop-seq and 10x Genomics Chromium) protocols [55, 80].. In order to explain the steps required for preprocessing single-cell RNA-seq data, and clarify the motivations related to the presence of noise in this type of data, the main steps required to perform a scRNA-seq experiment are presented below:

- **Cell dissociation**. Starting from a biological sample, such as cancer biopsies,

patient derived organoids or blood samples, a suspension of single cells is generated. This is a critical step in the experiment, as cells may be suffer from stress or damages that then reflect on their expression profiles. In the next section, some quality control metrics are presented that aim at detecting this highly stressed or damaged cells to remove them from the sample.

- **Cell isolation**. This step is specific to the protocol used for sequencing: usually, UMI-based sequencing protocols use droplet-based isolation (with some exceptions such as [60]), while full-length protocols exploit plate-based techniques. Droplet-based isolation methods create droplets that will contain one single cell, and they are able to isolate thousands of cells. Plate-based methods instead sort cells cell into a small plate, and they are able to isolate a lower number of cells that varies based on the dimensionality of the plate employed. Please refer to [101] for more details regarding cell isolation.

- **Library preparation.** in this step, the mRNA transcripts are extracted from each cell, broken into fragments, converted to cDNA and amplified. However, the fraction of transcripts that are captured in this phase is estimated to be $\sim 10-20\%$ of all transcripts present in a cell [92] and thus this constitutes a source of noise for scRNA-seq data. In this step there is a difference between UMI and full-length methods. In fact, the former attach small nucleotide sequences called Unique Molecular Identifiers (UMI) to every transcript before amplification, so that once fragments get amplified the information about which duplicates refer to the same original molecule can be found in the UMI [29, 81]. This thus eliminates the amplification bias, where some molecules my be preferentially amplified with respect to others, resulting in a distorted quantification of the corresponding gene expression values. UMIs are attached to the end of each transcript before fragmentation, and for this reason only those fragments that include the transcript end are sequenced. On the other hand, full-length methods don't include any identifier before amplification, enabling sequencing of all fragments (hence the name *full-length*. However, such technologies are affected by the amplification bias.

- **Sequencing.** Each cDNA fragment is sequenced producing a read, that are strings representing the sequence read from the cDNA fragment. Such reads are then assembled together in a FATSQ file, which together with the reads reports their quality scores.

Technical problems during the different steps of a sequencing experiment introduce noise in the final gene expression matrix, and thus in this work a comparative assessment between different denoising methods is presented. For the purpose of the comparison, both UMI and full-length protocols have been considered.

3.1.2 Denoising of gene expression profiles

Single-cell RNA sequencing experiments are affected by technical errors, due to problems such as low capture efficiency, amplification bias, low sequencing depth etc., and the resulting count matrices are characterized by high levels of noise. Thus, 0 values in the count matrix may correspond to either true biologically non expressed genes or to false negatives, meaning that the gene was expressed in the cell but was not captured and sequenced (i.e., dropout event).

Thus, over the last decade different computational methods have been developed with the goal of recovering the corrupted information and missing data from single-cell gene expression matrices. Such methods take in input the count matrix, and they return the denoised expression profiles, exploiting different techniques and making specific assumptions about data distribution. Without considering its application to single-cell RNA-seq data, denoising is a wide computational problem that aims at recovering corrupted information by building models that make assumptions about data distribution, and several distinct Machine Learning (ML) approaches can be applied for denoising, such as autoencoders and Bayesian models.

Given the plethora of methods developed for denoising of scRNA-seq profiles, a comparative assessment carried out by an independent research group would be useful in characterizing the performance of each method and identifying the most robust tools to be applied.

Thus, we carried out a review of 19 different denoising and imputation methods for single-cell RNA-sequencing data. We exploited both synthetic and real-world datasets, as the former are fundamental to assess the ability of methods in recovering corrupted information, due to the absence of ground truth values for real data. We categorized methods based on their assumptions and techniques used for denoising, and we assessed their performance considering multiple quantitative metrics to evaluate the ability of each method in (i) imputing dropout events (ii) recovering the true gene expression profile (iii) characterizing cell similarities (iv) improving the identification of Differentially Expressed Genes. To carry out the comparison, we considered gene expression data obtained with droplet and plate-based sequencing protocols for both simulated and real matrices. We aim at providing a fair and unbiased comparison, that can serve as a guideline for researchers to identify which method is the most suitable according to the data analysis task that needs to be performed and to the technology used to produce the expression matrices.

We notice that in the literature there are other works providing an unbiased comparison of imputation methods for scRNA-seq data, which are presented in [83, 107] and [142]. In detail, in [83] authors present an analysis where the main goal consists in quantifying whether by performing differential expression analysis on imputed expression profiles, false positives are introduced in the results. This work has two main limitations: first,

it only considers the imputation impact on one specific downstream task, that is differential expression analysis, and second, the set of tools considered in the comparison is limited to only 6 methods.
In the second work presented in [107], authors compare 8 different imputation methods to assess their ability in imputing dropout events and improving downstream analyses. With respect [83], this work considers a wider ranges of metrics to quantify the performance of each method, but it still suffers from two limitations: first, the set of tools tested is still limited to 8 leaving out some more recently developed methods. Second, while this work uses simulated data to assess the ability in recovering missing values, it does not take into consideration the denoising task, i.e., the task of correcting nonzero values in expression matrices.
Finally, the third work presented in [142] is the most complete, as it considers a broader range of methods and analyses their performance using multiple metrics. However, this work is also not taking into consideration the denoising task separately from the imputation. In fact, to generate synthetic data authors employ the tool Splatter [79], that simulates the output of a scRNA-seq experiment and enables to simulate dropout events. On the contrary, in our work we exploited SymSim, a tool that simulates first the biological ground truth and then the sequencing process, which enabled us to assess the impact of each denoising method also on the recovery of non-zero corrupted values. Thus, we believe that our comparative assessment provides a full overview over the impact of each denoising methods, and can provide a guideline for the identification of the best performing method.
From our work, we identified 4 methods, namely ENHANCE [128], MAGIC [102], SAVER [91], and SAVER-X [129], that achieve the overall best compromise across all the considered tasks. The methods considered can be divided in four categories based on the assumptions and computational strategy used to solve the denoising task, that are namely: data smoothing, Machine Learning, matrix theory and model based methods. In particular, both model based and a subset of ML based methods include prior knowledge on the counts distribution and about their biological variability,that is used to model the observed signal. In our work we show that two of the best performing methods - namely SAVER and SAVER-X - include such prior information, and we also analyse how among the ML based methods, those that include this type of prior knowledge achieve better performance than the rest of the methods in the same category.
In detail, the two distributions used to model counts in single cells are the Negative Binomial (NB) and Zero-Inflated Negative Binomial (ZINB), where the latter models count data with a high fraction of zero values. There is a currently ongoing debate about which of the two distributions is the most suitable to model scRNA-seq counts [152], as there is evidence that the NB is able to explain all the observed zeros in the data, without the need to rely on the zero-inflation [152, 151]. In our comparative as-

sessment we included a subset of methods that model counts both with a NB, such as SAVER-X [129], and a ZINB, such as scVI [95], and we showed that the latter method is less effective in recovering missing and corrupted information compared to the former that employs the NB. Thus, both the analyses presented in [152, 151] and the results in our comparative assessment, suggest that the NB is able to model counts data, without the need to include the zero inflation process.

Finally, when considering a scRNA-seq dataset, the choice between applying or non-applying a denoising step should be evaluated carefully. As it is indicated also in [120], in general it is not advisable to exploit denoised expression profiles to perform tasks such as the identification of differentially expressed genes: in fact, since the goal of such analyses is that of identifying the biological mechanisms responsible for the behaviour of specific cells sub-populations, we need to minimize the probability of distorting the true biological signal in the count matrices. However, in case one is particularly interested in detecting cells subpopulations in single cell datasets, performing denoising could aid the clustering task: in fact, in our work we showed how specific methods are able to enhance cell-to-cell similarities and can serve as an effective preprocessing step. Thus, a general practice that can be applied is that of exploiting denoised expression profiles during the clustering step, in order to identify cell subpopulations, and then employ the non-corrected expression profiles to compare the distribution of genes across cells assigned to different clusters.

Please notice that here the supplementary information of this work is not included, but it is available in the online version of the manuscript [149].

Briefings in Bioinformatics, 22(4), 2021, 1–19

https://doi.org/10.1093/bib/bbaa222
Method Review

A review of computational strategies for denoising and imputation of single-cell transcriptomic data

Lucrezia Patruno, Davide Maspero, Francesco Craighero, Fabrizio Angaroni, Marco Antoniotti and Alex Graudenzi

Corresponding authors: Marco Antoniotti, Department of Informatics, Systems and Communication, University of Milan-Bicocca, Milan, Italy. Tel: +39 0264487901; E-mail: marco.antoniotti@unimib.it; Alex Graudenzi, Institute of Molecular Bioimaging and Physiology, Consiglio Nazionale delle Ricerche (IBFM-CNR), Segrate, Milan, Italy. Tel: +39 0221717551; E-mail: alex.graudenzi@ibfm.cnr.it
Lucrezia Patruno and Davide Maspero are equal contributors. Marco Antoniotti and Alex Graudenzi are co-senior authors.

Abstract

Motivation. The advancements of single-cell sequencing methods have paved the way for the characterization of cellular states at unprecedented resolution, revolutionizing the investigation on complex biological systems. Yet, single-cell sequencing experiments are hindered by several technical issues, which cause output data to be noisy, impacting the reliability of downstream analyses. Therefore, a growing number of data science methods has been proposed to recover lost or corrupted information from single-cell sequencing data. To date, however, no quantitative benchmarks have been proposed to evaluate such methods. **Results.** We present a comprehensive analysis of the state-of-the-art computational approaches for denoising and imputation of single-cell transcriptomic data, comparing their performance in different experimental scenarios. In detail, we compared 19 denoising and imputation methods, on both simulated and real-world datasets, with respect to several performance metrics related to imputation of dropout events, recovery of true expression profiles, characterization of cell similarity, identification of differentially expressed genes and computation time. The effectiveness and scalability of all methods were assessed with regard to distinct sequencing protocols, sample size and different levels of biological variability and technical noise. As a result, we identify a subset of versatile approaches exhibiting solid performances on most tests and show that certain algorithmic families prove effective on specific tasks but inefficient on others. Finally, most methods appear to benefit from the introduction of appropriate assumptions on noise distribution of biological processes.

Key words: denoising; imputation; single-cell RNA-sequencing; machine learning

Lucrezia Patruno is a PhD student in computer science at the Department of Informatics, Systems and Communication of the University of Milan-Bicocca. Her studies focus on data analysis and machine learning methods for the study of complex biological phenomena.
Davide Maspero is a PhD student in computer science at the Department of Informatics, Systems and Communication of the University of Milan-Bicocca. His studies focus on data integration methods for complex biological systems.
Francesco Craighero is a PhD student at the Department of Informatics, Systems and Communication of the University of Milan-Bicocca. His research is devoted to deep learning and explainable AI, with the aim of explaining deep networks inner sparse representations.
Fabrizio Angaroni is a postdoc researcher at the Department of Informatics, Systems and Communication of the University of Milan-Bicocca. His research is focused on mathematical methods for modeling and data analysis of complex biological system.
Marco Antoniotti is an associate professor at the Department of Informatics, Systems and Communication of the University of Milan-Bicocca. His main research topics are bioinformatics, computational systems biology, simulation, verification and cancer data analysis.
Alex Graudenzi is a research fellow at the IBFM-CNR. His research integrates (bio)informatics, complex systems, statistics and systems biology to deliver computational methods for the investigation of complex biological phenomena.
Submitted: 26 May 2020; **Received (in revised form):** 7 August 2020

Introduction

In recent years, an increasing number of studies has involved data generated from single-cell RNA sequencing (scRNA-seq) experiments [1, 2], which quantify gene expression levels at single-cell resolution, thus providing insights into cell population heterogeneity [3]. scRNA-seq methods can be used to perform accurate transcriptome quantification with a relatively small number of sequencing reads, isolating a typically large number of single cells. In optimal conditions, scRNA-seq data can recapitulate the results of standard sequencing experiments from bulk samples, yet with a much higher resolution [4].

This is a great advantage, as many works report that even cells in a homogeneous population may have heterogeneous expression profiles [5–8]. For instance, scRNA-seq data can be used to characterize rare cell subpopulations that had been hidden in the output of bulk RNA sequencing experiments [9], as well as in the analysis of cancer evolution, where they can be exploited to study the heterogeneity of tumor cell subpopulations [10] and the processes that lead to drug resistance or metastasis [11]. The wide use of scRNA-seq technologies has also allowed the creation of cell atlases for simple organisms such as, for example, the *Caenorhabditis elegans* [12]; most importantly, there is an ongoing effort to create such map for the human organism, i.e. the Human Cell Atlas [13]. However, the analysis of single-cell sequencing data is affected by the complex combination of biological variation and technical noise, which typically result in sparse and noisy single-cell expression profiles.

On the one hand, stochasticity of gene expression is inherent in most biological systems, with respect to both the biochemical processes related to gene regulation and the fluctuations of other cellular components and phenomena [14]. For this reason, even cells of the same type within the same tissue may display different gene expression distributions, complicating the identification and characterization of cellular states and transitions [15].

On the other hand, currently available sequencing technologies are still hindered by various technical issues [2, 16, 17]. In particular, the most common approaches for scRNA-seq are based on either droplet platforms (e.g. Drop-seq [18], InDrop [19] and Chromium 10x [20]) or plate-based platforms (e.g. Smart-Seq2 [21], MATQseq [22], MARS-seq [23], CEL-seq [24] and SPLIT-seq [25]), while some further approaches rely on microfluidics (e.g. C1 SMARTer [26]) or nanowell arrays (e.g. SEQ-well [27]). Typically, droplet platforms allow to isolate a large number of single cells (from a few to many thousands), by sequencing the 3′-end and by employing unique molecular identifiers (UMIs) [28], which allow the tagging of each transcript before amplification, thus distinguishing original transcripts from amplification duplicates [29]. Conversely, plate-based platforms usually employ full-length sequencing protocols and, accordingly, allow to sequence a much lower number of single cells (∼hundreds), yet with a considerably higher coverage. Overall, all sequencing protocols are affected by a number of technological and experimental issues, which typically result in noisy measurements.

- Capture efficiency: due to (i) the low quantity of RNA in a given single cell, and (ii) the stochastic nature of gene expression patterns at the single-cell level, certain gene can display null expression level, since none of its transcripts may be captured, thus resulting in zero expression levels. These are the so-called dropout events [30] and might be particularly relevant for scarcely expressed genes. This issue causes both noise and a high sparsity in the data [9].
- Amplification bias: the amplification phase may be subject to potential PCR biases in the quantification of the abundance of each gene, such as preferential amplification of certain templates. UMI-based approaches are able to mitigate this issue, yet in any case, amplification biases can be a potential source of noise in the data.
- Sequencing depth: the number of sequenced reads per cell varies between different experimental settings and platforms, and this can result in noisy and sparse outputs, especially when the depth is relatively low [29].
- Batch effects: technical sources of systematic variation may add a confounding factor in downstream analysis. Batch effects can be generated by analyzing samples with different technologies, in different laboratories or in different runs [31, 32]. When multiple experiments are considered, it is appropriate to remove such bias. In recent years, many methods were proposed to reach this goal. However, the comparison of the performance of methods for batch removal requires an in-depth investigation that is beyond the scope of this work (see [33] for a recent review).

As a consequence, it is safe to suppose that (i) nonzero expression values may not coincide with the true transcript abundance in the cell and (ii) zero values observed in the gene expression profiles may be either due to truly non-expressed genes—in this case, we refer to structural zeros, as proposed in [34]—or to technical limitations of the sequencing technology, i.e. dropout events.

For this reason, many computational approaches have been developed to retrieve lost and corrupted information from scRNA-seq data, with the goal of returning an estimation of the correct expression levels in each single cell. Such methods are typically grouped in two major categories: (*i*) imputation methods, with the general goal of recovering the missing values in the data and (*ii*) denoising methods, aimed at adjusting the data by removing biological and technical noise. Very often, the two categories are mentioned indistinctly (see e.g. [35]), even though they comprise substantially different computational tasks.

To better distinguish the two categories, here, we propose a rigorous categorization of imputation and denoising methods for scRNA-seq data, in order to reduce the possible ambiguity in the definition of the underlying computational tasks (an analogous distinction was recently proposed in [36]).

- Imputation methods for scRNA-seq data include two major steps. The first step is aimed at distinguishing structural zeros (associated to non-expressing genes) from dropout events (i.e. genes whose transcripts were not captured during the sequencing process due to technical issues). Accordingly, in the second step, such methods strive to impute the values of dropout entries only. Nonzero entries and structural zeros are left unchanged.
- Denoising methods for scRNA-seq data ideally include both an imputation step (see above) and an additional computational step, which is aimed at modifying the entries which include falsely increased or decreased gene expression levels due to, e.g. biological variation or technical noise. According to this definition, all denoising methods are also imputation methods while the opposite is typically not true (a rigorous definition of the two categories is provided in section 1 of the Supplementary Material).

Methods in both categories rely on different assumptions and employ different algorithmic strategies to perform their tasks.

Thus, as reported in [37], a comprehensive comparison of all available approaches might be useful and timely to clarify which methods are more suitable for different circumstances and distinct data types. In particular, in [37], the different approaches are grouped in the following typologies.

- Data smoothing: the methods in this category aggregate the expression profiles of similar cells in order to perform denoising and imputation. In this category, we find DrImpute [38], DEWÄKSS [39], scHinter [40], kNN-smoothing [41], LSImpute [42], MAGIC [43], netSmooth [44], PRIME [45] and RESCUE [46]. Finally, other methods that use data smoothing to impute missing values are G2S3 [47] and scTSSR [48]. However, the former aggregates the information across similar genes to perform imputation, while the latter considers both similar cells and similar genes.
- External knowledge integrators: these methods exploit external knowledge to impute or denoise gene expression profiles. In this category, we find ADImpute [49], netSmooth [44], netNMF-sc [50], SAVER-X [51], SCRABBLE [52], scNPF [53] TRANSLATE [54] and URSM [55].
- Machine learning (ML): these methods employ ML techniques to correct for technical noise. We can find very recent methods that employ Artificial Neural Networks (ANNs) to infer the denoised or imputed version of the dataset, which are AutoImpute [56], DeepImpute [57], DCA [58], EnImpute [59], GraphSCI [60], LATE [54], scIGANs [61], SAUCIE [62], scScope [63], scVI [64] and SISUA [65]. Next, we have methods that use regression to correct for noise in the dataset, which are 2DImpute [66] and RIA [67].
- Matrix theory: these methods decompose the observed gene expression matrix in a low-dimensional space to remove noise. In this category, we find ALRA [68], ENHANCE [69], scRMD [70], CMF-Impute [71], deepMc [72], McImpute [73], PBLR [74], WEDGE [75], ZIFA [76] and Randomly [77].
- Model-based: these methods make assumption on the statistical model of the distribution of technical and biological variability and noise and perform denoising and imputation by estimating the parameters of the distributions. In this category, we find bayNorm [78], BISCUIT [79], BUSseq [80], CIDR [81], MISC [82], SAVER [83], scImpute [84], scRecover [85], SCRIBE [86], SIMPLEs [87] and VIPER [88].

We here present a comparative assessment of denoising and imputation methods for scRNA-seq data, with the goal of providing a general overview of their features, strengths and limitations, in order to understand in which data analysis task they are most computationally and statistically efficient. In particular, we selected a subset of 19 different methods out of the list mentioned above, by including some of the most widely used approaches and which fall in the following categories.

- Data smoothing methods: DrImpute [38], kNN-smoothing [41] and MAGIC [43].
- ML methods: AutoImpute [56], DCA [58], DeepImpute [57], SAUCIE [62], SAVER-X [51], SCScope[63] and scVI [64].
- Matrix factorization/theory methods: ALRA [68], ENHANCE [69], McImpute [73], Randomly [77] and scRMD [70].
- Model-based methods: bayNorm [78], SAVER [83], scImpute [84] and VIPER [88].

The comparative assessment was carried out both on simulated data, generated via the widely used tool SymSim [89], and four real-world scRNA-seq datasets from [90–93]. All computational methods were tested with respect to a number of metrics, in order to assess the effectiveness in imputing dropout events, recovering the true expression profiles, characterizing the similarity among cells and improving the identification of differentially expressed genes (DEGs), in addition to quantify their scalability. In the Results section, we present the results of the extensive comparative assessment, also by releasing a summary for a quick evaluation of the distinct techniques in different scenarios and experimental settings.

We note that previous works reviewing imputation methods have been proposed. In particular, in [94], the authors focus on understanding whether six different imputation strategies introduce false positives in the results of differential expression analysis. In [95], eight different methods are analyzed to understand whether they improve the result of clustering and differential expression analysis. Both works, however, do not include in the analysis the most recent methods and assess the performance of a relatively limited number of computational strategies. In addition, both works mainly focus on the imputation task, without assessing how denoising techniques may recover corrupted information. Finally, a recent preprint on a similar subject [35] exploits real-world data to assess the performance of imputation methods on downstream analyses. While this work includes a more extensive assessment of recent methods, it does not employ simulated data, which are necessary to evaluate a number of ground truth (GT)-based performance metrics. Further comments in this respect are provided in the Discussion section.

In the Methods section, we provide a brief description of each denoising and imputation method included in the study, discuss the performance assessment describing both the synthetic data generation and the real-world datasets and present the different metrics used in the analysis. In the Results section, we present the results of the comparative assessment on both simulated and real data, also by releasing a summary for a quick evaluation of the distinct techniques in different scenarios and experimental settings. Finally, in the Discussion section, we draw conclusions about the comparison and discuss possible future developments.

Methods

In this section, we describe in detail the 19 methods included in the comparative assessment; we discuss the synthetic data generation and present the 4 real-world scRNA-seq datasets from [90–93] employed in the analysis, as well as the performance metrics.

Description of denoising and imputation methods

The 19 methods that have been analyzed and tested can be partitioned into the following four families, according to their assumptions and modeling techniques: smoothing, model-based, matrix factorization/theory and ML. In the following sections, we provide a brief description of each method. For additional details, we refer the reader to the original papers.

Data smoothing methods

The first category includes methods that aggregate the expression profiles of similar cells, e.g. by averaging the expression values, in order to impute (DrImpute) or denoise (MAGIC and kNN-smoothing) their expression values.

DrImpute [38] imputes dropout events with the following three steps: first, it computes a distance matrix between cells, then it runs the k-means algorithm and, lastly, it defines the expected value of a dropout event as the average value of that

gene over the cells belonging to the same cluster. To make the estimations more robust, the similarity matrix is computed with both Pearson and Spearman correlations and a range of number of clusters is tested. The averaged estimation over all combinations is taken as the final imputation value, reducing the risk of over-imputation.

kNN-smoothing [41] improves the signal-to-noise ratio of single-cell expression profiles with a two-phase algorithm: first, the k-nearest neighbors (kNNs) of each cell are identified, then the gene expression profile of each cell is smoothed by considering its neighbor profiles. The initial step of the algorithm is performed by normalizing the expression profiles and stabilizing their variance. Then, to overcome the problem of finding the best assignment for k, smoothing is applied in a progressive fashion, by starting from $k = 1$ and increasing k step-by-step until the desired level of smoothness is reached.

MAGIC [43] extracts the true similarity between cells by amplifying biological trends, while simultaneously filtering out spurious correspondences due to noise in the data. First, to overcome the problem of data sparsity, a nearest neighbor graph based on cell–cell expression distance is built. Then, an affinity matrix is defined by applying a Gaussian kernel on the principal components of the graph. Lastly, a diffusion process [96] is applied on the similarity matrix to obtain a smoothed, more faithful affinity matrix. The final imputation involves computing the new expression of each gene as a linear combination of the same expression in similar cells, weighted by the similarity strength obtained in the previous steps.

ML methods

This group includes methods that apply ANNs to solve the denoising problem (further details on ANN types are reported in section 2 of the Supplementary Material). As reported in [37], an increasing number of methods fall in this category (see above). In particular, we selected DeepImpute, DCA, SAVER-X, SAUCIE, scScope, AutoImpute and scVI.

AutoImpute [56] employs a sparse autoencoder, to learn the distribution of the input gene expression matrix and perform imputation. With regard to the implemented loss function, this method takes advantage of standard reconstruction errors such as (root) mean squared error, applied only on the nonzero expressed genes. After training the autoencoder (AE), the reconstructed matrix is taken as the imputed output.

DCA [58] employs AEs to perform denoising. Instead of the classical AE decoder output, it defines a parametric decoder that models each gene count as a negative binomial (NB) or a zero-inflated negative binomial (ZINB) distribution; consequently, the reconstruction error is defined as a likelihood. The predicted distribution is then used to generate the denoised output.

DeepImpute [57] employs a deep feedforward network (DFN) to perform imputation. After the initial preprocessing, where only relevant genes are kept, N random groups of genes G_i are defined. Then, for each gene in each G_i, a set I_i with the top five Pearson correlated genes not in G_i is built. Lastly, each I_i will be an input for a different DFN, trained to output G_i. The output of each DFN is then used for imputing dropout events.

SAUCIE [62] is an AE-based denoising method that also supports batch correction and enhanced clustering and visualization capabilites. More in detail, the AE embedding layer is used for both low-dimensional visualization and batch correction, by minimizing the difference between the probability distribution of layer's activations belonging to different batches. Moreover, the activations of the decoding part are binarized to define an encoding of each cell, which is then used for clustering. Lastly, denoising is performed by minimizing the reconstruction error, i.e. the mean squared error, that deals both with noise and dropout events.

SAVER-X [51] is an extension of SAVER [83] that pairs the Bayesian model with an AE. A NB distribution is used to model technical and biological noise, while the AE is used to estimate the portion of gene expression that is predictable by the other genes. Lastly, Bayesian shrinkage is used to compute a weighted average of the predicted expression values and the observed data, to get the final denoised value. Additionally, SAVER-X allows transfer learning [97] across species, thanks to the flexibility of AEs, allowing to extract information from data belonging to different species and experimental conditions.

scScope [63] exploits a deep learning approach for imputation, combining an AE with a recurrent layer. The architecture of the neural network is composed by a first layer that performs batch correction. Successively, the encoding and decoding layers of the AE perform compression and reconstruction, respectively, of the batch corrected input. Lastly, the imputation layer corrects the missing values and sends back the imputed output to the encoding-decoding layers, to re-learn a compressed representation. The loss function is defined as a standard reconstruction error, on the nonzero entries.

scVI [64] employs a variational AE to specify a ZINB distribution, which models the true gene expression. More in detail, the neural network takes as input each batch-annotated cell expression and successively learns a variational distribution accounting for, separately, the cell-specific scaling factor and the remaining gene variation; furthermore, the defined latent space allows to perform both clustering and visualization. Lastly, the ZINB distribution is specified based on the learned latent representation and the cell scaling factor.

Matrix factorization and matrix theory methods

The third category comprises four methods that denoise (ENHANCE) or impute (ALRA, McImpute and scRMD) the observed gene expression data by solving a matrix factorization problem [98]. For the sake of simplicity, we added to this category also a method that performs imputation by exploiting random matrix theory (RMT): Randomly.

ALRA [68] performs imputation by low-rank matrix completion [99] of the observed gene expression matrix. The algorithm is composed by two phases: firstly, a low-rank approximation with Singular Value Decomposition [100] is computed. Then, to distinguish dropouts from true zeros, the authors observed that biological zeros in the computed low-rank matrix are assigned to small values around 0, due to the approximation error. Consequently, by taking the magnitude of the smallest negative value of each gene as an approximation of the error, it is possible to define a gene-wise threshold to distinguish dropouts and extract the imputed values.

ENHANCE [69] is a method that combines PCA and cell aggregation using kNNs to denoise the observed count matrix. The algorithm can be divided into two main steps. The first one accounts for reducing the bias toward highly expressed genes, by aggregating the expression of similar cells based on the distance between their principal component scores. The second phase projects the aggregate matrix on the first k principal components, where k is selected to represent only true biological differences. Lastly, the selected components are used to derive the final denoised matrix.

McImpute [73] is a low-rank matrix completion approach to impute missing values in a gene expression matrix. This method aims at finding a lower-dimensional decomposition of the input matrix. They formulated a low-dimensional nonnegative matrix factorization problem as an optimization problem, solved using the majorization-maximization technique [101]. To ensure the convexity of the problem, McImpute solves a relaxed version of the original objective: nuclear norm minimization. Lastly, the resulting decomposition is used to impute missing values.

Randomly [77] is a recent denoising method that extracts the true biological signal from the gene expression data by analyzing the eigenvector statistics predicted by RMT [102]. The algorithm is composed by three steps. In the pre-processing step, expression counts are normalized and genes contributing to a sparsity-induced nonbiological signal are removed; then, the random matrix accounting for the noise is estimated. Lastly, the eigenvalues carrying the true biological signal are extracted following RMT, providing a low-rank representation of the input data; additionally, the genes that are mostly responsible for the signal directions can be separated from the less relevant ones.

ScRMD [70] is a method that approaches the imputation task by means of a robust matrix decomposition (RMD) approach [103]. The authors assumed that we can decompose each gene expression in the following components: the mean expression of cells belonging to the same cluster, the specific cell variability, the measurement error and the dropouts events. The method defines each component as a matrix decomposition problem, solved with an alternating direction method of multiplier, by also applying a regularizer to account for the low-rank of the biological signal and the sparseness of the observed counts.

Model-based methods

This category is composed by methods that model the observed expression value of each gene in each cell as a random variable and perform imputation (scImpute and VIPER) and denoising (bayNorm and SAVER) by estimating the parameters of their distributions.

bayNorm [78] employs a Bayesian approach to perform denoising. The posterior distribution of the original counts is composed by (i) the likelihood of obtaining the observed transcripts, modeled as a Binomial distribution, and (ii) a prior on each gene expression value. In order to model biological variability, bayNorm employs a prior on the underlying true gene expression levels, by modeling them as variables following an NB distribution. Parameters can then be estimated locally or globally, depending on one's interest in amplifying or not, respectively, the intergroup differences between cells.

SAVER [83] estimates the true gene expression levels by modeling observed counts as a NB distribution. More in detail, the technical noise in the gene expression signal is approximated by the Poisson distribution, while the gamma prior accounts for the uncertainty in the true expression. The final recovered expression is a weighted average of the normalized observed counts and the predicted true counts.

scImpute [84] is a method that performs imputation, in a three-step algorithm. Initially, it identifies subpopulations of cells by first applying PCA and, successively, spectral clustering [104] on the remaining dimensions. To infer which genes are affected by dropout, it models genes in each subpopulation with a gamma-normal mixture model. Lastly, only highly probable dropout events are considered, to reduce over-imputation, and the final imputation value is computed as a linear combination of the expression of the other cells in the same subpopulation, weighted by the pairwise similarity.

VIPER [88] is an imputation method composed of four phases. The first step performs a pre-selection of candidate similar cells, to reduce overfitting. Then, a least-squares method is used to choose a local neighborhood for each cell. To prevent imputing missing values, VIPER estimates the dropout probability and the expected expression for each zero-valued neighbor. Furthermore, to adjust dropout events, the gene expressions in each neighborhood are assumed to follow a zero-inflated Poisson mixed model, estimated using expectation maximization. Lastly, imputation is performed by computing the weighted sum of the expression of each neighbor, by also taking into account the computed dropout adjustments.

Performance assessment

In the original articles, the imputation and denoising methods introduced above are often compared with competing approaches. However, such comparisons typically involve a limited number of denoising methods and a small number of selected experimental settings. In order to provide a comprehensive evaluation of performances, in this work, we tested all methods on a large number of both simulated and real-world datasets, with respect to several metrics.

In particular, we generated an extensive array of simulated data, for which the GT is available and which allow to quantify the ability of each method to actually recover the lost information (see Supplementary Material section 3 for details about the generation of such data). Moreover, we tested all methods on four real-world scRNA-seq datasets generated via distinct experimental protocols and settings.

Simulations

We employed the tool SymSim [89] to generate a large number of synthetic scRNA-seq datasets (for a total of 90 distinct synthetic datasets). SymSim takes as input the number of single cells, the number of genes, the number of cell subpopulations (characterized by distinct gene expression patterns) and a number of parameters that tune the amount of biological variability and technical noise.

The tool returns as output (*i*) a GT expression matrix, which includes biological variability but no noise; (ii) a theoretical expression profile (TEP) for each cell subpopulation, which is obtained by removing the biological variability from the GT; and (iii) a noisy (and sparse) expression matrix (NEM), which is finally derived by simulating the steps of a sequencing experiment.

In this work, we generated datasets simulating two main experimental scenarios and, in particular,

(i) non-UMI full-length datasets (i.e. high-coverage, high-amplification bias), including 100 single cells and modeling a typical plate-based full-length sequencing experiment (e.g. Smart-Seq2). Thirty datasets were generated with distinct parameter settings;
(ii) UMI datasets (i.e. low-coverage, low-amplification bias), including 3000 single cells (30 datasets) and 10000 single cells (30 datasets) and modeling a typical droplet sequencing experiment (e.g. Chromium 10x).

The different datasets in each scenario are characterized by distinct parameter settings, in terms of number of cell subpopulations ($\{3, 5\}$), noise level (5 levels), number of selected (most variable) genes ($\{500, 2000, 10000\}$) (Table 1). A detailed

Table 1. Summary of the simulated datasets. We simulated a total of 90 datasets, with the following combinations of parameters: 3 values of sample size (number of single cells) ×2 different numbers of subpopulations ×5 levels of noise ×3 numbers of selected most variable genes

Protocol	UMI	Non-UMI full-length
No datasets	60	30
No cells	{3000; 10000}	100
GT sub-populations	{3; 5}	{3; 5}
Capture efficiency	Low	High
Amplification Bias	No	Yes
Coverage	Low	High
Noise level[a]	{1; 2; 3; 4; 5}	{1; 2; 3; 4; 5}
No genes	{500; 2000; 10000}	{500; 2000; 10000}

[a] The levels of noise present in the simulated datasets are defined in section 3 of the Supplementary Material, which we refer for further details on synthetic data generation.

description of synthetic data generation can be found in the Supplementary Material section 3.

Real-world datasets

All methods were tested also on four distinct real-world scRNA-seq datasets, generated with distinct protocols and experimental specifications. In detail, we have the following.

- **RW-D** #1 (PBMCs – 10x) [90]: this widely employed scRNA-seq dataset is generated via 10x Genomics platform [20] and includes 68579 peripheral blood mononuclear cells (PBMCs), which are annotated with 11 cell types of the immune system, via correlation with benchmark gene expression profiles. This dataset was used in our analysis to assess the performance of imputation and denoising methods in characterizing cell similarities (for further details on the dataset, please refer to [90]; instructions for download are provided in the Supplementary Material).
- **RW-D** #2 (lung cell lines – 10x) [91]: this scRNA-seq dataset is generated via the 10x Genomics platform and includes 3918 cells from 5 distinct cell lines, which were assigned to its corresponding identity by exploiting known genetic differences (i.e. SNPs) between cell lines [91]; this allows not to rely on gene expression profiles for cell labeling. We employed this dataset to assess the robustness of the characterization of cell similarity.
- **RW-D** #3 (pancreatic islets – Smart-Seq2) [92]: this scRNA-seq dataset is generated via the full-length Smart-Seq2 protocol and includes 3514 cells from human pancreatic islets of four diabetic patients and five healthy samples. We employed this dataset to assess the performance of imputation and denoising methods with respect to cell similarity characterization when processing data from non-UMI full-length protocols.
- **RW-D** #4 (melanoma cell lines – 10x, Fluidigm/Smart-Seq, bulk) [93]: this dataset includes three different measurements from the same biological samples, namely (i) bulk RNA-seq experiments, (ii) 10x Genomics scRNA-seq experiments with 737280 barcodes, (iii) Fluidigm/Smart-Seq scRNA-seq experiments with approximately 100 single cells. Since no cell type labels are provided in this dataset, we here used the data to compare the performance of imputation and denoising methods with respect to the correct identification of DEGs, by setting the results obtained on bulk data as baseline.

All real-world datasets were preprocessed to consider only high-quality single cells, and downsampled, to ensure a uniform assessment scheme for all methods. In Table 2, one can find the main features of all datasets employed in the analyses (see Supplementary Material section 4 for further details on preprocessing and downsampling).

Performance metrics

To evaluate the performance of the 19 selected methods, we employed a number of metrics, which were assessed with respect to either simulated or real-world data, according to the specific cases. All metrics are further detailed in section 5 of the Supplementary Material.

Imputation of dropout events (simulations) The effectiveness of the methods in identifying and correcting dropouts events can be evaluated by employing the GT expression matrix obtained from simulations (see Supplementary Material section 5 for additional details). In order to quantify the correct imputation of the dropout entries present in the GT, we employed three distinct metrics.

In particular, we computed (i and ii) precision and recall on dropout entries only (i.e. entries that are > 0 in the GT and are $= 0$ in the NEM), (iii) the Spearman correlation delta between the imputed/denoised expression matrix (for the sake of readability, we will refer to as denoised expression matrix, from now on) and the GT with respect to all the zero entries in the NEM, which allows to evaluate how imputed entries are correlated with GT values (this metric is shown in the Supplementary Material section 5).

Notice that the false discovery rate (FDR) can be easily determined from precision (FDR $= 1-$precision) and, in this case, allows to evaluate the effectiveness of the methods in not imputing structural zeros (i.e. entries that are 0 both in the GT and in the NEM).

Recovery of true gene expression profiles (simulations) To estimate the ability of each method in recovering the true single-cell gene expression profiles, we relied on both the GT and the NEM obtained from simulations.

In particular, we computed the difference between the Spearman correlation coefficient ρ computed after imputation or denoising (i.e. ρ between denoised expression matrix and GT) and that computed before imputation or denoising (i.e. ρ between NEM and GT). This measure is denoted as delta correlation in the following, $\Delta\rho$.

Characterization of cell similarity (simulations and real-world data) In order to evaluate the effectiveness of each method in capturing the similarity among cells, we computed the average silhouette coefficient (or width) [105] by grouping single cells according to the GT labels, i.e. cell subpopulations labels for both simulated data, and cell type/line labels for real data. Higher values of the average silhouette coefficient indicate that cells are grouped consistently with GT labels. Therefore, we here measured the difference between the average silhouette coefficient obtained from denoised data and that computed from the NEM (i.e. silhouette delta). Further detail about the evaluation of such metric is given in the Supplementary Material section 5.

We finally remark that, with regard to simulations, we here employed the TEP of all cell subpopulations as performance benchmark.

Table 2. Features of real-world datasets. Main features of the four real-world datasets used in the assessment of imputation and denoising methods: **RW-D**#1 [90], **RW-D**#2 [91], **RW-D**#3 [92] and **RW-D**#4 [93]

Dataset			Number of cells		
RW-D	Name	Protocol	Original	Employed	Task
#1	PBMC [90]	UMI	68579	3000	Cell sim.
		UMI	68579	10000	Cell sim.
#2	Lung cell lines [91]	UMI	3918	3918	Cell sim.
#3	Pancreatic islets [92]	Non-UMI	352	245	Cell sim.
		Non-UMI	383	243	Cell sim.
		Non-UMI	383	197	Cell sim.
		Non-UMI	383	224	Cell sim.
		Non-UMI	383	196	Cell sim.
		Non-UMI	383	263	Cell sim.
		Non-UMI	383	93	Cell sim.
		Non-UMI	384	275	Cell sim.
		Non-UMI	384	293	Cell sim.
#4	Sake (Parent.) [93]	UMI	737280	3178	DEGs
		Non-UMI	113	113	DEGs
	Sake (Resist.) [93]	UMI	737280	3085	DEGs
		Non-UMI	84	84	DEGs

Identification of DEGs (real-world data)
To assess the improvement on the identification of DEGs due to the application of imputation/denoising methods, we employed real-world dataset **RW-D**#4 which includes two independent cell populations, namely parental and resistant, for which single-cell 10x, single-cell Fluidigm/Smart-Seq and bulk sequencing experiments were executed.

We proceeded as follows: for each single-cell dataset (10x and Fluidigm/Smart-Seq), we performed a standard Wilcoxon test to select the DEGs ($p < 0.05$) between parental and resistant populations, with respect to both the NEM and the denoised expression matrix, and which results in two distinct lists of DEGs.

The expression profiles of the DEGs are then used to calculate the Spearman correlation coefficient between each single cell and the corresponding bulk profile. The distribution of the difference of the Spearman correlation coefficient as computed on denoised data and that on the NEM is used to evaluate the performance for this task.

Computation time (simulations) We finally analyzed the computational time of each tested method to impute or denoise datasets with distinct numbers of observations (i.e. single cells) and of variables (i.e. genes), with respect to a selected number of simulated datasets. All computations were performed on a HP® Z8 G4 Workstation equipped with two Intel® Xeon® Gold 6240 processors at 2.60 GHz, 1 TB DDR4 RAM at 2933 MHz and Linux Mint 19.2 Tina.

We note that, in the original papers, the authors do not declare any theoretical worst-case performance in terms of $O(\cdot)$ notation; although for many of them, it would be derivable from literature. We therefore present an empirical study of the relative performances of the methods.

Parameter settings of computational methods

Most methods were run on both simulated and real-world datasets using default settings and following guidelines provided from the authors, if any. For additional details on parameter settings of all methods, please refer to section 6 of the Supplementary Material and to Supplementary Table 4.

Note that we report the results SAVER-X without pre-training, as its performance seems to be only slightly affected by pre-training on real-world datasets, as shown in Supplementary Figure 9. Besides, for analyses involving synthetic datasets, we did not run AutoImpute, McImpute, scImpute and VIPER on datasets with 10 000 cells and 10 000 genes, and we did not execute VIPER on **RW-D**#1 (downsampled to 10 000 cells and 10 000 genes), due to the high computational time required by such methods. Furthermore, for 10 out of 30 non-UMI full-length simulated datasets, SAUCIE collapsed all cells into one unique profile. Thus, such datasets were not included in the analysis. Finally, please note that for Fluidigm/Smart-Seq datasets in **RW-D**#4, the computation of bayNorm and ENHANCE raised errors and, therefore, their results are not reported.

Results

We start by providing a qualitative example of the effect of the tested imputation and denoising methods: Figures 1 and 2 show the tSNE low-dimensional representation [106] of a synthetic dataset (3000 cells, 5 subpopulations and 2000 genes) and of one real dataset (**RW-D**#1, downsampled to 3000 cells and 2000 genes; see the Methods section for further details). For the synthetic dataset, we show the GT expression matrix, the NEM and the denoised datasets returned by each method, whereas for **RW-D**#1 we show its original expression matrix and the corresponding denoised versions.

From this qualitative analysis, one can appreciate the substantial different data transformations which are determined by the distinct methods.

While it is difficult to draw conclusion from single experiments, certain methods apparently tend to reduce the variability of gene expression profiles, resulting in more compact representations on the tSNE space (e.g. kNN-smoothing, SAUCIE, MAGIC), some others appear to enhance the inter-cluster distance (scImpute, SAVER and ENHANCE), while most methods seem to preserve the original disposition in the transcriptomic space, with some exceptions (note that in this and subsequent analyses, AutoImpute seems not to have reached convergence, with default parameters).

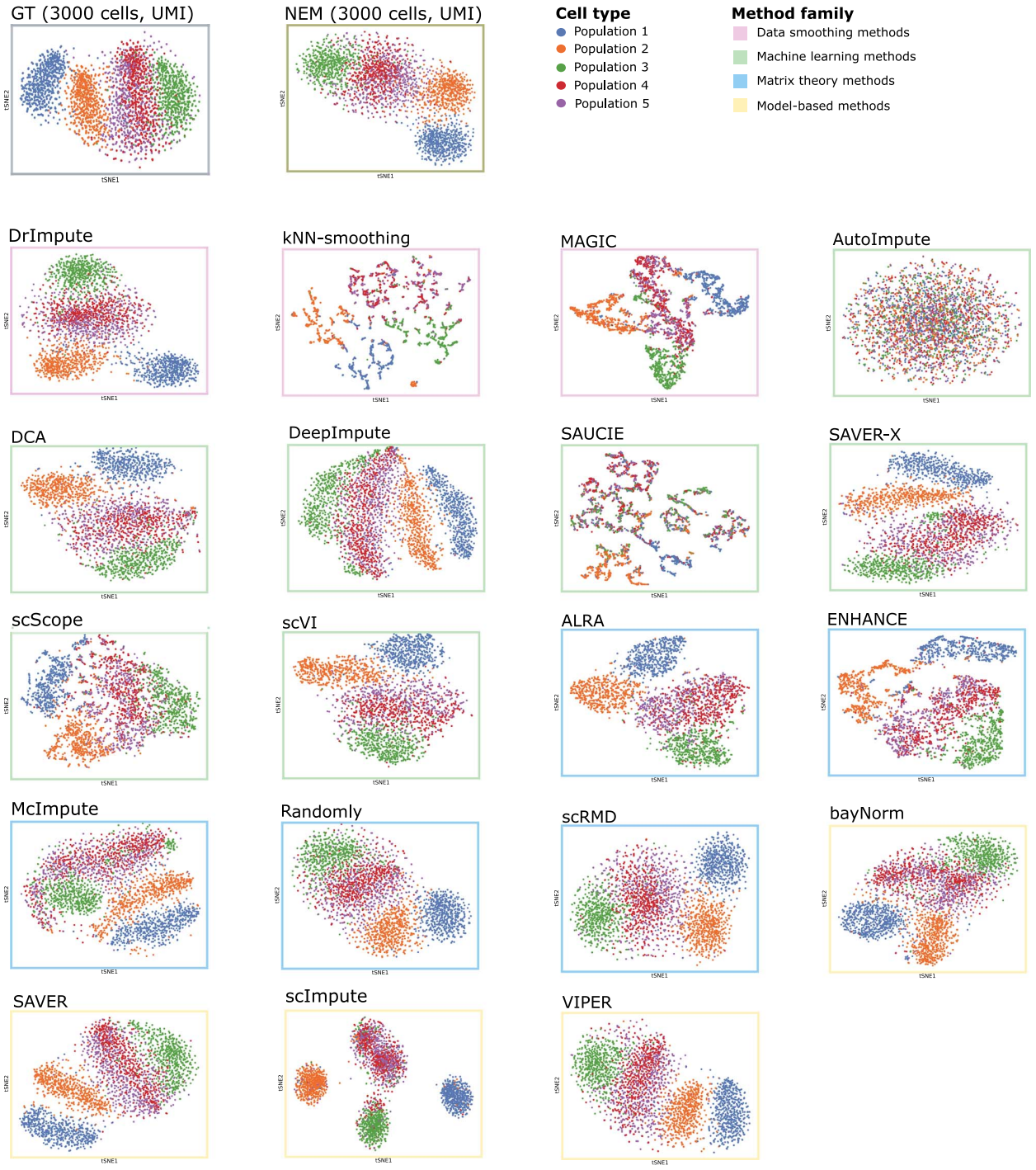


Figure 1. Effect of 19 imputation and denoising methods on a selected simulated scenario via tSNE low-dimensional representation. tSNE low-dimensional representation [106] of the gene expression profile of 3000 single cells of a selected synthetic UMI dataset with 5 subpopulations and 2000 genes. For this dataset, we present the tSNE plot of the GT expression matrix generated via SymSim and the NEM obtained after simulating the sequencing experiment. The remaining tSNE plots represent the gene expression of the cells after the application of all tested denoising and imputation methods to the NEM.

The visualization of three further synthetic datasets and of real-world datasets **RW-D**#2 and **RW-D**#3 are shown in Supplementary Figures 1–5. The results of the quantitative assessment with respect to the metrics described in the Methods section are presented in the following.

Imputation of dropout events (simulations)

We first assessed the performance of all methods in imputing dropout events (i.e. entries $= 0$ in the NEM but > 0 in the GT expression matrix), leaving structural zeros unchanged (i.e. entries $= 0$ both in the NEM and the GT). The parameters of all

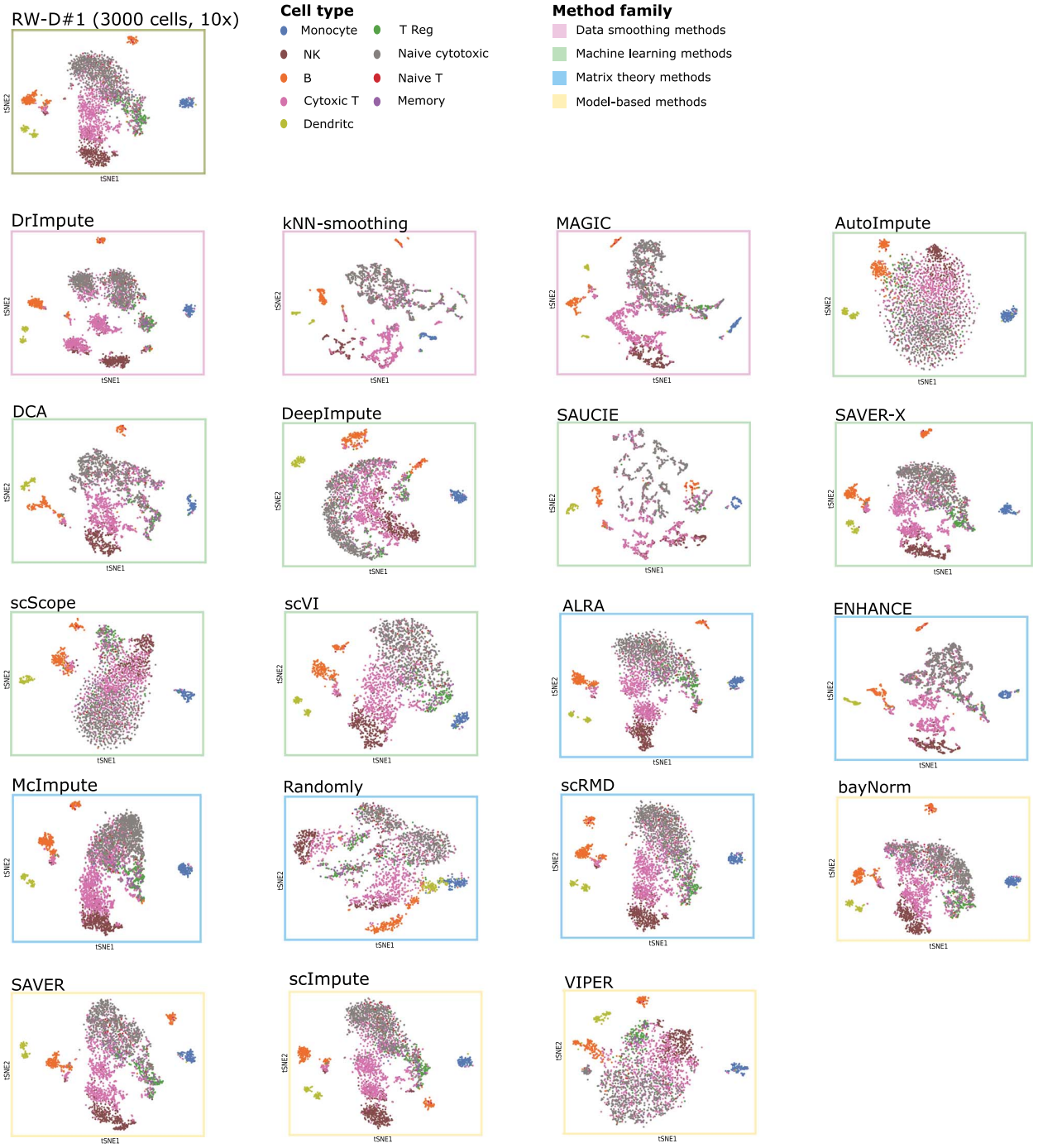


Figure 2. Effect of 19 imputation and denoising methods on real-world dataset RW-D #1 via tSNE low-dimensional representation. tSNE low-dimensional representation [106] of the gene expression profile of 3000 selected cells from RW-D #1 (PBMCs – 10x) [90] as computed on the 2000 most variable genes. For this dataset, we present the tSNE projection of the original dataset, which includes nine cell types and the tSNE plots of the single-cell expression profiles after the application of all methods under analysis.

simulations are recapitulated in Table 1 and in Supplementary Tables 1 and 2. Please refer to the Methods section and to Supplementary Material sections 3 and 5 for details on synthetic data generation and performance metrics. Note that Randomly was not included in this test, since it provides an already scaled expression matrix as output.

In Figure 3, one can find, for each method, the median precision and recall on correctly imputed dropouts (in this case, a true positive is an entry > 0 both in the GT and in the denoised expression matrix but $= 0$ in the NEM), grouped according to the number of (most variable) selected genes ($\{500, 2000, 10\,000\}$) and the number of single cells (100 for non-UMI full-length

Figure 3. Performance assessment on imputation of dropout events (simulations). Assessment of imputation of dropouts, as evaluated on non-UMI full-length simulated datasets (100 single cells) and UMI simulated datasets ({3000, 10000} single cells), with {500, 2000, 10000} genes. In each panel, we display a scatter-plot returning, for each imputation and denoising method, the median precision (y-axis) and recall (x-axis) as computed on correctly imputed dropouts (computed on 10 simulations per setting). In this case, a true positive, is an entry that is > 0 in the denoised expression matrix and in the GT but is $= 0$ in the NEM (see the Methods section for further details and Supplementary Table 3 for the confusion matrix). The squared shade indicates methods with precision and recall > 0.80. In Supplementary Figure 6, the distribution of precision and recall is displayed.

and {3000, 10000} for UMI datasets). In order to identify the methods showing high precision (i.e. how many imputed entries are dropouts) and high recall (i.e. how many dropouts are imputed) scatter-plot areas corresponding to high values for both measures ($>$ 0.80) were highlighted (in Supplementary Figure 6 the distributions of precision and recall on settings are displayed).

As a first result, most methods struggle when dealing with non-UMI full-length datasets (with 100 cells), as proven by the relatively lower value of average precision. This aspect is likely due to the low number of observations (single cells) as compared with the number of variables (genes) and consistently affects the performance of all methods on most tasks (see below).

Conversely, we observe a subset of methods that achieve extremely positive performances (both precision and recall $>$ 0.80) for UMI datasets with 3000 and 10 000 cells. In detail, VIPER provides the best performance with datasets with 500 genes, while for datasets with 2000 and 10 000 genes, ALRA, bayNorm, DrImpute, ENHANCE, kNN-smoothing, MAGIC, SAVER, SAVER-X and scVI consistently provide optimal and analogous performances. In particular, such methods show values of recall very close to 1 in all experimental settings (with the exception of kNN-smoothing). While this effect might be due to over-imputation, such methods also display significantly high precision in most settings. Notice also that higher values of precision implicate a lower fraction of wrongly imputed structural zeros (entries $= 0$ both in the GT and the NEM), as measured by the false discovery rate (FDR $= 1-$ precision).

Finally, we note that scRMD and scImpute display the highest values of precision in most settings, which, however, are most likely due to the conservative nature of the approaches, which tend to limit the number of imputed values. This observation is strengthened by considering the low values of recall for both methods: indeed, as recall corresponds to the fraction of imputed dropouts, a value close to 0 indicates that the method did not impute most of the events.

To further extend the analysis on imputation of dropouts, in Supplementary Material section 7 (Supplementary Figure 7), we return the analysis of the Spearman correlation coefficient computed considering zero entries of the NEM and which allows to quantify the correlation between imputed entries and the corresponding GT expression values. On the one hand, bayNorm, DrImpute, ENHANCE, MAGIC, SAVER and SAVER-X provide the most accurate and robust results in most scenarios, proving effective in correctly recovering the true expression values of imputed entries. On the other hand, ALRA, kNN-smoothing and scVI and VIPER, which exhibit good values of precision and recall on imputed dropouts (see above), display a relatively lower performance in terms of correlation of the imputed entries with respect to the GT expression values.

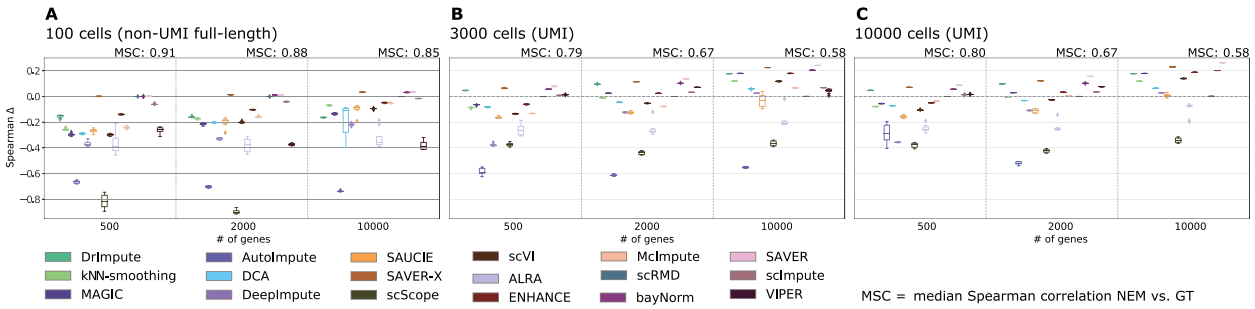


Figure 4. Performance assessment on recovery of true gene expression profiles (simulations). Assessment of recovery of true expression profiles, as evaluated on non-UMI full-length simulated datasets (100 single cells, panel A) and UMI simulated datasets ({3000, 10000} single cells, panels B and C), with {500, 2000, 10000} genes. The boxplots return the distribution of correlation delta, $\Delta\rho$, i.e. the difference between the Spearman correlation coefficient computed between the denoised expression matrix and the GT and that computed between the NEM and the GT, for all methods in each experimental setting. The baseline median Spearman correlation coefficient (MSC) between the NEM and GT is reported on top of the panels, for each setting, while in Supplementary Figure 8, the relative distributions are returned.

Recovery of true gene expression profiles (simulations)

We next tested the capability of each method in recovering the GT gene expression profiles, by using simulated data. In Figure 4, one can find the difference of the Spearman correlation coefficient as computed between the GT and the denoised expression matrix after the application of all 19 methods and that computed between the GT and the NEM. Such difference is denoted as correlation delta, $\Delta\rho$, from now on (see the Methods section and Supplementary Material section 5 for further details).

In particular, the results are displayed according to the number of genes, {500, 2000, 10000} and number of cells, 100 for non-UMI full-length and {3000, 10 000} for UMI experiments, as this allows to analyze the performance under different experimental settings. Note that, as for the analysis on imputed entries, we here do not include the output of Randomly, which provides a scaled output matrix.

As expected, sample size and protocol-type highly influence the capability of any method to recover corrupted information, as the performance of all methods generally improves with datasets with a larger number of single cells and generated via UMI-based protocols. More specifically, most methods appear to struggle when processing non-UMI full-length datasets characterized by a low number of cells (i.e. = 100), delivering unreliable and often erroneous denoised expression profiles, as proven by the negative Spearman correlation delta observed in most cases (up to −0.45 for some methods).

Conversely, correlation deltas progressively improve with UMI datasets including larger numbers of cells and/or genes, and, in particular, all methods with the exception of ALRA and scScope, achieve a positive median delta with datasets with 10 000 genes and 10 000 cells.

Examining the methods in greater detail, we observe that bayNorm, SAVER and SAVER-X are the methods with the best overall performance, as they always provide a positive correlation delta and achieve the best results with both non-UMI full-length and UMI datasets. Furthermore, we note that such approaches show an extremely low variance, suggesting that the results are robust. Among the other approaches, we note that DrImpute displays a high correlation delta with UMI datasets, whereas both ENHANCE and MAGIC exhibit remarkable performances with datasets with more than 3000 cells and more than 2000 genes.

All in all, the results of this and the previous analyses suggest that bayNorm, SAVER and SAVER-X might be an adequate choice for both imputing dropouts and recovering corrupted information, as they show the most accurate and stable performances with both UMI and non-UMI full-length datasets, whereas DrImpute, ENHANCE and MAGIC are similarly effective when processing UMI datasets.

Characterization of cell similarity (simulations and real-world data)

When analyzing scRNA-seq data, one might be interested in characterizing the possible heterogeneous populations included in the dataset, typically performing unsupervised clustering. For example, the Scanpy [107] and Seurat [108] packages for single-cell analyses incorporate the Louvain and Leiden algorithms for community detection [109], which identify clusters based on a nearest neighbors graph constructed from the profiles of each single cell. Therefore, it is clear that improving the identification of cell similarities might result in better clustering performances. To this end, we assessed the effectiveness of all tested methods in enhancing cell similarity with respect to both simulated and real data.

In Figure 5, we show the difference between the average silhouette coefficient computed on denoised expression matrix and that obtained from the NEM, by grouping single cells according to the GT labels. Higher values of the average Silhouette coefficient indicate that cells are close to other cells of the same subpopulation and separated from those belonging to other subpopulations. In particular, GT labels are provided by cell subpopulation labels for simulated data and by cell type/line labels for real-world datasets (see the Methods section and the Supplementary Material for further details). We remark that the silhouette coefficient allows one not to rely on arbitrarily chosen clustering approaches, to evaluate the correct grouping of single cells. In fact, currently available clustering methods for scRNA-seq data are characterized by different properties, goals and specifications and produce results that are extremely sensitive to parameter choices and variations, and which might, in turn, undermine the comparison of denoising and imputation methods on this specific task.

Results are shown for simulated datasets with {500, 2000, 10000} genes and 100 (non-UMI full-length) or {3000, 10000} single cells (UMI), as well as for real-world datasets **RW-D**#1, **RW-D**#2 and **RW-D**#3. Note that we employed the TEP of all cell subpopulations as benchmark for the assessment on simulated datasets: in particular, the silhouette coefficient delta between the TEP and the NEM represents the largest theoretical improvement in each setting.

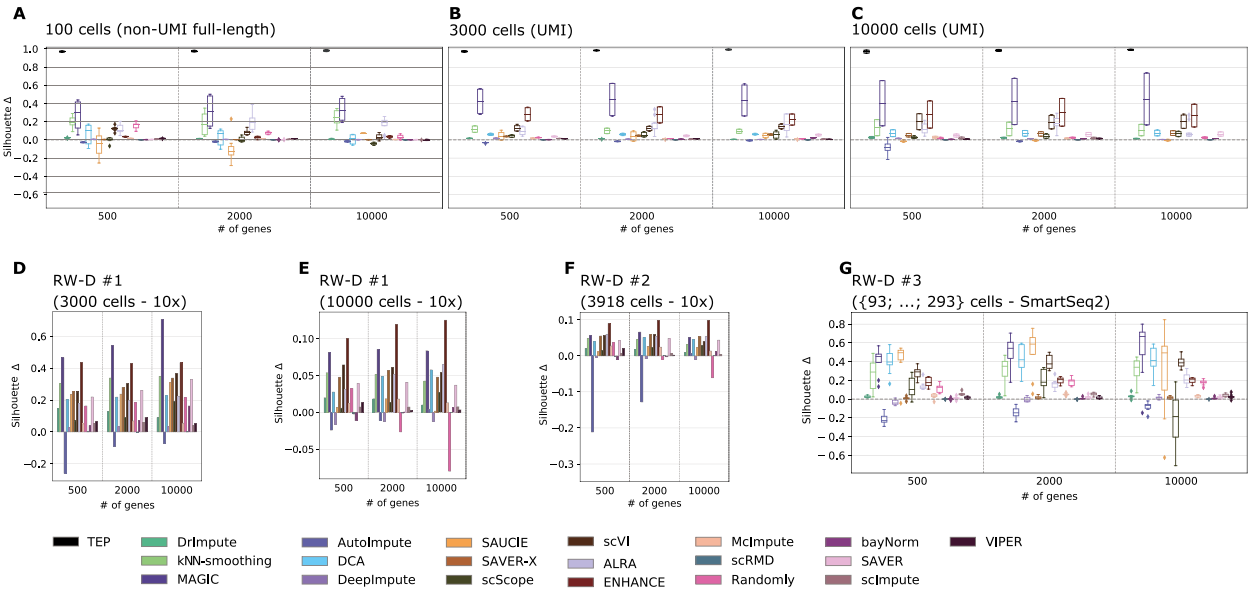


Figure 5. Performance assessment on cell similarity characterization (simulations and real-world data). Assessment of enhancement of cell similarity characterization after denoising, as evaluated on (i) simulated datasets (non-UMI full-length with 100 single cells and UMI-simulated datasets with {3000, 10000} single cells, panels A–C) and (ii) real-world datasets RW-D#1 (downsampled to {3000, 10000} cells, 10x platform, panels D and E), RW-D#2 (3918 cells, 10x platform, panel F) and RW-D#3 ({93, 196, 197, 224, 243, 245, 263, 275, 293} cells, Smart-Seq2, panel G). The boxplots (respectively, barplots) in all panels, depict the distribution (respectively, values) of the Silhouette delta, i.e. the difference between the average silhouette coefficient computed on the denoised expression matrix and that computed on the NEM, for all methods. The difference between the average silhouette coefficient evaluated on the TEP and that computed on the NEM is also shown for all simulated datasets.

Overall, most methods cause an increase of the average silhouette coefficient in most settings, suggesting that imputation and denoising approaches are indeed effective in enhancing the similarity of the expression profiles of cells belonging to the same sub-populations.

This effect is significantly intensified with datasets with larger sample size and generated (or simulated) with UMI protocols, as proven by the overall increase in delta magnitude. In particular, MAGIC and ENHANCE appear to produce the best results, with respect to both simulated and real-world datasets, yet with noteworthy variance in some scenarios, and with the latter method improving its performance with UMI datasets. We further notice that ALRA, kNN-smoothing and scVI deliver notable performances in most scenarios, closely followed by DCA. Surprisingly, SAUCIE exhibits a negative delta with simulated non-UMI full-length datasets but produces good results with real-world Smart-Seq2 dataset RW-D#3.

We recall that, among the best performing methods for the imputation and expression recovery tasks (see above), in addition to the aforementioned MAGIC and ENHANCE, SAVER-X and SAVER consistently produce improvements of the average silhouette delta in most simulated and real-world scenarios, whereas bayNorm and DrImpute appear to be less effective with respect to this specific task.

We finally specify that the results on simulated and real-world datasets are mostly coherent across experimental scenarios, further proving the suitability of simulations in assessing the performance of imputation and denoising methods.

Identification of DEGs (real-world data)

In order to quantify the effect of denoising and imputation methods on the identification of DEGs, we leveraged on bulk RNA-sequencing data included in real-world dataset RW-D#4 [93]. In detail, we first computed the DEGs between the parental and resistant samples included in the dataset, with respect to both the original expression matrix and the denoised matrix (via Wilcoxon test, $P < 0.05$), and which resulted in two distinct lists of DEGs. The analysis was repeated for both the Fluidigm/Smart-Seq dataset (84 and 113 single cells for resistant and parental cell lines, respectively) and the 10x datasets (3085 and 3178; see the Methods section and the Supplementary Material section 4 for further details).

In Figure 6, we display the difference of the Spearman correlation coefficient between the expression profile of the DEGs obtained from the denoised expression matrix and the bulk expression profile (computed for each single cell), and the one computed on the profiles of DEGs determined from the original expression matrix.

Noteworthy, most approaches produce an increase of the correlation with respect to the bulk expression profile. In particular, kNN-smoothing, MAGIC and SAUCIE deliver a median Spearman delta > 0.10 for both the Fluidigm/Smart-Seq and the 10x datasets, while bayNorm, ENHANCE, SAVER, SAVER-X and scVI show a median Spearman delta > 0.10 for the latter protocol only.

Overall, this result indicates that, in many cases, imputation and denoising methods might be effective in improving downstream analyses, such as the identification of DEGs.

Computation time (simulations)

Figure 7 reports the results of the computational time assessment on three simulated datasets: (i) non-UMI full-length (100 cells) (ii) UMI (3000 cells), and (iii) UMI (10 000 cells), with respect to {500, 2000, 5000} genes, plotted in logarithmic scale.

We can observe that all methods suffer an approximately exponential increase of computational time with respect to the number of cells and the number of genes, with extremely significant difference in magnitude. Overall, the most scalable

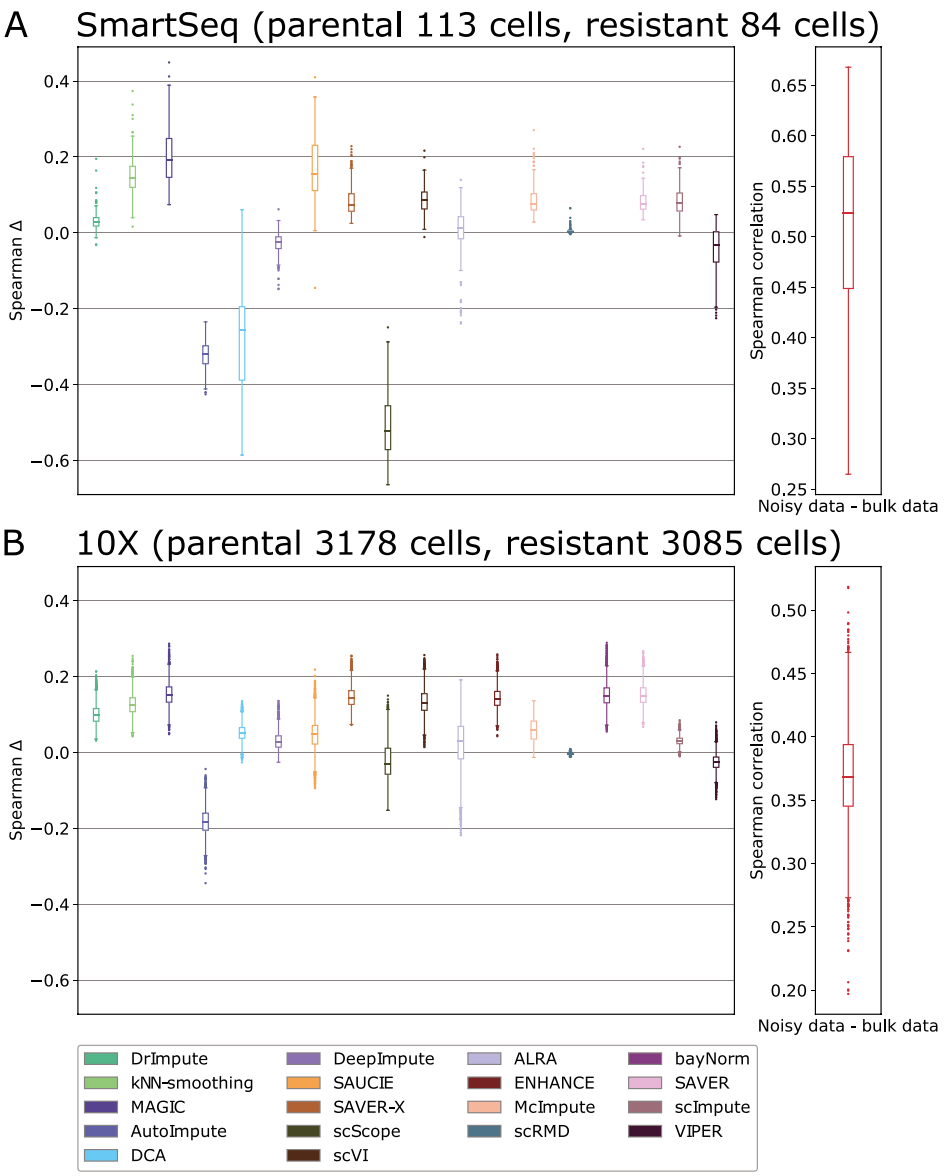


Figure 6. Performance assessment on identification of DEGs (real-world data). Assessment of identification of DEGs, as computed on **RW-D**#4 [93]. DEGs between parental and resistant cell lines of RW-D#4 are identified via Wilcoxon test ($P < 0.05$), both starting from the original scRNA-seq dataset and from the corresponding denoised matrices, for both Fluidigm/Smart-Seq and 10x datasets (panel A and B). The Spearman correlation coefficient between the expression profile of all single cells and the corresponding bulk expression profile is computed with respect to all the DEGs included in the distinct lists. The distribution (on all single cells) of the difference between the Spearman correlation coefficient computed with original data matrix and that computed with the denoised version is then shown as boxplots for both 10x and Fluidigm/Smart-Seq datasets. In the rightmost panels, the baseline distribution of the Spearman correlation coefficient between the NEM and bulk data (with respect to the corresponding list of DEGs) is shown, for both scenarios.

algorithms appear to be ALRA, kNN-smoothing and scRMD while, in general, matrix theory appears to be the most computationally efficient category.

Summary of the performance assessment on denoising and imputation methods

In Figure 8, we present a recapitulation of the performance assessment. The schema includes seven panels, structured as follows:

- imputation of dropout events,
- recovery of gene expression profiles,
- characterization of cell similarity,
- identification of DEGs,
- computation time,
- task,
- release code quality.

In particular, we selected a subset of simulated datasets, characterized by selected parameter settings in terms of single-cell number ($\{100, 3000, 10000\}$), sequencing protocol {non-UMI full-length, UMI} and number of genes (2000 for all settings)—and all four real-world datasets (see the Methods section), which we employed to compute a schematic ranking of all methods with respect to the distinct tasks.

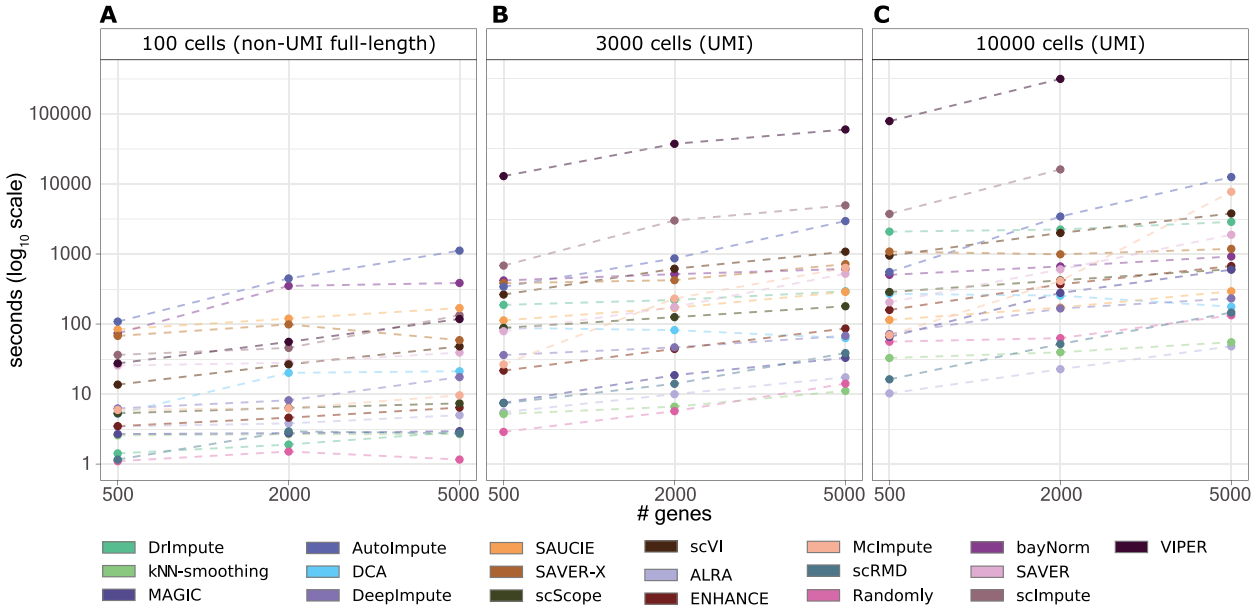


Figure 7. Computational time assessment. Running time of each method in denoising/imputing datasets with increasing number of cells and genes. In **(A)** results with 100 cells, in **(B)** results with 3000 cells and in **(C)** results with 10 000 cells. Values are plotted in logarithmic scale.

More in detail, for each selected parameter setting of the simulated dataset and for each real-world dataset, we ordered all 19 methods with respect to the average values of the following metrics:

(i) average Spearman correlation delta for zero entries of the NEM (for imputation of dropout events),
(ii) average Spearman correlation delta on the whole expression matrix (for recovery of gene expression profiles),
(iii) average silhouette delta (for characterization of cell similarity),
(iv) average Spearman correlation delta (for identification of DEGs),
(v) computation time.

The ranking is visually represented with dots with respect to each experimental setting, where the largest dot corresponds to the best performing method (green) and the smallest dot to the worst performing method (red).

The task panel indicates whether each method performs either denoising or imputation (see the Introduction section and Supplementary Material section 1 for a rigorous classification of the two tasks). Finally, the last panel reports a summary of selected quality code metrics, which were used to evaluate the different tools. In particular, usability and documentation range from 1, i.e. the worst result, to 4, corresponding to the best score. Usability is calculated by considering a set of characteristics that contribute in worsening the overall usability of the tool: (i) either input preprocessing, preliminary operation, e.g. clustering, or output post-processing, e.g. re-normalization, are required to the user; (ii) at least one parameter depends on the input, i.e. a grid-search is required; (iii) parameters meaning is not intuitive, e.g. it has no biological meaning; (iv) the tool is not available on a package distribution platform, e.g. `Bioconductor` or `pip`/`conda`. If a tool has none of the previously introduced features is assigned to the maximum score of 4; otherwise, the scoring is reduced to a minimum of 1. Documentation score is assigned as follows: 1 indicates that the authors did provide neither a documentation nor a detailed tutorial, 2 indicates that the authors provided a tutorial but did not write a detailed explanation for the parameters, 3 indicates that a detailed tutorial is available and 4 indicates that the authors provided both a detailed tutorial and a full explanation of all parameters. Finally, we indicate both whether the program is maintained, i.e. updated in the past 2 years, and the programming language on which the tool was implemented.

Discussion

We presented a review of the current state-of-the art of computational approaches for denoising and imputation of scRNA-seq data. Extensive tests on both real and synthetic datasets allowed to evaluate the performances and the robustness of each method under different experimental scenarios.

In light of the presented results, distinct methods appear to be more suitable for different tasks. In particular, ENHANCE, MAGIC, SAVER, and SAVER-X provide the best overall compromise and show robust performances with respect to all considered tasks. In addition to such methods, bayNorm and DrImpute are especially effective in recovering the true expression profiles and imputing dropout entries, while kNN-smoothing and scVI in improving the characterization of cell similarity and the grouping of single cells in coherent subpopulations, as well as the identification of DEGs.

We also note that, as expected, most methods appear to struggle with non-UMI full-length datasets, likely due to the low number of observations (cells) as compared with the high number of variables (genes). Furthermore, as already mentioned and as reported in [110], denoised expression values returned by any method should be considered with caution, due to the presence of possible artifacts, as proven by the low correlation with GT expression profiles from simulations recorded in many cases and, particularly, with non-UMI full-length datasets.

By focusing on machine learning frameworks, we notice that methods that employ assumptions on biological variability and technical noise (i.e. DCA, SAVER-X, scVI) typically exhibit

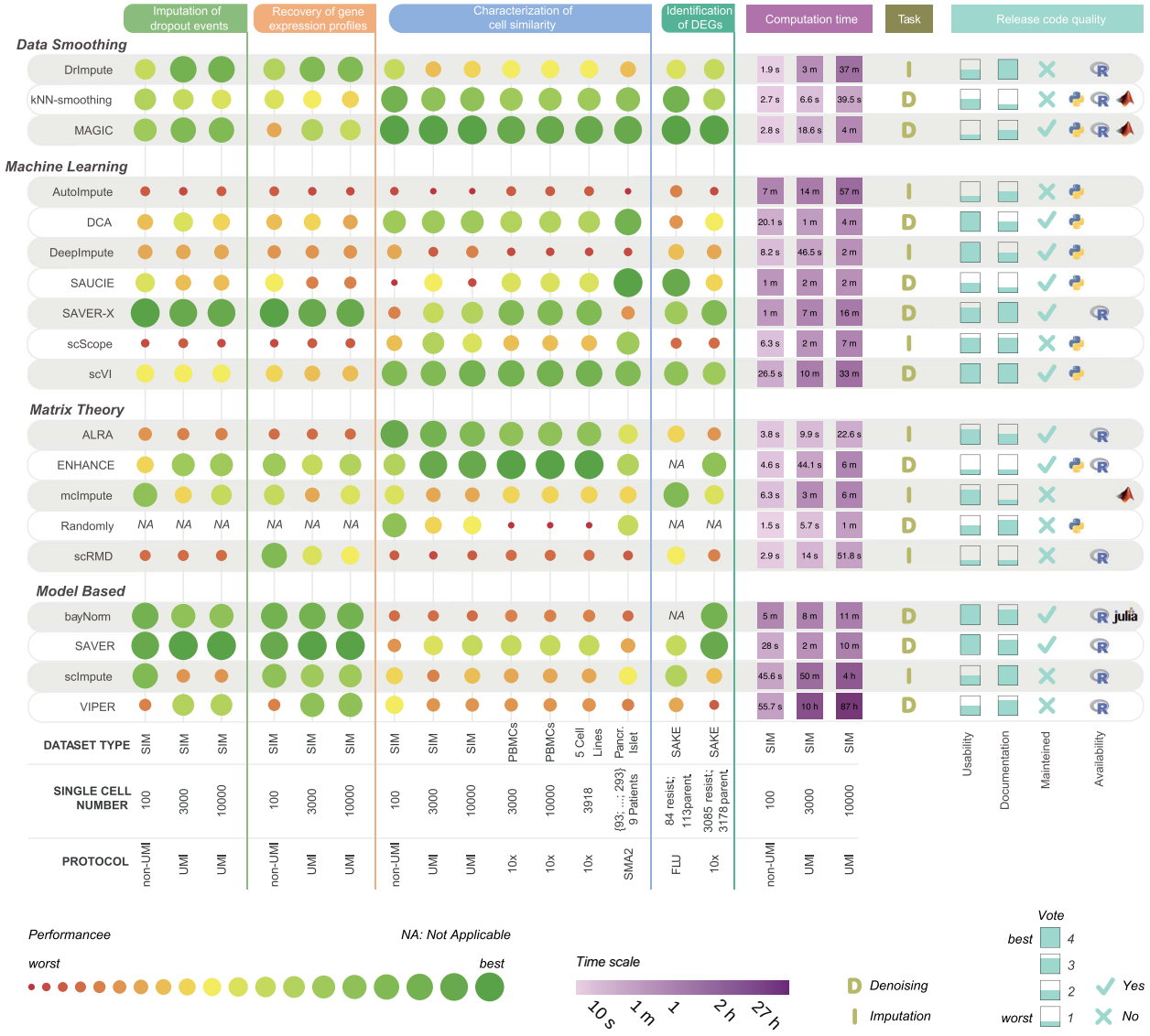


Figure 8. Summary of the performance assessment on denoising and imputation methods. The five leftmost panels report a schematic ranking of all 19 tested denoising and imputation methods, as computed on a selected panel of synthetic and real-world datasets, in terms of average Spearman correlation delta for zero entries of the NEM (for imputation of dropout events), average Spearman correlation delta on the whole expression matrix (for recovery of gene expression profiles), average silhouette delta (characterization of cell similarity), average Spearman correlation delta (identification of DEGs) and computation time. The size of each round marker is proportional to the ranking, with the largest (green) dots corresponding to the best performing tool and the smallest (red) dots to worst performing tool, with respect to the considering metric. The task panel indicates whether the method can perform either denoising or imputation tasks. Finally, the rightmost panel reports a summary of a quality code metrics that were used to evaluate the different tools in terms of usability, documentation, maintenance and availability (please refer to the Methods section (Summary of the performance assessment on denoising and imputation Methods) for further details).

better performances, hinting at the importance of including prior knowledge to inform the learning algorithms. Model-based methods present a typically good performance in both imputation and expression recovery, yet at a usually high computational cost, and generally showing suboptimal performance in cell similarity enhancement. Matrix theory-based techniques show good performance in terms of characterization of cell similarity, in addition to noteworthy scalability, even with large datasets. Finally, data smoothing approaches present typically good performances, yet with significant differences according to the specific task.

All in all, the performance of all methods appear to be highly dependent on the specific features of the dataset, as very distinct results are observed for the same method in different experimental scenarios, as recapped in Figure 8. This summary should guide potential users in selecting an optimal method according to the research needs and the available data types.

We further note that a review on a similar subject can be found as a preprint in [35]. Despite achieving similar conclusions on several methods included in our review, such work does not include comparisons on simulated data, which allow to evaluate a number of metrics with respect to the GT. For instance, certain methods that were identified as highly performing in [35], appear to struggle in dealing with true expression profiles recovery, an effect that can be evaluated only via simulations. The virtually unlimited number of in silico scenarios that can

be generated via methods such SymSim [89] suggests that simulations should be increasingly used to quantitatively assess the performance of data science methods and especially to test the robustness of their results.

Possible limitations of our assessment might be related to the application of most methods with default parameters, while one can expect improvements when fine tuning the parameters. In this respect, setting guidelines provided by the authors were followed when present and appear to be extremely beneficial to increase the overall usability and performance of the methods.

We also recall that for some methods, such as those based on AEs, it would be possible to use the latent variable space to perform single-cell clustering, while in our analysis we chose to use the denoised expression profiles, to provide a fair comparison for all methods.

We finally remark that scalable methods for denoising of single-cell transcriptomic data might pave the way for refined downstream analyses, for instance, by improving the reliability and accuracy of variant calling pipelines from scRNA-seq data to provide an accurate mapping of genotype and phenotype of single cells [111, 112], as well as by allowing a better estimation of metabolic fluxes from scRNA-seq data in the investigation of cancer metabolism [113, 114].

Key Points

- Extensive tests on synthetic and real datasets provide a quantitative assessment of the performance of denoising and imputation methods in distinct scenarios.
- Some methods are effective in improving the characterization of cell similarity, some others in recovering the true gene expression profiles and imputing dropouts.
- Appropriate assumptions on the noise model are beneficial to recover lost information.
- Overall, ENHANCE, MAGIC, SAVER and SAVER-X constitute a good compromise on all tasks.
- Corrected expression values returned by any method should be considered with caution in downstream analyses.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Data availability

The source code used to replicate all our analyses, including synthetic and real datasets, is available at this link: https://github.com/BIMIB-DISCo/review-scRNA-seq-DENOISING.

Acknowledgments

We thank Chiara Damiani, Daniele Ramazzotti and Giulio Caravagna for helpful discussions.

Funding

This work was supported by the Cancer Research UK and Associazione Italiana per la Ricerca sul Cancro (CRUK/AIRC) "Accelerator Award" (award #22790) 'Single-cell Cancer Evolution in the Clinic'. Partial support was also provided by the Italian node of the Elixir network (https://elixir-europe.org/about-us/who-we-are/nodes/italy) and the SysBioNet project, a Ministero dell'Istruzione, dell'Universitá e della Ricerca initiative for the Italian Roadmap of European Strategy Forum on Research Infrastructures.

References

1. Dalerba P, Kalisky T, Sahoo D, *et al*. Single-cell dissection of transcriptional heterogeneity in human colon tumors. *Nat Biotechnol* 2011;**29**(12):1120.
2. Vieth B, Parekh S, Ziegenhain C, *et al*. A systematic evaluation of single cell RNA-seq analysis pipelines. *Nat Commun* 2019;**10**(1):1–11.
3. Angela RW, Norma F, Neff TK, *et al*. Quantitative assessment of single-cell RNA-sequencing methods. *Nat Methods* 2014;**11**(1):41.
4. Kalisky T, Blainey P, Quake SR. Genomic analysis at the single-cell level. *Annu Rev Genet* 2011;**45**:431–45.
5. Huang S. Non-genetic heterogeneity of cells in development: more than just noise. *Development* 2009;**136**(23):3853–62.
6. Li L, Clevers H. Coexistence of quiescent and active adult stem cells in mammals. *Science* 2010;**327**(5965):542–5.
7. Shalek AK, Satija R, Shuga J, *et al*. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* 2014;**510**(7505):363–9.
8. Hwang B, Lee JH, Bang D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med* 2018;**50**(8):1–14.
9. AlJanahi AA, Danielsen M, Dunbar CE. An introduction to the analysis of single-cell RNA-sequencing data. *Mol Ther Methods Clin Dev* 2018;**10**:189–96.
10. Lawson DA, Kessenbrock K, Davis RT, *et al*. Tumour heterogeneity and metastasis at single-cell resolution. *Nat Cell Biol* 2018;**20**(12):1349–60.
11. Shaffer SM, Dunagin MC, Torborg SR, *et al*. Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance. *Nature* 2017;**546**(7658):431.
12. Cao J, Packer JS, Ramani V, *et al*. Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* 2017;**357**(6352):661–7.
13. Regev A, Teichmann SA, Lander ES, *et al*. Science forum: the human cell atlas. *elife* 2017;**6**:e27041.
14. Elowitz MB, Levine AJ, Siggia ED, *et al*. Stochastic gene expression in a single cell. *Science* 2002;**297**(5584):1183–6.
15. Marinov GK, Williams BA, McCue K, *et al*. From single-cell to cell-pool transcriptomes: stochasticity in gene expression and RNA splicing. *Genome Res* 2014;**24**(3):496–510.
16. Haque A, Engel J, Teichmann SA, *et al*. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med* 2017;**9**(1):75.
17. Ziegenhain C, Vieth B, Parekh S, *et al*. Comparative analysis of single-cell RNA sequencing methods. *Mol Cell* 2017;**65**(4):631–43.
18. Macosko EZ, Basu A, Satija R, *et al*. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* 2015;**161**(5):1202–14.
19. Klein AM, Mazutis L, Akartuna I, *et al*. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* 2015;**161**(5):1187–201.

20. Fraction of mRNA transcripts captured per cell. https://kb.10xgenomics.com/hc/en-us/articles/360001539051-What-fraction-of-mRNA-transcripts-are-captured-per-cell.
21. Ramsköld D, Luo S, Wang Y-C, *et al*. Full-length mRNA-seq from single-cell levels of RNA and individual circulating tumor cells. *Nat Biotechnol* 2012;**30**(8):777.
22. Sheng K, Cao W, Niu Y, *et al*. Effective detection of variation in single-cell transcriptomes using MATQ-seq. *Nat Methods* 2017;**14**(3):267–70.
23. Jaitin DA, Kenigsberg E, Keren-Shaul H, *et al*. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* 2014;**343**(6172):776–9.
24. Hashimshony T, Wagner F, Sher N, *et al*. CEL-Seq: single-cell RNA-Seq by multiplexed linear amplification. *Cell Rep* 2012;**2**(3):666–73.
25. Rosenberg AB, Roco CM, Muscat RA, *et al*. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* 2018;**360**(6385):176–82.
26. Pollen AA, Nowakowski TJ, Shuga J, *et al*. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat Biotechnol* 2014;**32**(10):1053.
27. Gierahn TM, Wadsworth MH, II, Hughes TK, *et al*. Seq-well: portable, low-cost RNA sequencing of single cells at high throughput. *Nat Methods* 2017;**14**(4):395–8.
28. Islam S, Zeisel A, Joost S, *et al*. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nat Methods* 2014;**11**(2):163.
29. Ziegenhain C, Vieth B, Parekh S, *et al*. Comparative analysis of single-cell RNA sequencing methods. *Mol Cell* 2017;**65**(4):631–43.
30. Haque A, Engel J, Teichmann SA, *et al*. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med* 2017;**9**(1):75.
31. Goh WWB, Wang W, Wong L. Why batch effects matter in omics data, and how to avoid them. *Trends Biotechnol* 2017;**35**(6):498–507.
32. Tung P-Y, Blischak JD, Hsiao CJ, *et al*. Batch effects and the effective design of single-cell gene expression studies. *Sci Rep* 2017;**7**:39921.
33. Tran HTN, Ang KS, Chevrier M, *et al*. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol* 2020;**21**(1):12.
34. Zhu L, Lei J, Devlin B, *et al*. A unified statistical framework for single cell and bulk RNA sequencing data. *Ann Appl Stat* 2018;**12**(1):609.
35. Hou W, Ji Z, Ji H, *et al*. A systematic evaluation of single-cell RNA-sequencing imputation methods. *bioRxiv* 2020. doi: 10.1101/2020.01.29.925974.
36. Agarwal D, Wang J, Zhang NR, *et al*. Data denoising and post-denoising corrections in single cell RNA sequencing. *Stat Sci* 2020;**35**(1):112–28.
37. Lähnemann D, Köster J, Szczurek E, *et al*. Eleven grand challenges in single-cell data science. *Genome Biol* 2020;**21**(1):1–35.
38. Gong W, Kwak I-Y, Pota P, *et al*. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC Bioinformatics* 2018;**19**(1):220.
39. Tjaernberg A, Mahmood O, Jackson CA, *et al*. Optimal tuning of weighted kNN- and diffusion-based methods for denoising single cell genomics data. *bioRxiv* 2020. doi: 1101/2020.02.28.970202.
40. Ye P, Ye W, Ye C, *et al*. scHinter: imputing dropout events for single-cell RNA-seq data with limited sample size. *Bioinformatics* 2020;**36**(3):789–97.
41. Wagner F, Yan Y, Yanai I. K-nearest neighbor smoothing for high-throughput single-cell RNA-seq data. *bioRxiv* 2018. doi: 10.1101/217737.
42. Moussa M, Măndoiu II. Locality sensitive imputation for single cell RNA-seq data. *J Comput Biol* 2019;**26**(8):822–35.
43. Van Dijk D, Sharma R, Nainys J, *et al*. Recovering gene interactions from single-cell data using data diffusion. *Cell* 2018;**174**(3):716–29.
44. Ronen J, Akalin A. netSmooth: network-smoothing based imputation for single cell RNA-seq. *F1000Res* 2018; 7:8.
45. Jeong H, Liu Z. PRIME: a probabilistic imputation method to reduce dropout effects in single cell RNA sequencing. *Bioinformatics* 2020;**36**(13):4021–9.
46. Tracy S, Yuan G-C, Dries R. RESCUE: imputing dropout events in single-cell RNA-sequencing data. *BMC Bioinformatics* 2019;**20**(1):388.
47. Wu W, Dai Q, Liu Y, *et al*. G2S3: a gene graph-based imputation method for single-cell RNA sequencing data. *bioRxiv* 2020. doi: 10.1101/2020.04.01.020586.
48. Jin K, Ou-Yang L, Zhao X-M, *et al*. scTSSR: gene expression recovery for single-cell RNA sequencing using two-side sparse self-representation. *Bioinformatics* 2020;**36**(10):3131–8.
49. Leote AC, Wu X, Beyer A. Network-based imputation of dropouts in single-cell RNA sequencing data. *bioRxiv* 2019. doi: 10.1101/611517.
50. Elyanow R, Dumitrascu B, Engelhardt BE, *et al*. netNMF-sc: leveraging gene–gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome Res* 2020;**30**(2):195–204.
51. Wang J, Agarwal D, Huang M, *et al*. Data denoising with transfer learning in single-cell transcriptomics. *Nat Methods* 2019;**16**(9):875–8.
52. Peng T, Zhu Q, Yin P, *et al*. SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data. *Genome Biol* 2019;**20**(1):88.
53. Ye W, Ji G, Ye P, *et al*. scNPF: an integrative framework assisted by network propagation and network fusion for preprocessing of single-cell RNA-seq data. *BMC Genomics* 2019;**20**(1):347.
54. Badsha MB, Li R, Liu B, *et al*. Imputation of single-cell gene expression with an autoencoder neural network. *Quant Biol* 2020;1–17.
55. Zhu L, Lei J, Devlin B, *et al*. A unified statistical framework for single cell and bulk RNA sequencing data. *Ann Appl Stat* 2018;**12**(1):609.
56. Talwar D, Mongia A, Sengupta D, *et al*. AutoImpute: autoencoder based imputation of single-cell RNA-seq data. *Sci Rep* 2018;**8**(1):1–11.
57. Arisdakessian C, Poirion O, Yunits B, *et al*. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biol* 2019;**20**(1):1–14.
58. Eraslan G, Simon LM, Mircea M, *et al*. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* 2019;**10**(1):390.
59. Zhang X-F, Ou-Yang L, Yang S, *et al*. EnImpute: imputing dropout events in single-cell RNA-sequencing data via ensemble learning. *Bioinformatics* 2019;**35**(22):4827–9.

60. Rao J, Zhou X, Lu Y, *et al.* Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. *bioRxiv* 2020. doi: 10.1101/2020.02.05.935296.
61. Xu Y, Zhang Z, You L, *et al.* scIGANs: single-cell RNA-seq imputation using generative adversarial networks. *bioRxiv* 2020. doi: 10.1101/2020.01.20.913384.
62. Amodio M, Van Dijk D, Srinivasan K, *et al.* Exploring single-cell data with deep multitasking neural networks. *Nat Methods* 2019;**62**:1139–45.
63. Deng Y, Bao F, Dai Q, *et al.* Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nat Methods* 2019;**16**(4):311–4.
64. Lopez R, Regier J, Cole MB, *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018;**15**(12):1053–8.
65. Mehtonen J, González G, Kramer R, *et al.* Semisupervised generative autoencoder for single-cell data. *J Comput Biol* 2019;**27**(8):1190–203.
66. Zhu K, Anastassiou D. 2DImpute: imputation in single-cell RNA-seq data from correlations in two dimensions. *Bioinformatics* 2020;**36**(11):3588–9.
67. Tran B, Tran D, Nguyen H, *et al.* Ria: a novel regression-based imputation approach for single-cell RNA sequencing. In: *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, 2019. pp. 1–9. New York City, NY, USA: IEEE.
68. Linderman GC, Zhao J, Kluger Y. Zero-preserving imputation of scRNA-seq data using low-rank approximation. *bioRxiv* 2018. doi: 10.1101/397588.
69. Wagner F, Barkley D, Yanai I. Accurate denoising of single-cell RNA-seq data using unbiased principal component analysis. *bioRxiv* 2019;**655365**. doi: 10.1101/655365.
70. Chen C, Wu C, Wu L, *et al.* scRMD: imputation for single cell RNA-seq data via robust matrix decomposition. *Bioinformatics* 2020;**36**(10):3156–61. 03.
71. Xu J, Cai L, Liao B, *et al.* CMF-Impute: an accurate imputation tool for single-cell RNA-seq data. *Bioinformatics* 2020;**36**(10):3139–47.
72. Mongia A, Sengupta D, Majumdar A. deepMc: deep matrix completion for imputation of single-cell RNA-seq data. *J Comput Biol* 2019;**27**(7):1011–9.
73. Mongia A, Sengupta D, Majumdar A. McImpute: matrix completion based imputation for single cell RNA-seq data. *Front Genet* 2019;**10**:9.
74. Zhang L, Zhang S. PBLR: an accurate single cell RNA-seq data imputation tool considering cell heterogeneity and prior expression level of dropouts. *bioRxiv* 2018. doi: 10.1101/379883.
75. Hu Y, Li B, Liu N, *et al.* WEDGE: recovery of gene expression values for sparse single-cell RNA-seq datasets using matrix decomposition. *bioRxiv* 2019;864488. https://doi.org/10.1101/864488.
76. Pierson E, Yau C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol* 2015;**16**(1):241.
77. Aparicio L, Bordyuh M, Blumberg AJ, *et al.* A random matrix theory approach to denoise single-cell data. *Patterns* 2020;;**1**(3):100035.
78. Tang W, Bertaux F, Thomas P, *et al.* bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics* 2020;**36**(4):1174–81.
79. Azizi E, Prabhakaran S, Carr A, *et al.* Bayesian inference for single-cell clustering and imputing. *Genomics Comput Biol* 2017;**3**(1):e46–6.
80. Song F, Chan GM, Wei Y. Flexible experimental designs for valid single-cell RNA-sequencing experiments allowing batch effects correction. *Nat Commun* 2020;**11**:3274.
81. Lin P, Troup M, Ho JWK. CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol* 2017;**18**(1):59.
82. Yang MQ, Weissman SM, Yang W, *et al.* MISC: missing imputation for single-cell RNA sequencing data. *BMC Syst Biol* 2018;**12**(7):114.
83. Huang M, Wang J, Torre E, *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nat Methods* 2018;**15**(7):539.
84. Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat Commun* 2018;**9**(1):997.
85. Miao Z, Li J, Zhang X. scRecover: discriminating true and false zeros in single-cell RNA-seq data for imputation. *bioRxiv* 2019;665323. https://doi.org/10.1101/665323.
86. Zhang Y, Liang K, Liu M, *et al.* SCRIBE: a new approach to dropout imputation and batch effects correction for single-cell RNA-seq data. *bioRxiv* 2019;793463. https://doi.org/10.1101/793463.
87. Hu Z, Songpeng Z, Liu JS. SIMPLEs: a single-cell RNA sequencing imputation strategy preserving gene modules and cell clusters variation. *bioRxiv* 2020. doi: 10.1101/2020.01.13.904649.
88. Chen M, Zhou X. VIPER: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. *Genome Biol* 2018;**19**(1):1–15.
89. Zhang X, Xu C, Yosef N. Simulating multiple faceted variability in single cell RNA sequencing. *Nat Commun* 2019;**10**(1):2611.
90. Zheng GXY, Terry JM, Belgrader P, *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**(1):1–12.
91. Tian L, Dong X, Freytag S, *et al.* Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat Methods* 2019;**16**(6):479–87.
92. Segerstolpe Å, Palasantza A, Eliasson P, *et al.* Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab* 2016;**24**(4):593–607.
93. Ho Y-J, Anaparthy N, Molik D, *et al.* Single-cell RNA-seq analysis identifies markers of resistance to targeted BRAF inhibitors in melanoma cell populations. *Genome Res* 2018;**28**(9):1353–63.
94. Andrews TS, Hemberg M. False signals induced by single-cell imputation. *F1000Res* 2018;**7**:1740.
95. Zhang L, Zhang S. Comparison of computational methods for imputing single-cell RNA-sequencing data. *IEEE/ACM Trans Comput Biol Bioinform* 2018;**17**(2):376–89.
96. Coifman RR, Lafon S. Diffusion maps. *Appl Comput Harmon Anal* 2006;**21**(1):5–30.
97. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge, MA, USA: The MIT Press, 2016.
98. Wang Y-X, Zhang Y-J. Nonnegative matrix factorization: a comprehensive review. *IEEE Trans Knowl Data Eng* 2012;**25**(6):1336–53.
99. Candès EJ, Recht B. Exact matrix completion via convex optimization. *Found Comput Math* 2009;**9**(6):717.
100. Eckart C, Young GM. The approximation of one matrix by another of lower rank. *Psychometrika* 1936;**1**:211–8.
101. Sun Y, Babu P, Palomar DP. Majorization-minimization algorithms in signal processing, communications, and

machine learning. *IEEE Trans Signal Process* 2016;**65**(3):816–794.
102. Livan G, Novaes M, Vivo P. *Introduction to Random Matrices: Theory and Practice*, Vol. 26. London, UK: Springer, 2018.
103. Hsu D. Kakade SM, Zhang T. Robust matrix decomposition with sparse corruptions. *IEEE Trans Inf Theory* 2011;**57**(11):7221–34.
104. Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, 2002. pp. 849–56. Vancouver, BC, Canada: MIT press.
105. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.
106. van der Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.
107. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018; **19**(1):15.
108. Butler A, Hoffman P, Smibert P, *et al*. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;**36**(5):411.
109. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* 2019;**9**(1):1–12.
110. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol Syst Biol* 2019;**15**(6):e8746.
111. Ramazzotti D, Angaroni F, Maspero D, *et al*. Longitudinal cancer evolution from single cells. *bioRxiv* 2020. doi: 10.1101/2020.01.14.906453.
112. Zhou Z, Xu B, Minn A, *et al*. DENDRO: genetic heterogeneity profiling and subclone detection by single-cell RNA sequencing. *Genome Biol* 2020;**21**(1):1–15.
113. Damiani C, Maspero D, Di Filippo M, *et al*. Integration of single-cell RNA-seq data into population models to characterize cancer metabolism. *PLoS Comput Biol* 2019;**15**(2):e1006733.
114. Graudenzi A, Maspero D, Damiani C. FBCA, a multiscale modeling framework combining cellular automata and flux balance analysis. *J Cell Autom* 2020;**15**:75–95.

.

3.2 [DNA]Inference of clonal trees from single-cell mutational profiles

3.2.1 Introduction

Phylogenetics. A phylogenetic tree is a graph that describes the pattern of descent among individuals (e.g., species, cancer clones, etc.) by considering their similarities in terms of phenotypic or genetic characteristics. This graph in typically a tree, where the root corresponds to the common ancestor of the population under study, and it contains those features that are common to all individuals. The leaves in the tree are the observed individuals and the internal nodes correspond to unobserved events, as each node is the unobserved most recent common ancestor of all its descendants. Each branch in the tree corresponds to the accumulation of one or more events that distinguish the child from the parent node. An example is Darwin's *tree of life* [58, 62], which represents the evolution of life and describes the temporal relationships between living and extinct organisms. Reviews for the reconstruction of phylogenetic models are presented in [5, 6].

Cancer phylogenetics. The standard phylogenetic analysis can be applied to cancer data, leveraging DNA sequencing technologies that allow to measure the genomic sequence in biological samples. As described in section 2.1.4.1, cancer is an evolutionary process characterized by the accumulation of mutations, which can correspond to SNVs, more complex Structural Variants (SV) or CNAs. The process is initiated by one cell that acquires an alteration conferring a selective advantage with respect to the rest of the population, which gets propagated to its successors. Disease progression leads to the emergence, competition and (positive/negative) selection of genetically distinct subpopulations of cells called *clones* [3], as cancer evolves into multiple heterogeneous subpopulations that accumulate different mutations.
Thus, we can distinguish between clonal events that are acquired early in the disease and are present throughout the tumor, and subclonal events that are acquired later on and affect only subsets of the tumor cells. Please note that above we gave a general definition of a clone as a set of genetically distinct cells. However, cancer is the result of a complex interplay between different factors (e.g., genome, chromatin, gene expression), and the exact definition of cancer clone is still subject of debate [138]. Given that during disease progression only a subset of the mutations (the so-called *driver* mutations) confer a selective advantage to cells, and we don't want to consider as a clone a set of cells that is characterized by a distinct mutational pattern but has no biological difference from

the other tumor cells, we can define a clone as a set of cells that are characterized by a growth/survival advantage with respect to the rest of the population [138].

Thus, it is of great interest to reconstruct the evolutionary relationship between clones that emerge during disease progression. We can identify three distinct tasks: (i) detecting which mutations are present in cells, (ii) detecting the clonal mutations and identifying the subclones present in the sample and (iii) reconstructing the evolutionary history of the tumor. The types of alterations specified in the previous section (SNV, CNV and SV) need specific computational tools to be detected in each sample, and CNVs require specific frameworks to model their evolution [106, 42, 71], which are outside the scope of this work. Thus, throughout this chapter we will refer to works that deal with SNVs and SV, and we use the term *mutation* to indicate these two types of mutations. Task (i) is carried out exploiting technologies for DNA sequencing to get the genomic sequence from the biological sample and using computational methods to compare the obtained sequence with a reference for detecting SNVs and Structural Variants. Many efforts have been recently carried out to reconstruct the clonal structure of a sample and to build models of cancer evolution from available data. Some approaches tackle both the problem of detecting clones in the sample and reconstructing their evolutionary relationships [53] while other works present approaches to reconstruct the clonal structure [138, 44]. The reason because there is great interest for such methods is that by characterizing the evolutionary history of a tumor it is possible to: (*i*) distinguish between types of evolution e.g., linear, branching, neutral, or punctuated [70]; (*ii*) assess the effect of therapies [185]; (*iii*) evaluate the fitness pressure of clones (e.g., selection coefficients or clonal prevalence variation) [178]; (*iv*) assess the presence of preferential temporal ordering (e.g, selective advantage relations)[19, 114, 158, 162]; (*v*) identify prognostic bio-markers [175]; (*vi*) date the key cancer evolution events [143]; (*vii*) investigate the genotype-phenotype relation; (*viii*) predict the possible future evolution of the tumor [36, 113].

Clonal trees Given the final goal of reconstructing a tree describing the ancestral relationship between clones, it is possible to reconstruct two types of trees: phylogenies and clonal trees. On the one hand, the former have been described in the previous paragraph, and are characterised by having observed clones on the leaves in the tree. Clonal trees on the other hand are not characterised by that constraint: edges correspond to ancestral relationships between two clones and they are labeled with the mutations that distinguish the parent from the child [53]. Given one node v, the mutations found on the shortest path between the root and v define the genotype of the corresponding clone, i.e., its set of driver mutations.

3.2.2 Improving the clonal tree inference with COB-tree.

Over the last years different computational approaches have been presented to reconstruct clonal trees from bulk [53] and single cell data [196, 63]. For instance, LACE [196] is a method that leverages single-cell data to reconstruct longitudinal models of cancer evolution, by solving a matrix factorization problem optimizing a weighted log-likelihood through a Markov chain Monte Carlo (MCMC) search schema. The final output of the algorithm is one model describing the evolutionary history of the tumor.
The approximated computational complexity of LACE to reach convergence is $\mathcal{O}(nm^3 \log(m))$, where n is the number of observations and m is the number of mutations. Thus, for large values of n and m the amount of time and computational resources required would become too high to reach convergence. model may fail to reach convergence. In addition to this problem, there may be multiple equivalent solutions that have the same likelihood but correspond to distinct topologies, and by returning only one maximum likelihood solution all the equivalent ones explored during the MCMC search are ignored. Thus, we worked on a method to (i) explore the solution space explored during the MCMC search and (ii) exploit regularities in the search space to return one *Consensus Optimum Branching Tree* (`COB-tree`) that summarises the trees explored during the MCMC. This method is not designed to work strictly with the output of LACE, but it takes in input a list of trees generated with any MCMC based algorithm and it returns the COB-tree solution. In order to assess the performance of our COB-tree algorithm we simulated multiple synthetic datasets with a different number of mutations, and we also present a case study on a real dataset containing longitudinal samples. To explore the solution space we build a 3-dimensional representation that takes into account both the similarity between solutions and their likelihood value: 2 dimensions encode the relative distance among the different trees, and a third dimension corresponds to their Negative Log Likelihood. This work entitled "Exploring the solution space of cancer evolution inference frameworks for single-cell sequencing data" was presented in the 16th International Workshop on Artificial Life and Evolutionary Computation (WIVACE 2022), and it has been accepted for publication in the conference proceedings.

Exploring the solution space of cancer evolution inference frameworks for single-cell sequencing data

Davide Maspero[1,2][0000−0001−8519−4331], Fabrizio Angaroni[2][0000−0002−3375−6686], Lucrezia Patruno[2][0000−0002−3721−9984], Daniele Ramazzotti[3][0000−0002−6087−2666], David Posada[4,5,6][0000−0003−1407−3406], and Alex Graudenzi[2,7][0000−0001−5452−1918]

[1] CNAG-CRG, Centre for Genomic Regulation (CRG), Barcelona Institute of Science and Technology (BIST), Barcelona, Spain
[2] Dept. of Informatics, Systems and Communication, Univ. of Milan-Bicocca, Milan, Italy
[3] Dept. of Medicine and Surgery, Univ. of Milan-Bicocca, Monza, Italy
[4] Centre of Biomedic Investigation (CINBIO), Univ. Vigo, Vigo, Spain
[5] Department of Biochemistry, Genetics, and Immunology, Univ. de Vigo, Vigo, Spain
[6] Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO, Vigo, Spain
[7] Bicocca Bioinformatics, Biostatistics and Bioimaging Centre – B4, Milan, Italy
Correspondence: alex.graudenzi@unimib.it

Abstract. In recent years, many algorithmic strategies have been developed to exploit single-cell mutational profiles generated via sequencing experiments of cancer samples, to return reliable models of cancer evolution. Here, we introduce the COB-tree algorithm, which summarizes the solutions explored by state-of-the-art methods for clonal tree inference, to return a unique consensus optimum branching tree. The method proves to be highly effective in detecting pairwise temporal relations between genomic events, as demonstrated by extensive tests on simulated datasets. We also provide a new method to visualize and quantitatively inspect the solution space of the inference methods, via Principal Coordinate Analysis. Finally, the application of our method to a single-cell dataset of patient-derived melanoma xenografts shows significant differences between the COB-tree solution and the maximum likelihood ones.

Keywords: Cancer evolution · Single-cell sequencing · Markov Chain Monte Carlo

In cancer data science, many efforts are devoted to the design of methods for the reconstruction of cancer evolution models from sequencing data [2, 27, 21, 16, 22, 3]. Indeed, such models are becoming essential to identify possible regularities and repeated evolutionary patterns across tumors, as well as to investigate the impact of therapeutic strategies [6]. In particular, single-cell DNA and RNA sequencing experiments, performed, e.g., on biopsies or on patient-derived models, are a priceless source of high-resolution data on individual tumors.

Despite the typically high levels of noise, mostly due to technical and experimental limitations [20], with such data it is possible to call genomic variants (e.g., single-nucleotide variants, indels, structural alterations, copy-number alterations) via consolidated pipelines [18, 28, 24]. Accordingly, the most widely used methods to reconstruct the evolutionary history of tumors from single-cell mutational profiles rely on robust statistical frameworks to return accurate models by reducing the impact of noise (see [16, 23, 22]).

In this lively field, key attention is devoted to the characterization of the solution space of the inference frameworks. This can be of help both from the theoretical and the application perspectives [27, 1, 8]. In brief, the approximate estimate of the computational complexity of the statistical frameworks proposed in some of the most recent works in the field, namely SCITE [16], LACE [22], VERSO [25], is $\mathcal{O}(nm^3 \log(m))$ to reach MCMC convergence, where n is the number of cells/samples (observations), and m is the number of mutations (variables), as initially discussed in [17]. Thus, for mutation tree reconstructions, it is evident that the complexity mainly depends on the number of mutations included in the final model. Unfortunately, as this number increases (i.e., more mutations are present) it may becomes unfeasible to reach convergence. In addition, in many cases, the algorithm may return equivalent solutions, which share the same likelihood value, but with different topologies.

For these reasons, in this work we aim at: (i) characterizing the space of solutions explored during the MCMC inference of state-of-the-art algorithms for the reconstruction of clonal trees from single-cell mutational profiles; (ii) summarising the collection of solutions, so to return a unique *Consensus Optimum Branching* tree (COB-tree), instead of the Maximum Likelihood one (ML tree). In other words, the goal is to design an algorithm that takes as input a collection of trees sampled during the MCMC of an arbitrary method for clonal tree inference, and exploits the regularities of the solution space to return a unique COB-tree solution.

Note that similar approaches have already been employed in classical phylogenetic studies. For example, BEAST 2 [5] uses the Maximum Clade Credibility method, whereas in [19] the authors propose to employ the Majority Rule. Such approaches could not be directly employed in our analyses due to the intrinsic differences between phylogenetic trees and clonal trees. In particular, the former are binary trees, and thus the number of edges is fixed and depends on the number of samples. This is not valid in clonal trees, where any node could have an arbitrary number of outgoing edges.

We also point out that the opportunity of computing consensus trees in cancer phylogeny is debated. For example, in [1] the authors argue that summary methods returning only a single tree may not accurately represent the topological features of the solution space. By assuming that the solution space is rugged and includes different local minima related to clonal trees displaying distant topologies, the authors suggest to cluster the tree solutions and, successively, apply a summary method for each cluster. Such approach is not computationally feasible for our goal, because the clustering step is limited to a small number of (small) trees, which is orders of magnitudes lower than the trees sampled during an MCMC. We also note that, in their conclusion, the authors state that a proper characterization

of the solution space under an error model of single-cell mutation profiles has not been presented yet.

1 Materials and Methods

COB-tree: a new algorithm for clonal tree inference. In this work, we introduce a new algorithm for the inference of clonal trees from binary mutational profiles. The general idea is to return a unique consensus tree, obtained by exploiting the solutions explored during the MCMC of an arbitrary algorithm for clonal tree reconstruction, such as LACE [22], VERSO [25] or SCITE [16].

In detail, given an edge-weighted digraph in which any weight is the number of times that a given parental relation is returned during the MCMC (i.e., the frequency of such edge as sampled by the MCMC, which underlying its posterior probability), we identify a unique consensus tree by applying algorithms for optimum branching. The outcome COB-tree model includes all nodes connected with a set of edges that maximises the weight sum. To do this, in our case we employed the efficient implementation of Tarjan [29] of the optimum branching tree method originally proposed in [9, 12]. This method is analog to the minimum spanning tree problem, but when considering a directed graph. Note that our algorithm is: (*i*) deterministic; (*ii*) computationally efficient, to handle the vast number of trees sampled during an MCMC; (*iii*) independent of the order of the input tree list.

The algorithmic steps are detailed in the following:

- The COB-tree algorithm takes as input a binary data matrix D, with n rows representing samples (i.e., single cells or biological samples) and m columns representing genomic mutations. Each entry of D is equal to 1 if the mutation is present in a given sample, 0 if it is absent, NA if the information is missing (e.g., due to low coverage). Such data format is widely used in cancer phylogeny studies. For example, the same data format was used in [22].
- In the first step, a generic algorithm for the reconstruction of clonal/mutational trees (e.g., LACE, VERSO or SCITE) is applied to input data D, recording all the solutions sampled during the MCMC. In the output tree T nodes represents mutations (clones) and edges represent parental relations, as in [22].
- For each tree T^p sampled during the MCMC, we generate the corresponding adjacency matrix M^p with dimension $m \times m$, where $p \in [1, \dots r]$ and r is the number of MCMC iterations.
- We compute a weighted adjacency matrix W with dimensions $m \times m$, where each entry is defined as: $W_{i,j} = \frac{\sum_{p=1}^{r} M^p_{i,j}}{r}$. So, W_{ij} stores a weight that corresponds to the frequency by which the mutation i is found as parent of mutation j in all sampled trees. W represents a edge-weighted digraph.
- Finally, we apply the Tarjan algorithm to W in order to find the COB-tree model.

Synthetic data generation. In order to both characterize the solution space and test the performance of the COB-tree algorithm, we generated a number of simulated datasets with the following procedure. We first randomly generated a number of

ground-truth topologies T_{gt}. In particular, given a specific number of nodes m (i.e., mutations or clones), 20 topologies are created. Starting from the root, we attached a random number (between 2 to 5) of children nodes. We then selected one of the children nodes and repeated the process until all the nodes were attached. Trees including 50, 100 and 200 mutations were considered, thus, a total of 60 topologies are eventually generated. An example of a tree with 50 mutations is reported in figure 3.

We sampled 1000 single cells to generate the ground-truth single-cell genotypes matrix D_{gt} (cells $\times$ mutations) from each topology T_{gt}. In particular, we populated each row of D_{gt} (i.e., cell genotype) by randomly selecting a node. Then, the genotype of a cell (i.e., row of D_{gt}) is populated by assigning 1, if the mutation is included in the shortest path from the selected node to the root, or 0, otherwise. Notice that this path is unique because each node can have only one parent. Since it is unrealistic to observe a high number of clones in a cancer sample, we increased the probability of selecting any of the leaf nodes (i.e., the most recent clones) with respect to that of selecting one of the internal nodes. The former probability is 5 times higher than the latter. As a result, D_{gt} is a binary matrix with 1000 rows and m columns (notice that each clone can be defined by the last mutation accumulated, so the number of clones equals the number of mutations).

In order to include data-specific noise in the simulated datasets, we defined *low*, *middle*, and *high* noise levels by setting the rates of False Positives (α), False Negatives (β), and Missing values (γ) as follows:

- Low noise level: $\alpha = 0.005$, $\beta = 0.05$, and $\gamma = 0$
- Middle noise level: $\alpha = 0.01$, $\beta = 0.1$, and $\gamma = 0.1$
- High noise level: $\alpha = 0.02$, $\beta = 0.2$, and $\gamma = 0.2$

From each D_{gt}, 3 different noisy datasets D are generated, by randomly selecting α of the entries equal to 1, β of 0 entries and changing them into 0 and 1, respectively. Then, a fraction γ of all the entries are replaced with missing values (NA).

Simulation settings. The procedure described above yields a total of 180 noisy datasets. In this preliminary analyses, we employed SCITE [16] as inference framework, since it is one of the state-of-the-art approaches for single-cell mutational tree inference. In particular, each inference is performed multiple times for each dataset, with distinct values of MCMC iterations. We performed 10 independent SCITE runs (with 10 restarts), with the following MCMC iterations:

- for models with $m = 50 \rightarrow [1000, 2000$ *(short)*$, \ldots, 6000$ *(average)*, $\ldots, 10000$ *(long)*] MCMC iterations,
- for models with $m = 100 \rightarrow [5000, 10000$ *(short)*$, \ldots, 30000$ *(average)*, $\ldots, 50000$ *(long)*] MCMC iterations,
- for models with $m = 200 \rightarrow [50000, 100000$ *(short)*$, \ldots, 300000$ *(average)*, $\ldots, 500000$ *(long)*] MCMC iterations.

Performance Metrics. In order to compare the solution provided by the COB-tree algorithm and the corresponding ML solutions, we considered the differences with respect the ground-truth topologies on simulated data. To this aim, we computed

two different metrics (i.e., Parent-Child distance PC and Clonal Genotype errors GC), which assess either the local or the global structure in terms of errors between the obtained COB and ML trees, and the ground-truth topologies.

- Parent-Child distance (PC). This metric is widely used to compare different trees, for example in [1, 13]. In brief, the parent-child distance between two trees enumerates the edges unique in either trees. Small values of this metric reflect a correct recovery of the relations between two consecutive nodes, but disregard their position in the topology, so it is considered a local measure. We compute the PC_{ML}, and PC_{COB} for evaluate the goodness of maximum likelihood and optimal branching tree respectively.
- Clonal Genotype errors (CG). As explained above, each clone can be associated with the node representing the last accumulated mutation. So, its genotype includes all the mutations in the path from such node to the root. Thus, clonal genotypes depend on the overall topology. For each inference, we transformed the ground-truth topology, the ML tree, and the COB-tree model into the corresponding clonal genotype matrices. Then, we computed the Hamming distance, i.e., the total number of errors, between the clonal genotype of the ground-truth and either the ML tree (CG_{ML}) or the COB-tree model (CG_{COB}).

Finally, we define $\Delta PC = PC_{ML} - PC_{COB}$ distance and $\Delta CG = CG_{ML} - CG_{COB}$. Positive values indicate an improvement of our approach with respect to the ML tree.

Characterization of the solution space. In order to provide a way of characterizing and visualizing the solution space of the inference framework, we decided to plot the distribution of the sampled trees during the MCMC. To do this, we applied Principal Coordinate Analysis (PCA) [15, 14] on the distance matrix computed considering the PC distance.

This approach returns a 2-dimensional representation of the tree space, where the relative distance among each point (i.e., sampled trees) is maintained. We added the value of the likelihood L of each tree as a third dimension, by computing the $-\ln(L)$. Notice that, after the transformation, the best likelihood values are the lowest.

Real data processing As a proof of principle, we applied the COB-tree algorithm to a real-world longitudinal scRNA-seq dataset of patient-derived xenografts (PDXs) of $\mathsf{BRAF}^{\mathsf{V600E/K}}$ mutant melanomas produced in [26]. The authors generated four datasets collected at different time points, before, during, and after therapy administration of a BRAF/MEK-inhibitor with a total of 674 cells. For the aim of the current work, we pre-processed them as independent datasets to defined a set of highly confided SNVs, by using GATK pipeline [10] for alignment and variant calling, and by using filters based on statistical significance (explained in the following). Thus, we selected a set of 55 mutations with a highly significance, well separated from the background noise.

For variant calling from scRNA-seq data, we applied the same steps performed in [22] which are here briefly reported:

1. Considering the expression data, we discarded cells with a high fraction of mitochondrial reads. So, we kept 475 high-quality cells.
2. Via *SRA toolkit* we downloaded the FASTQ files (one for each cell) from the GEO dataset using the accession number GSE116237.
3. Using Trimmomatic (v. 0.39) [4], we removed the nucleotides with a poor quality score.
4. Using 2-pass mode STAR aligner [11], we aligned the reads to the human reference genome (GRCh38 release). This step generate one SAM file for each cell.
5. We added read groups, we sorted, we marked duplicates and we indexed the reads in each BAM file via Picard tools
6. With GATK (v. 3.8.1) we hard clipped intronic regions (via SplitNCigarReads command) and we re-calibrate the base alignment (via BaseRecalibrator command)
7. Finally, we used HaplotypeCaller and VariantFiltration to call Single Nucleotide Variants and to remove the ones with poor quality score (we applied default parameters).

The above steps generated a VCF file for each cell which we merged together and load it into an R envirorment, as matrix with cells as row and mutations as columns, to perform downstream filtering steps. In particular:

1. Considering each entry of the mutational matrix, we set as NA (i.e., missing data) mutations that fall in a position covered with less than 3 reads
2. We removed mutations (columns) which display a frequency of missing data higher than 0.4 in at least one time point.
3. We compute the frequency of every mutation in each time point and we remove rare mutations by keeping only those with a frequency sum higher than 0.15

We selected 55 high-quality mutations, and we use those mutational profiles for generating the input file to infer the tumour evolution via SCITE. We run one inference using a false discovery rate equal to 0.02, an allele dropout rate of 0.2, and 30 restarts of Markov Chain Monte Carlo with a length of 20000 steps.

SCITE considers the first 25% trees sampled as burning and discards them. Thus, we sampled the remaining 450000 trees from which we removed the ones with a likelihood value less than 1.3× Maximum log-Likelihood (we highlight that the log-likelihood is negative so we are discarding trees with a low likelihood value). Finaly, we draw the solution landscape by considering 4740 unique trees using the approach described in the above section.

2 Results

Characterization of the search space. We first characterized the search space of the clonal trees inferred via one of the state-of-the-art approaches for mutational tree inference – SCITE [16]. SCITE was selected for the extremely efficient implementation and the good computational costs. The experiments were executed

by repeatedly performing the inference on synthetic datasets generated from a number of distinct topologies (the whole simulation settings is described in the Materials and Methods section).

In figure 1 we report the results of the PCA applied on the trees sampled from the inference of a selected dataset, generated from a tree topology including $m = 100$ mutations. We show the solution considering three different MCMC lengths i.e., 10000 (short), 30000 (average), and 50000 (long) steps.

As one can see, the 10 independent chains tend to reach the same global minimum but, when the MCMC is short, the trees with better likelihood are far away from each other. We also marked the COB-tree solutions with red dots, and they appear to be placed in a central position among the trees sampled late in the inference. This interesting result suggests that, in this case, applying a consensus approach that does not depend on a clustering step seems to be a reasonable algorithmic choice.

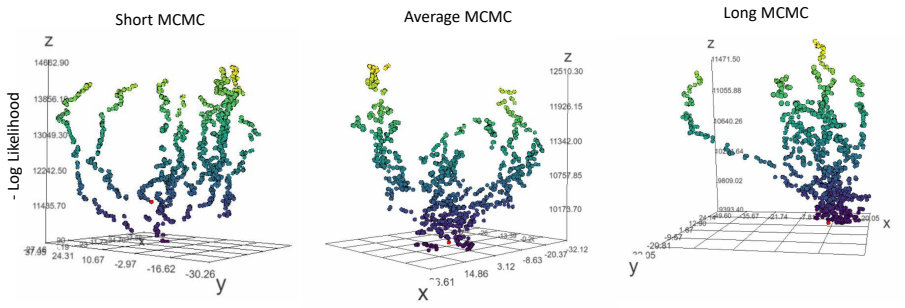


Fig. 1. Visual representation of the solution space explored during the inference of a clonal tree from the same synthetic single-cell dataset (with 100 mutations and 1000 cells), in three independent MCMC runs with 10 restarts (via SCITE). From left to right, the total number of MCMC increases (10000, 30000, and 50000 steps). The solution space is defined by computing a PCA on the distance matrix (using parent-child distance) of the trees sampled during each inference. Z-axis reports the corresponding likelihood value. Red dots indicate the position of the COB-tree solutions.

Performance assessment of COB-tree. We applied the COB-tree algorithm to the synthetic datasets described in the Materials and Methods section. To this end, we considered all the trees sampled during the MCMC. Since SCITE discards the first 25% trees, we only considered the remaining 75% and kept only the trees with a likelihood between L_{best} and $L_{best} \times 1.3$, so to focus on the final part of the MCMC. The Tarjan algorithm was finally applied to retrieve the unique COB-tree model. Notice that we also kept track of the ML tree (L_{best}). We considered three MCMC lengths to evaluate how this affects the performance the COB-tree method. Results are reported in figure 2.

It is possible to observe how our approach improves the local phylogenetic structure, by recovering a better ordering of the accumulation of mutations. Instead, the improvement is not that evident when considering the CG metric underlying the global phylogenetic structure. Even though the COB-tree method often improves this metric as well, it sometimes returns trees with a global structure very far from the ground-truth.

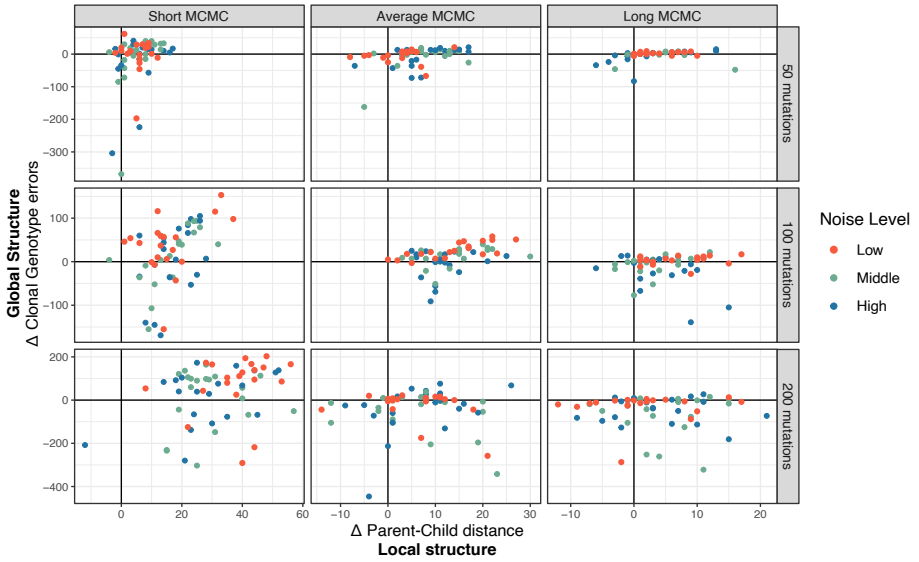


Fig. 2. Differences between Maximum Likelihood and COB trees are reported. Positive values of ΔCG errors or ΔPC distance indicate an improvement for the global structure or the local structure of the COB-tree solutions over the ML ones. Colours indicate the level of noise (i.e., rate of FP events, FN events, and missing values) included in the simulated datasets.

Note that it is possible to have trees with high PC distance values and low CG metric values. A possible explanation is illustrated in the example depicted in figure 3. In the plot, it is possible to observe how COB-tree retrieves a better ordering of mutation pairs. Still, the few errors drastically changed the global topology of the tree by shifting an entire subtree.

Application of COB-tree to PDX melanoma datasets. We finally applied the COB-tree algorithm to a real-world dataset of patient-derived xenografts (PDXs) of $\text{BRAF}^{\text{V600E/K}}$ mutant melanomas. The preprocessing steps are described in the Materials and Methods section.

The model includes 55 nodes. In Figure 4 one can find both the COB-tree solution and three equivalent ML solutions. As one can see, significant differences are present in the model.

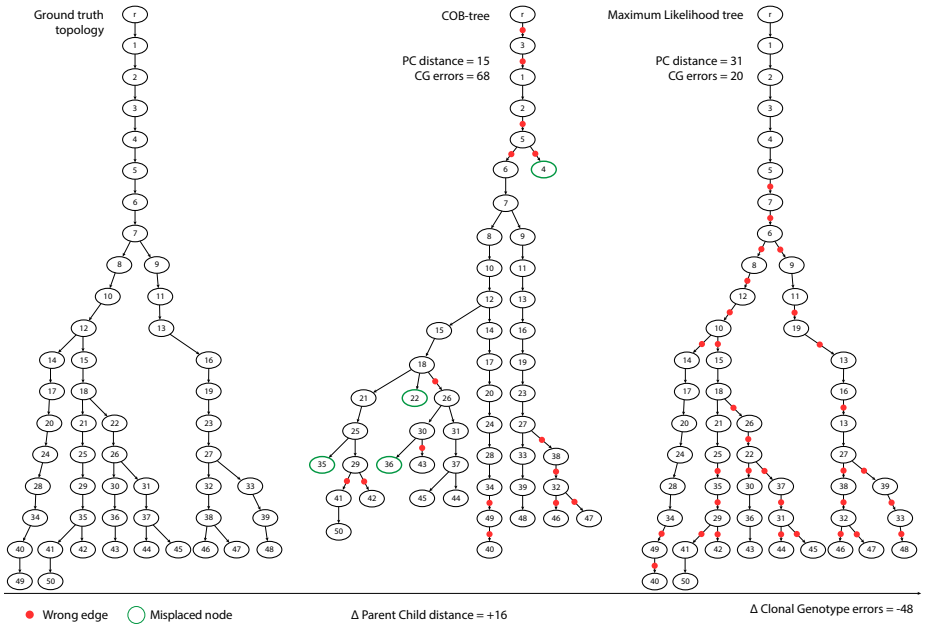


Fig. 3. The 3 clonal tree comparison highlights the difference between local and global tree structures. The order of mutational events are improved in the COB-tree solution (wrong edges marked with red dots, $\Delta PC = +16$), while the global structure is worse ($\Delta CG = -48$) due to a error propagation of few nodes being misplaced (most relevant are highlighted with green circle). The numbers in the nodes indicate distinct mutations.

3 Conclusions

The preliminary analyses illustrated in this work show that in case one is interested in defining the ordering of mutational events the COB-tree algorithm is a reliable and effective option.

This aspect might be of particular relevance, for instance, if someone is interested in detecting possible patterns of repeated cancer evolution across different patients [6], as this might help in identifying possible therapeutic targets, as well as weak points of specific tumor types.

Currently, other methods to generate aggregated trees are available [7]. They are based on specific assumptions and exploit different strategies to summarize the solution space. A comparison between them and COB-tree could be useful to highlight the specific behavior of each method.

Further development of the COB-tree algorithm are underway, aimed, e.g., at better exploiting the properties of the solution space so to improve the performance with respect to the global structure too. However, on the basis of the simulation results and on the fact the – by definition – the COB-tree algorithm returns a

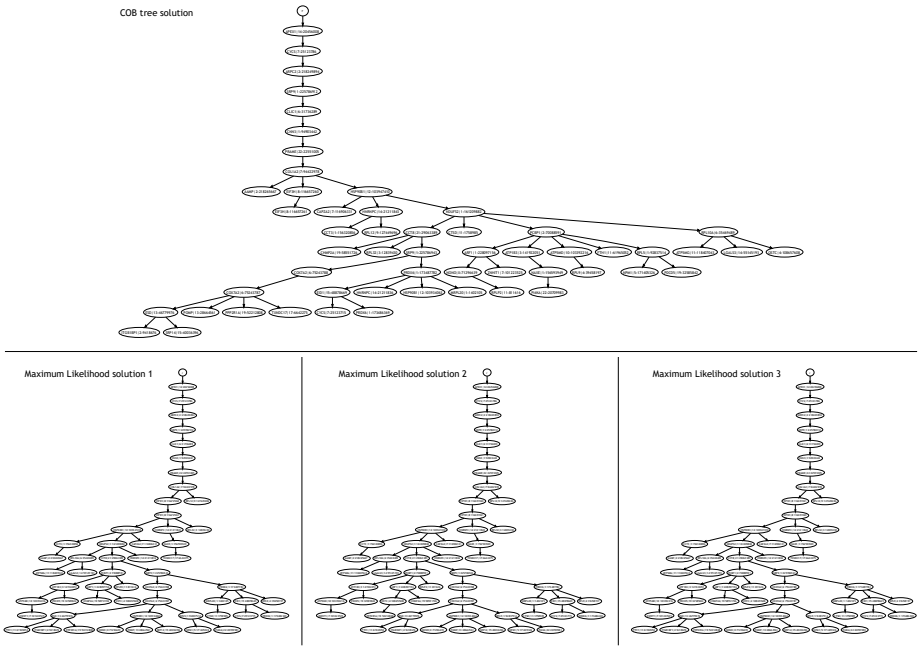


Fig. 4. Clonal tree comparison between the COB-tree solution and three equivalent maximum likelihood solutions returned by SCITE considering the mutational profiles from the melanoma PDX dataset. The node labels define the gene involved and the genome position of the SNVs. The inference was performed with SCITE using a false discovery (fd) rate of 0.2, an allele dropout (ad) rate equal to 0.2, 30 restarts, and a MCMC with 20000 steps.

unique solution, we believe that it could be a reliable and robust option for clonal tree inference.

4 Acknowledgments

This work was supported by a Bicocca 2020 Starting Grant and Google Cloud Academic Research Grant to DR and FA. Partial support is also granted by the CRUK/AIRC Accelerator Award #22790 "Single-cell Cancer Evolution in the Clinic". The funders had no role in the design and conduct of the study, analysis, and interpretation of the data, preparation of the manuscript, and decision to submit the manuscript for publication.

References

[1] Nuraini Aguse, Yuanyuan Qi, and Mohammed El-Kebir. "Summarizing the Solution Space in Tumor Phylogeny Inference by Multiple Consensus Trees".

In: *Bioinformatics* 35.14 (July 15, 2019), pp. i408–i416. DOI: 10.1093/bioinformatics/btz312.

[2] Philipp M Altrock, Lin L Liu, and Franziska Michor. "The mathematics of cancer: integrating quantitative models". In: *Nature Reviews Cancer* 15.12 (2015), pp. 730–745.

[3] Fabrizio Angaroni et al. "PMCE: Efficient Inference of Expressive Models of Cancer Evolution with High Prognostic Power". In: *Bioinformatics* (Oct. 14, 2021), btab717. DOI: 10.1093/bioinformatics/btab717.

[4] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. "Trimmomatic: a flexible trimmer for Illumina sequence data". In: *Bioinformatics* 30.15 (2014), pp. 2114–2120.

[5] Remco Bouckaert et al. "BEAST 2: A Software Platform for Bayesian Evolutionary Analysis". In: *PLOS Computational Biology* 10.4 (Apr. 10, 2014), e1003537. DOI: 10.1371/journal.pcbi.1003537.

[6] Giulio Caravagna et al. "Detecting Repeated Cancer Evolution from Multi-Region Tumor Sequencing Data". In: *Nature Methods* 15.9 (9 Sept. 2018), pp. 707–714. DOI: 10.1038/s41592-018-0108-x.

[7] Sarah Christensen et al. "Detecting Evolutionary Patterns of Cancers Using Consensus Trees". In: *Bioinformatics* 36 (Supplement_2 Dec. 30, 2020), pp. I684–I691. DOI: 10.1093/bioinformatics/btaa801. pmid: 33381820.

[8] Sarah Christensen et al. "Detecting evolutionary patterns of cancers using consensus trees". In: *Bioinformatics* 36.Supplement_2 (2020), pp. i684–i691.

[9] Yoeng-Jin Chu. "On the Shortest Arborescence of a Directed Graph". In: *Scientia Sinica* 14 (1965), pp. 1396–1400.

[10] Mark A DePristo et al. "A framework for variation discovery and genotyping using next-generation DNA sequencing data". In: *Nature genetics* 43.5 (2011), p. 491.

[11] Alexander Dobin et al. "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21.

[12] Jack Edmonds. "Optimum Branchings". In: *Journal of Research of the National Bureau of Standards, B* 71 (1967), pp. 233–240.

[13] Kiya Govek, Camden Sikes, and Layla Oesper. "A Consensus Approach to Infer Tumor Evolutionary Histories". In: *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. BCB '18. New York, NY, USA: Association for Computing Machinery, Aug. 15, 2018, pp. 63–72. ISBN: 978-1-4503-5794-4. DOI: 10.1145/3233547.3233584.

[14] J. C. Gower. "Adding a Point to Vector Diagrams in Multivariate Analysis". In: *Biometrika* 55.3 (Nov. 1, 1968), pp. 582–585. DOI: 10.1093/biomet/55.3.582.

[15] J. C. Gower. "Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis". In: *Biometrika* 53.3-4 (Dec. 1, 1966), pp. 325–338. DOI: 10.1093/biomet/53.3-4.325.

[16] Katharina Jahn, Jack Kuipers, and Niko Beerenwinkel. "Tree Inference for Single-Cell Data". In: *Genome Biology* 17.1 (May 5, 2016), p. 86. DOI: 10.1186/s13059-016-0936-x.

[17] Jack Kuipers and Giusi Moffa. "Uniform Random Generation of Large Acyclic Digraphs". In: *Statistics and Computing* 25.2 (Mar. 1, 2015), pp. 227–242. DOI: 10.1007/s11222-013-9428-y.

[18] Ben Langmead and Steven L Salzberg. "Fast gapped-read alignment with Bowtie 2". In: *Nature methods* 9.4 (2012), pp. 357–359.

[19] Joseph E O'Reilly and Philip C J Donoghue. "The Efficacy of Consensus Tree Methods for Summarizing Phylogenetic Relationships from a Posterior Sample of Trees Estimated from Morphological Data". In: *Systematic Biology* 67.2 (Mar. 1, 2018), pp. 354–362. DOI: 10.1093/sysbio/syx086.

[20] Lucrezia Patruno et al. "A Review of Computational Strategies for Denoising and Imputation of Single-Cell Transcriptomic Data". In: *Briefings in Bioinformatics* 22.4 (July 1, 2021), bbaa222. DOI: 10.1093/bib/bbaa222.

[21] Daniele Ramazzotti et al. "CAPRI: Efficient Inference of Cancer Progression Models from Cross-Sectional Data". In: *Bioinformatics* 31.18 (Sept. 15, 2015), pp. 3016–3026. DOI: 10.1093/bioinformatics/btv296.

[22] Daniele Ramazzotti et al. "LACE: Inference of cancer evolution models from longitudinal single-cell sequencing data". In: *Journal of Computational Science* 58 (2022), p. 101523. DOI: https://doi.org/10.1016/j.jocs.2021.101523. URL: https://www.sciencedirect.com/science/article/pii/S1877750321001848.

[23] Daniele Ramazzotti et al. "Learning Mutational Graphs of Individual Tumour Evolution from Single-Cell and Multi-Region Sequencing Data". In: *BMC Bioinformatics* 20.1 (Apr. 25, 2019), p. 210. DOI: 10.1186/s12859-019-2795-4.

[24] Daniele Ramazzotti et al. "Variant calling from scRNA-seq data allows the assessment of cellular identity in patient-derived cell lines". In: *Nature communications* 13.1 (2022), pp. 1–3.

[25] Daniele Ramazzotti et al. "VERSO: A Comprehensive Framework for the Inference of Robust Phylogenies and the Quantification of Intra-Host Genomic Diversity of Viral Samples". In: *Patterns* 2.3 (Mar. 12, 2021), p. 100212. DOI: 10.1016/j.patter.2021.100212.

[26] Florian Rambow et al. "Toward Minimal Residual Disease-Directed Therapy in Melanoma". In: *Cell* 174.4 (Aug. 9, 2018), 843–855.e19. DOI: 10.1016/j.cell.2018.06.025.

[27] Russell Schwartz and Alejandro A. Schäffer. "The Evolution of Tumour Phylogenetics: Principles and Practice". In: *Nature Reviews Genetics* 18.4 (4 Apr. 2017), pp. 213–229. DOI: 10.1038/nrg.2016.170.

[28] Jochen Singer et al. "Bioinformatics for precision oncology". In: *Briefings in Bioinformatics* 20.3 (2019), pp. 778–788.

[29] R. E. Tarjan. "Finding Optimum Branchings". In: *Networks* 7.1 (1977), pp. 25–35. DOI: 10.1002/net.3230070103.

.

4

Task B: Computational methods for omics data integration

As presented in section 1.1, in order to have a complete picture of the system under study, in addition to perform single-omics experiments, it is necessary to integrate the information extracted from the multiple layers. In the context of cancer research, data integration is fundamental. In fact, mechanisms that are responsible for drug resistance are the result of a complex interplay between different layers, as this might lead to explanatory and predictive models of disease evolution, supporting experimental and clinical research in the definition of diagnostic, prognostic and therapeutic strategies.
In this chapter we discuss the main contribution regarding task (B), that is diagonal integration of multi-omics data. We present two methods, CONGAS and CONGAS+ that integrate respectively RNA-DNA and RNA-DNA-ATAC.

4.1 [DNA] + [RNA] CONGAS

In order to perform data integration the general goal is to map measurements in a shared latent space, which can be built following two types of approaches: first, it is possible to exploit deep-learning based methods such as variational autoencoders [84, 121, 136, 189, 193]. These approaches constitute a powerful tool, as they can accommodate known covariates and external domain knowledge [121] and they can be used to perform multiple tasks such as the reconstruction of a joint space and data denoising [189]. By including covariates and external knowledge, such methods are able to perform data integration

in two directions: first, by including the batch label as a covariate they can perform horizontal integration (See Section 1.1 for more details) and remove the non-biological variability in the data introduced by the batch effect. Second, they are able to integrate data from different modalities, accommodating both the integration of multiomics measurements performed on the same set of cells (vertical integration) and on different subset of cells (diagonal integration) [193].
However, the major drawback of deep learning based methods is that the latent space is not interpretable, and thus in case we want it to reflect a specific biological property we need to rely on other approaches. For example, given the hierarchy described in section 2.1.4, it is possible to assume that the Copy Number status of a specific genomic region predicts linearly the number copies of RNA transcripts for the genes mapped to that region [111], and this relation can be exploited to define a mapping and integrate the multiple omics layers. Over the years, different computational methods have been proposed to infer CNAs from single-cell RNA sequencing data. HoneyBadger [89], CaSpER [150], InferCNV [38] and CopyKAT [165], are designed to be applied on datasets that consist of a mixture of both tumor and normal cells. In fact, normal cells are used to compute the baseline expression value for each gene, so that differences in expression between tumor and normal cells can be imputed to CNAs and are used to infer the copy number state of tumor cells. However, the presence of normal cells is not guaranteed in biological samples such as PDOs and cell lines, where the prevalence of tumor cells is 100%.

One method that doesn't rely on such assumption is clonealign [111], that instead of using solely scRNA-seq data to extract CNAs, it aims at finding a map between two measurements: one is single-cell DNA sequencing, from which it is possible to measure CNAs at the single cell resolution and thus identify the copy number profiles of the multiple subclones present in a tumor sample. Next, under the assumption that the copy number status of a segment is a linear predictor for the gene expression of genes mapping to that segment, clonealing aims at associating to each single-cell gene expression profile the corresponding CNA clone profile detected from scDNA-sequencing.

Clonealign is a supervised algorithm which requires in input the profiles of all clones that we aim at detecting in the data, and thus it relies on expensive single-cell DNA sequencing experiments. Therefore, we investigated whether we could apply the same linear model linking CNAs and gene expression in an unsupervised framework to detect new clusters from scRNA-seq data, relying on the output of a bulk DNA sequencing experiment. We developed CONGAS, a Bayesian method that clusters single-cell gene expression profiles using CNAs. Our method integrates scRNA-seq with bulk DNA data, using the latter to reliably identify genomic segments and set Bayesian priors on the CNAs values for the different clusters. We tested CONGAS on both synthetic and real datasets, comparing its performance with CopyKAT, InferCNV and clonealign.

Please notice that the supplementary information of this work is not included here, but it is available in the online version of the manuscript [194].

Bioinformatics, 38(9), 2022, 2512–2518
https://doi.org/10.1093/bioinformatics/btac143
Advance Access Publication Date: 17 March 2022
Original Paper

Gene expression

A Bayesian method to cluster single-cell RNA sequencing data using copy number alterations

Salvatore Milite[1], Riccardo Bergamin[1], Lucrezia Patruno[2], Nicola Calonaci[1] and Giulio Caravagna [1,*]

[1]Department of Mathematics and Geosciences, University of Trieste, Trieste 34127, Italy and [2]Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milano 20125, Italy

*To whom correspondence should be addressed.

Associate Editor: Can Alkan

Received on March 17, 2021; revised on January 31, 2022; editorial decision on February 21, 2022; accepted on March 16, 2022

Abstract

Motivation: Cancers are composed by several heterogeneous subpopulations, each one harbouring different genetic and epigenetic somatic alterations that contribute to disease onset and therapy response. In recent years, copy number alterations (CNAs) leading to tumour aneuploidy have been identified as potential key drivers of such populations, but the definition of the precise makeup of cancer subclones from sequencing assays remains challenging. In the end, little is known about the mapping between complex CNAs and their effect on cancer phenotypes.

Results: We introduce CONGAS, a Bayesian probabilistic method to phase bulk DNA and single-cell RNA measurements from independent assays. CONGAS jointly identifies clusters of single cells with subclonal CNAs, and differences in RNA expression. The model builds statistical priors leveraging bulk DNA sequencing data, does not require a normal reference and scales fast thanks to a GPU backend and variational inference. We test CONGAS on both simulated and real data, and find that it can determine the tumour subclonal composition at the single-cell level together with clone-specific RNA phenotypes in tumour data generated from both 10× and Smart-Seq assays.

Availability and implementation: CONGAS is available as 2 packages: CONGAS (https://github.com/caravagnalab/congas), which implements the model in Python, and RCONGAS (https://caravagnalab.github.io/rcongas/), which provides R functions to process inputs, outputs and run CONGAS fits. The analysis of real data and scripts to generate figures of this paper are available via RCONGAS; code associated to simulations is available at https://github.com/caravagnalab/rcongas_test.

Contact: gcaravagna@units.it

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Cancers grow from a single cell, in an evolutionary process modulated by selective forces that act upon cancer genotypes and phenotypes (Greaves and Maley, 2012; McGranahan and Swanton, 2015). The fuel to cancer evolution is genotypic and phenotypic cellular heterogeneity, and much is yet to be understood regarding its effect on evolution and response to therapy (McGranahan and Swanton, 2017; Turajlic *et al.*, 2019). Notably, the heterogeneity observed in cancer can also be produced during normal tissue development, and therefore the quest for understanding heterogeneity has implications far beyond cancer (Martincorena, 2019; Martincorena *et al.*, 2015).

While the evolutionary principle of cancer growth is intuitive to conceptualize and replicate *in vivo* (Acar et al., 2020), it is still hard to precisely measure clonal evolution using sequencing technologies (Caravagna, 2020). Even if popular single-omic assays from 10× and Smart-Seq achieve higher resolution than bulk counterparts (Picelli *et al.*, 2014; Wang *et al.*, 2021), their analysis poses many challenges (Lähnemann *et al.*, 2020). Nowadays, much hope is put into multiomics technologies that probe multiple molecules from the same cell (Macaulay *et al.*, 2015). Multiomics data explicitly gather DNA and RNA measurements per cell; unfortunately, however, such assays are still too expensive to scale to more than hundreds of cells. An interesting opportunity is attempting the integration of different types of single-omic assays that, individually, already scale to thousands of cells. At least conceptually, the statistical integration of independent assays comes from mapping one dataset on top of another, leveraging a quantitative model for the relation between the sequenced molecules (e.g. we may wish to predict DNA from RNA, or vice versa).

In this work, we develop a Bayesian method for total copy number genotyping from single cells (CONGAS), which integrates total copy number alterations (CNAs) obtained from bulk DNA sequencing and single-cell RNA (scRNA) data from independent cells (Fig. 1a). Our method is similar to clonealign, which uses two single-cell assays to assign scRNA profiles to tumour clones predetermined from low-pass single-cell DNA sequencing (Campbell *et al.*, 2019). CONGAS and clonealign conceptualize the same linear model to link total CNAs—i.e. the sum of the major and minor allele copies (Househam *et al.*, 2021)—with RNA counts, but while clonealign fixes the set of clones from its input and is therefore supervised (Supplementary Fig. S1), CONGAS is unsupervised and finds new clusters by leveraging Bayesian priors from the input bulk (Fig. 1b). Precisely, CONGAS uses input CNAs to define the genome segmentation and parametrize a prior for total CNAs of each segment—then each cluster has its posterior distribution over the ploidy of the segments. We note that the extra input for CONGAS can be generated from a routine low-pass bulk DNA assay, which is much cheaper than the single-cell counterpart required by clonealign.

With CONGAS we formulate an unsupervised clustering problem: we seek to group cells with segment-level RNA profiles that can be explained by similar CNAs (Fig. 1c and d), inferring CNAs and clusters jointly. There are methods that are alternative to CONGAS, for instance InferCNV, HoneyBADGER, CASPER and copyKAT, that detect CNAs by segmenting scRNA counts (Fan *et al.*, 2018; Serin Harmanci *et al.*, 2020). These methods, however, decouple CNA detection from clustering, requiring to select the number of optimal clusters with some heuristic. Instead, CONGAS detects subclonal CNAs and clusters cells in a unified model, therefore integrating uncertainty with its Bayesian formulation. Compared to some of the alternative methods CONGAS also has the advantage of working without reference scRNA expression; this avoids using RNA tissue databases, or requiring normal cells in the input scRNA assay. Therefore, CONGAS can be applicable in designs where the normal signal is difficult to obtain, e.g. with cancer organoids. CONGAS can associate CNA-associated cancer subclones to transcriptomic profiles, providing an explicit mapping between genotype and phenotype at the clone level. This is particularly important in cancer, where we want to characterize how chromosomal instability drives tumour evolution (Watkins *et al.*, 2020), or, where we want to understand how precancerous cells can be causally linked to the onset of cancer (Martincorena, 2019).

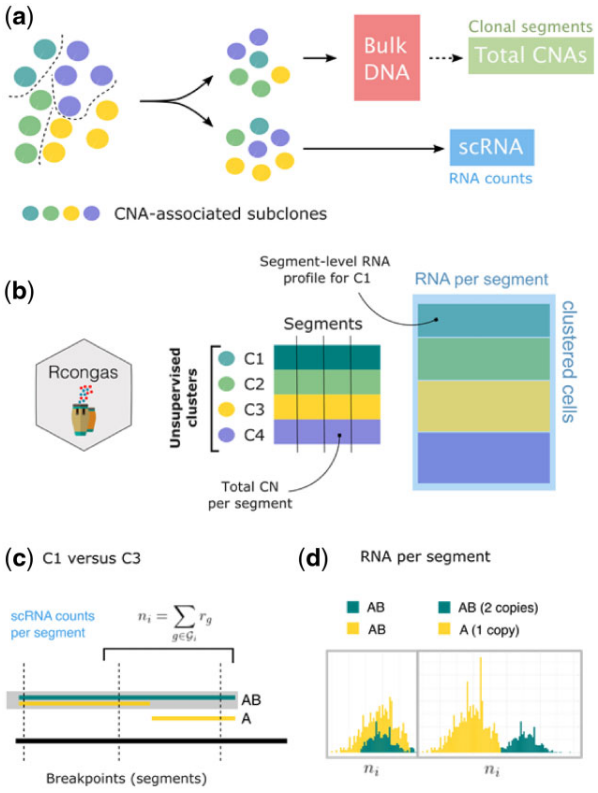


Fig. 1. (**a**) CONGAS works with (1) total CNA data (ploidy values per segment) from a bulk DNA assay and (2) scRNA sequencing data. The two assays are generated from independent cells of the same starting sample. The aim is to identify CNA-associated subclones from RNA counts. (**b**) CONGAS is a Bayesian unsupervised method to identify clusters of cells whose differences in scRNA counts can be explained by total CNAs. Subclonal CNAs are here inferred at the resolution of the input segments. (**c** and **d**) Assume subclones C1 and C3 differ for a portion of DNA (right segment): C3 has a subclonal LOH (A genotype), where C2 is heterozygous diploid (AB genotype). CONGAS identifies CNAs by examining total RNA counts mapped to segments: subclone C3 shows fewer RNA counts on the deleted segment, and the subclones have similar RNA counts on the segment where both clones are heterozygous diploid (left segment)

2 Materials and methods

The aim of CONGAS is to statistically integrate DNA and RNA measures for every cell, deriving a measure of total DNA abundance per segment (i.e. total copy number) and RNA counts per cell. This accounts for emulating a DNA–RNA multiomics assay, which we use to cluster cells whose RNA profile can be explained by similar copy numbers.

2.1 CONGAS

CONGA is a Bayesian method that 'genotypes' bulk CNAs on top of scRNA data; The term genotyping elicits the use in CONGAS of an input set of CNAs obtained from bulk DNA sequencing, here used to create Bayesian priors. A vector of input total copy number profiles drives the calling of subclonal CNAs from single cells, in a way that new CNAs can be obtained as ploidy changes with fixed breakpoints. In particular, breakpoints are used to pool RNA counts per segment, and bulk-level total copy numbers constitute a Bayesian prior per segment. Therefore, the model is able to infer variations of single-cell CNAs around the input bulk. The CONGAS likelihood is a mixture of $K > 0$ Poisson distributions for scRNA counts per segment, and works also with data normalized in common units; the likelihood is conditioned on the latent CNAs that we infer for each of the K cluster, and normalizes counts for library size and number of genes per segment if required.

A low-pass bulk DNA assay to generate the input CNAs required for CONGAS is inexpensive. If this is unavailable, RNA segmentation can also be attempted, or an arm-level segmentation with constant ploidy 2 can be used to detect large CNAs. Input CNAs simplify the statistical inference problem and avoid the segmentation of noisy scRNA data. The model chases subclonal populations that show different total CNAs at the resolution of the input segments. For instance, it can detect a subclonal population underlying a loss of heterozygosity (Fig. 1b). After pooling RNA counts on every segment (Fig. 1c), under a linear model that links DNA abundance to RNA counts (Campbell *et al.*, 2019), we use Poisson distributions parameterized by unobserved copy number values to explain counts (Fig. 1d). By this definition, clonal CNAs—i.e. present in 100% of the input cells—show the same RNA signal and cannot be detected unless normal cells are in the sample (e.g. tumour versus normal). Nonetheless, the difference across subpopulations can be still captured wherever present (e.g. tumour subpopulation 1 versus tumour subpopulation 2).

The model likelihood with the usual independence assumption among cells and segments is

$$p(Y|\theta, \mu, C, Z, \pi) = \prod_{n=1}^{N} \prod_{i=1}^{I} p(y_{ni}|\theta, \mu, C, Z, \pi)$$

.

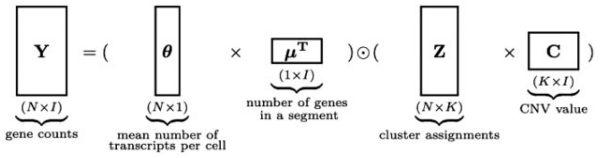
Here, $\mathbf{Y}$ is the $N \times I$ input data matrix of RNA counts, which describe N sequenced cells and I input segments (mapped anywhere on

the genome). Counts on a segment $y_{n,i}$ are summed up by pooling all genes that map to the segment; with cumulative counts we rarely observe 0-counts segments, which allows us to avoid zero-inflated distributions (Sarkar and Stephens, 2021). The segment likelihood is

$$p(y_{ni}|\theta,\mu,C,Z) = Pois\left(\frac{\theta_n \cdot \mu \prod_{k=1}^{K} C_{ik}^{z_{nk}}}{\sum_{i=1}^{I}\prod_{k=1}^{K} C_{ik}^{z_{nk}}}\right),$$

where the model uses θ_n, a Gamma-distributed latent variable which models the library size for cell n, and μ_i for the number of genes in segment i (a constant determined from data). In CONGAS, $\mathbf{C}$ is the clone CNA profile for k clones, where each clone is defined by I segments and associated CNAs; the prior for $\mathbf{C}$ is a log-transform of a normal distribution, consistently with the fact that ploidies are positive values. In this formulation, $\mathbf{Z}$ are the $N \times k$ latent variables that assign cells to clusters, and π the k-dimensional mixing proportions (Supplementary Fig. S2). Based on this modelling idea, we also built alternative models that can process input data when these are already corrected for library size (e.g. in units of transcripts or read fragments per million) using Gaussian likelihoods; see Supplementary Materials.

Another way of thinking of the denominator in the formula is, given that all the effects are linear, as a matrix decomposition of the input. Note that here the denominator is omitted.



CONGAS parameters are learnt via stochastic variational inference (Blei *et al.*, 2017). The model joint distribution can be factored as

$$p(Y,Z,C,\theta,\pi) = p(Y|C,\theta,\pi)p(Z|\pi)p(\pi)\prod_{ik}p(c_{ik})\prod_{n}p(\theta_n)$$

and in the variational framework, latent variables are approximated as variational distributions $q(Z,C,\theta,\pi)$, supposed to be independent and factorizable. The prior distributions for our latent variables are

- $p(c_{ik}) \sim \text{LogNorm}(m_{ik}, v)$, where m_{ik} is the input CNA value *from* bulk DNA, and the variance v governs how far the *actual* CNAs can be compared to input (default $v = 0.5$);
- $p(\theta_n) \sim \text{Gamma}(e_s, e_r)$, a scarcely informative prior that works well in most cases (default $e_s = 3$, $e_r = 1$);
- $p(\pi) \sim \text{Dirichlet}(r)$, a prior over cluster distributions, by *default* all assumed to have equal proportions (i.e. $r = 1/k$).

The CONGAS model is implemented in 2 open-source R/Python packages. One, called CONGAS, implements the model in the Pyro probabilistic programming language, a backend that allows running on both CPU and GPU (Bingham *et al.*, 2019). A frontend R package, called RCONGAS, provides functions for data preprocessing, visualization and model inference.

3 Results

3.1 Synthetic simulations

Generative model: We tested CONGAS by simulating synthetic data from its generative model, emulating a common 10× assay (1000 cells) for tumours of various complexities. Overall, we could retrieve the generative model in a number of scenarios for tumours with up to five CNA-associated subclones, evolving both linearly and branching (Fig. 2a). The performance was measured via the adjusted rand index (ARI), the ratio of agreements over disagreements in cell clustering assignments, and was consistent with other information-theoretic scores (Supplementary Fig. S3). Clustering assignments were stable across a number of configurations of different subclonal complexities (Fig. 2b).

CONGAS could also work with negative binomial overdispersed data, a violation of its Poisson model. Performance clearly increased for lower dispersion, plateauing for non-dispersed data (Supplementary Figs S4 and S5). We also tested how errors in the input segmentation affects deconvolution. Precisely, we generated subclonal CNAs that were shorter than the input bulk segments, so that only a percentage of genes mapping to a segment were showing a signal in RNA data (from 10% to 90% of mapped genes). This is another test-case where the assumptions of CONGAS are violated. We observed good performance when >40% of the genes that map to a segment are associated to the subclonal CNA (Supplementary Figs S4 and S6), which suggests that genotyping focal amplifications that involve a handful of genes might be hard, while larger CNAs are generally identifiable even with imperfect segmentation.

scRNA-based tools: We compared CONGAS against InferCNV and copyKAT, two popular CNA-calling methods for scRNA, using an independent scRNA simulator to avoid biases (Zappia *et al.*, 2017). We tested the performances with 500 cells from a variable number of subclones, assuming a linear model for the CNA-expression dependency. Overall, CONGAS obtained the highest ARI (always above 0.75 in all configurations), showing the ability to recover the real clusters in most cases. In general, CONGAS performance was particularly good in settings with <7 subclones, with clear difference to inferCNV. In those cases, inferCNV showed a tendency to overestimate the real number of clusters by a factor of 2 (i.e. one false cluster for every true one), while CONGAS retrieves on average the exact number of subclones in the data. copyKAT showed slightly worse performance than inferCNV. From tests, we also observed that the probability of miscalling a cluster goes to zero as the size of the cluster increases, as corroborated by the fact that most of the clusters missed by CONGAS had less than 25 cells, and were therefore too small (<5% of the simulated cells) to detect (Fig. 2c and f, Supplementary Materials). To avoid our conclusions being derived solely from using different model selection criteria, we compared the performance of inferCNV and copyKAT on the same dataset of simulations used previously, but this time giving the dendrogram cutting algorithm the true number of clusters. We indeed observed that the performances, especially for inferCNV, increase a lot for low k. Instead for $k > 10$, the ARI does not improve and in some cases ($k = 12$) decreases (Fig. 2d, Supplementary Materials).

clonealign: We compared CONGAS (unsupervised) against clonealign (supervised) using synthetic simulations and three possible inputs (Fig. 2e, details in Supplementary Materials) in order to capture different qualities for the supervision set of clonealign. We considered (i) the ideal input, when clonealign knows all the simulated clonal profiles (perfect clustering from scDNA-seq), (ii) a noisy input, where we applied noise to the clonal profiles, simulating more realistic noisy scDNA-seq clustering and (iii) a partial information, where only a subset of the real input profiles is given to clonealign. This last case simulates imperfect clustering from scDNA-seq (missing clones); this type of input could also mimic usage of a subclonal copy number caller from bulkDNA-seq (instead of scDNA-seq), where we call certainly fewer clusters than with a single-cell assay (Supplementary Materials).

We again generated assays with 500 cells using the same CNA model as the previous simulations. As expected, with prefect data clonealign has better ARI when the number of clones increases; in these cases, since cluster size decreases with fixed number of cells, CONGAS is not able to separate well some clusters. Clonealign seems also very robust with respect to the adopted noise. On the other hand, when we simulate more realistic partial input profiles, the performance of clonealign decreases rapidly and proportionally to the number of clusters in the original data, and the performance of CONGAS is higher. Further, comparison between the two tools is

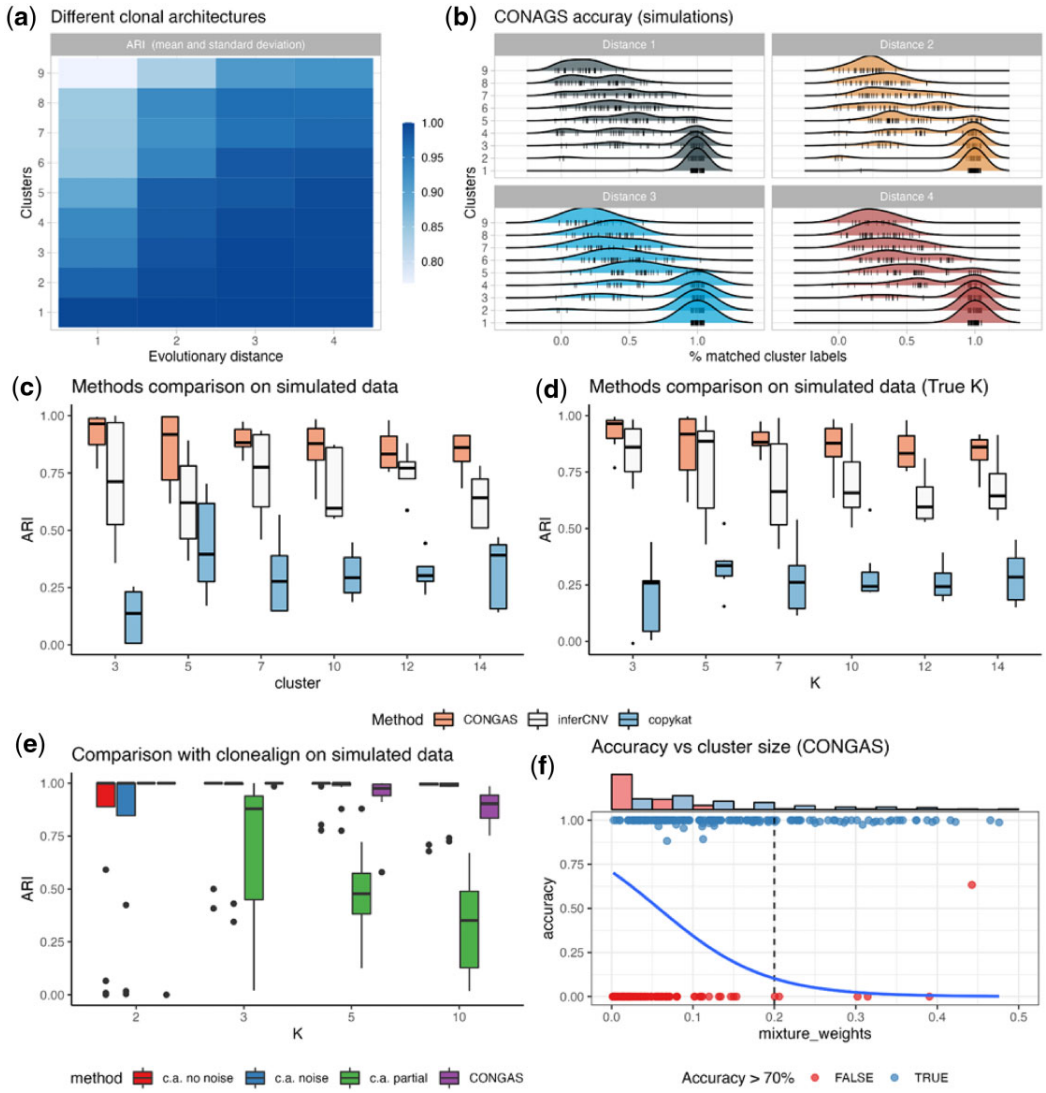


Fig. 2. (**a**) CONGAS synthetic tests with different subclonal architectures, obtained sampling clone trees with variable number of nodes. The degree of tumour heterogeneity is tuned by an evolutionary distance, which counts the number of CNAs that a subclone acquires, relative to its ancestor. The bulk input profile for CONGAS is generated by considering CNA segments from the most prevalent clone. We scan models with up to nine clones, with distance ranging from 1 to 4. The performance is measured by using the ARI between simulated and retrieved cell assignments. The heatmap reports the mean. (**b**) Smoothed density for the percentage of cluster labels matched in every simulation, split by simulated tumour trees of increasing distance to mimic subclonal complexity. (**c**) CONGAS, inferCNV and copyKAT were run on a set of synthetic scRNA-data-set with 500 cells and a linear model for CNA effect on expression. Overall, CONGAS obtained the highest ARI score, and all other methods overestimated the true number of clusters in the data. (**d**) The same simulations from panel (c) were reclustered by cutting the dendrogram generated in output by copyKAT and inferCNV using the actual number of clusters. Despite the improvement in performance, especially for little k, CONGAS (unsupervised) continues to perform better than the other two tools. (**e**) CONGAS and clonealign (supervised) were tested on a set of simulated dataset with the same generative process as in panel (b). For clonealign we tested three scenarios where we input the ground truth data ('no noise'), the ground truth data upon stochastically flipping subclonal CNAs ('noise'), a subset of the original subclones ('partial'). We observe that the ARI of both methods equates until many clones are present (>5); the performance of clonealign partial degrades largely. (**f**) Performance of CONGAS in detecting clusters based on cellular proportion. Cases with accuracy above 70% are marked, showing the relation between cluster size and probability of detection. The line represents a logistic regression curve fit on the observed probability. The data are the same as panels (c, d)

discussed below on real data collected from one triple-negative breast cancer.

3.2 Subclonal CNAs in a triple-negative breast xenograft

We used CONGAS to analyze a triple-negative breast cancer dataset generated with 10× technology; we use this case study to validate our method against single-cell low-pass DNA data, used initially for clonealign (Campbell *et al.*, 2019). This dataset is the patient-derived xenograft SA501X2B collected from patient SA501, and has been used before to determine clone-specific phenotypic properties that associate with a complex clonal architecture, also validated by reproducing clonal dynamics over serial xenograft passages (Eirew *et al.*, 2015). From low-pass whole-genome CNA calling, the authors estimated three genetically distinct clones (prevalence 82.3%, 10.8% and 6.9%); one clone sweeping in next engraftments.

To run CONGAS, we retrieved the input genome segmentation from the largest clone identified in the original paper (82.3% of the cells). After retaining segments with at least 10 mapped genes and performing quality control, we retained $n = 503$ cells from which we could identify two of the three clones (Fig. 3a). The signals identified by CONGAS are clear across multiple segments, with particular strength on chromosomes 15, 16 and 18 (two-sided Poisson test, $p < 0.001$, Fig. 3b). This is consistent with low-pass analysis

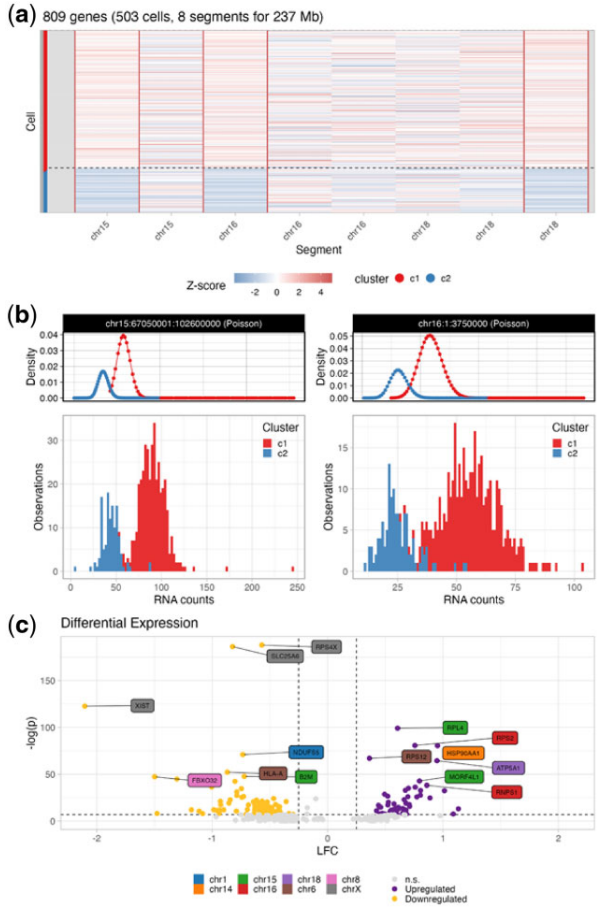


Fig. 3. (**a**) CONGAS analysis of $n = 503$ cells from a 10× assay from a triple-negative breast xenograft, where $k = 2$ populations are identified with 380 (∼75%), and 123 cells (∼25%). The heatmap shows input raw RNA counts (normalized per segment, with z-score) on chromosome 15, 16 and 18 where differences among CNAs are detected across the two subclones (red boxes). (**b**) RNA transcripts count for the genes mapping to a segment on chromosome 15, and one on 16. The densities on top of the histograms are the Poisson mixtures inferred by CONGAS. (**c**) Genome-wide clone-specific differential expression analysis highlights $n = 212$ dysregulated genes with adjusted $p < 0.01$ and absolute log-fold change (LFC) >0.25 (up-regulation) or <0.25 (down-regulated); notice that some of those genes do not overlap with CNAs that characterize the populations

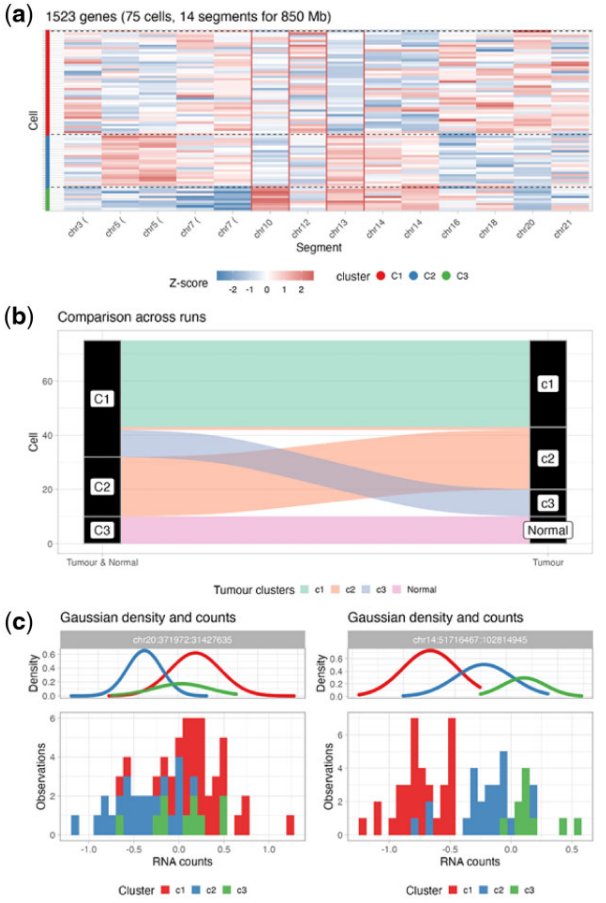


Fig. 4. (**a**) CONGAS two-steps analysis of $n = 75$ cells from a Smart-Seq assay of a glioblastoma. The analysis first identifies normal cells in the sample, and then reclusters tumour subclones; in the end, $k = 3$ subclones are identified with 32, 23 and 10 cells. The heatmap shows input raw RNA counts (normalized per segment, with z-score) for a segmentation obtained directly from B-allele frequencies in RNA, and clusters from the first run (normal cells have no CNAs). (**b**) Sankey plot of clustering assignments for the two runs. Cluster C3 from the first run are normal cells; tumour clusters are consistent across both steps of the analysis. (**c**) RNA transcripts count for the genes mapping to a segment on chromosome 20, and one on 14. The densities on top of the histograms are the Gaussian mixtures inferred by CONGAS, here used instead of Poisson because data were normalized

(Campbell *et al.*, 2019), validating our inference (Supplementary Figs S7 and S8). Our analysis, however, does not detect the third subclone from the original analysis; this was explained by observing that subclonal CNAs defining that population contain <10 genes, and have been removed from data. We note, however, that this cluster is poorly supported also in Campbell *et al.* (2019), which reports assignment uncertainty between the second largest clones and this population. Moreover, we tested if clonealign could have been used with a bulk whole-genome, instead of a low-pass single-cell one, to detect such cluster. In particular, we run the subclonal copy number caller ReMixT (McPherson *et al.*, 2017) on bulk data from the primary tissue of SA501, and used its results as input for clonealign. Consistently with our analysis, in this case, the tool was unable to discriminate the different populations (Supplementary Fig. S7).

The populations identified by CONGAS show significant differences in RNA counts (Fig. 3b and Supplementary Fig. S9): the largest subclone consists of $n = 380$ cells (_75%), and the smallest one of $n = 123$ (_25%). We also performed clone-specific differential expression analysis with the DeSeq2 (Love *et al.*, 2014) and found (Fig. 3c) 122 genes significantly up-regulated or down-regulated using a Wald test over negative binomial coefficients (adjusted $p < 0.001$ via Benjamini–Hochberg correction), imposing absolute log-fold change (LFC) >0.25 to determine the regulatory state (Fig. 3c). Note that some of these genes do not overlap with CNAs, and therefore could only be marginally explained by genetic changes. Instead, they might be explained by more complex regulatory mechanisms indirectly linked to these, and other events. Library factors were also found quite variable across cells (Supplementary Fig. S11).

We tested these data with inferCNV and copyKAT as well. Consistently with trends observed in simulations, while the true CNAs are identified even by these methods, the final number of clones is overestimated and spurious clusters are reported (Supplementary Materials and Figs S12 and S13).

3.3 Tumour normal deconvolution in primary glioblastoma

We used CONGAS to analyze the glioblastoma Smart-Seq data released in Patel *et al.* (2014). This dataset consists of $n = 430$ cells from five primary glioblastoma, from which we analyzed patient MGH31 ($n = 75$ cells). MGH31 was chosen as it harbours subclones, according to both the original paper and a successive analysis (Fan *et al.*, 2018). With this scRNA CONGAS was mainly

challenged by (i) the lack of an input CNA segmentation and (ii) the presence of normal cells in the assay (also known from the original analysis). To process this sample, we have created a simple pipeline around CONGAS.

We have first developed a variational Hidden Markov Model to segment B-allele frequencies from germline single nucleotide polymorphisms called by scRNA (Supplementary Material). In this way, we obtained segments with evident losses of heterozygosity, as well as large amplifications on chromosomes 7, 10, 13 and 14 (Supplementary Fig. S14). In a first run (Fig. 4b), CONGAS identifies $k = 3$ clusters from all cells (normal plus tumour); one of them ($n = 10$) lacked any CNA. The very same set of cells were classified as 'normal' by a comparison with a healthy reference (Fan *et al.*, 2018). We removed normal cells and rerun CONGAS on the remaining tumour cells, finding $k = 3$ distinct subclones (Fig. 4a–c). These two-steps results were consistent with a solution with $k = 4$ clusters, obtained in the first run. Manual phylogenetic reconstruction after CONGAS suggested an early branching from an ancestor harbouring chromosome 7+ (amplification) and 10− (deletion). Clones then branched out: one sustained by 5+ (34% of cells), while a linear path described the evolution of two nested clones with increasing aneuploidy (with the largest subclone with 34% of cells), distinguished by 14− but harbouring the same deletion on chromosome 13 (Supplementary Fig. S14).

The DE analysis of these few cells was inconclusive due to the small number of sequenced cells (data not shown); nonetheless this two-steps analysis shows how CONGAS can perform signal deconvolution in the presence of normal contamination of the input sample. This is interesting and consistent with the fact that the method can work without a reference normal expression.

We note that this data have been analyzed also with inferCNV, honeyBADGER and CaSpER (Fan *et al.*, 2018; Patel *et al.*, 2014; Serin Harmanci *et al.*, 2020). In all three cases, however, only two clones were found, one characterized by 5+, and another characterized by 13− and 14−, in substantial agreement with our analysis. However, our analysis is higher resolution, since it splits the latter clone based on the presence or absence of 13− (Supplementary Fig. S13).

3.4 Monosomy of chromosome 7 in haematopoietic cells

To show the versatility of CONGAS we have also analyzed mixtures of non-cancer cells collected within one experiment associated with the Human Cell Atlas project (Rozenblatt-Rosen *et al.*, 2017). In this case, the dataset provides scRNA from haematopoietic stem and progenitor cells from the bone marrow of healthy donors and patients with bone marrow failure. We focussed on one patient (patient 1) with severe aplastic anaemia that eventually transformed in myelodysplastic syndrome, and for which cytogenetic analyses revealed monosomy of chromosome 7, a condition that increases the risk of developing leukaemias (Zhao *et al.*, 2017).

To analyze this data, we pooled patient 1 together with one healthy donor, gathering $n = 101$ total cells (Supplementary Fig. S15). This gives CONGAS both diploid cells (control, from the health patient), and cells with chromosome 7 deletion.

This dataset comes without a suitable input segmentation, so we used full chromosomes (arm-level segments) with a diploid prior. Aneuploid cells were clearly distinguished from diploid cells by CONGAS, which found $k = 2$ clusters. One, containing diploid cells from both patients, the other cells from patient 1 that are associated with monosomy of chromosome 7. Clone-specific differential expression performed as for the breast xenograft reported 99 genes differentially expressed at significance level $p < 0.01$, and with absolute log-fold change >0.25. Interestingly, the top dysregulated genes were not expressed in the aneuploid chromosome, suggesting that an integrated study of transcriptomics and CNAs could lead to a better understanding of how these genomic events—which have considerable dimension—can alter cellular behaviour across different pathways and functional modules.

4 Discussion

In this article, we presented CONGAS, a Bayesian method to detect CNAs that can cluster scRNA sequencing profiles, opening the way to study tumour subclonal composition at the single-cell copy number level. CONGAS requires inputs that can be generated by following a split design, leveraging both bulk and single-cell assays. In this way, the inference is easier and more precise compared to methods that call CNAs directly from scRNA. The method compares also against methods that assign scRNA to subclonal CNA profiles, with the main advantage of being unsupervised. In this sense, input clonal CNAs are used to build a Bayesian prior to detect subclonal CNAS in single cells. In other approaches, instead, the clusters are predetermined and cannot mutate during the cell-assignment process.

CONGAS also has other interesting features. First, it does not require a normal RNA reference from a matched tissue, or the presence of normal cells in the sample. This means that it can find subclones with different CNAs regardless of reference expression, a major advantage in organoids designs where we do not collect non-tumour cells (Vlachogiannis *et al.*, 2018). Second, CONGAS reconciles copy number heterogeneity from RNA using a probabilistic model for cell assignment. Compared to callers that do not attempt clone detection or that separate calling from clustering, the advantage is that uncertainty is modelled in a unique framework, both for copy number estimation and clustering assignments. Third, the method uses a powerful probabilistic programming backend to scale to thousands of cells, overcoming computational limitations of other methods (Supplementary Figs S15 and S16).

CONGAS can be used to curate clonal evolution models (Caravagna *et al.*, 2016, 2018), or to assess clone-specific phenotypic signatures at the RNA level. This mapping comes out as a byproduct of the integration of genetic copy number events together with RNA data. With CONGAS one detects CNA-associated subclones and their patterns of differential expression, a key step to study how selective pressures shape genotype and phenotype evolution in cancers (Caravagna *et al.*, 2020). In addition, CONGAS is also able to correctly estimate the magnitude of subclonal copy number events. Which together with the input segmentation obtained from bulk sequencing, allow the estimation of the subclonal karyotypic profiles (Supplementary Fig. S17 and Materials).

This work offers a complementary perspective to DNA-only methods, for which many single-cell CNA detection algorithms have been developed (Garvin *et al.*, 2015; Kuipers *et al.*, 2020; Macintyre *et al.*, 2018; Wang *et al.*, 2018; Zaccaria and Raphael, 2020). Working with DNA, these methods can infer a *de novo* segmentation of the tumour genome—i.e. without prior input segmentations—and in the future, it will be key to integrate ideas at the core of these models together with RNA-genotyping methods such as CONGAS. Notably, in this work, we also show—across multiple case studies—that we can determine clone-specific differentially expressed genes that can be explained only partially by copy numbers, pointing to complex non-trivial regulatory mechanisms that link genotype states with expression patterns. Our method provides a solid statistical framework to approach this type of investigation, which is crucial to determine disease clonal dynamics, as well as cell plasticity and patterns of drug response from the large wealth of single-cell data available nowadays.

Author contributions

S.M., R.B. and G.C. conceptualized and created CONGAS, with support from L.P. and N.C. S.M. and R.B. implemented the tool; S.M. ran synthetic tests, and collected data for the case studies with support from N.C. and R.B. All authors analyzed the data and interpreted the results. G.C. and S.M. drafted the manuscript, which all authors approved in final form.

Funding

The research leading to these results has received funding from AIRC under MFAG 2020-ID. 24913 project—P.I. Caravagna Giulio.

Conflict of Interest: none declared.

References

Acar,A. *et al.* (2020) Exploiting evolutionary steering to induce collateral drug sensitivity in cancer. *Nat. Commun.*, **11**, 1923.

Bingham,E. *et al.* (2019) Pyro: deep universal probabilistic programming. *J. Mach. Learn. Res.*, **20**, 1–6.

Blei,D.M. *et al.* (2017) Variational inference: a review for statisticians. *J. Am. Stat. Assoc.*, **112**, 859–877.

Campbell,K.R. *et al.* (2019) clonealign: statistical integration of independent single-cell RNA and DNA sequencing data from human cancers. *Genome Biol.*, **20**, 54.

Caravagna,G. (2020) Measuring evolutionary cancer dynamics from genome sequencing, one patient at a time. *Stat. Appl. Genet. Mol. Biol.*, **19**, 20200075.

Caravagna,G. *et al.* (2016) Algorithmic methods to infer the evolutionary trajectories in cancer progression. *Proc. Natl. Acad. Sci. USA*, **113**, E4025–E4034.

Caravagna,G. *et al.* (2018) Detecting repeated cancer evolution from multi-region tumor sequencing data. *Nat. Methods*, **15**, 707–714.

Caravagna,G. *et al.* (2020) Subclonal reconstruction of tumors by using machine learning and population genetics. *Nat. Genet.*, **52**, 898–907.

Eirew,P. *et al.* (2015) Dynamics of genomic clones in breast cancer patient xenografts at single-cell resolution. *Nature*, **518**, 422–426.

Fan,J. *et al.* (2018) Linking transcriptional and genetic tumor heterogeneity through allele analysis of single-cell RNA-seq data. *Genome Res.*, **28**, 1217–1227.

Garvin,T. *et al.* (2015) Interactive analysis and assessment of single-cell copy-number variations. *Nat. Methods*, **12**, 1058–1060.

Greaves,M. and Maley,C.C. (2012) Clonal evolution in cancer. *Nature*, **481**, 306–313.

Househam,J. *et al.* (2021) Integrated quality control of allele-specific copy numbers, mutations and tumour purity from cancer whole genome sequencing assays. *bioRxiv 2021.02.13.429885*; doi: https://doi.org/10.1101/2021.02.13.429885.

Kuipers,J. *et al.* (2020) Single-cell copy number calling and event history reconstruction. *BiorXiv*, 2020.04.28.065755.

Lähnemann,D. *et al.* (2020) Eleven grand challenges in single-cell data science. *Genome Biol.*, **21**, 31.

Love,M.I. *et al.* (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.

Macaulay,I.C. *et al.* (2015) G&T-seq: parallel sequencing of single-cell genomes and transcriptomes. *Nat. Methods*, **12**, 519–522.

Macintyre,G. *et al.* (2018) Copy number signatures and mutational processes in ovarian carcinoma. *Nat. Genet.*, **50**, 1262–1270.

Martincorena,I. (2019) Somatic mutation and clonal expansions in human tissues. *Genome Med.*, **11**, 35.

Martincorena,I. *et al.* (2015) Tumor evolution. High burden and pervasive positive selection of somatic mutations in normal human skin. *Science*, **348**, 880–886.

McGranahan,N. and Swanton,C. (2015) Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. *Cancer Cell*, **27**, 15–26.

McGranahan,N. and Swanton,C. (2017) Clonal heterogeneity and tumor evolution: past, present, and the future. *Cell*, **168**, 613–628.

McPherson,A.W. *et al.* (2017) ReMixT: clone-specific genomic structure estimation in cancer. *Genome Biol.*, **18**, 140.

Patel,A.P. *et al.* (2014) Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, **344**, 1396–1401.

Picelli,S. *et al.* (2014) Full-length RNA-seq from single cells using Smart-seq2. *Nat. Protoc.*, **9**, 171–181.

Rozenblatt-Rosen,O. *et al.* (2017) The Human Cell Atlas: from vision to reality. Nature, **550**, 451–453.

Sarkar,A. and Stephens,M. (2021) Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nat. Genet.*, **53**, 770–777.

Serin Harmanci,A. *et al.* (2020) CaSpER identifies and visualizes CNV events by integrative analysis of single-cell or bulk RNA-sequencing data. *Nat. Commun.*, **11**, 89.

Turajlic,S. *et al.* (2019) Resolving genetic heterogeneity in cancer. *Nat. Rev. Genet.*, **20**, 404–416.

Vlachogiannis,G. *et al.* (2018) Patient-derived organoids model treatment response of metastatic gastrointestinal cancers. *Science*, **359**, 920–926.

Wang,X. *et al.* (2018) DNA copy number profiling using single-cell sequencing. *Brief. Bioinform.*, **19**, 731–736.

Wang,X. *et al.* (2021) Direct comparative analyses of 10x genomics chromium and Smart-seq2. *Genomics Proteomics Bioinformatics*, **19**, 253–266.

Watkins,T.B.K. *et al.* (2020) Pervasive chromosomal instability and karyotype order in tumour evolution. *Nature*, **587**, 126–132.

Zaccaria,S. and Raphael,B.J. (2020) Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nat. Biotechnol.*, **39**, 207–214.

Zappia,L. *et al.* (2017) Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.*, **18**, 174.

Zhao,X. *et al.* (2017) Single-cell RNA-seq reveals a distinct transcriptome signature of aneuploid hematopoietic cells. *Blood*, **130**, 2762–2773.

.

4.2 [DNA] + [RNA] + [ATAC] CONGAS+

4.2.1 Introduction

Cancer is a disease in which cell subpopulations with enhanced functional capabilities emerge, evolve and undergo selection against the immune system response and treatment [191]. The investigation of the evolution and heterogeneity of single tumours in terms of omics layers – e.g., genome, transcriptome, proteome, epigenome, metabolome – has key translational repercussions [135], and can benefit from the widespread availability of single-cell sequencing technologies [147] such as those that can probe RNA (scRNA-seq), DNA (scDNA-seq) and ATAC (scATAC-seq), generated from biopsies and patient-derived model systems [192]. With the current technologies, the most recent protocols can extract multiple measurements from the very same cell (e.g., G&T macaulay2015g or GoT [122] for matched scRNA and scDNA, or 10x multiome [160] for scATAC/ scRNA), even if "multimodal" technologies have still limited diffusion because they are very expensive and relatively low throughput. For this reason, a much more common single-cell design is based on separating cells before sequencing, with many computational efforts focused on integrating, a posteriori, data generated from different pools of cells.

In this second scenario, sometimes referred to as diagonal integration [159], the general idea is to map the two (or more) measurements in a latent space, using some unsupervised integration method. If we do not need the latent space to reflect any specific biological quantity, methods based on variational autoencoders or factor analysis can be adopted [84, 121, 136]. Otherwise, when it is required that the latents are biologically interpretable, other approaches should be preferred. In the context of cancer genomics and tumour evolution studies, one possibility to reconcile RNA and ATAC measurements stems from the observation that both capture distinct aspects of the same DNA molecule. In this sense, RNAs are products of the transcriptional processes that initiate from DNA, and ATAC is an assessment of chromatin conformation, a physical feature of DNA. Therefore, an interesting attempt – which is the one we follow in this work – is mapping RNA and ATAC on latent DNA states. Moreover, because we are interested in cancer genomics, another layer of complexity is introduced by observing that the latent DNA configuration can differ among tumour clones, i.e., subgroups of cells that are characterised by multiple types of genetic lesions such as point mutations and larger Copy Number Alterations (CNAs).

While point mutations are difficult to characterise and link to RNA and ATAC, the opportunity of modelling latent CNAs seems more feasible. In this case, the possibility of inferring latent tumour subclones from scRNA-seq has been already widely investigated [194, 38, 150, 111, 165, 89], and some preliminary attempts at working with scATAC-

seq are also present [183, 166, 174]. In this framing, we recently introduced CONGAS, a method to perform this CNA-based integration from scRNA-seq data. CONGAS, starting from a pre-defined genome segmentation (set of breakpoints), used a Bayesian probabilistic model to infer latent total CNAs (i.e, per segment ploidy estimates) while clustering input cells. CONGAS was the first model to join signal deconvolution (i.e., clustering), while detecting subclonal patterns of aneuploidy, and worked much better than methods like InferCNV [38], HoneyBADGER [89], CopyKAT [165] and Numbat [190] that performed first CNA inference, and then decoupled clustering. The solution achieved with CONGAS was however only partially satisfactory, because the statistical signal of CNAs in scRNA-seq is generally found to be affected by strong confounders such as allele-specific expression and post-transcriptional regulation, two biological phenomenon that are only partially understood and play an important role in cancer [15, 177, 157]. In practice, the distribution of read counts in RNA space (the inverse of the latent mapping), is not a perfect predictor of CNAs, and a better-quality signal can instead be achieved by examining chromatin conformation, a direct measurement of DNA. As in CONGAS, one can leverage the intuition that the more alleles (i.e., copies) of a chromosome region are open, the stronger the signal of ATAC on the region should be. In this sense, a model a-la-CONGAS could be developed to link the latent CNA to the observable ATAC peaks. As far as we understand, such an intuition has never been leveraged before, missing the possibility of integrating independent scRNA-seq and scATAC-seq measurements using a biology-informed latent model of DNA alterations.

Building on this intuition, in this paper we develop CONGAS+, a Bayesian probabilistic graphical model to map single-cell RNA and ATAC measurements in a latent copy number space, clustering cells across the two data modalities, and predicting clones with a well-defined discrete copy number profile. Doing so, the CONGAS+ framework naturally allows one to estimate and compare both the gene expression and the chromatin accessibility profiles of copy number clones, also separating tumour from normal cells when the former are characterised by aneuploidy, a very common situation in cancer. The model is unsupervised, and the likelihoods of RNA and ATAC are computed separately but conditioned on the same latent CNAs, but combined thanks to a shrinkage statistics that allows to weight the contribution of the data modalities unevenly, which helps when one of the two modalities (usually RNA) is a worst predictor of CNAs. The overall model uses stochastic variational inference and gradient descent to learn parameters from data, and enjoys a fast implementation via probabilistic programming in Pyro [110]. This allows deploying CONGAS+ on GPUs seamlessly, analysing datasets with tens of thousands of cells in matters of minutes thanks to the massively parallel architectures offered by graphical devices. In this paper we show the application of CONGAS+ to three real-world datasets from (i) basal cell carcinoma (1200 cells RNA, 1200 cells ATAC), (ii) B-cell lymphoma (6400 cells RNA+ATAC multimodal), (iii) prostate can-

cer cell line LNCaP (7600 cells RNA, 8800 cells ATAC). In all cases we observe that CONGAS+ is a solid statistical method to distinguish tumour from normal cells, and to detect subclonal copy number events that distinguish distinct populations of cancer cells.

4.2.2 The CONGAS+ model

4.2.2.1 Relationship between single-cell signal and copy number.

The goal of CONGAS+ is that of identifying subsets of cells that are characterized by the same copy number value over each segment. We employ a bulk DNA sequencing experiment output to identify the segments, and we use their coordinates to aggregate the RNA and ATAC signal of each single cell over each segment. This choice is motivated by the goal of identifying a tradeoff between reliability of the model and overall cost of the analysis: compared to a single-cell assay, bulk DNA-seq experiments are characterized by a lower cost and they can be exploited to confidently identify Copy Number segments through a wide range of well-established tools, such as Sequenza [48], CNVkit [66] and GATK best practices [154]. Through this step we can model segments as independent entities, using as input to the model the aggregated counts for each single cell. CONGAS+ models the observed counts for each single-cell in each segment as variables dependent on the copy number state of that segment: once counts have been aggregated, we assume that the hidden copy number state influences the gene expression and chromatin accessibility signal through a linear relation: this is based on the intuition that if one tumor cell loses one of the two alleles in one segment, the genes on that segment will be characterized by a lower signal compared to the same genes in a normal (i.e., diploid) cell. This dependence has already been succesfully exploited on RNA in [111] and in our previous work [194], and we extended this to the chromatin accessibility signal: higher or lower Copy Number values for a specific segment predict more or less transcripts for those genes mapped to the segment, and amount of open chromatin on the segment.

4.2.2.2 Full formulation of CONGAS+ statistical model

The model takes in input two single cell datasets X^{atac} and X^{rna} of sizes $N^{atac} \times I$ and $N^{rna} \times I$, containing respectively the signal of RNA and ATAC cells per segments. These two matrices are the result of a pre-processing step, were the counts are aggregated over each segment i by summing all the features that map to i, with $i = 1, \dots I$. RNA features correspond to genes, while ATAC features correspond to peaks or fixed-length genome bins. Both these two matrices can either be non-normalised integer counts or normalised values, where in the latter case, we compute the z-score for each input feature (i.e., gene for scRNA and peak/genomic bin for scATAC). The type of input provided determines the distribution used to model the data, as it is explained in the next section.

CONGAS+ is a finite Dirichlet mixture of $K \geq 1$ distributions that model the K clones present in the single-cell samples. The likelihood of our model has the following form:

$$p(X^t|\Delta^t, \Phi, \pi^t) = \prod_{n=1}^{N^t} \sum_{k=1}^{K} \pi_k^t \prod_{i=1}^{I} f(x_{n,i}^t|\Delta^t, \Phi), \tag{4.1}$$

where N^t is the number of cells for modality t, K the number of clusters and I the number of segments. Here f is a generic likelihood function which models the observed signal for the omic, π^t are the clusters mixing proportions and Φ the probability distribution over discrete copy number values for each cluster and each segment: each of the k clusters is associated to a probability distribution per segment $\phi_{k,i,h} = P(C_{k,i} = h)$ over the possible copy number values $h = 1, .., H$ that the $i - th$ segment may assume, where by default, $H = 5$. The prior on the probabilities is a Dirichlet distribution

$$\Phi_{k,i,h} \sim Dirichlet(\alpha), \qquad \alpha = (\alpha_1, ..., \alpha_5), \;\; \alpha_i \in \mathbb{R}_{>0}, \tag{4.2}$$

where the concentration vector α is a hyperparameter chosen by the user. Given the ploidy p of the segment, one may choose $\alpha = (\alpha_1 = 0.1, ...\alpha_p = 0.6, ...\alpha_5 = 0.1)$. Note that the tensor Φ does not change between modalities since cells from both omics are assigned to the same set of clusters.
CONGAS+ accommodates settings where the two omics have clusters in different proportions, which is achieved by using two different vectors π^{atac} and π^{rna}, that model the mixing proportion for each cluster. Each entry π_k is sort by another Dirichlet distribution:

$$\pi_k^{rna} \sim Dirichlet(\beta^{rna}), \qquad \pi_k^{atac} \sim Dirichlet(\nu^{atac}) \tag{4.3}$$

where we choose $\nu^{rna} = \nu^{atac} = (1/K, ..., 1/K)$. Our model allows also to have a shared parameter π to control the mixing proportions in ATAC and RNA jointly.

The generic likelihood function f in Equation eq. (4.1) is defined based on the type of input matrix provided. On the one hand, for integer count matrices use Negative Binomial (NB) distributions to model the observed signal. On the other hand, with normalised counts each feature is z-scored prior to aggregating the signal over each segment and f is defined as a Gaussian likelihood.

Joint likelihood The joint scRNA/scATAC CONGAS+ log-likelihood has a shrinkage form:

$$P(\mathbf{X}|\mathbf{\Delta}, \Phi) = \lambda \times \log(p(X^{rna}|\Delta^{rna}, \Phi)) + (1 - \lambda) \times \log(p(X^{atac}|\Delta^{atac}, \Phi)), \tag{4.4}$$

where λ is an hyperparameter to weight the likelihood of both modalities, $\mathbf{X}$ are datasets X^{atac} and X^{rna}, and Δ are parameters of the model Δ^{atac} and Δ^{rna}.

Integer count matrix With segment-specific integer counts we use the Negative Binomial (NB) likelihood,

$$p(x_{ni}^t|\rho_n^t, \theta_i^t, r_i^t, \Phi_{k,i}) = NB\left(\frac{\mu_{k,i,n}}{\mu_{k,i,n} + r_i^t}, r_i^t\right), \tag{4.5}$$

where $\mu_{k,i,n}$ and r_i^t are the mean and size of the NB, respectively. The prior distribution on r_i is $r_i \sim Unif(a_i^t, b_i^t)$ for some choice of the extrema a_i^t, b_i^t. The mean of the NB is defined as

$$\mu_{k,i,n} = \theta_i^t \cdot (\textstyle\sum_h \omega_{k,i,h} \cdot h) \cdot \rho_n^t \tag{4.6}$$

Here θ_i^t are omic-specific variables that represent the average signal of a single copy of the i-th segment. For these quantities we choose a Gamma prior

$$\theta_i^t \sim \Gamma(\alpha_i^t, \beta_i^t), \tag{4.7}$$

where the hyperparameters α_i^t, β_i^t can be estimated from the data.
Through the definition of the Negative Binomial mean described in eq. (4.6), we are modeling our assumption that the observed signal for each cluster in each segment is proportional to the number of DNA copies of that segment for that specific cluster.

In fact, given the distribution $\Phi_{k,i}$ over possible discrete CNA values $h \in 1, \ldots, H$ (by default $H = 5$) for each cluster k and segment i, samples from this distribution are represented by $\Omega_{k,i}$, that are one-hot vectors of length H whose positions $h = 1, \ldots, H$ correspond to the copy number value. Thus, to obtain the CNA for cluster k on segment i we sum each entry $\omega_{k,i,h}$ multiplied by h, and then we multiply the obtained CNV by the latent parameter θ_i, scaling counts by DNA ploidy.
We also introduce the cell specific normalization factors ρ_n^t, which take into account possible expression differences due to sequencing. These are hyperparameters of the model and can be estimated from the data.

Normalized count matrices CONGAS+ supports also datasets where where prior to aggregation, each feature has been z-scored. In this case we assume the aggregated signal over each segment to be normally distributed, with the mean equal to the copy number value:

$$p(x_{ni}^t|\Phi_{k,i}, \sigma_i^t) = N\left(\mu_{k,i}, \sigma_i^t\right) \tag{4.8}$$

where

$$\mu_{k,i} = \sum_h \omega_{k,i,h} \cdot h \tag{4.9}$$

and the standard deviation σ_i is sorted by a uniform distribution

$$\sigma_i \sim Unif(a_i^{rna}, b_i^{rna}). \tag{4.10}$$

The definition of mean of the Gaussian presented in eq. (4.9) is analogous to that of the Negative Binomial in eq. (4.6): $\Omega_{k,i}$ are indicate samples from the distribution $\Phi_{k,i}$ over possible discrete CNA values $h \in 1, \ldots, H$ (by default $H = 5$) for each cluster k and segment i. $\Omega_{k,i}$ are one-hot vectors of length H where the positions $h = 1, \ldots, H$ correspond to the copy number value, and by summing each entry $\omega_{k,i,h}$ multiplied by h we obtain the CNA for cluster k on segment i.

Parameters estimates Our inference algorithm requires marginalizing the likelihood with respect to the clustering and copy number assignment and calculating the Maximum A Posteriori (MAP) MAP estimates of the continuous parameters.
Once the MAP estimators have been computed, we can compute the copy number profile of each cluster $C_{k,i}$ by taking $C_{k,i} = \arg\max_h (\phi_{k,i,h})$. Given also the copy number states, one can compute the clustering assignment probabilities $P^t_{n,k}$ of the cells for both modalities. These are

$$P^t_{n,k} = \frac{\pi^t_k \prod_i f\left(x^t_{n,i} | \Phi_{k,i}, \Delta^t\right)}{\sum_k \pi^t_k \prod_i \left(f\left(x^t_{n,i} | \Phi_{k,i}, \Delta^t\right)\right)} \tag{4.11}$$

Using the above probabilities one can estimate for each cell the assignment vector $z^t_{n,k}$

$$z^t_{n,k} = \begin{cases} 1 \text{ if } k = \text{argmax}\left(P^t_{n,k}\right) \\ 0 \text{ otherwise} \end{cases} \tag{4.12}$$

The Probabilistic Graphical Model(PGM) of CONGAS+ for the Negative Binomial distribution is presented in fig. 4.1.

4.2.2.3 Variational inference for parameter estimation

Given the model definition presented above, we want to estimate the values for all parameters by learning their posterior distribution, defined using the Bayes rule. For simplicity, we use U to indicate all the parameters in the model and thus we can write the posterior as

$$P(U|X) = \frac{P(X|U)P(U)}{P(X)} \tag{4.13}$$

Where $P(X)$ is the marginal likelihood, also called evidence. The denominator is usually intractable, and thus we need to approximate the real posterior. CONGAS+, like CONGAS uses Stochastic Variational Inference (SVI) [31] to get the approximation of the true posterior $p(U|X)$. The aim of SVI is to find a variational distribution $q(U)$ that belongs to a family of probability distributions $\mathcal{Q}$ and can approximate the real posterior. This can be formulated as an optimization problem, where the goal is to minimize the Kullback-Leibler (KL) divergence between $p(U|X)$ and $q(U)$ [31, 57]:

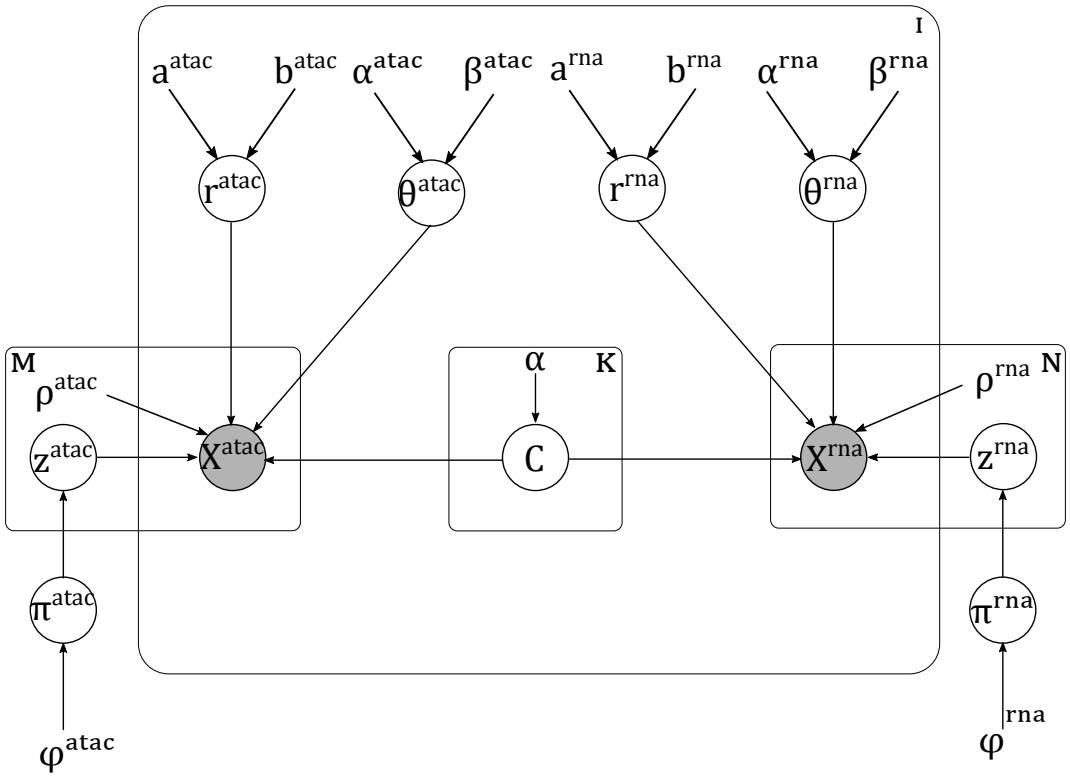


Figure 4.1: PGM of CONGAS+

$$q^*(U) = \arg\min_{q(U)\in\mathcal{Q}}\{KL(q(U)||p(U|X))\} \tag{4.14}$$

However, this term is still untractable, as it requires to compute the posterior. Thus, the objective function that gets optimized in SVI is the Evidence Lower Bound (ELBO):

$$\text{ELBO}(q) = \mathbb{E}\left[\log p(U,X)\right] - \mathbb{E}\left[\log q(U)\right]. \tag{4.15}$$

and maximizing this quantity is equivalent to minimizing the KL divergence [9, 13, 31]. In detail, the variational distribution q is parametrized by γ, which is what we want to learn during the inference, and in order to optimize the ELBO, SVI computes gradient descent optimization taking a Monte Carlo estimates of the gradient, which is defined as:

$$\nabla\gamma\text{ELBO} = \nabla_\gamma \mathbb{E}_{q_\gamma(U)}\left[\log p(x,u) - \log q_\gamma(U)\right]. \tag{4.16}$$

To reduce the search space we first identify segments with at least two clusters ($K > 1$) from an independent run, and then we retain only multi-modal segments. CONGAS+ is implemented in `Pyro` [110], a probabilistic programming language based on `Python` which implements SVI, we use *Adam* as an optimizer.

4.2.2.4 The Gumbel-Softmax distribution

CONGAS+ uses SVI to approximate the posterior by performing gradient descent. However, our model contains a discrete random variable Φ, that encodes the probability distribution over the possible discrete copy number values. In order to be able to estimate the gradient for this categorical variable, we use the Gumbel-softmax [64] (GSM), which is a continuous distribution that can approximate samples from a categorical distribution.
Considering the categorical probability distribution Φ which has H probability classes $\alpha_1, \alpha_2, \ldots, \alpha_H$ and models the distribution over the possible copy number values, a sample ω from such distribution can be seen as a one-hot vector that lies on the corners of a $(h-1)$-dimensional simplex Δ^{h-1}:

$$\omega = one\_hot(\text{argmax}_h[g_h + log\alpha_h]) \tag{4.17}$$

where g_h are i.i.d. samples drawn from Gumbel$(0,1)$ [64].

In order to approximate the argmax to make it continuous and differentiable, the softmax is employed. Thus, a H-dimensional sample vector from the Gumbel-softmax distribution is a vector $\omega \in \Delta^{h-1}$ defined as:

$$\omega_h = \frac{\exp\left((log(\alpha_h) + g_h)/\tau\right)}{\sum_{j=1}^{k}\exp\left((log(\alpha_j) + g_j)/\tau\right)}$$

Where $i = 1, 2, \dots k$ and τ is the temperature parameter. As τ approaches 0, the samples from the Gumbel Softmax become one-hot vectors and thus sampling from the GSM becomes identical to drawing samples from the categorical distribution Φ.

In the GSM definition, α is the vector of parameters of the distribution, and it corresponds to the vector of the probabilities for the H classes of the categorical distribution. For values of the temperature greater than zero, the GSM has a well defined gradient with respect to its parameters, and thus if we replace the categorical samples with the Gumbel Softmax it is possible to use backpropagation during training to compute the gradients.
However, the samples from the GSM are not identical to samples from the corresponding categorical distribution when τ is not zero and thus there is the need for identifying a trade-off between large and small temperatures. In fact, on the one hand for temperatures close to zero samples are close to one-hot, but the variance of the gradients is large. On the other hand, large temperatures yield small gradient variance but smooth samples. The solution is to decrease the temperature following a schedule: in Pyro we start from a value τ_{start}, and then at each step j of gradient descent optimization we use the temperature $\tau_j = \tau_{start} / \log(j + 0.1)$.

4.2.2.5 Model selection

CONGAS+ optimises the number of clusters in the finite mixture K via model selection, letting the user choose from three well-known metrics: Bayesian Information Criterion (BIC) [4], Akaike Information Criterion (AIC) [8] and Integrated Completed Likelihood Criterion (ICL) [10]. In particular, given the complete likelihood $P(X)$, the number of parameter $v(k)$ for a model with k components we can compute:

- BIC $= v(k)ln(n) - 2ln(L(k))$ where n is the number of cells in the dataset
- AIC $= 2v(k) - 2ln(L(k))$
- ICL $= BIC + H(Z)$ where Z is the latent variable modelling cell clustering assignments and $H(Z)$ is the entropy defined as $-\sum_i p(z_i) \log(z_i)$

Given a vector of possible K as a hyperparameter, we pick as the optimal one the value lowest score for the Information Criterion chosen.

4.2.2.6 Implementation

CONGAS+ is implemented in 2 open-source packages: one, in Python, implements the model in the probabilistic programming language Pyro [110], while the other, in `R`, provides data processing and model visualisation functions, interacting with Python through Reticulate.

4.2.3 Model validation and parameterization

4.2.3.1 Synthetic data simulations

Synthetic data simulations. We performed extensive simulations to assess the performance of CONGAS+ in scenarios with increasing complexity in terms of clonal composition, mimicking real data analysis settings. Noting that CONGAS+ is also a generative model, i.e., it can sample simulated data, in order to obtain a performance assessment that is unbiased and realistic, we opted to simulate scRNA and scATAC data outside of our tool.

We first simulated scRNA and scATAC of a normal cell population via simATAC [172] and SPARsim [109], two tools for synthetic data generation that recapitulate signals consistent with modern sequencing technologies. Then, we added CNAs for K clones (minimum 2, maximum 10) assembled from a clonal tree [196] with K nodes. Starting with a random CNA profile for the root (ancestral clone), we iteratively attached the remaining clones at random, generating its copy number profile by randomly changing the value of D=5 segments from the parent. Finally, we generated the clusters mixing proportions from a Dirichlet distribution with uniform concentration, and assigned cells to clusters. We simulated 10 replicas for each value of K, for a total of 90 synthetic datasets, each one including 1500 scRNA-seq and 1500 scATAC-seq cells.

We applied CONGAS+ to 90 datasets, searching for up to 10 clusters in each run, and measuring several standard performance metrics including (i) the Adjusted Rand Index (ARI), which assesses the similarity between the ground truth and the inferred cluster memberships, and (ii) the mean absolute error (MAE) between the inferred cluster-specific CNA profile, and the simulated one. Together, by combining (i) and (ii) we assessed the ability of CONGAS+ to retrieve the cells from each clonal population, and their copy number profiles. In Figure 4.2 we observe a very good performance, with a median ARI higher than 0.8 for every value of and a MAE consistently lower than 1, which is a good result, suggesting that, on average, CONGAS+ does not estimate copy number profiles that are too extreme with respect to the ground truth (Figure 4.2B).

4.2.3.2 The importance of using a joint assay.

CNA-associated signal quality is not necessarily even across ATAC and RNA, with the latter showing higher overdispersion due to difference in sample preparation, library size, gene expression variability, and sequence-specific biases [179, 188]. High overdispersion can act as a confounder, making it difficult to infer poorly separable clusters from scrNA-seq data alone, eventually leading to failures in detecting subclones. Using $\sim$ 1800 scRNA and $\sim$ 600 scATAC profiles from the Basal Cell Carcinoma (BCC) SU008 [133, 124], we created a dataset of tumour and normal cells in even proportions, subsetting the genome to two diploid and two with aneuploidy, with bimodal signal poorly evident in

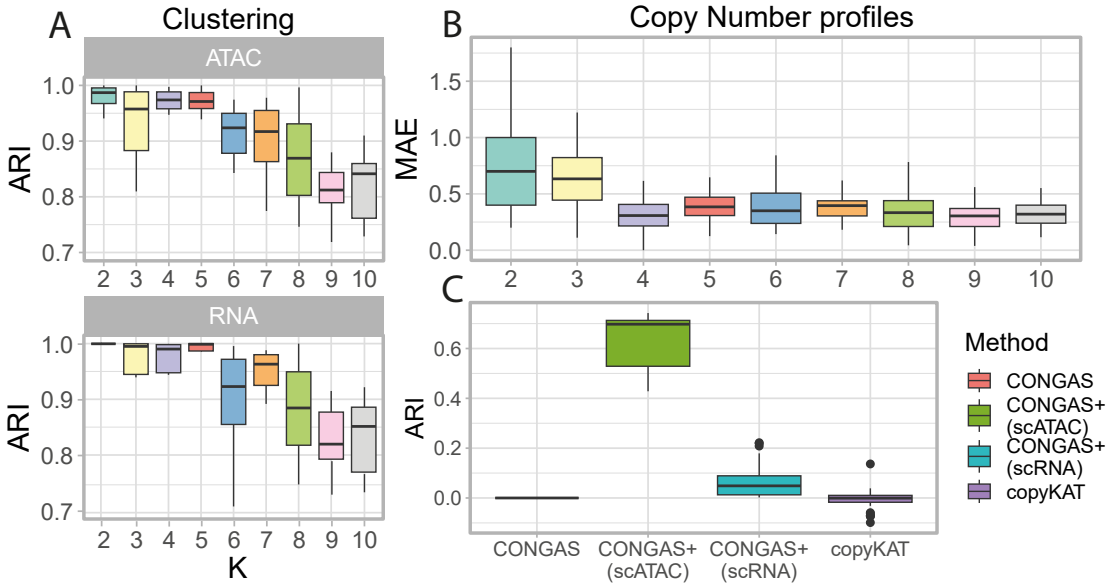


Figure 4.2: Validation of CONGAS+. A: Adjusted Rand Index (ARI) computed comparing the ground truth labels of simulated cells with the clustering assignments returned by CONGAS+. 90 datasets with 1500 cells for scRNA and 1500 cells for scATAC were simulated. For any number of ground truth clusters , the median ARI is always higher than 0.8. B: For the same data in panel (A), Mean Absolute Error (MAE) between the ground truth and the inferred copy number profiles. C: Boxplot showing the ARI for copyKAT [165], CONGAS [194] and CONGAS+ (computed on scRNA and scATAC separately) obtained running the tools on bootstrap samples characterised by a bimodal signal poorly evident in RNA.

RNA. Then, we performed non-parametric bootstrapping for the genes in each segment, and compared 30 inferences with CONGAS+ (RNA plus ATAC), CONGAS (RNA) and copyKAT (RNA). Using a joint ATAC-RNA assay, CONGAS+ with $\lambda = 0.1$ detected CNAs that distinguish tumour from normal cells, obtaining a median ARI ≈ 0.7 on ATAC but a lower ARI on RNA (Figure 4.2C). In general, due to the weaker RNA signal, all tools that looked only at RNA struggled separating tumour and normal cells, with copyKAT and CONGAS unable to detect the split (Figure 4.2C). In this test, copyscAT failed to execute with standard parameters. Overall, this shows that with a joint inference on the ATAC and RNA modalities we can detect the clonal structure of the dataset also when one data modality has a weak signal.

4.2.3.3 Shrinkage effect with Basal Cell Carcinoma data.

CONGAS+ likelihood depends on a shrinkage hyperparameter which serves to weigh differently the evidence available from the two datasets, which might not be even. This serves as a natural hyperprior to decrease the importance given to a modality that we believe is more noisy or affected by some consistent bias. A natural question is how does this affect the inference, and what value for should be suggested in the general case.
To this aim, we used CONGAS+ in two controlled scenarios: (i) one where the signal is present only in one modality and (ii) another where the signal is present in both. To make this more realistic, we exploited data from [133] and [124] which include scRNA-seq ($\sim$ 6400 cells), scATAC-seq ($\sim$ 6400 cells) and bulk WES data of Basal Cell Carcinoma (BCC) samples. The two datasets (SU006 and SU008) include a mixture of tumour and normal cells, and were subsampled to obtain similar cell proportions.
For sample SU006, we selected 2 diploid segments and 2 segments in a loss of heterozygosity (LOH) state, where the latter are characterised by a bimodal distribution in both ATAC and RNA counts. As shown in Figure 4.3A, the peaks in the bimodal distribution for LOH segments have lower dispersion in the ATAC with respect to the RNA. For this reason, we assessed whether this greater variability in the RNA signal might impact the inference results when varying λ. To do so, we set K=2 and performed 10 independent runs with $\lambda = \{0.05, 0.15, ..., 0.95\}$. For each run, we compared the cluster assignments with the ground truth labels via ARI (Figure 4.3C), observing for both modalities values stable against changes in λ, and CONGAS+ is always able to separate tumour from normal cells. For sample SU008, we selected 2 diploid segments and 2 segments with an amplification. In this case, the amplification is reflected only on the ATAC signal, which exhibits a neat bimodal distribution (Figure 4.3E). We employed the same design used for SU006, but this time we observed (Figure 4.3F) that for $\lambda < 0.5$ the ARI for ATAC cells is stable (ARI), whereas it decreases as λ approaches 1. In Figure 4.3G-H we report the inference result for the best and worst ARI values, respectively ($\lambda < 0.25$ and $\lambda < 0.95$), showing the percentage of tumour and normal cells assigned to each cluster.

With $0.25 < \lambda < 0.95$ CONGAS+ cannot fit the ATAC bimodal signal, merging 63% of tumour cells and 90% of normal cells in cluster C2. Instead, for $\lambda < 0.25$ it retrieves the bimodality and correctly identifies tumour cells from ATAC. As expected, the model is never able to retrieve tumour and normal clusters from RNA, due to its unimodal distribution.
Overall, these tests show that, with real data, the quality of the two assays might be different, and that tuning λ can be important to favour one assay over the other. Based on our experience, scATAC-seq tends to have a more clearly multimodal signal, arguably because it is a direct measurement of DNA, whereas RNA expression is more subject to complex nonlinearities. For this reason, values lower of λ can be used to favour ATAC signals. CONGAS+ offers a principled approach based on likelihood to inspect the optimal value of λ, and a final decision has to be taken on each and every datasets, also inspecting the quality of the fits.

4.2.4 CNA-associated drug-resistance clones in a prostate cancer cell line

Copy number events can lead to the emergence of complex cancer phenotypes, sometimes even capable of resisting negative selection induced by anticancer drugs. To test if CONGAS+ could identify clonal populations with associated CNAs that resist treatment, we collected data of the prostate cancer cell line LNCaP from [182], where scRNA-seq and scATAC-seq were performed on independent cells to study drug resistance.
In detail, Taavitsainen et al. sequenced the LNCaP parental cell line (DMSO), one line treated for 48 hours with AR antagonist enzalutamide (ENZ), and two resistant lines (RES-A and RES-B) derived after long-term exposure to ENZ and RD-162, respectively. To search for high-resolution subclonal CNAs we downloaded the copy number segmentation of the parental LNCaP from the DepMap portal [28], and used it to obtain breakpoint coordinates and priors for copy number values. We merged the 4 samples (parental, ENZ-48, RES-A, RES-B), and filtered out segments with more than 10% of cells showing zero counts.

CONGAS+ on the 4 samples identified 3 clusters present in both ATAC and RNA modalities. As shown in Figure 4.4B, the DMSO and ENZ-48 lines were clustered together in C1, while RES-A and RES-B were split in two clusters (C2 and C3), respectively. This is perfectly consistent with the experimental design [182]: in fact, ENZ48 has not yet acquired resistance to therapy due to the short-term drug exposure, and cluster C1 is composed only of cells from this line and the parental. Moreover, the two other clusters are composed of almost fully-resistant cells, with C2 sharing the greatest overlap with RES-A, and C3 mainly composed of cells from RES-B. These two clusters share specific CNAs such as an amplification (21q+) on the q-arm of chromosome 21 (Figure 4.4A), which is the main event to distinguish resistant RES-A and RES-B cells from

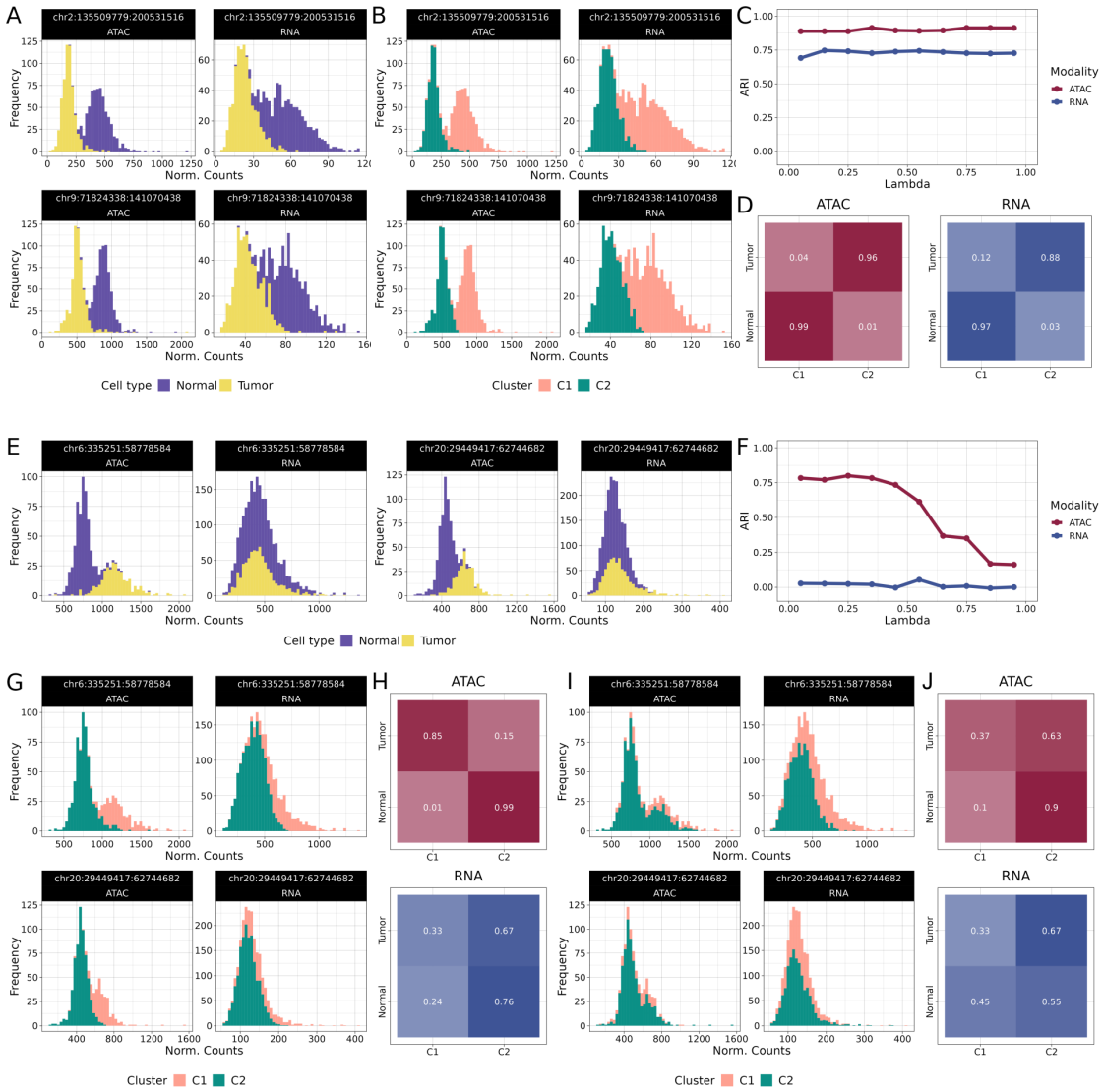


Figure 4.3: Impact of lambda variation on CONGAS+ performance using Basal Cell Carcinoma data. A,B,C,D: Basal Cell Carcinoma sample SU006 from , where we selected segments with bimodal signal in both scRNA-seq and scATAC-seq profiles. E,F,G,H: BCC sample SU008 from [133, 124], where we selected segments in which ATAC signal is bimodal and RNA unimodal. A,E: normalised counts distribution coloured by the ground truth cell labels. B,G: distributions coloured according to clustering assignments obtained from the solution showing the highest ARI. C,F: ARI value for each modality, computed for every lambda ranging from 0.05 to 0.95. D,H overlap between clustering assignments and ground truth labels for the solution with highest ARI . I-J: normalised distribution (I) and clustering overlap with ground truth labels (J) for the worst solution in terms of ARI.

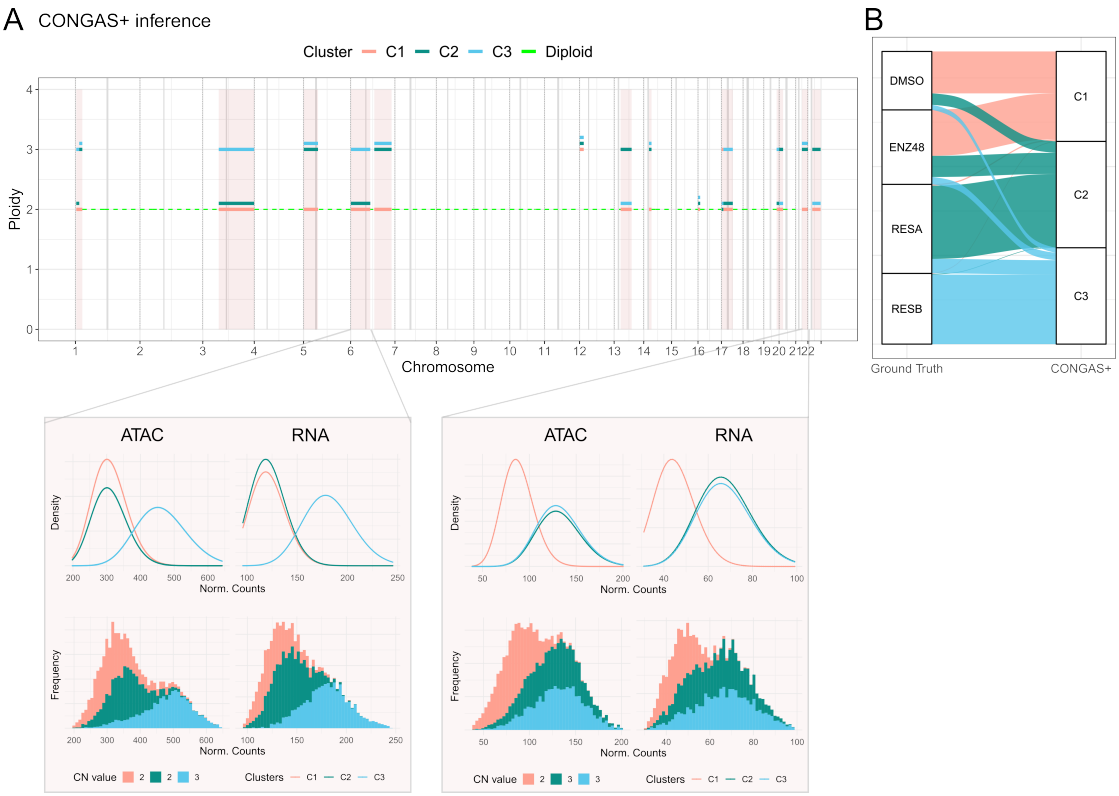


Figure 4.4: CONGAS+ application to a prostate cancer dataset from [45], composed of a mixture of four cell lines with 7600 scRNA-seq cells and 8800 scATAC-seq cel. A: copy number profiles inferred by CONGAS+ for each cluster, and density plots and histograms of normalised counts coloured according to cluster assignments for chromosome 6p where an amplification event is private to cluster C3, and chromosome chr21q where an amplification is shared by clusters C2 and C3. B: Sankey plot showing the overlap between the sample of origin and the cluster inferred by the model.

sensitive ones. Additionally, since the two resistant populations are clustered separately, this suggests that there are additional CNAs that characterise the two populations. In fact, CONGAS+ finds that RES-B cells have, on top of 21q+, also an amplification on the p-arm of chromosome 6 (6p+).

Overall, this analysis shows that complex lineage relation associated to tumour subclones with copy number alterations can be effectively detected by CONGAS+, laying the foundations for more systematic investigation on the causal roles of certain CNAs against therapy resistance.

4.2.5 ATAC and RNA phasing in B-cell lymphoma multimodal data

We used a multimodal dataset of B-cell lymphoma [1] sequenced with the 10x multiome kit [160] to test if CONGAS+ can identify clusters across the two modalities, and assign cells the same copy number clusters. This is the ideal test because this kit can measure both gene expression and chromatin accessibility from the same cell, which means that we can exploit the one-to-one correspondence between scRNA-seq and scATAC-seq barcodes to phase cells across modalities. Our expectation is that, even if we do not input it, all cells in the same RNA cluster should also be part of the same ATAC cluster. In other words, there should not be mismatches in clustering assignments across modalities.

We processed 6400 cells after quality control via Seurat [167] and Signac [181], for which cell types were manually annotated by the authors. Cell types are distinguishable in the UMAP low-dimensional representation [96] obtained by integrating RNA and ATAC via Seurat (Figure 4.5B): two tumour cell populations (B and B-cycling) are present and cluster together, whereas normal cells form three clusters that correspond to Monocytes, T and B cells. Note that, while we can expect to have distinct copy number profiles to tell apart normal from tumour cells, the distinction among B and B-cycling tumour subpopulations is unlikely to be explainable by CNAs, because cell cycle entry dynamics are linked to regulation in the ligation of the B-cell receptor complex and other receptor agonists [16].

We run CONGAS+ with chromosome arm-level segmentations and diploid default states a priors, and searched for K=2 clusters. We first compared the inferred clusters with the ground truth labels by computing the fraction of tumour and normal cells detected in each cluster. In Figure 4.5D and 3E, we see that 96% of tumour cells are assigned to cluster C2 for both modalities, whereas 96% and 98% of normal cells in ATAC and RNA respectively are assigned to cluster C1. This reflects the fact that CONGAS+ detects copy number events that distinguish tumour from normal cells. Moreover (Figure 4.5F), high values of ARI ≈ 0.85 and consistent confusion matrices are observed for the inferred clustering assignments and the cell types annotated in the original paper.

Overall, this tests demonstrated that CONGAS+ can assign independent observations to the same clustering structure with real multimodal data, opening the opportunity of using both modalities as well as independent assays within one unique framework. To the best of our understanding, in terms of CNA detection, CONGAS+ is the only model to work seamlessly with these two types of assays.

[1] `https://www.10xgenomics.com/resources/datasets/fresh-frozen-lymph-node-with-b-cell-lymphoma-14-k-sorted-nuclei-1-standard-2-0-0`

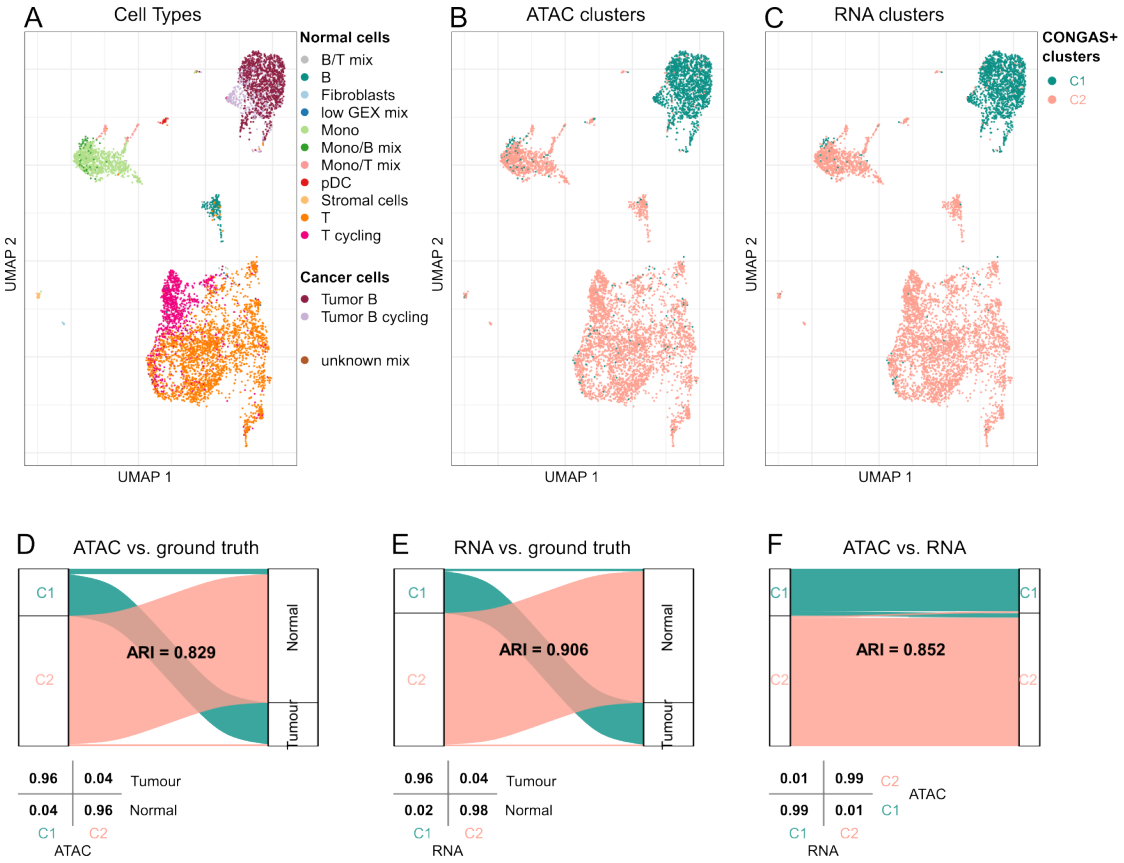


Figure 4.5: Application of CONGAS+ to prostate cancer cell line LNCaP with independent assays. CONGAS+ application to a prostate cancer dataset from [182], composed of a mixture of four cell lines with 7600 scRNA-seq cells and 8800 scATAC-seq cells. A: The copy number profiles inferred for each clone, focusing on two subclonal copy number events. The panels in the top row contain the density plots, whereas on the bottom row the histograms display the normalised counts for the inferred clusters. B: Sankey plot showing the overlap between the sample of origin and the cluster inferred by the model.

4.2.6 Discussion

We introduced CONGAS+, the first Bayesian method designed to cluster single-cell gene expression and chromatin accessibility profiles on copy number clones. Extensive simulations allowed us to prove its robustness and accuracy in retrieving both the ground truth clonal composition and the corresponding copy number profiles, by exploiting the signal of simulated scRNA-seq and scATA-seq data. This result was further confirmed through the application to real-world datasets.

By applying CONGAS+ on a B-cell lymphoma multiomic dataset, we could prove that our framework correctly assigns single cells from independent assays to the same copy number clones, also in absence of input WGS/WES bulk data. The case study on prostate cancer cell lines showed the capability of our method in inferring a complex subclonal architecture, which may allow one to compare both the gene expression and the chromatin accessibility states of the inferred genetic clones, with straightforward translational applications. Finally, the application of CONGAS+ to basal cell carcinoma data enabled us to quantitatively assess the effect of tuning the shrinkage hyperparameter λ, which can be used to shift the inference weight towards the most "reliable" data type. In this regard, CONGAS+ includes functions to inspect data distribution before performing the inference, so to set λ accordingly.

Extensions of CONGAS+ are underway. For example, the identification of single-nucleotide polymorphisms from scATAC-seq data would allow one to compute the B-Allelic Frequency (BAF) profiles, which would in turn enable the detection of complex copy number events, such as copy-neutral losses of heterozygosity. The BAF profiles might also be exploited in conjunction with the read counts signal to implement an algorithm for genome segmentation and copy number calling, without requiring any input bulk DNA data.

5

Task C: A comprehensive pipeline for reproducible single-cell analyses

Giving the availability of extensively used tools for pipeline development, such as Nextflow and Snakemake, it is possible to build modular frameworks to generate reproducible analyses and flexible pipelines. In this chapter we discuss the main contribution regarding task (C), presenting a currently ongoing effort to convey the work presented here into SIgMOIDAL, one reprodcible, modular, flexible and scalable pipeline for multi-omics data analysis and integration. SIgMOIDAL is being developed following the principles of open science and the FAIR guidelines for data management, with the goal of working towards the reproducibility of the results. We note that data analysed through the pipeline are not shared in this thesis, as they have been produced as part of the Accelerator Award project #22790 and will be made available upon publication of the results.

SIgMOIDAL combines different modules that were developed over the past three years. However, the pipeline is modular and it will be adapted to include new strategies, based on most recent developments in the field of computational biology. The current modules combine both State-Of-The-Art (SOTA) methods and frameworks that were developed to improve existing approaches, that were presented as new contribution in the previous chapters. In detail, the structure of SIgMOIDAL with the multiple sub-modules is the following:

SIgMOIDAL (Jan 2023 implementation)

Module	Omics data type	Task	Implementation
Module 1: pre-processing	RNA	Quality Check	SOTA
		Denoising/Imputation	New Result - Section 3.1
	DNA	Quality Check	SOTA
	ATAC	Quality Check	SOTA
Module 2: single-omic specific analyses	RNA	Low-dimensional projection	SOTA
		Clustering	SOTA
		Batch correction	SOTA
		Cell cycle phase	SOTA
		Diapause	New Result - Section 5.1
		CMS	New Result - Section 5.1
		Cell type	New Result - Section 5.1
		Differential Expression Analysis	SOTA
	DNA	Cancer evolution models	New result - Section 3.2
	ATAC	Low-dimensional projection	SOTA
		Clustering	SOTA
		Differential peak accessibility	SOTA
Module 3a: diagonal data integration	RNA + DNA	CONGAS	New result - Section 4.1
		clonealign	SOTA
		LACE	SOTA
	RNA + DNA + ATAC	CONGAS+	New result - Section 4.2
Module 3b: vertical data integration	RNA + DNA	G&T-seq data analysis	Note 1
Module 4: downstream analyses	Note 2		

Note 1: notice that during the Accelerator project G&T-seq data (see Section 1.1) have been generated and might be exploited to assess both the gene expression and the clonal composition of the sequenced samples. SIgMOIDAL pipeline can accommodate the analyses of this multi-omics data, but we leave this to further works.

Note 2: given the characterization of cancer subpopulations obtained via data analysis and/or integration, it is possible to perform downstream analyzes. Ongoing efforts are aimed at defining an automated framework for clone-specific differential expression analysis, where we want to automatise the investigation of differences in gene expression patterns among clones defined via data integration, via diagonal integration (i.e., CONGAS+, CONGAS, etc.) or vertical integration approaches.

The analyses are implemented in `Python` and `R`, using the package `Scanpy` to manipulate single-cell RNA-sequencing objects, and the package `Signac` to work with scATAC-seq data.

In this chapter, we describe the complete analysis workflow, showing its application on two different experimental settings. Please notice that the case studies do not handle all the data types. In fact, one case study involves scRNA-seq data and the second processes scRNA-seq and scDNA-seq data. The motivation behind the absence of a case study involving scATAC-seq data is that the project involving the generation of that data type from PDOs is still under development, as new data are being produced during the last year of the Accelerator award project and the currently ongoing analyses are still in their preliminary phase.

5.1 SIgMOIDAL pipeline description

5.1.1 Module 1: pre-processing

5.1.1.1 Preprocessing of scRNA-seq data

The main raw-data output from sequencing are FASTQ files containing reads information for every single-cell. These files need to be aligned to the reference genome, meaning that each read has to be mapped to its position in the genome to then quantify the presence of each gene by counting the number of reads that map to its position. Among the most popular alignment tools for scRNA-seq, `STARsolo` was employed in this pipeline. This option was chosen with respect to another popular pipeline for alignment which is CellRanger for its higher efficiency in terms of runtime and number of detected genes per cell [187]. `STARsolo` performs multiple steps to produce the final count matrix encoding the expression level of each gene in each single cell [171], which are described below. Please note that each step can be performed with multiple options, and here the ones employed in this work are reported.

- **Read mapping**: each read is aligned to the reference genome, and the corresponding gene is annotated.
- **Barcode passlisting**: each cell is identified by a barcode, that is a sequence of nucleotides that uniquely identifies a cell. `STARsolo` takes as input a list of possible barcodes and considers reads containing those barcodes. To take into account possible nucleotide substitutions due to technical errors, barcodes that are characterized by one mismatch with respect to the provided list are also considered.
- **UMI deduplication**: in order to take into account possible sequencing errors, UMIs with a Hamming distance ≤ 1 are collapsed together.
- **Quantification**: for each gene in each cell, the expression is quantified by the number of UMI that map to it.
- **Cell filtering**: most of the barcodes contained in the matrix produced in the step above correspond to droplets containing ambient RNA. Thus, a preliminary filter based on counts is performed to filter out these empty droplets: a threshold t corresponding to the $99 - th$ percentile of UMI counts in all barcodes is computed, and then barcodes with a total count lower than 10% of t are removed.

Quality control and denoising The output of the previous step is a genes-by-cells expression matrix, that has already been preliminary filtered by the alignment tool. However, additional metrics need to be computed to perform Quality Control of the

obtained matrix and remove low-quality observations. First, we filter out cells cells with zero counts for more than 90% of the genes. Indeed, a very low number of counts for a specific observation may be caused by cell death, premature cell rupture, or the capture of random mRNA which "escaped" from other cells (i.e., the observation is not an actual cell, but a mixture of random mRNA molecules) [82]. Then, we remove doublets filtering cells based on their total counts and we remove cells with high mitochondrial activity. In fact, observations with (i) a low number of total counts, (ii) few detected genes, and (iii) a high fraction of mitochondrial counts may correspond to cells whose mRNA has leaked out through a broken membrane, leaving only the mRNA located inside the mitochondria membrane [120]. This factors are thus an indicator of the death of the cell, therefore it is important to consider the fraction of mitochondrial gene counts when filtering out cells in the observed matrix.

5.1.1.2 Preprocessing of scDNA-seq data

Raw scDNA-seq data are FASTQ files containing the reads, i.e., short sequences that contain the DNA sequence of one genome fragment. These reads need to be aligned to a reference sequences, in order to identify mutations. Our analysis was implemented exploiting the output of 10x Genomics kit [1], that enables sequencing of the DNA to estimate Copy Number Variation profiles at single-cell level. We note that the production and sale of the 10x solution has been discontinued. However, our analysis is not binded to this technology, as any other scDNA-seq protocol that enables copy number calling at the single-cell resolution [119] could in principle be applied.
The reads contained in the FASTQ files are aligned to the reference genome, using 10x Genomics software `CellRanger`, which performs the alignment and calls copy number alterations. Given the heterogeneity of the profiles inferred at the single cell resolution, one fundamental step is to perform clustering of the obtained profiles, that is the subsequent step in CellRanger in which hierarchical clustering is applied.

5.1.1.3 Preprocessing of scATAC-seq data

The FASTQ files that result from of a scATAC-seq experiment are aligned with `CellRanger`, that in addition to map reads on the reference genome performs also the peak calling step, where contiguous regions of open chromatin (peaks) are detected, and are used as features in the output read count matrix. Together with this matrix, CellRanger returns a fragment file, that contains read count information of fixed-length genomic bins, that are used to perform peak calling. Like in scRNA-seq, The obtained data needs to be pre-processed to remove low quality observations. In case of ATAC, multiple metrics described in the following paragraphs are employed.

[1] `https://www.10xgenomics.com/products/single-cell-cnv`

Nuclesome/nucleosome-free regions It is possible to exploit the distribution of fragment lengths to detect low quality cells. In fact, high quality cells are expected to have that distribution enriched at 100 and 200 bp, while low quality cell exhibit different patterns. For more details regarding this filter, please refer to [156]. So for a successful experiment one should observe a fragment-length distribution that is enriched around 100 and 200 bp (the peaks should decrease).

Transcription Starting Site enrichment The other metric used to detect low quality cells is the Transcription Start Site (TSS) Enrichment score. In fact, TSS are open chromatin regions and they are expected to have an enriched number of reads. Thus, by computing the TSS enrichment score it is possible to detect those cells for which there is no enrichment in TSS and discard them as low quality. For more detail on this quality check metric please refer to [156].

Total number of fragments in peaks analogous to what's performed with the RNA, barcodes with a low number of fragments in peaks are removed. Then, to remove doublets cells with high values of fragments in peaks are excluded.

Fraction of fragments mapping to peaks Barcodes with a low fraction of fragments that map to peaks are considered as low-quality and they are removed.

5.1.2 Module 2: single-omic specific analyses

The second module of SIgMOIDAL contains data-type specific analyses, that are used to extract knowledge from one single data type. In this section we review the methods from the SOTA and we describe the new modules that were implemented.

5.1.2.1 Analyses of scRNA-seq data

Once quality control metrics have been computed and data have been pre-processed, multiple downstream analyses can be performed with the goal of extracting knowledge from the data and characterize the observations. In the context of Colorectal Cancer, over the last years different analysis have been proposed aiming at characterizing cancer cells populations. In this work we merged multiple building blocks re-implementing different metrics, with the goal of providing an end-to-end solution for an in depth analysis and characterization of high dimensional single-cell gene expression data. The data extracted from our workflow can be used by clinicians and wet-lab experts to formulate and/or validate biological hypothesis.
In detail, the analyses that have been implemented aim at associating to each single-cell multiple features, namely Consensus Molecular Subtype (CMS), diapause, cell type,

and cell cycle, and additional analyses implemented from the state-of-the-art enable to perform clustering, low-dimensional projection of cells, batch correction and differential expression analysis. In the following paragraphs we will give an overview of all analyses.

Low-dimensional projection Given the high dimensionality of the count matrix, whose number of features is in the order of 10^4, it is necessary to apply sound methods to project single-cell profiles onto a low-dimensional space, that should capture similarities between cells. The standard dimensionality reduction method is Principal Component Analysis (PCA). Next, two methods have been proposed and have been extensively applied, namely UMAP [96] and t-SNE [17], for projecting high dimensional data in a low-dimensional space where similar points in the original space have similar embeddings, while distant points in the projection correspond to distant elements in the original space.

Clustering As it has been presented in Section 2.2, the most popular clustering methods of scRNA-seq data are Louvain [14] and Leiden [127], that are methods designed for community detection and thus require data to be embedded in a k-Nearest Neighbor (k-NN) graph. `scanpy` [104] and `Seurat` [167] implement the embedding step by first computing PCA and UMAP/t-SNE, and use the obtained low-dimensional representations to build the k-NN graph. Finally, they use this graph to run Louvain or Leiden and identify clusters.

Batch correction When scRNA-seq experiments, are performed at different times or in different laboratories, one additional source of noise needs to be taken into account, namely the batch-effect, which adds one additional source of variation that causes single-cell profiles from different experiments to appear different in the read count matrix, even though they correspond to the same biological population. Over the years multiple frameworks have been introduced to perform batch effect correction, which have the aim of both removing technical variability while preserving true biological signal. In order to avoid removing true biological variability, it is fundamental to carefully evaluate the impact of such methods on the data under investigation, and in [153] authors present a benchmark of 14 recent methods, where they employ both visualization techniques (t-SNE and UMAP) and quantitative metrics to assess the methods performance, and identify 3 best-performing methods, namely Harmony [116], LIGER [132], and Seurat v3 [126].

Consensus Molecular Subtype. Colorectal Cancer is characterized by a high degree of heterogeneity, and by extracting specific features from gene expression data it is possible to classify this disease using a stratification that reflects important biological features, tumor phenotype and clinical outcome [50]. The classes are defined as Consen-

sus Molecular Subtypes (CMS), and until 2020 they were extracted only from bulk RNA sequencing data. Then, in [146], authors applied the concept of CMS to single-cell gene expression datasets, with the goal of understanding whether each tumor is characterized by multiple CMSs in a subset of its cells. Thus, the first analysis that we included in our workflow is the association of a subtype to each gene expression profile, following the approach presented in [146]. In detail:

1. We consider signatures for 4 CMS from the package `CMSclassifier`, where each CMS s has 5 profiles derived from 5 different datasets d. Each signature is a vector containing the expected expression value of each gene in the corresponding CMS: $v_{d,s} = v_{d,s,1}, \ldots, v_{d,s,f}$.

2. We compute the Pearson correlation between each single-cell gene expression profile $x_c = \{x_{c,1}, \ldots, x_{c,f}\}$, $\rho_{d,s} = corr(x_c, v_{d,s})$.

3. We associate to each single-cell, the CMS s showing the highest correlation mean: $cms_c = \arg\max_s \langle \rho_{d,c,s} \rangle_d$.

Cell-type association In order to assess whether PDO cancer cells recapitulate with the expression signatures of intestinal subtypes, we have employed the cell-type signatures defined with the canonical markers in [146] to associate a cell type to each PDOs gene expression profile. They provide the signatures for 6 cell types composed of 11 genes, where each signature is a vector of length 11 containing the scaled expression value for the corresponding genes:

$$v_s = \{v_{s,1}, \ldots, v_{s,11}\} \tag{5.1}$$

Thus we perform the following steps:

- For each PDO, we compute the z-score of each gene in the signature.

- For each single-cell profile $x_c = \{x_{c,1}, \ldots, x_{c,11}\}$, we compute the Pearson correlation coefficient between each cell and each signature $\rho_{c,s} = corr(x_c, v_s)$.

- Finally, we assign to x_c the cell-type s showing the highest correlation $type_c = \arg\max_s(\rho c, s)$.

consider each organoid and we compute the z-score we computed the Pearson correlation coefficient of every single-cell and cell-type signature, and we return the cell-type showing the highest correlation.

Diapause state. It has been shown that one mechanism exploited by Cancer Cells to survive treatment is that of entering a Drug Tolerant Persister state (DTP), where cells are slow-cycling or quiescent and then start growing again during drug holiday. Thus, cells that enter DTP are able to survive treatment without acquiring any genomic mutation that makes them different from the parental population, and once they exit the quiescent state they are still sensitive to treatment. In [176], authors show that the transcriptomic profiles of cells in DTP state recapitulate with that of embryonic cells in a paused state. This state mimics a natural phenomenon where development is arrested, known as diapause. Then, in [176] authors develop a diapause state signature score which can be associated to each single cell. This score is a vector of +1 and -1, computed using 124 genes that are de-regulated in embryonic stem cells and diapaused embryos: genes that are up-regulated in diapaused state cells are assigned +1, and conversely -1 is associated to every down-regulated gene. Thus, by (i) taking the z-score of each gene and (ii) computing the scalar product between each single-cell and the diapause signature, we compute a diapause state score.

Cell cycle phase The life of a cell can be seen as a cycle made of different phases, where each phase is characterized by a different behaviour of the cell. Understanding in which phase each single-cell is located, can provide great insights on the overall composition of the sample, as it can help in understanding whether specific subsets of cells are quiescent or actively dividing. In order to associate a cell cycle phase to each single-cell gene expression profile, suites like `Scanpy` and `Seurat` implement methods to compute a score for three different phases exploiting the expression level of known genes involved in the cell-cycle. Thus, following the best practices for single-cell data analysis we have also incorporated the cell-cycle phase computation in our workflow.

Differential gene expression analysis In order to identify putative processes responsible for drug resistance, the first step consists in identifying those genes that are differentially expressed between different conditions, i.e., genes whose distribution across two condition is significantly different. Given the functions provided in `Scanpy` to perform differential expression testing and filtering the results, we implemented a wrapper that uses a stratification of gene expression profiles into multiple groups (e.g., clustering labels, metadata on therapy resistance, etc.), and performs all pairwise comparisons to detect DEGs. Next, we filter results according to the following attributes: minimum number of cells expressing the genes in either one of the two groups, p-value and Log-Fold Change (LFC). Finally, in case of multiple pairwise comparisons, we perform a post-processing step where we aim at identifying those genes that are consistently differentially expressed across conditions.

5.1.2.2 Analyses of scDNA-seq data

To investigate the evolutionary history of a tumor, applying algorithms for the reconstruction of clonal trees enables to reconstruct the order of accumulation of mutations [53]. When single-cell mutational profiles are available, it is possible to apply MCMC based frameworks such as LACE [196] that return a maximum likelihood tree. From the inference result, it is possible to exploit our COB-tree algorithm to compute the consensus tree (Please see Section 3.2.1 for additional details on the algorithm).

5.1.2.3 Analyses of scATAC-seq data

Single-cell ATAC profiles need to be processed differently from scRNA-seq profiles. In fact, as it has been presented in Section 2.2, there are several approaches to perform normalisation and dimensionality reduction. One approach consists in binarizing the count matrix and applying Term Frequency-Inverse Document Frequency transformation [181], and projecting data on a low-dimensional space using Singular Value Decomposition (SVD). Once these steps have been computed, like for single-cell RNA sequencing (scRNA-seq) data, single-cell profiles are projected onto UMAP coordinates, and Louvain can be applied to perform clustering.

5.1.3 Module 3a: diagonal data integration

5.1.3.1 Integration of DNA and scRNA data

clonealign In order to integrate scRNA-seq and scDNA-seq data, in 2019 clonealign was developed [111], which is a framework that taks in input (i) the single cell gene expression profiles and (ii) the profiles of the copy number clones inferred from scDNA-seq experiment, mapping each profile to the corresponding clone. Clonealign uses a statistical model that assumes a linear relation between copy number and gene expression.
This method does not detect de novo the clonal profiles, but it is a supervised method that aims at associating to each known copy number clone detected via scDNA-seq the corresponding set of gene expression profiles.

CONGAS We developed a new Bayesian framework, presented in Section 4.1 that infers copy number states from scRNA-seq data. Unlike clonealign, our framework is unsupervised and does not require to know a priori which clones to detect in the data. Please refer to Section 4.1 for a through discussion about CONGAS.

5.1.3.2 Integration of DNA, scRNA and scATAC data

CONGAS+ We also presented a method to perform multimodal clustering and extend CONGAS to incorporate single-cell ATAC sequencing (scATAC-seq) signal, which

can be applied to characterise copy number clones considering their gene expression and chromatin accessibility profiles.

5.1.4 Module 3b: vertical data integration

5.1.4.1 Integration of DNA and RNA data

When a pool of cells is sequenced via full-length technologies, it is possible to exploit frameworks that accommodate using variants called from scRNA-seq data to reconstruct models of cancer evolution, such as LACE [196]. Such models are an example of vertical integration, as the information about genetic mutations and gene expression is associated to the same set of cells.

5.2 Case study #1: 20 Patient Derived Organoids of Colorectal Cancer [RNA]

5.2.1 Data

Experimental design The analyses explained above have been applied to a biobank of patient derived organoids of Colorectal Cancer cells. We have exploited our workflow to analyse 20 samples in total, that correspond to 20 PDOs derived from metastatic sites of 16 colorectal cancer. Prior to the the biopsy used to grow the corresponding PDOs, the patients underwent different rounds of treatment. However, the stratification and analysis of this cohort according to previous treatment regimens is outside the scope of this work.

Figure 5.1 reports a summary of the PDO cohort, where we report for each organoid the stratification into sensitive or resistant to 3 different therapies, that are Oxaliplatin (Oxa), SN-38 and 5-FU. Such stratification was carried out by wet lab experts, who tested different drug concentrations and reported the IC50 value, that corresponds to the mean concentration leading to the death of 50% of the original population of cells. Given that high values of IC50 correspond to higher drug concentrations, organoids with higher values of IC50 were classified as drug resistant, while lower values were associated with drug sensitivity. For each drug, PDOs were sorted according to their IC50 values: the organoids with the lowest 5 IC50 values were classified as *sensitive*, and those with the top-5 IC50 were in turn classified as *resistant*. All the scRNA-seq data from the PDOs were sequenced before treatment with any of the aforementioned mentioned drugs.

5.2.2 Results

Alignment and preprocessing We aligned scRNA-seq data to the GRCh38 reference sequence using `STARsolo`, as described in Section 5.1. Next, we performed quality check

to remove low quality observations using `Scanpy`: we removed cells with more than 90% of zero counts, cells that may correspond to doublets, cells with a high fraction of mitochondrial counts and genes that had 0 expression across all cells. For mitochondrial fraction filtering, we computed for each organoid the mean μ and standard deviation σ of the distribution, with the goal of taking into account possible differences in the distribution across multiple samples. In fact, we computed $t = \mu + \sigma$, and for values of t lower than 20%, we use t as upper bound for filtering, otherwise we set the threshold to 20%.

After preprocessing, we log-normalized each gene expression profile c_i, dividing all values for cell c_i by its total counts $\sum_i c_i$, multiplying by a scaling factor $s = 10^4$ and computing the log-transformation of the obtained values. In order to visualize the distribution of counts in the gene expression space, we computed UMAP coordinates, by (i) scaling the input features, (ii) computing Principal Components Analysis, (iii) embedding the single-cells in a neighborhood graph and (iv) finally, computing the UMAP dimensionality reduction. The result is presented in Figure 5.2. Considering the distribution in the UMAP space of the different single-cell profiles, we observe high inter-patient variability, as the organoids are grouped according to the patient of origin. In fact, only samples associated to the same patient such as 1021BL and 1021PD are characterised by expression profiles closer in the UMAP space.
Given that the main goal of our analysis is to characterise each organoid and assess whether there are any features that are consistently deregulated between sensitive and resistant organoids, we did not perform any batch correction, to avoid introducing any bias in gene expression distributions[120]. The subsequent analyses that we performed on the PDO cohort consists in the identification of DEGs using the stratification of the organoids according to drug resistance, and in the characterization of each organoid using multiple metrics.

Differential Expression Analysis The first analysis that we performed has the goal of identifying, for each treatment, those genes that are consistently up or down regulated between sensitive and resistant populations. We exploited the presence of multiple samples associated to different drug sensitivity values in the cohort to perform multiple pairwise comparisons, which may provide an opportunity to identify shared mechanisms responsible for drug resistance, identifying those genes that are consistently de-regulated between multiple pairs of organoids. Thus, given the three drugs, namely Oxaliplatin, 5-FU and SN-38, we considered the stratification into sensitive and resistant (Figure 5.1) samples, and we performed the following steps:

1. For each pair of sensitive and resistant organoids (s_i, r_i), we performed the Wilcoxon test on each gene, comparing its distribution across the two conditions.

2. We filtered the results of the previous step, selecting those genes characterised by

Organoid ID	Patient	Oxa	SN-38	5-FU
1018	1018	Sensitive		
1027	1027	Sensitive	Sensitive	Sensitive
1032	1032		Resistant	
1034	1034		Sensitive	
1001	1001		Sensitive	Resistant
1004	1004	Resistant		
1010	1010	Resistant	Resistant	Sensitive
1014 BL	1014			Sensitive
1014 PD A				
1014 PD B				
1015	1015	Sensitive		Resistant
1021 BL	1021		Sensitive	Sensitive
1021 PD				
1031 BL		Resistant		Resistant
1037 PD	1037		Sensitive	Sensitive
1046 BL	1046	Sensitive	Resistant	
1046 PD Seg2		Resistant	Resistant	Resistant
1046 PD Seg5		Sensitive	Resistant	
1047	1047			
3994-117	117	Resistant		Resistant

Resistant Sensitive

Figure 5.1: PDOs cohort considered in the analysis. There are 20 PDOs, which can be stratified into sensitive and resistant to 3 different therapies (namely Oxaliplatin, SN-38 and 5-FU) according to IC50 data. Data generated during the SCEICC AIRC/CRUCK Accelerator project #22790

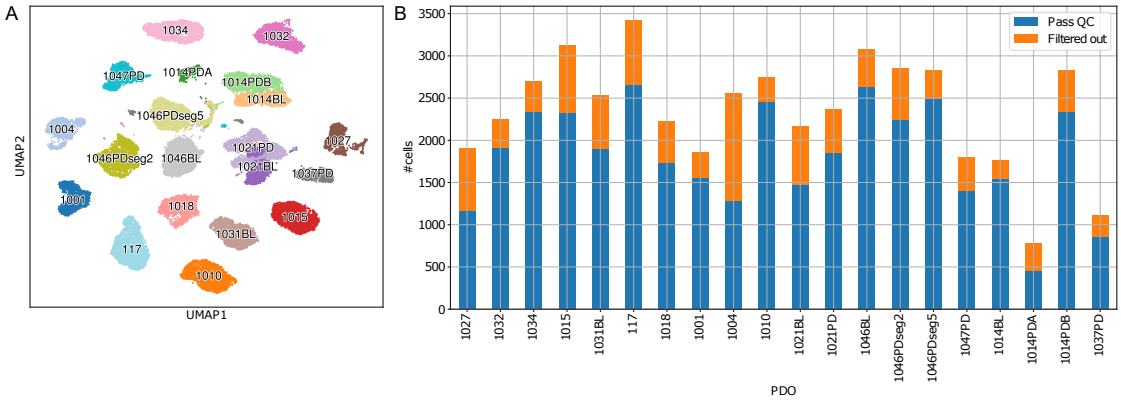


Figure 5.2: Patient Derived Organoids cohort. A: UMAP dimensionality reduction of the PDOs cohort. There is high inter-patient variability, as organoids are grouped in space according to the patient of origin. B: barplot showing that are respectively kept and filtered out after quality check.

a p-value ≤ 0.05, $|\text{LFC}| \geq 1$ and expressed in at least 30% of cells in either one of the two organoids.

3. For each sensitive organoid, we kept only genes deregulated against all resistant organoids, and we finally selected genes found consistently deregulated (previous step) for at least 3 out of 5 sensitive organoids.

From this analysis we obtained one list of genes per drug type, containing the features that are consistently up (or down) regulated between the multiple pairs of sensitive and resistant organoids. The two-step filter enables to (i) detect for each sensitive organoid those genes that consistently exhibit a higher or lower expression compared to the sensitive organoids, and (ii) merge the results obtained from each sensitive PDO to identify genes that are consistently de-regulated for multiple sensitive organoids. We present results in Figure 5.3, where we show the genes that were selected from our analysis for each drug. These lists have been subsequently analysed by wet lab experts to identify possible targets that can be experimentally validated.

Knowledge extraction from multiple metrics In order to characterize the full cohort, we extracted multiple features from the data using the measures described in Section 5.1, and we show the result in Figure 5.4.
Considering the stratification into multiple CMS, in Figure 5.4A we show that the highest proportion of cells is associated to CMS2, and only three organoids present heterogeneous composition in term of CMS2. In Figure 5.4 we report the distribution of the diapause score across the organoids in the cohort, and we show that there are a subset of the organoids whose cells are characterised by a higher diapause score. Unfortunately, when

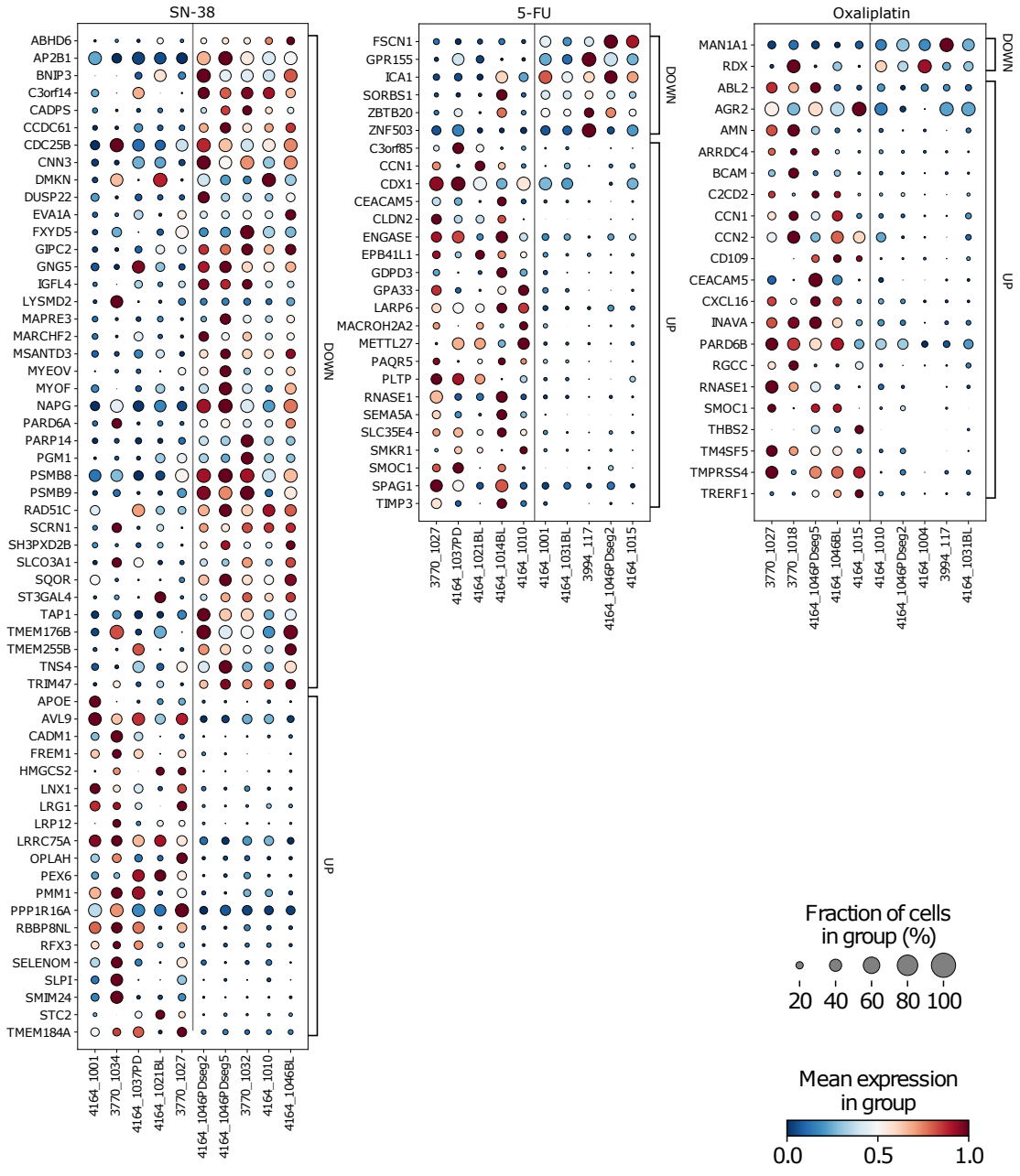


Figure 5.3: Results of the differential expression analysis performed by merging the results of multiple pairwise comparisons to identify the genes that are consistently up or down regulated.

we couple this information with the stratification into sensitiveness and resistance to the therapies tested, we observe that the median scores are randomly distributed across the sensitive and resistant states for all the drugs. In Figure 5.4C-D we report the distribution of the 11 cell types identified through the cell lineage analysis presented in Section 5.1.2.1, and the distribution of cell cycle phases. In this case, we observe low heterogeneity among the multiple organoids that compose the cohort.

Conclusions The analysis of the PDOs cohort enabled us to design and setup an analysis pipeline that handles the preprocessing of raw data, quality control to remove low quality observations and downstream analyses. In order to improve the knowledge extraction step and favour reproducible results, we believe it's valuable to provide a comprehensive workflow that merges together multiple strategies presented in different state-of-the-art publications, with the goal of making such analyses reproducible and enhance the quality of the knowledge extracted from single-cell data applied on cancer research.

5.3 Case study #2: Data integration of longitudinal samples from 4 Patient Derived Organoids of colorectal cancer [RNA + DNA]

5.3.1 Data

The next case study we present is the analysis of 4 PDOs, that are all derived from the same parental organoid, which was treated with 2 drugs, namely MK2206 and AZD5363 at two different concentrations. In Figure 5.5A we show the experimental design: (i) the *Parental* is the untreated organoid, which (ii) was treated with MK2206 and AZD5363 at 1uM concentration for 35 days, in order to obtain two resistant organoids. The drug was then removed, the two organoids were left expanding for 35 days and then sequencing was performed on both organoids. Finally, a second round of treatment was applied with a higher drug concentration of 5uM for 40 days, and the organoid treated with AZD5363 was sequenced.
In the next part of this section we will use the following IDs to refer to each organoid: Parental is the original untreated PDO; MK-1uM and AZD-1uM are the two resistant organoids obtained after treatment with 1uM concentration of MK2208 and AZD5363 respectively and AZD-5uM is the resistant sample obtained after treatment with 5uM concentration of AZD5363. In detail we have the following samples: (i) scRNA-seq of Parental, AZD-1uM and AZD-5uM and (ii) scDNA-seq of Parental, MK-1uM and AZD-5uM. Both scRNA-seq and scDNA-seq experiments were carried out using 10x genomics kit.

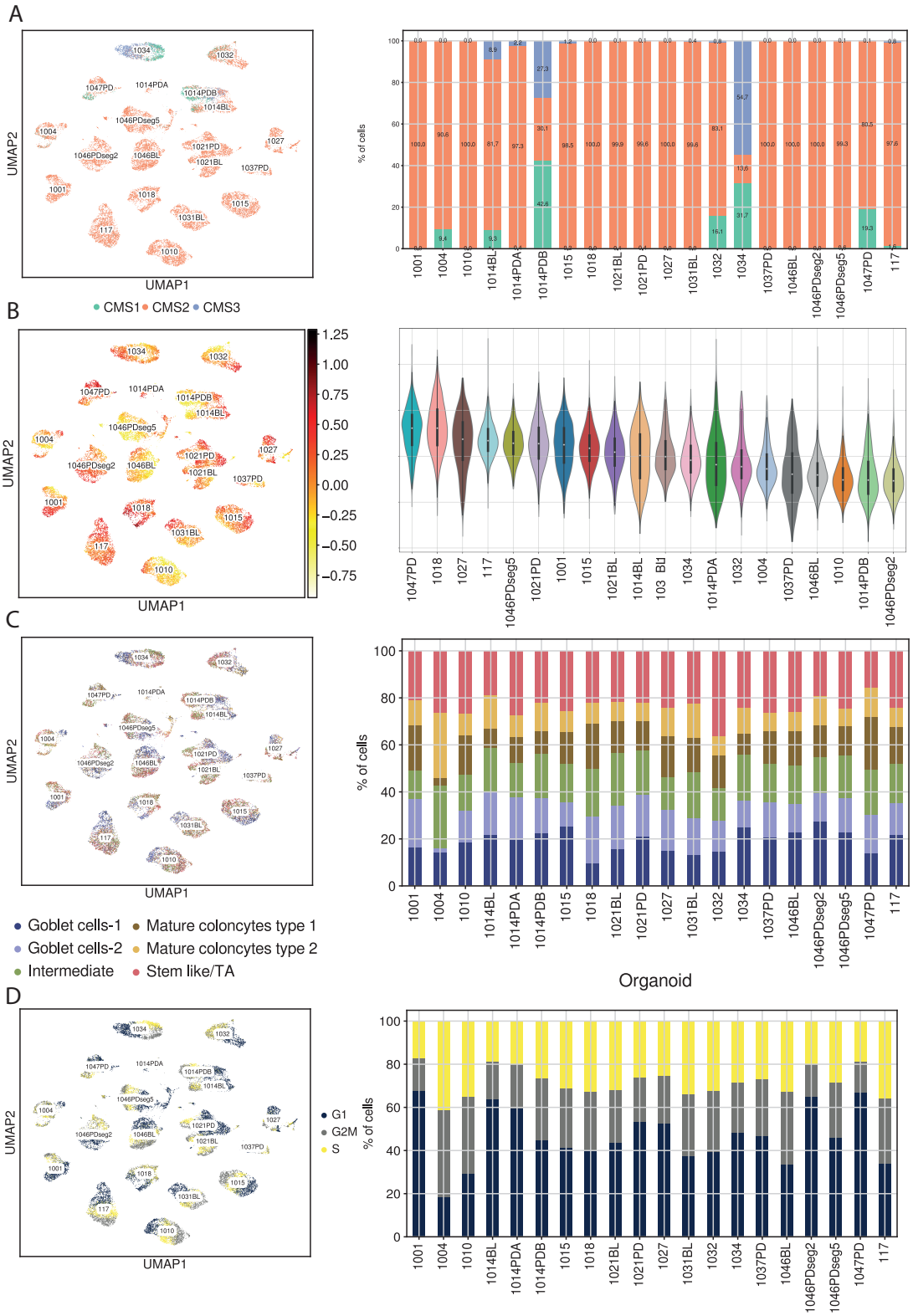


Figure 5.4: Results of downstream analyses of the organoids cohort. A: Visualization of the distribution of the different CMS profiles across the organoids in the cohort. B: Distribution of the diapause score. The violinplot shows how the score values are distributed in each organoid, sorted according to the median score. C: composition of the organoids in terms of epithelial cell types. D: cell cycle phase distribution.

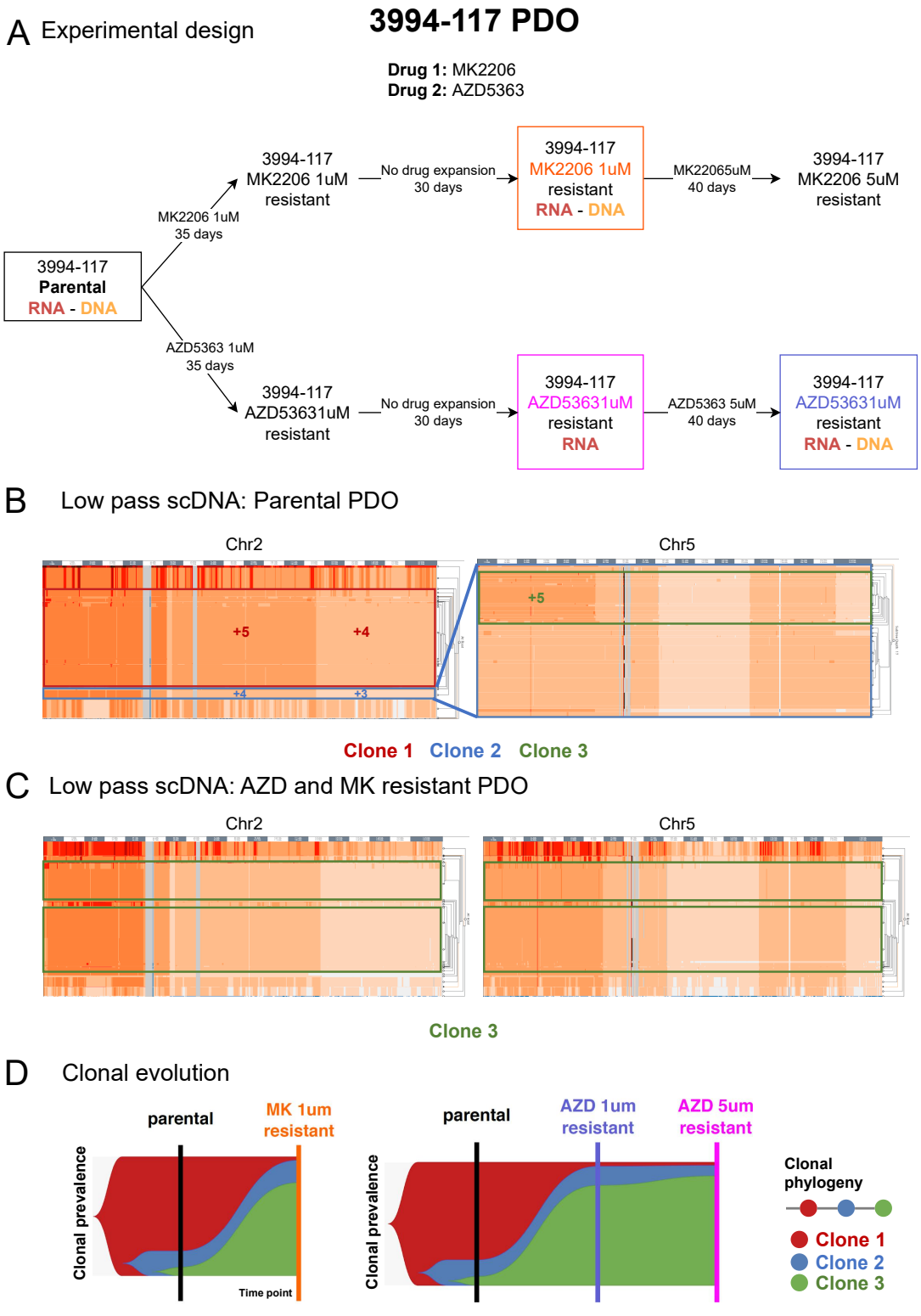

A Experimental design
3994-117 PDO
Drug 1: MK2206
Drug 2: AZD5363
3994-117 Parental RNA - DNA
MK2206 1uM 35 days
3994-117 MK2206 1uM resistant
No drug expansion 30 days
3994-117 MK2206 1uM resistant RNA - DNA
MK22065uM 40 days
3994-117 MK2206 5uM resistant
AZD5363 1uM 35 days
3994-117 AZD53631uM resistant
No drug expansion 30 days
3994-117 AZD53631uM resistant RNA
AZD5363 5uM 40 days
3994-117 AZD53631uM resistant RNA - DNA
B Low pass scDNA: Parental PDO
Chr2
Chr5
+5
+4
+4
+3
+5
Clone 1 Clone 2 Clone 3
C Low pass scDNA: AZD and MK resistant PDO
Chr2
Chr5
Clone 3
D Clonal evolution
parental
MK 1um resistant
Clonal prevalence
Time point
parental
AZD 1um resistant
AZD 5um resistant
Clonal prevalence
Clonal phylogeny
Clone 1
Clone 2
Clone 3

Figure 5.5

5.3.2 Results

In this setting, is was possible to exploit the two single-cell datasets for performing data integration, which was particularly interesting given the observed clonal dynamics. In detail, Figure 5.5B shows the clonal profile observed in the parental organoid: while the majority of cells were clustered into the clone that we define as clone 1, there was an additional clone (namely, clone 2), that exhibited a different behaviour from clone1 in specific segments of chromosome 2.This clone, was also characterized by a subclone, whose copy number profile differed from clone 2 in chromosome Chr 5. Then, Figure 5.5C shows the results on the resistant organoid AZD5-5uM, which is the same as we observe in AZD-1uM and MK-1uM: clone 3 that was present in a low portion of cells in the parental, becomes the most prevalent. These observations can be summarised in the clonal evolution plot reported in Figure 5.5: the first observed time-point corresponds to the untreated parental organoid, where we observe clone 1 with the highest prevalence and lower proportion of cells are associated to clone 2 and clone 3. Then, the following timepoints correspond to the resistant organoids derived after treatment, where clone 3 is the most prevalent clone.
Thus, while clone 1 is a sensitive population that doesn't survive after treatment, clone 3 corresponds to the resistant population, that becomes the most prevalent after treatment. Given this clear clonal evolution, where we could characterize the resistant and sensitive cells in terms of their copy number profile, we wanted to exploit single-cell gene expression profiles to characterize these clones. In detail, given the following inputs (i) the clonal and subclonal copy number profiles detected with single-cell resolution and (ii) the single-cell gene expression profiles of each organoid, we could apply `clonealign` [111] to find a mapping between the two input data.
Thus, we extracted the copy number profile of clone 1, clone 2 and clone 3, and we ran the tool on the 4 organoids, mapping each gene expression profile to the corresponding clone. The results, shown in Figure 5.6, are consistent with the clonal evolution history inferred from scDNA-seq: in fact, while most of the cells in the parental PDO are mapped to clone 1, in the resistant PDOs clone 3 shows the highest prevalence.
Performing a mapping between the two data types, makes it possible to study differences between clones defined from DNA-sequencing, considering their gene expression patterns, which would not be possible without using a framework that models the statistical dependence between data types and performs the mapping. Thus, this shows how tools that perform data integration between independent samples are fundamental in order to obtain a comprehensive overview of the system that is under investigation.
In order to study difference among copy number clones considering their gene expression, we performed differential gene expression analysis, using the mapped copy number clones to stratify cells into sub-groups. In detail, we considered the following classes of cells:

- Clone 1 pre treatment: cells in the parental organoid mapped to clone 1

- Clone 3 pre treatment: Cells in the parental organoid mapped to clone 3.
- Clone 1 post treatment: cells in any of the resistant organoids mapped to clone 1.
- Clone 3 post treatment: cells in any of the resistant organoids mapped to clone 3.

Considering both the clone label and the PDO label, we performed differential expression analysis to study the following differences:

- Pre-existing diversity: genes that are differentially distributed between clone 1 and clone 3 in the parental organoid.
- Plasticity of clone 1: given that we have cells assigned to clone 1 both before and after treatment (that are thus identical in terms of copy number profiles), we can detect if there are any differences between cells from this clone that arise during treatment.
- Plasticity of clone 3: as we did for clone 1, we can study the changes in gene expression among clone three in the sensitive and resistant organoids.

We performed the Wilcoxon test, setting a threshold of 0.05 to the p-value and $|\text{LFC}| > 1$. We identified a list of genes that were subsequently analysed by wet-lab experts and clinicians, to determine whether any of the genes could be selected as a putative target for subsequent experiments to investigate further the drug resistance mechanisms.

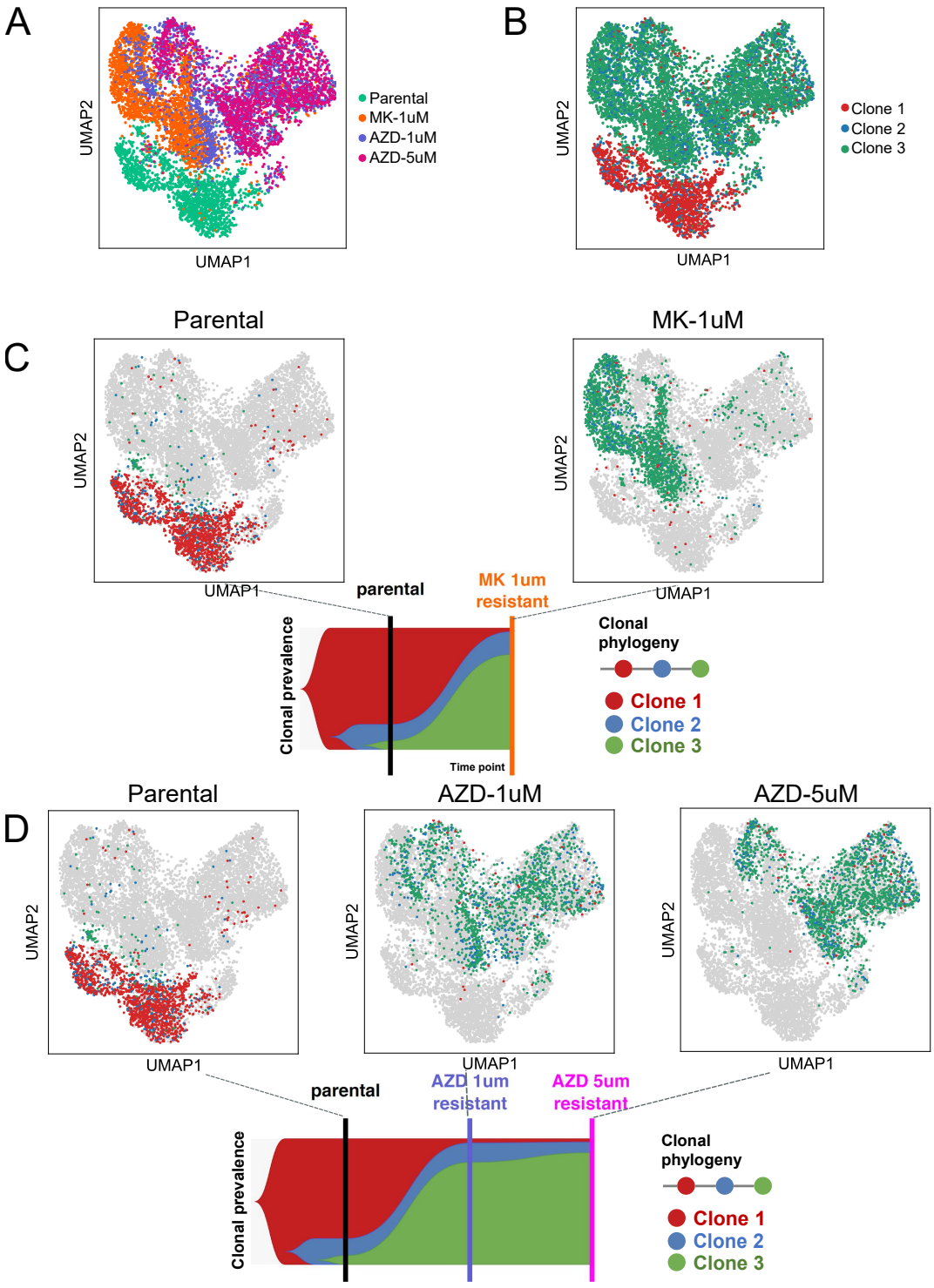


Figure 5.6: Results of mapping copy number clones to each gene expression profile. Consistently with what is observed in scDNA data, the majority of cells in the Parental organoids are associated with clone 1, while resistant organoids have the highest prevalence of clone 3. A: UMAP of the gene expression profiles colored by organoid ID. B: UMAP colored by copy number clone mapped by clonealign. C: zoom on the clonal evolution with MK2206 drug. D: zoom on the clonal evolution with AZD5363 drug. In both C and D, the results obtained with clonealign are consistent with the clonal evolution inferred using scDNA data.

6 Conclusions

This PhD project has been focused on the design and application of computational strategies to perform data analysis and integration of high-dimensional and heterogeneous multi-omics data. The work was divided in three main tasks: (Task A) methods for single-omics data types, (Task B) methods for the integration of multi-omics data and (Task C) a pipeline for reproducible analyses. Each of these categories delivered important achievements. Overall, the contributions have a two-fold impact, both on the progress of the SCCEiC project and in the advancement in the broad field of computational sciences. A detailed discussion of the impact follows.

6.1 Impact

Task A: Methods of single-omics data types This work led to two major achievements with respect to methods tailored to the analysis of specific omic data types.

i **Benchmarking of denoising methods for scRNA-seq data** (Section 3.1): we carried out an extensive comparison of state-of-the-art tools for denoising and imputation of scRNA-seq data, which in most cases rely on machine learning strategies to perform the task. Extensive simulations were performed in order to assess the ability of each method in recovering the corrupted information. Also, multiple real-world datasets were employed to discuss the impact of denoising in real scenarios. By exploiting both real and simulated datasets we computed a number of different metrics, and provided a comprehensive evaluation of all methods, provid-

ing a ranking of the methods according to different dimensions and perspectives. This line led to publication [149].

Contribution to the SCCEiC project: given the extensive availability of scRNA-seq data within the project, that are generated from PDOs derived from samples collected from colorectal cancer patients, this benchmarking has enabled to characterise which methods could be included in the final data analysis pipeline. From this work, we also identified which method provide inaccurate results by artefactually reducing the biological variability in the data or inferring inaccurate missing values, and as this has allowed to avoid the application of preprocessing strategies that would bias subsequent analyses. As a result, downstream analyses of scRNA-seq data benefitted from improved quality of the data, enhancing the reliability of the results.

Impact on computational sciences: in our work, we implemented an extensive comparison of multiple computational tools, that were grouped into different categories according to the assumptions and techniques used for denoising. In [149] we present a thorough and schematic summary of all the results, which can serve a guideline for computational scientists using denoising and imputation tools.

The generation of realistic data to be used in testing of computational methods is a particularly hot topic [186]. We contributed to this field by defining a rigorous and reproducible workflow for data generation. the design of extensive simulations settings, that takes into account differences in the data generation process can also be exploited by other researchers for the validation of new computational strategies.

Finally, other fields such as deep learning (computer vision or natural language processing), where similar denoising or imputation methods might be applied, can benefit from the analysis of the performance assessment presented in our work, where we highlight strength and weaknesses of multiple methods, discussing also their assumptions and the models exploited for denoising.

ii **Consensus approach for the inference of clonal trees from single-cell mutational profiles** (Section 3.2.1): we presented a method that can be exploited to improve current approaches for the reconstruction of clonal trees. The presented approach consists of (i) a new data visualization approach that sumarises the solution space explored during the MCMC search and (ii) an algorithm that exploits regularities in the search space to return one Consensus Optimum Branching Tree (COB-tree) that summarises the trees explored during the MCMC. The work led to the publication presented in Section 3.2.1, which has been accepted for publication in the conference proceedings of the *16th International Workshop on Artificial Life and Evolutionary Computation* (WIVACE 2022).

Contribution to the SCCEiC project: given that our proposed method was

designed to work with the output of MCMC methods that infer clonal evolution models using single-cell mutational profiles, the COB-tree algorithm can be applied on mutation data produced within the project, for instance by calling variants (single-nucleotide, indels, copy numbers) from DNA- RNA- or ATAC-seq data, in order to improve the inference of cancer evolution models.

Impact on computational sciences: this method represents an advancement in the field of cancer phylogenetics, as it represents the first method designed to explore and summarise the solution space of MCMC based algorithm for clonal tree reconstruction. The framework could also be applied to other computational methods that infer solutions through MCMC sampling, as it highlighted noteworthy differences with the standard maximum likelihood strategies. Finally, by reconstructing clonal trees using our approach it is possible to build better explanatory models of the evolutionary history of tumors. This in turn might allow to improve the the quality, robustness and reliability of the inference, possibly leading to the identification of repeated evolutionary patterns [87] within and between patients that could be exploited for anticipating tumor evolution.

Task B: Methods multi-omics data integration In task C2, the main achievements regarded two Bayesian methods that solve the following integration tasks:

i **CONGAS** (Section 4.1) integrates scRNA-seq data with bulk DNA-sequencing measurements with the goal of inferring genetic copy number clones from gene expression data, by clustering single-cell profiles in groups characterised by the same copy number state. It implements a model that links the copy number to the RNA signal, and it infers the parameters of the model through SVI. This line led to publication [194].

ii **CONGAS+** (Section 4.2) is the first method that maps two distinct omics (scRNA-seq and scATAC-seq) on the latent space of copy numbers, It enables to detect and characterise copy number clones considering their gene expression and chromatin accessibility patterns. The integration of both scRNA-seq and scATAC-seq profiles is implemented through a shrinkage coefficient, that is an hyperparameter that controls the weight given to the two omics in the final likelihood. This hyperparameter enables to incorporate external knowledge in the model regarding the reliability of the two signals.

Another interesting feature of the framework is that it models the latent copy number states using a probability distribution over discrete values. Given that the parameters are learnt via SVI, and the optimization requires to estimate the gradients of all the variables, to sample from the copy number state distribution we implement the Gumbel-softmax distribution [64] that is a continuous distribution that can approximate samples from a categorical distribution.

Both Bayesian frameworks model the copy number values as latent variables, and aim at assigning single-cells (ATAC/RNA) by exploiting a linear relation between the copy number state and the observed single-cell signal. We assessed the robustness of our methods using both simulated and real datasets, showing that both methods are able to accurately retrieve copy number clusters and correctly map single-cell profiles to the clones.

Contribution to the SCCEiC project: our integration methods fit well within the project experimental setting, and are one of the building blocks of the Sigmoidal pipeline (see Chapter 5). In fact, they provide a strategy to map genetic clones on single-cell datasets (either RNA or ATAC), enabling the characterization of biological subpopulations in each PDO considering their genetic alterations. This will be applied to datasets from PDOs of 16 cancer patients, from the cohort presented in Chapter 2, eventually allowing to assess and compare both the gene expression and the chromatin accessibility of the genetic clones of the different samples. Importantly, this will allow to evaluate the impact of the distinct therapeutic strategies tested in the project both on the genotype and the phenotype of cancer subpopulations.

Impact on computational sciences: our methods are Bayesian frameworks that exploit SVI to infer the posterior of the parameters. CONGAS+ is characterised by two interesting features: the shrinkage coefficient and the use of a categorical variable to infer copy number states. The first feature can be applied in principle to other computational problems where the goal is to combine together multiple data types and that would benefit from the introduction of external knowledge about each data type.

Second, given the presence of a discrete variable in the model, we implemented the Gumbel-softmax distribution that is a continuous distribution that can approximate categorical samples. Thus, our work may be useful to other computational scientists who need to model discrete factors while using approaches that require computing gradients for optimization.

Task C: SIgMOIDAL, a pipeline for reproducible single-cell analysis The efforts summarised above have brought to the definition of a pipeline that combines different strategies to deliver reproducible and standardized analyses of biological samples.

Contribution to the SCCEiC project: this pipeline has been employed for the analyses of samples generated within the project, and will be used in the sampled generated in the near future. The two case studies in which we exploited our pipeline are presented in Section 5.2 and Section 5.3. Through SIgMOIDAL we could extract usable knowledge from scRNA-seq samples, identifying multiple features to characterise each sample, also performing data integration to characterise genetic clones considering their gene expression. This results have allowed to generate data-driven experimental hypotheses on cancer evolution and especially on the motivations underlying drug resistance in col-

orectal cancer.

Impact on computational sciences: the design of `SIgMOIDAL` provides a scaffold for the design of comprehensive pipelines that need to take into account complex experimental settings with multiple data types, multiple experiments and possibly longitudinal data. We remark that with respect to widely used tools such as Seurat [167] and Scanpy [104], the design of `SIgMOIDAL` is specifically tailored for cancer research. Its design combines multiple modules highlighting current state-of-the-art techniques to perform knowledge extraction from cancer samples, that can be modified or updated to accomodate new frameworks. The effort of designing `SIgMOIDAL` goes in the direction of the principles of open science and the FAIR guidelines [67] for data management, towards an improvement of the reproducibility of data analysis workflows.

6.2 Limitations and future works

Despite the various achievements, the methods presented in this thesis have several limitations.

With respect to the COB-tree tree algorithm and the visualisation approach that describes the solution space explored during MCMC search Section 3.2.1, we note that in [140], another consensus approach for clonal trees is presented, that is applied to the different problem of inferring one consensus solution from the clonal trees in cohort of patients. Thus as future developments we aim at testing our COB-tree algorithm on the problem tackled in [140] and [87], with the goal of understanding whether using one consensus tree per patient improves the inference of the evolution models, which might in turn be used to find the repeated patterns of evolution via models such as REVOLVER [87] that aim at studying repeated patterns of evolution.

Regarding the task of diagonal integration, our frameworks CONGAS and CONGAS+ take in input the segmentation from bulk sequencing, and thus they are able to detect copy number clones at the resolution of input segments inferred from the bulk. One limitation of these works is that, given the segment-level resolution, we may not observe subclones that are affected by one or more copy number events present on a sub-portion of an input segment. However, given that scATAC-seq reads correspond to fragments of the DNA in open chromatin regions, it may be possible to extend our framework to incorporate a segmentation step that would exploit this data type and would enable to detect subclones in the data. From preliminary analyses, we observed that it might be possible to extend our approach to a two-step inference, where we first separate tumor from normal cells, we employ the ATAC signal on the obtained clusters to detect SNPs and we finally compute the BAF, that is be used in conjunction with the read depth to perform the segmentation and copy number calling to feed to the next run of the inference.

Regarding the design of SIgMOIDAL, one limitation that arised within the SCCEiC project concerns the experimental settings. In fact, the case study presented in Section 5.2 where multiple PDOs from different patients are analysed, is characterised by high inter-patient variability, that introduces multiple confounding factors in the data that hinder the identification of putative target genes for personalised therapy. In order to improve methods for knowledge extraction, it is necessary to design experimental settings that enable the in depth characterization of each PDOs, collecting multiple omics data type for each organoid.
Finally, a progressive development of our pipeline SIgMOIDAL is ongoing. At the moment, multiple building blocks are implemented in `bash`, `R` and `Python`, but we are working on (i) a comprehensive `Nextflow` implementation of the preprocessing steps and (ii) an interactive module to perform data integration and clone-specific differential expression analyses. We believe that implementing an interactive solution to perform integration and downstream analyses is key to improve the reproducibility and usability of our approaches.

A

Appendix: Additional papers

A.1 Deep Learning model for Predicting Relative Fluxes in Reaction Systems

When studying differences between healthy and pathological systems from the point of view of their metabolic reactions transforming one chemical into another, it is important to understand differences between their fluxes. However, current technologies allow only to measure the abundance of metabolites involved in the reactions, making the task of predicting the variation of fluxes across steady states very challenging. One solution is to use constrained optimization to predict flux variations from relative chemical abundances [65]. However, performing such an optimization requires many assumptions and simplifications. Thus, we developed a multi-target Deep Neural Network predicts the variation of reaction fluxes between two states of a reaction system, using as input features the variation of metabolites abundances.
Given limitations in technologies to measure flux variations, we used a dataset of a yeast metabolic network simulated using using kinetic modeling [69]. We also tested whether flux variations are affected also by chemicals that are not directly involved in the reaction, by reducing the set of chemicals given in input to the model. We show that our DNN is able to predict variations in fluxes also when we reduce the input features. The code for all the experiments performed in the paper is available at `https://github.com/BIMIB-DISCo/FLUX-PREDICT`.

Information and Computation 281 (2021) 104798

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Combining multi-target regression deep neural networks and kinetic modeling to predict relative fluxes in reaction systems

Lucrezia Patruno [a,1], Francesco Craighero [a,1], Davide Maspero [a,b], Alex Graudenzi [b,c], Chiara Damiani [d,e,*]

[a] Department of Informatics, Systems and Communication, University of Milan-Bicocca, Milan, Italy
[b] Institute of Molecular Bioimaging and Physiology, Consiglio Nazionale delle Ricerche (IBFM-CNR), Segrate, Milan, Italy
[c] Bicocca Bioinformatics Biostatistics and Bioimaging Centre – B4, Milan, Italy
[d] Department of Biotechnology and Biosciences, University of Milano-Bicocca, Milan, Italy
[e] SYSBIO/ISBE.IT Centre for Systems Biology, Milan, Italy

ARTICLE INFO

Article history:
Received 13 October 2020
Received in revised form 20 May 2021
Accepted 29 August 2021
Available online 6 September 2021

Keywords:
ODE-based modelling
Monte Carlo sampling
Deep neural networks
Metabolomics
Metabolic fluxes

ABSTRACT

The strong nonlinearity of large and highly connected reaction systems, such as metabolic networks, hampers the determination of variations in reaction fluxes from variations in species abundances, when comparing different steady states of a given system. We hypothesize that patterns in species abundance variations exist that mainly depend on the kernel of the stoichiometric matrix and allow for predictions of flux variations. To investigate this hypothesis, we applied a multi-target regression Deep Neural Network (DNN) to data generated via numerical simulations of an Ordinary Differential Equation (ODE) model of yeast metabolism, upon Monte Carlo sampling of the kinetic parameters. For each parameter configuration, we compared two steady states corresponding to different environmental conditions. We show that DNNs can predict relative fluxes impressively well even when a random subspace of input features is supplied, supporting the existence of recurrent variation patterns in abundances of chemical species, which can be recognized automatically.

1. Introduction

The determination of the rate at which a substance is transformed into another through a given reaction or pathway (i.e. the flux) on the basis of routine measurements of the abundance of chemical species, when the mechanistic dynamics of the systems is not fully characterized, is an important problem in different fields, from life [1] to environmental sciences [2]. Knowledge on relative fluxes is important, as it may translate into knowledge about the controllable mechanisms underlying the differences between two steady states of a system (e.g. pathological vs healthy state). This translation occurs more directly and successfully than in the case of abundances of chemicals, which provide a mere snapshot of the system [1]. Yet, fluxes are hardly measurable with current technologies, whereas abundances can be largely measured with high throughput methods.

* Corresponding author at: Department of Biotechnology and Biosciences, University of Milano-Bicocca, Milan, Italy.
E-mail address: chiara.damiani@unimib.it (C. Damiani).
[1] Equal contribution.

https://doi.org/10.1016/j.ic.2021.104798
0890-5401/

We investigate the problem of estimating relative fluxes from relative abundances, by means of both theoretical reasoning and simulations. We show that the problem can be solved analytically only if enzymatic kinetics are neglected. When enzymatic kinetics are taken into account, extra information is required, namely relative abundances of enzyme-substrate complexes when mass action rate law formulation is used, kinetic constants (i.e. the binding affinity of enzymes) when the Michaelis-Menten approximation is used. Both types of information are currently not measurable on a large scale. Moreover, analytical solutions require knowledge on the abundance of each single substrate involved in a reaction, whereas the current sensitivity of abundance quantification techniques (as e.g., mass spectrometry) typically allows detecting only a subset of them at a time.

To overcome this lack of knowledge, current approaches mainly rely on optimization subject to constraints to identify feasible solutions out of a very large set of candidates. Along with stoichiometric constraints and mass balance, such approaches incorporate constraints on relative abundances of metabolites in the form of constraints on relative fluxes. For example, iReMet-flux [3] seeks to minimize the distance between any pair of flux distributions in the feasible region, whose ratio between each flux is within upper and lower bounds, derived from the ratios of metabolic and enzyme abundances, according to the mass action formulation (see Section 2). Pandey et al. [4], instead, convert the variation in the abundance of a given metabolite into a constraint on the generic variation in the fluxes responsible of either its production or consumption and seek to maximize the consistency with such constraints, along with other constraints on relative fluxes assumed from relative gene expression data.

By requiring relative metabolic abundances to be incorporated in the form of constraints on relative fluxes, the above approaches require many assumptions and simplifications. Another limitation of these approaches is that they require the definition of an objective function. Moreover, it is difficult to find the optimal trade-off between narrow constraints leading to infeasible solutions and loose constraints leading to too wide feasible regions.

Here, we propose a different approach based on the combination of kinetic modeling and Machine Learning (ML). The combination of computational modeling, and in particular constraint-based modeling, with machine learning techniques is an emerging field which is revealing great potential. Recent approaches exploit the mechanistically linked information provided by context-specific models as the input of either supervised or unsupervised machine learning approaches, as reviewed in [5–7]. Although some of these studies have used neural network to predict e.g., individual fluxes from enzyme or gene expression data [8] or abundances of metabolites from other -omics data [9,10], to our knowledge, this is the first time that ML is used to predict overall flux variations from overall relative abundances.

The approach that we are proposing originates from the hypothesis that recurrent patterns resulting from stoichiometric and mass balance constraints exist. Hence, we can exploit information of the vector of abundance variations $\boldsymbol{\delta x}$ or, possibly, of a subspace of it, in order to predict with a good confidence level the vector of flux variations $\boldsymbol{\delta v}$. We expect these patterns to be learned and recognized by ML regression algorithms.

However, given that fluxes are hardly measurable in real-world scenarios, it is unrealistic to obtain a large and heterogeneous experimental training set of $(\boldsymbol{\delta x}, \boldsymbol{\delta v})$ pairs to properly train any ML algorithm. To overcome this limitation, we propose to simulate $(\boldsymbol{\delta x}, \boldsymbol{\delta v})$ pairs with kinetic modeling, namely via standard ODEs. Notice that reaction rate equations and constants are largely undetermined, otherwise it would suffice to directly simulate $\boldsymbol{\delta x}$ with a ODE model to predict $\boldsymbol{\delta v}$. Here, we assume that, in light of the steady state constraint, $f(\boldsymbol{\delta x}) = \boldsymbol{\delta v}$ is largely independent from the specific values of kinetic constants.

To investigate the validity of our assumption, we propose to randomly sample the space of kinetic parameters, as in [11–13]. For each sampled set of parameters, $\boldsymbol{\delta x}$ and $\boldsymbol{\delta v}$ can be collected, by comparing the state of the ODE model in two different environmental conditions in a time invariant condition (i.e., the steady state). In a preliminary phase, we employed the simulated dynamics of a previously published yeast metabolic network [12,11], undergoing nutritional perturbations, to train, validate and test different configurations of Deep Neural Networks (DNNs). We also evaluated the predictive performance of DNNs in the realistic scenario in which the abundance of metabolites can be measured for a limited subset of the model species.

2. Motivation and main assumptions

A biochemical reaction system is defined by a set $\mathcal{X} = \{X_1, \ldots, X_M\}$ of molecular species occurring in the system, and a set $\mathcal{R} = \{R_1, \ldots, R_N\}$ of chemical reactions taking place among the species. We define reactions as: $R_r : \sum_{q \in Q_r} \alpha_q X_q \Rightarrow \sum_{t \in T_r} \beta_t X_t$ where $\alpha_q, \beta_t \in \mathbb{Q}^+$ are stoichiometric coefficients associated, respectively, with the q-th reactant and the t-th product of the r-th reaction, and Q_r and T_r are the set of reaction substrates and products of reaction r, respectively. Let $[X_m](t)$, with $m = 1, \ldots, M$ be the abundance of X_m at a given time t in the system's evolution, either expressed as number of molecules or as concentration. Let V_r, with $r = 1, \ldots, N$ be the rate (or flux) through reaction R_r in a unit of time, i.e. the number of times R_r occurs in that unit of time. Such a system is said to be at steady state if $\partial[X_m](t)/\partial t = 0, \forall m$. Steady state is thus the condition in which fluxes may occur but the concentration of all species does not change in time. Let S be a $M \times N$ matrix, referred to as stoichiometric matrix, whose element $s_{m,r}$, takes value $-\alpha_{m,r}$ if species X_m is a substrate of reaction R_r (i.e., $X_m \in Q_r$), $\beta_{m,r}$ if species X_m is a product of reaction R_r (i.e., $X_m \in T_r$), 0 otherwise. Let $\boldsymbol{v} = (V_1, \ldots, V_N)$ be the vector of reaction fluxes, then a system is at steady state when $S\boldsymbol{v} = 0$.

It is worth mentioning that, if a reaction R_r is *reversible*, a reaction R_b exists such that $s_{m,r} = -s_{m,b}, \forall m$. Typically, life scientists use the term flux to indicate the net rate $V_r - V_b$, that is, the rate of the forward reaction minus the rate of the

reverse reaction. However, in this work reversible reactions are represented with two distinct and complementary forward reactions, thus the terms flux and rate coincide.

Now let i and j be two different steady states of the system and $\boldsymbol{x^i} = ([X_1]^i, \ldots, [X_M]^i)$ be the vector of abundances of the chemical species in steady state i and $\boldsymbol{v^i} = (V_1{}^i, \ldots, V_N{}^i)$ be the vector of reaction fluxes in steady state j, and let $\boldsymbol{x^j} = ([X_1]^j, \ldots, [X_M]^j)$ be the vector of abundances of the chemical species in steady state j and $\boldsymbol{v^j} = (V_1{}^i, \ldots, V_N{}^j)$ be the vector of reaction fluxes in steady state j. We define the species abundance variation between state i and j as $\boldsymbol{\delta x^{i,j}} \equiv (\delta[X_1]^{i,j}, \ldots, \delta[X_M]^{i,j}) = \boldsymbol{x^j} - \boldsymbol{x^i}$, and the variation of reaction fluxes as $\boldsymbol{\delta v^{i,j}} \equiv (\delta V_1^{i,j}, \ldots, \delta V_N^{i,j}) = \boldsymbol{v^j} - \boldsymbol{v^i}$. In the following, $\boldsymbol{\delta x^{i,j}}$ and $\boldsymbol{\delta v^{i,j}}$ are also referred to as relative abundances and relative fluxes, respectively.

The aim of this work is to deduce flux variations from species abundance variations, that is, $\boldsymbol{\delta v^{i,j}}$ from $\boldsymbol{\delta x^{i,j}}$. In the following, we will make use of a very simple and specific example of reaction system to motivate, without loss of generality, the complexity of the problem and the non linearities that one may encounter when trying to deduce flux variations from species abundance variations.

Example 1. Let us assume a very simple system composed of 4 chemical species $\mathcal{X} = \{A, B, C, D\}$ (e.g., metabolites) and 3 reactions $\mathcal{R} = \{R_1, R_2, R_3\}$ defined as follows:

$$R_1: \ A \Rightarrow B$$
$$R_2: \ C \Rightarrow B$$
$$R_3: \ B \Rightarrow D$$

In order for the system above to be able to reach a steady state, unbalanced reactions (also referred to as exchange reactions) must be included, for A and C – which must be fed into the system ($\emptyset \Rightarrow A$; $\emptyset \Rightarrow C$) – and for D – which must be depleted ($D \Rightarrow \emptyset$).

At the steady state, the rate of production and consumption of the species must balance. Hence, if any event (e.g., an external perturbation of the system) determines the increase of either V_1 and/or V_2, then V_3 must eventually increase to reach a new steady state. Consequently, when comparing two steady states of the system, if $\delta V_1 + \delta V_2 > 0$ then $\delta V_3 > 0$.

Let us now suppose that information on $\boldsymbol{\delta x^{i,j}}$ is given and, for instance, that an increase in $[B]$ ($\delta[B]^{i,j} > 0$) and an increase in $[D]$ ($\delta[D]^{i,j} > 0$) were observed. This must be imputed to one of the following cases:

1. an increase in V_3 and an increase in V_1,
2. an increase in V_3 and an increase in V_2,
3. an increase in V_3 and an increase in both V_1 and V_2.

Information on $\delta[A]$ and $\delta[C]$ does not let us to exclude case 1 or case 2. In fact, case 2 is compatible with both: (i) $\delta[C] > 0$, i.e., an increase in the reaction's substrate $[C]$ and (ii) $\delta[C] = 0$, if the higher depletion of C, resulting from an increase in $\delta V_2 > 0$, is compensated by a higher influx of C. Along similar lines, case 1 is compatible with both: $\delta[A] > 0$ and $\delta[A] = 0$.

Example 1 demonstrates the complexity of the problem of determining analytically flux variations from relative abundances. The complexity is expected to increase with the number of interconnected reactions and when reactions of higher order and/or feedback loops come into play, as it is typically observed in real-world scenarios.

However, the following assumptions would allow one to analytically estimate relative fluxes from relative abundances:

- for each reaction r in the system, the mass action law is assumed: $V_r = k_r * \prod_{q \in Q_r} [X_q]^{|S_{q,r}|}$, where k_r is the kinetic constant of reaction r, X_q is the q^{th} substrate of the set Q_r of substrates of reaction r, and $S_{q,r}$ is the stoichiometric coefficient of substrate X_q i.e., how many molecules of the substrate partake to the reaction;
- at (steady) states i and j, the kinetic constant k_r of any reaction r of the system is assumed to be identical.

Given such assumptions, the variation between the flux of an irreversible reaction r in two steady states i and j can be analytically determined as the ratio V_r^i / V_r^j:

$$\frac{V_r^i}{V_r^j} = \prod_{q \in Q_r} \left(\frac{[X_q]^i}{[X_q]^j} \right)^{|S_{q,r}|} \tag{1}$$

which does not depend on k_r.

The above situation completely neglects enzymatic kinetics, which are an important factor in the dynamics of chemical systems. Let us suppose, for instance, that reaction R_1 in the previous example is catalyzed by enzyme E^1. Hence, the series of steps through which reactants bind to specific enzymes before being transformed into products should be taken into account, as follows:

$$R_1 : A + E^1 \Leftrightarrow AE^1 \Rightarrow B + E^1$$

In principle, one can apply equation (1) to the last reaction step, but knowledge of the relative abundance of intermediate complexes (δAE^1) is required. Yet, owing such a level of detail of information is unrealistic with current technologies.

When dealing with cellular metabolic reactions, it is reasonable to assume that they are far from thermodynamic equilibrium and that substrates are in excess over enzyme-substrate complexes. Hence, the enzyme kinetics is generally approximated with Michaelis-Menten rate laws [14,15]. The Michaelis-Menten formulation does not explicitly take into account the abundance of enzymes, but it models saturation kinetics, by describing the variation of the rate of a reaction as a function of substrate's abundance. In the simplest scenario, the rate of a reaction R_r involving a single substrate X_r with unitary stoichiometric coefficient would be described as:

$$V_r = \frac{V_r^{MAX}[X_r]}{K_r^M + [X_r]} \tag{2}$$

where, briefly, V_r^{MAX} is the maximum rate of reaction r (when the enzyme is saturated with substrate), whereas K_r^M is the concentration of substrate that permits the enzyme to achieve half V_r^{MAX}, which depends inversely on the affinity of the enzyme for its substrate. In this scenario, the ratio V_r^i/V_r^j describing the variation between the flux of a irreversible reaction r in two steady states i and j is defined as follows:

$$\frac{V_r^i}{V_r^j} = \frac{[X_r]^i(K_r^M + [X_r]^j)}{[X_r]^j(K_r^M + [X_r]^i)} \tag{3}$$

which does depend on K_r^M. Given the incomplete knowledge of the value of K_r^M of metabolic reactions, $\boldsymbol{\delta v^{i,j}}$ cannot be analytically derived from $\boldsymbol{\delta x^{i,j}}$ in this scenario.

Moreover, both equations (1) and (3) require information on the variation of the abundance of each substrate partaking in reaction R_r. However, in real-life scenarios only a fraction of the species involved in a reaction system is detected by chemical quantification technologies. Hence, we here investigate the possibility of using information about variations in other species in the network to improve predictions of δV_r when information on the abundance of substrates of R_r is lacking.

Our hypothesis originates from the consideration that all the steady states of a biochemical reaction system abide by the constraint $S\boldsymbol{v} = 0$ and, therefore, relationships among $\boldsymbol{v}^i$ and $\boldsymbol{v}^j$ exist which are independent from specific rate laws and kinetic constants of reactions. Hence, we speculate that similar relationships between $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$ may also exist, which do not depend on kinetic constants values. In this work, we investigate such hypothesis by means of simulation experiments and machine learning.

The general idea of the proposed approach is depicted in Fig. 1.

3. Methods

3.1. Synthetic dataset

To preliminarily investigate our hypothesis, we used a dataset previously generated via numerical simulations of an ODE model [11], in which elementary mass action law was assumed for every reaction rate. The model is defined by a set of $N = 48$ reactions and a set of $M = 34$ metabolites. The metabolic network model is available in SBML format at this link: github.com/BIMIB-DISCo/FLUX-PREDICT, and a graphical representation of it is shown in Fig. 1. For the sake of notation simplicity, in the following, we will refer to the name of specific reactions with the name of the first substrate and the name of the main product separated by the underscore symbol. For instance the reaction in the top left corner of the map ($Glc + ATP \Rightarrow G6P + ADP$) will be referred to as Glc_G6P. Reverse reactions are considered separately and are indicated with the suffix $\_reverse$. $P = 100\,000$ sets of kinetic constants $\mathcal{K}_1 = \{k_1, k_2, \ldots, k_N\}$, $\mathcal{K}_2 = \{k_1, k_2, \ldots, k_N\}$, $\ldots$, $\mathcal{K}_P = \{k_1, k_2, \ldots, k_N\}$ for the model reaction rates were generated randomly from a uniform distribution in [0,1). Initial abundances of metabolites were defined according to data in literature and are reported in [11]. For each parameter set $\mathcal{K}_p$, with $p = 1, \ldots, P$, we retrieved two steady states of the model corresponding to two different nutritional conditions: condition i – low glucose (2.8 mM), condition j – high glucose (25 mM). Glucose concentration is maintained fixed during the simulation. The model was simulated via integration of ODEs by means of the LSODA solver [16] for a simulated time of 50 seconds. The quasi-steady state condition was determined according to a small threshold (0.01) on the average standard deviation (σ) of the value of species concentration for the last 10% of time dynamics. For further details on the simulations the reader is referred to [12]. For each parameter set $\mathcal{K}_p$, with $p = 1, \ldots, P$, we obtained the vector of abundances at steady state of the chemical species in condition i $\boldsymbol{x_p^i} = ([X_1]_p^i, [X_2]_p^i, \cdots, [X_M]_p^i)$ and in condition j $\boldsymbol{x_p^j} = ([X_1]_p^j, [X_2]_p^j, \cdots, [X_M]_p^j)$ and the vectors of fluxes $\boldsymbol{v_p^i} = (V_{1,p}^i, V_{2,p}^i, \cdots, V_{N,p}^i)$ and $\boldsymbol{v_p^j} = (V_{1,p}^j, V_{2,p}^j, \cdots, V_{N,p}^j)$ and we computed the vector of variations of abundances $\boldsymbol{\delta x_p}^{i,j} = \boldsymbol{x_p^j} - \boldsymbol{x_p^i}$ and of fluxes $\boldsymbol{\delta v_p^{i,j}} = \boldsymbol{v_p^j} - \boldsymbol{v_p^i}$ between conditions i and j. We decided to compute the difference rather than the ratio between conditions to avoid problems related to divisions by zero. We finally obtained 100 000 pairs of metabolites-flux variations ($\boldsymbol{\delta x_p}^{i,j}, \boldsymbol{\delta v_p}^{i,j}$), with $p = 1, 2, \cdots, 100\,000$.

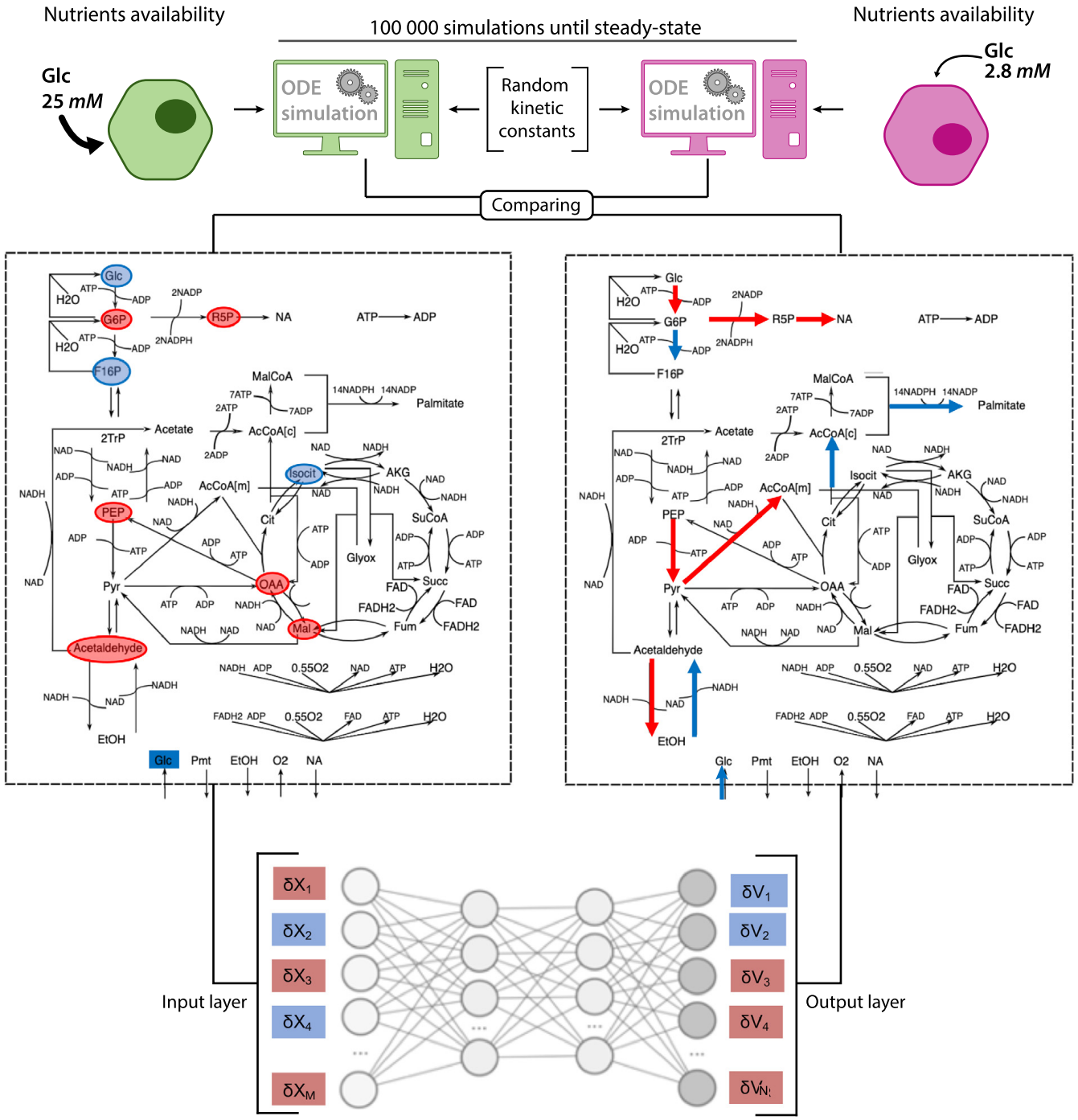


Fig. 1. Schematic representation of the proposed approach and diagram (adapted from [11]) of the yeast metabolic network used to generate the synthetic dataset. Blue/red represent positive/negative variations in species abundances and fluxes. The DNN diagram is for representative purposes only. In our setting, the input layer has less nodes ($M = 34$) that the output layer ($N = 48$). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

For the sake of simplicity, in the following we will refer to $\boldsymbol{\delta x} = (\delta x_1^{i,j}, \ldots, \delta x_p^{i,j})$ as the matrix of size $P \times M$, where $\boldsymbol{\delta x_p}$ is the vector of variation of abundances for constants $\mathcal{K}_p$ and $\boldsymbol{\delta x_{*,m}}$ is the vector of variation of abundances of metabolite m for all the P parameters. Similarly, we will refer to $\boldsymbol{\delta v} = (\delta v_1^{i,j}, \ldots, \delta v_p^{i,j})$ as the matrix of size $P \times N$, where $\boldsymbol{\delta v_p}$ is the vector of variation of fluxes for constants $\mathcal{K}_p$ and $\boldsymbol{\delta v_{*,r}}$ is the vector of variation of flux r for all the P parameters. In addition, we will refer to a specific metabolite variation $\boldsymbol{\delta x_{*,m}}$ and a specific flux variation $\boldsymbol{\delta v_{*,r}}$ with the name of the corresponding metabolite and reaction.

Data pre-processing Prior to training the neural network, the input dataset was pre-processed. We removed zero variance predictors [17] from the relative metabolites $\boldsymbol{\delta x}$ and we removed zero variance output features from the relative fluxes $\boldsymbol{\delta v}$. In detail, we removed 1 out of 34 relative metabolites (namely, $O2$) and 2 out of 48 relative fluxes (namely the exchange

reactions of *Glc* and $O2$). Finally, in order to proceed with the training phase of our DNN model, we standardized the input variables (*i.e.* metabolite variations), by removing the mean and scaling to unit variance.

3.2. Model definition

In order to define a model that predicts the vectors of flux variations $\boldsymbol{\delta v_p}$ from the vectors of abundance variations $\boldsymbol{\delta x_p}$, we built a multi-target regression DNN. Such model is depicted in Fig. 1: its input layer contains as many nodes as the number of abundance variations (i.e., M), and the output layer is composed of a number of nodes equal to the number of flux variations that need to be predicted (i.e., N). As commonly done, networks are trained in order to minimize the Mean Squared Error (MSE) between true and predicted flux variations.

We considered a multi-target regression DNN rather than single-target to reduce computational costs and hopefully to exploit relationships among the output variables to improve the goodness of fit [18].

3.3. Cross-validated grid search

The selection of the hyperparameters that define the DNN was performed by a cross-validated grid search over a set $\mathcal{H}$ of 48 possible hyperparameters configurations. More in detail, for each configuration $h \in \mathcal{H}$ we estimate the performance on unseen data by cross-validation and then define our chosen model with the best performing h.

Train and test sets We split our dataset in training (*outer training set*) and *test set* with a percentage of $90\% - 10\%$. The former partition is used to fit the neural network and to perform hyperparameter selection, while the latter partition is used to provide an unbiased evaluation of the prediction error, as it is used neither during the training phase, nor for hyperparameter optimization.

Hyperparameters The main aim of our grid search is to explore whether there is a need for deep networks or if wide networks with one single hidden layer may suffice, and to exclude configurations with low predictive power. To this aim, we varied the following hyperparameters:

- Hidden layers sizes, to take into account different widths (number of neurons for each layer) and depths (number of layers). In detail, we considered the following settings:

  $$\{(100), (200), (500), (100, 100), (200, 200), (100, 100, 100)\},$$

  with each tuple indicating the number of neurons for each hidden layer.
- Optimization algorithms: $\{Adam, SGD\}$.
- Initial learning rate: $\{0.01, 0.001\}$.
- With and without the dropout regularization technique, that is commonly used to improve generalization. When used, we set the dropping rate to 50%.

In addition to varying the hyperparameters just listed, for each neural network configuration h we kept the following elements fixed: (i) batch normalization method to normalize the input of each activation function, with the aim of improving the stability of the training process. (ii) *ReLU* activation function. (iii) Early stopping heuristic to halt training if the model doesn't improve in 200 epochs, in order to prevent overfitting. (iv) Exponential decay schedule for the learning rate. (v) Batch size of 128. (vi) Mean Squared Error (MSE) as loss function. For a detailed explanation of all the techniques please refer to [19].

Cross validation The grid search procedure was combined with a 5-fold cross validation procedure (*i.e.* cross-validated grid search), see Fig. 2 for a schematized representation. In detail, the *outer training set* was split into 5 groups of equal size, the so-called *folds* (Step 1 in Fig. 2). Then, each neural network configuration $h \in \mathcal{H}$ was trained using 4 folds for the training process (*inner training set*) and the last one for performance evaluation (*valid set*). This procedure was repeated 5 times (CV Loop), so that each fold is employed once as validation set.

In order to have an unbiased estimation of when performing early stopping (i.e. without taking into account the *valid set*), for each iteration the *inner training set* was partitioned in two sets with a percentage of 90%-10% (Step 2*b*), using the former for training and the latter for early-stopping.

Then, when the training phase was concluded, the performance of the network was tested over the *valid set* (Step 2c). In detail, for our experiments we calculated the coefficient of determination R^2 for each flux r:

$$R^2(\delta v_{*,r}, \hat{\delta} v_{*,r}) = 1 - \frac{\sum_{f=1}^{F} (\delta v_{f,r} - \hat{\delta} v_{f,r})^2}{\sum_{f=1}^{F} (\delta v_{f,r} - \bar{\delta} v_{*,r})^2} \tag{4}$$

where F is the number of samples, $\delta v_{*,r}$ is the vector of variations for flux r, $\hat{\delta} v_{*,r}$ is the corresponding vector of predicted values, $\delta v_{f,r}$ and $\hat{\delta} v_{f,r}$ are the values for sample f and $\bar{\delta} v_{*,r}$ is the mean variation for flux r. R^2 measures the goodness

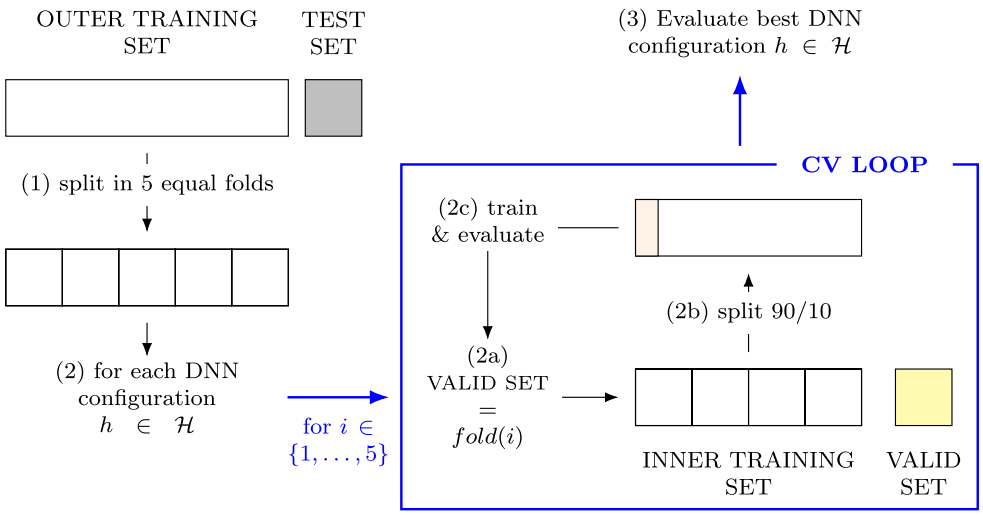


Fig. 2. Diagram of the 5 fold cross-validation procedure for hyperparameter selection with DNNs, where we take into account also an additional split for the early-stopping heuristic (step 2b).

of fit of the model by normalizing the sum of squared errors by the total deviance observed in ground truth data. After calculating R^2 for each flux r, we calculated the average R^2.

The final result of this cross-validated grid search consisted in 5 performance scores (average R^2) for each DNN configuration. Finally the model $h \in \mathcal{H}$ showing the best mean score was selected and evaluated on the *test set* (Step 3).

For details about the selected models see the Results section. See also Supplementary algorithm 1, for the pseudo-code of the Cross Validated Grid Search.

4. Results

4.1. Selected features

The main goal of this work is to investigate whether the prediction of the flux variation in a given reaction can benefit from information in the abundance variation of metabolites not directly involved in such reaction. To address this issue, we assessed the effect of reducing the number of input metabolites on the output prediction. To select a fraction g of the original features $\boldsymbol{\delta x_p}$ to be removed, with $g = 70\%$ and $g = 50\%$, at first instance, we followed the common practice of basing the choice on their redundancy. Firstly, we computed the pairwise Pearson correlation between metabolites. The heatmap in Fig. 3 shows the correlation of each pair. As already pointed out in [11], it can be observed that correlations between metabolites are not obvious. For example H_2O correlates strongly with $NADP$ even though they are not directly involved in the same reaction. Next, we ranked the pairs of metabolites by decreasing order of their absolute correlation. Starting from the most correlated one, we removed one feature from each pair until only a fraction g of the original metabolites was left.

4.2. Selected hyperparameters

We applied the overall methodology to train, cross-validate and test the best model, given simulated ($\boldsymbol{\delta x_p}$, $\boldsymbol{\delta v_p}$) pairs, with $p = 1, 2, \cdots, 100\,000$ (as illustrated in Section 3) for different fractions g of input metabolites.

We first applied the methodology by using the entire set of features, i.e., the variation of all the metabolites in the simulated network. In this case, the cardinality of $\boldsymbol{\delta x_p}$ coincides with the number of metabolites in the model $|\boldsymbol{\delta x_p}| = M$. The cross-validated grid-search procedure selected as best model the DNN with 2 hidden layers of 200 neurons each, i.e. $(200, 200)$, no dropout, learning rate $= 0.001$ and optimizer $=$ *Adam*. By analyzing the performances achieved by the different configurations (see Supplementary Table 1) we observed that models with one hidden layer provide similar performances regardless of their width. Thus, increasing the width is not enough to improve the performance and it is fundamental to explore deeper configurations. Indeed, an improvement in the predictive power of the model is observed when increasing the depth of the network. Finally, the best configuration provides an increase in the performance of $\approx 4\%$ with respect to the second best $(100, 100, 100)$. This result indicates that, by employing either wider or deeper models, the regression performance may be further improved. However, since the goal of this article is proving the potentiality of a Neural Network-based approach for predicting flux variations, exploring additional architectures is beyond the scope of this work. See Supplementary Table 1 for details about the performance of all the other DNN configurations tested.

We then tested the performance of the best model for $g = 70\%$ and $g = 50\%$ of features. On the one hand, the best DNN model selected by the cross-validated grid search procedure for $g = 70\%$ has 3 hidden layers with 100 neurons each, no dropout, learning rate $= 0.001$ and optimizer $=$ *Adam*. On the other hand, for $g = 50\%$ the best selected model has 2 hidden

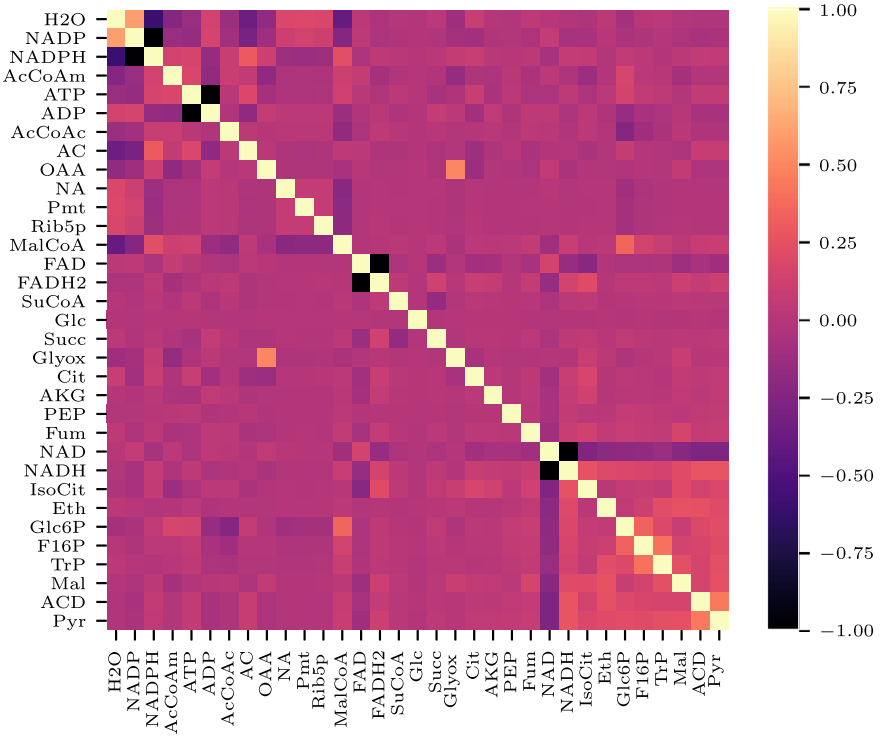


Fig. 3. Pearson correlation coefficient ρ for each pair of relative metabolites abundance variations $\boldsymbol{\delta x_{*,m}}$. The ρ correlations were computed by considering, for each metabolite, the vector of abundance variations over all samples in the dataset, with the aim of removing redundant features. For simplicity, we use metabolite names to refer to their $\boldsymbol{\delta}$s, and for improved readability we sorted metabolites by their average absolute correlation $|\rho|$.

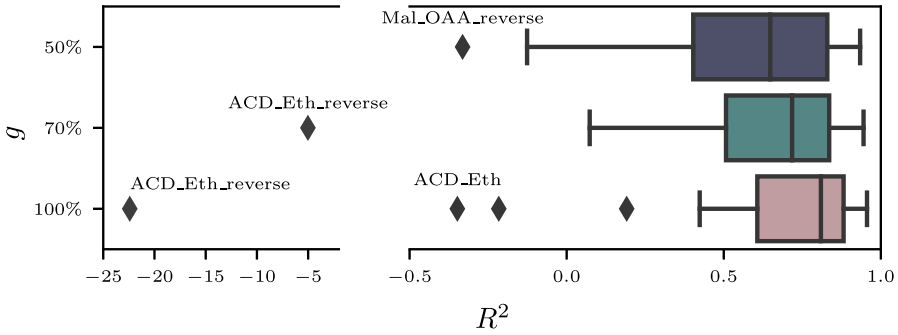


Fig. 4. Distribution of the R^2 coefficients calculated between true and predicted values of each flux r, and for different percentages g of features. The sensitivity of R^2 to outliers generates some negative outliers; as an example, ACD_Eth for $g = 100\%$, is plotted in Fig. 5. For simplicity, we use the fluxes names to refer to their $\boldsymbol{\delta}$s.

layers with 200 neurons each, no dropout, learning rate $= 0.001$ and optimizer $=$ *Adam*. See Supplementary Table 2 for the MSE values of the best DNNs.

It is worth noticing that in all the experiments, we observed that configurations with no dropout, learning rate of 0.001 and *Adam* as optimizer outperformed the other possible combinations of those hyperparameters. This result indicates that we may rely upon this selection for downstream analyses, without repeating the hyperparameters selection procedure.

4.3. Performance evaluation

The three best configurations selected were retrained on the *outer training set* and, then, used to predict flux variations $\boldsymbol{\delta v_p}$ on the *test set*.

The performance was evaluated computing, for each output feature, the R^2 score, which is a measure of the amount of variance in the target values captured by the values predicted by the model (see Eq. (4)). The median value of R^2 obtained for $g = 100\%$ is 0.8, while for $g = 70\%$ it is $= 0.71$ and for $g = 50\%$ it is equal to 0.64. The distribution of R^2 values for the three cases is reported in Fig. 4. As expected, the R^2 decreases as the fraction g of selected input features gets smaller. Interestingly, the reduction in the number of features by 30% and 50% corresponds to a modest reduction of R^2 by 11% and 20%, respectively.

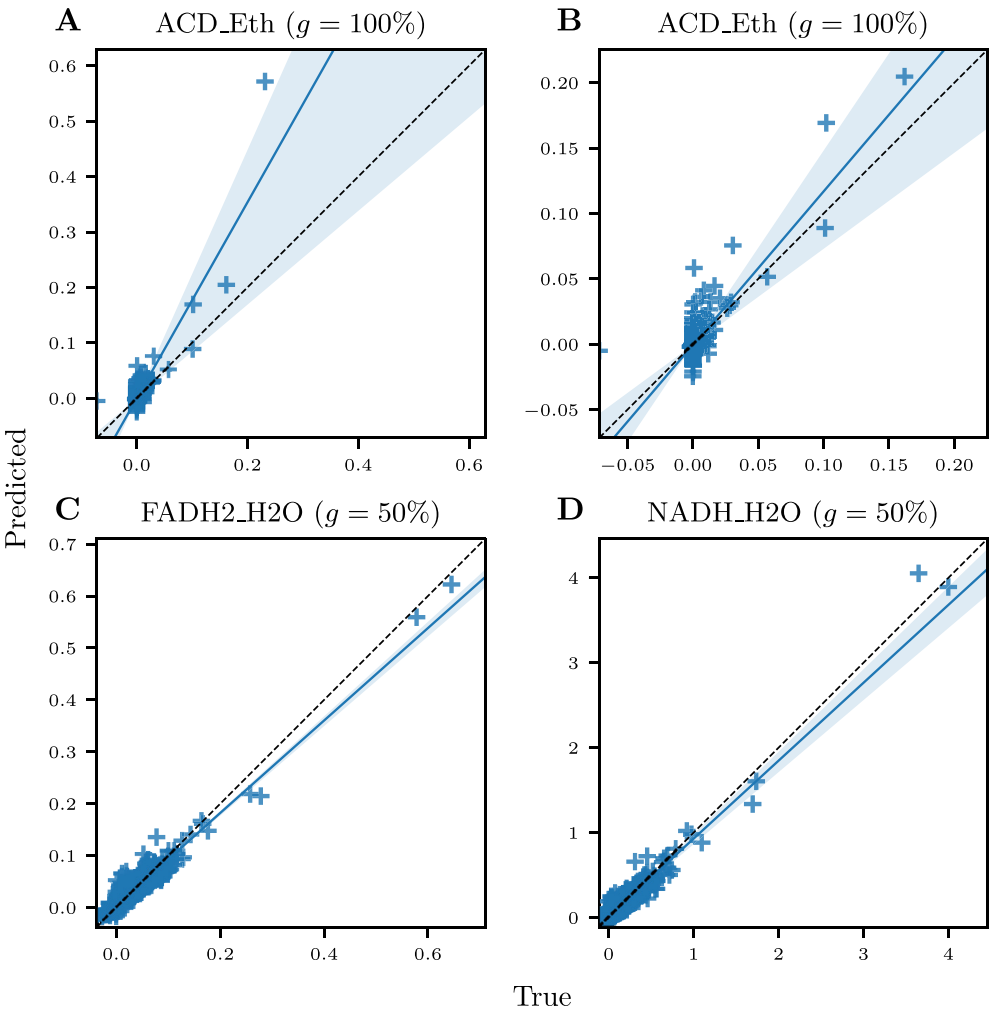


Fig. 5. Scatterplots of true and predicted values for selected fluxes. The 95% confidence interval of the regression line is visualized. For simplicity, we use the flux names to refer to their δs. (**A**) ACD_Eth for $g = 100\%$ with $R^2 = -0.35$, (**B**) ACD_Eth for $g = 100\%$ with the greatest outlier removed improves to $R^2 = 0.34$, (**C**-**D**) The two fluxes with best Pearson coefficient ($\rho = 0.971$) for $g = 50\%$.

Thus, results in terms of R^2 are overall good, except for a very few output features, such as the flux ACD_Eth in the $g = 100\%$ setting (see boxplots in Fig. 4). The existence of outlier reactions may be motivated with the choice of multi-target modeling. As motivated in Sec. 3.2, in our experiments we relied on MSE over all the predicted fluxes, and we weighed the error of each flux equally. As a result, the selected model corresponds to the one that performed better on average over all the fluxes, but the goodness of fit of different hyperparameters may vary across fluxes, as it can be observed in Supplementary Figure 1. Besides, low values of R^2 can be due to outliers in the error distribution of a single flux, especially when the variance of the true data is small. In fact, if we consider the predictions of the flux ACD_Eth (Fig. 5A), it stands out the presence of one single point for which the prediction error is many times higher than the deviance of the true values. The removal of this single outlier makes R^2 improve from $R^2 = -0.35$, up to 0.34 (Fig. 5B).

We also considered the Pearson correlation coefficient between true and predicted values (briefly ρ) to evaluate the models. It can be observed in Fig. 6 that the Pearson correlations are high. The median value of ρ is 0.90 for $g = 100\%$, 0.86 for $g = 70\%$ and 0.82 for $g = 50\%$.

To investigate in detail how the DNN performance is affected by a reduction of the number of features considered, we compared the ρ of each flux for the three g cases in Fig. 7. It can be observed that the worsening of the performance (decrease in ρ) is not homogeneously distributed among the different fluxes. On the contrary, the capability to predict many fluxes is nearly not affected by the change in g, whereas it dramatically worsens for a few fluxes.

It is natural to wonder whether the goodness of fit directly depends on the choice of the features that have been removed. The list of features that have been removed when $g = 50\%$ is: ACD, ATP, $AcCoAc$, Eth, $F16BP$, $FADH2$, $Glyox$, $H2O$, $MalCoA$, Mal, $NADH$, $NADPH$, $NADP$, NAD, Pyr, TrP, $O2$.

Surprisingly, the list includes most of the metabolites directly involved in the reactions with the best predictions in the $g = 50\%$ case, namely $FADH2\_H2O$ ($FADH2 + ADP + 0.55O_2 \Rightarrow FAD + ATP + H20$) and $NADH\_H2O$ ($NADH2 + ADP + 0.55O_2 \Rightarrow NAD + ATP + H20$). The ability of the DNN to predict well these two fluxes is also evident in the scatterplots

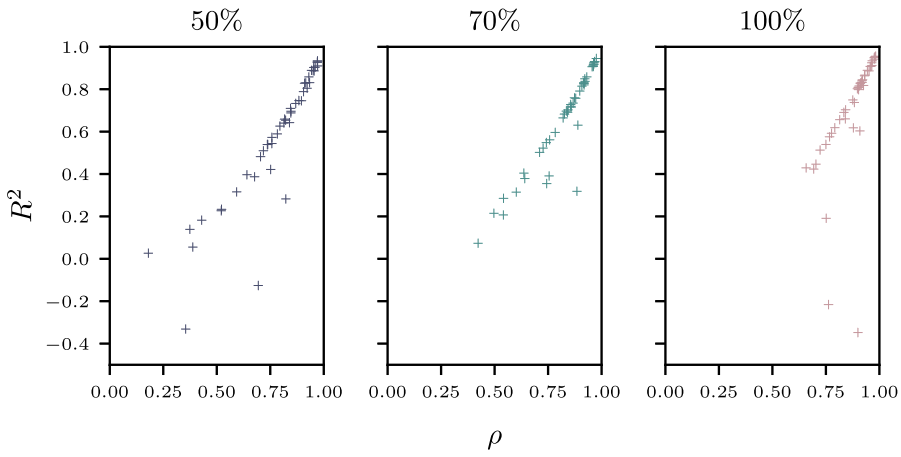


Fig. 6. Relation between Pearson and R^2 coefficients for true and predicted variation of each flux r, for different fractions of the original metabolites. We removed in advance the outliers for R^2 with a coefficient inferior to -5 (see Fig. 4).

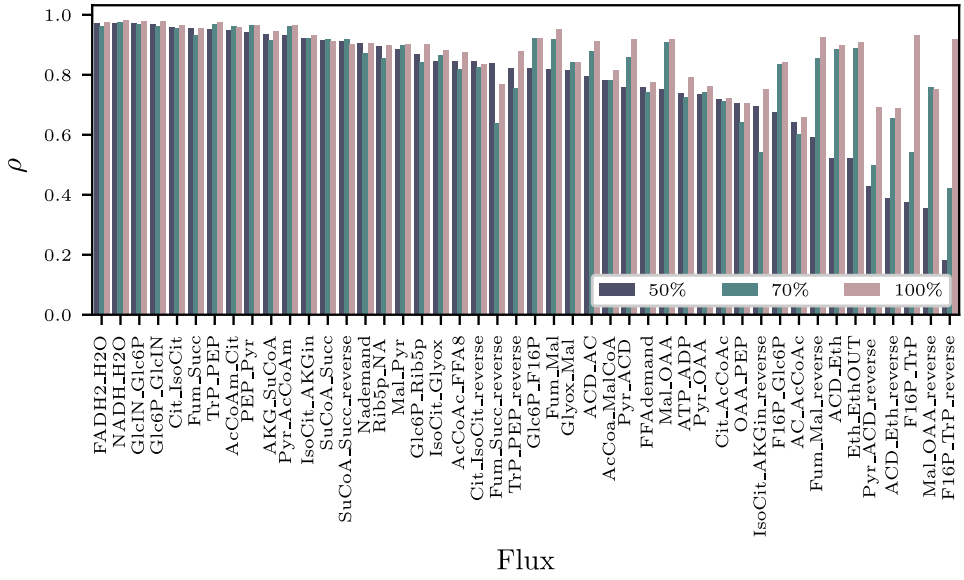


Fig. 7. Pearson correlation coefficient ρ for true and predicted variation of each flux r, for different fractions of the original metabolites. Results are sorted by decreasing ρ for the 50% fraction. For simplicity, we use the flux names to refer to their δs.

of true and predicted relative fluxes in Fig. 5B-C. This is a remarkable result, because it demonstrates that information on other metabolites in the network supports predictions in case of missing features.

However this consideration does not always hold. In fact, the list also includes all the metabolites directly involved in the reaction $F16P\_TrP\_reverse$ ($2TrP \Rightarrow F16BP$), and the substrate of the reaction $Mal\_OAA\_reverse$ ($MAL + NAD \Rightarrow OAA + NADH$), which display the worse prediction in the $g = 50\%$ case.

Taken together, the results of the performance evaluation confirm our hypothesis that patterns in metabolite abundance exist and that information in abundance variation can support the prediction of the flux variation even in reactions not directly involving those metabolites.

4.4. Performance is robust to feature reduction

Not all variation in the abundance of the different metabolites can be measured in a real system, and the variations that can be measured are hardly likely to coincide with our set of selected features. For this reason, it is relevant to investigate the effect of removing a random subset of features, instead of selecting them by looking at their pairwise correlation. To this aim, we evaluated the model performance for 10 randomly selected subsets of $g = 50\%$ and $g = 70\%$ metabolites with the overall best performing hyperparameters (i.e. hidden layers sizes (200,200), learning rate 0.001, optimizer *Adam* and no dropout). The results are reported in Fig. 8, where we show the distribution of the median R^2 of the test predictions for each random subset. It can be observed that by keeping $g = 50\%$ of the metabolites, the model performance showed greater variability (standard deviation = 0.058) than the case with $g = 70\%$ (standard deviation = 0.034). Interestingly, the model performance observed for our selected set of features falls within the first quartile and the median, in both cases.

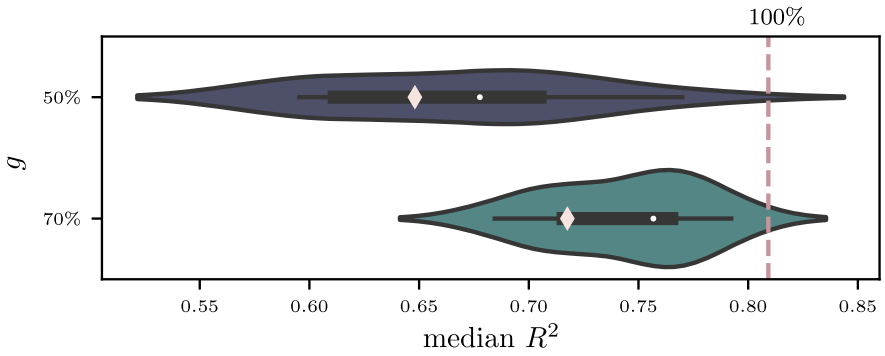


Fig. 8. Median test R^2 of 10 random subsets of $g = 50\%$ and $g = 70\%$ the original metabolites. The diamond indicates the median test R^2 obtained by our selection of features based on the absolute correlation and the dashed line corresponds to the median test R^2 achieved by keeping all the features, i.e. $g = 100\%$.

Remarkably, the median performance for R^2 drops by 6.4% only, when decreasing the cardinality of the set of features from $g = 100\%$ to $g = 70\%$, whereas it exhibits a drop of 16% when 50% of the features are removed.

It is also interesting to investigate whether the sensitivity to feature reduction differs across fluxes. In Supplementary Figure 2, the distribution of the reduction in the DNN prediction performance is reported for each flux. If we consider $g = 50\%$, it can be observed that some fluxes tend to be more sensitive to feature reduction, including the *F16P_TrP_reverse* flux, which in fact resulted sensitive also in Fig. 7. On the contrary, flux *SuCoA_Succ_reverse* does not seem to be particularly sensitive in Fig. 7, while displaying lower goodness of fit on average for random selections, suggesting that the low sensitivity of this flux observed in our selection was a result of the particular set of selected features.

Taken together, these results demonstrate that DNNs can predict flux variation well for most fluxes, regardless of the given subset of features. However, a few fluxes are intrinsically sensitive to feature reduction and would deserve further investigation.

4.5. Availability and scalability

The procedure described above was implemented using the `Python` libraries `Keras` [20] and `scikit-learn` [21]. The code and data used are available at github.com/BIMIB-DISCo/FLUX-PREDICT.

All tests were carried out on a machine with CPU 3.50 GHz Intel Xeon E3-1245 v5 and RAM 32 GB. The mean time required to train a configuration on the inner training set was 23.83 ± 11.58 minutes, while training the best model on the outer training set took 8.64 minutes.

Of course, the most computationally demanding step of our approach is the generation of the synthetic dataset by means of numerical simulations. In the specific case of the model used in this work, the total computational time to produce the data set was reasonable [11] (i.e., 5.5h to run ODEs simulations on a MacBookPro with CPU 2.6 GHz Intel Core i7, RAM 16 GB and to produce 268 Mb of data). Yet the computational time of this step depends on many factors, including the kinetic laws, the kinetic parameter values, the number of reactions and the number of simulations. An insufficient number of simulations, as well as the chosen variation range of each parameter, may impact on the goodness of fit of the ML model. However, given that we keep parameter values fixed when comparing two steady-states, the impact of under-sampling is expected to be limited. Furthermore, the computation of a large number of model trajectories may be reduced by exploiting GPU-accelerated algorithms.

5. Conclusions

We trained different configurations of deep neural networks to predict overall changes in the fluxes of a reaction system at the steady-state ($\boldsymbol{\delta v_p}$) from variations in the abundances of all or of some involved species ($\boldsymbol{\delta x_p}$). As training set, we used 100 thousands ($\boldsymbol{\delta x_p}, \boldsymbol{\delta v_p}$) pairs, obtained by sampling the parameter space and by simulating for each parameterization the steady state of a small metabolic network model under two different environmental conditions [12]. We have shown that DNNs can predict with a good level of confidence (median Pearson correlation between true and predicted values up to 0.9) changes in most reaction fluxes in the synthetic test dataset.

The fit remains good ($\rho = 0.82$) when up to 50% of the features are removed from the training set. When analyzing the goodness of fit for each output feature, we observed that the DNN predicts impressively well the variation in some fluxes, even when information on variation of the abundance of any species directly involved in the reaction is not given.

Our results indicate that patterns in relative abundances emerge from kinetic simulations of metabolic networks, with Monte Carlo generation of kinetic constants. These patterns reflect stoichiometric as well as mass balance constraints. The main advantage of using a DNN model to recognize such patterns *a posteriori*, instead of imposing constraints *a priori* as in constraint-based modeling, is the possibility to directly include information on relative abundances, instead of including them indirectly in the form of constraints on relative fluxes, which require limitations on admitted reaction rate laws (e.g.,

mass action) and are prone to feasibility problems. Analytical solutions, on the other hand, solve reactions individually, thus neglecting mass balance constraints, which are responsible to make predictions less sensitive to missing data.

A validation of the approach with experimental datasets and different metabolic models is of course desirable. Anyway, our approach has already the potential to pave the way for a systematic evaluation of alterations in metabolic fluxes, which is expected to guide drug target discovery, without the need for ad hoc laborious and expensive experiments and for explicit knowledge on kinetic parameters for dynamic simulations.

Our approach is not free from limitations. In order for fluxes to be predicted, the user must provide to the DNN deltas between two different conditions, whose difference must be controllable in order to be simulated (as e.g., difference in glucose availability). However, one may want to compare conditions whose triggering differences are not known *a priori*, as for instance pathological versus physiological conditions. A solution that we might envision and test in the future is to simulate many random perturbations to generate more heterogeneous ($\boldsymbol{\delta x_p}$, $\boldsymbol{\delta v_p}$) pairs and train a generic DNN.

In the future, we will also test our approach when more complex enzymatic kinetics are simulated. Difference in enzyme activities may be also taken into account by including proteomics or transcriptomics data. Finally, alternatives to DNNs, such as multi-target regression trees could also be evaluated.

CRediT authorship contribution statement

LP: Methodology, Software, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization; **FC**: Methodology, Software, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization; **DM**: Methodology, Formal Analysis, Writing - Review & Editing, Visualization; **AG**: Methodology, Formal Analysis, Writing - Review & Editing; **CD**: Conceptualization, Methodology, Formal Analysis, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision.

Declaration of competing interest

The authors declare that they have no competing interests.

Acknowledgments

We warmly thank Dario Pescini for providing the dataset and Giancarlo Mauri for the useful suggestions provided during the revision process.

Funding

The institutional financial support to SYSBIO.ISBE.IT within the Italian Roadmap for ESFRI Research Infrastructures and the FLAG-ERA grant ITFoC is gratefully acknowledged. Financial support from the Italian Ministry of University and Research (MIUR) through grant 'Dipartimenti di Eccellenza 2017' to University of Milano Bicocca is also greatly acknowledged. Support was also provided by the CRUK/AIRC Accelerator Award #22790: "Single-cell Cancer Evolution in the Clinic".

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ic.2021.104798.

References

[1] C. Damiani, D. Gaglio, E. Sacco, L. Alberghina, M. Vanoni, Systems metabolomics: from metabolomic snapshots to design principles, Curr. Opin. Biotechnol. 63 (2020) 190–199.
[2] S. Galmarini, J.V.-G. de Arellano, J. Duyzer, Fluxes of chemically reactive species inferred from mean concentration measurements, Atmos. Environ. 31 (15) (1997) 2371–2374.
[3] M. Sajitz-Hermstein, N. Töpfer, S. Kleessen, A.R. Fernie, Z. Nikoloski, IReMet-flux: constraint-based approach for integrating relative metabolite levels into a stoichiometric metabolic models, Bioinformatics 32 (17) (2016), i755–i762.
[4] V. Pandey, N. Hadadi, V. Hatzimanikatis, Enhanced flux prediction by integrating relative expression and relative metabolite abundance into thermodynamically consistent metabolic models, PLoS Comput. Biol. 15 (5) (2019) e1007036.
[5] P. Rana, C. Berry, P. Ghosh, S.S. Fong, Recent advances on constraint-based models by integrating machine learning, Curr. Opin. Biotechnol. 64 (2020) 85–91.
[6] G. Zampieri, S. Vijayakumar, E. Yaneske, C. Angione, Machine and deep learning meet genome-scale metabolic modeling, PLoS Comput. Biol. 15 (7) (2019) e1007084.
[7] M. Cuperlovic-Culf, Machine learning methods for analysis of metabolic data and metabolic pathway modeling, Metabolites 8 (1) (2018) 4.
[8] A. Ajjolli Nagaraja, N. Fontaine, M. Delsaut, P. Charton, C. Damour, B. Offmann, B. Grondin-Perez, F. Cadet, Flux prediction using artificial neural network (ann) for the upper part of glycolysis, PLoS ONE 14 (5) (2019) e0216178.
[9] A. Zelezniak, J. Vowinckel, F. Capuano, C.B. Messner, V. Demichev, N. Polowsky, M. Mülleder, S. Kamrad, B. Klaus, M.A. Keller, et al., Machine learning predicts the yeast metabolome from the quantitative proteome of kinase knockouts, Cell Syst. 7 (3) (2018) 269–283.
[10] Z. Costello, H.G. Martin, A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data, NPJ Syst. Biol. Appl. 4 (1) (2018) 1–14.

[11] C. Damiani, R. Colombo, M. Di Filippo, D. Pescini, G. Mauri, Linking alterations in metabolic fluxes with shifts in metabolite levels by means of kinetic modeling, in: Italian Workshop on Artificial Life and Evolutionary Computation, Springer, 2016, pp. 138–148.
[12] R. Colombo, C. Damiani, G. Mauri, D. Pescini, Constraining mechanism based simulations to identify ensembles of parametrizations to characterize metabolic features, in: International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics, Springer, 2016, pp. 107–117.
[13] R. Colombo, C. Damiani, D. Gilbert, M. Heiner, G. Mauri, D. Pescini, Emerging ensembles of kinetic parameters to characterize observed metabolic phenotypes, BMC Bioinform. 19 (7) (2018) 45–59.
[14] A. Cornish-Bowden, One hundred years of Michaelis–Menten kinetics, in: Proceedings of the Beilstein ESCEC Symposium - Celebrating the 100th Anniversary of Michaelis Menten-Kinetics, Perspect. Sci. 4 (2015) 3–9, https://doi.org/10.1016/j.pisc.2014.12.002.
[15] M.A. Savageau, Michaelis-Menten mechanism reconsidered: implications of fractal kinetics, J. Theor. Biol. 176 (1) (1995) 115–124.
[16] L. Petzold, Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, SIAM J. Sci. Stat. Comput. 4 (1) (1983) 136–148, https://doi.org/10.1137/0904010.
[17] M. Kuhn, K. Johnson, et al., Applied Predictive Modeling, vol. 26, Springer, 2013.
[18] H. Borchani, G. Varando, C. Bielza, P. Larrañaga, A survey on multi-output regression, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 5 (5) (2015) 216–233, https://doi.org/10.1002/widm.1157.
[19] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[20] F. Chollet, et al., Keras, https://keras.io, 2015.
[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

A.2 Optimization framework for personalised drug scheduling

Over the years we are witnessing an increasing availability of cancer patients clinical data, from which it is possible to extract data to build computational methods that design optimized drug scheduling protocols. We developed a closed-loop optimization framework, that exploits the theory of optimal control to optimize the administration schedule of Imatinib in Chronic Myeloid Leukemia patients. The framework is based on a population pharmacokinetics (PK) and pharmacodynamic (PD) ODEs model that is parametrized and optimized to minimize the adverse effects in a specific patient while maintaining high efficacy. We assessed the performance of our method using simulated data, showing the improvement in the decay of Cancer Stem Cells and a reduction in therapy toxcity with respect to standard Imatinib dosage.

The code used to perform the analyses presented in the publication is available at `https://github.com/BIMIB-DISCo/closedLoop-CT4TD`.

A closed-loop optimization framework for personalized cancer therapy design

Fabrizio Angaroni*
Dept. of Informatics, Systems and Communication
Universitá degli Studi di Milano-Bicocca, Milan, Italy
* equal contributor

Mattia Pennati*
Dept. of Informatics, Systems and Communication
Universitá degli Studi di Milano–Bicocca, Milan, Italy
* equal contributor

Lucrezia Patruno*
Dept. of Informatics, Systems and Communication
Universitá degli Studi di Milano–Bicocca, Milan, Italy
* equal contributor

Davide Maspero*
Dept. of Informatics, Systems and Communication
Universitá degli Studi di Milano–Bicocca, Milan, Italy
* equal contributor

Marco Antoniotti+,†
Dept. of Informatics, Systems and Communication and
Bicocca Bioinformatics, Biostatistics and Bioimaging Center (B4),
Universitá degli Studi di Milano–Bicocca, Milan, Italy
+ co-senior author
† correspondence: marco.antoniotti@unimib.it

Alex Graudenzi+,†
Institute of Molecular Bioimaging and Physiology
Consiglio Nazionale delle Ricerche, Segrate, Italy
+ co-senior author
† correspondence: alex.graudenzi@ibfm.cnr.it

***Abstract*—A current challenge in cancer research is the development of therapeutic strategies aimed at reducing the toxicity of treatments, since *Adverse Events* (AEs) typically cause substantial problems and long-term damages to the patients. A possible solution to this issue lies in the personalization of therapy dosages according to demographic factors and in the employment of optimized data-driven drug administration protocols. Control theory can be exploited to this end, as its application in pharmacology allows to define optimized dosages and schedules, aimed at minimizing AEs and maximizing the therapy efficacy. However, an effective application of control theory approaches to this issue is constrained by our ability in inferring the parameters of the mathematical models from currently available data.**

We here present a closed-loop optimization framework of patient-specific *pharmacokinetics* (PK) and *pharmacodynamics* (PD) models, combined with a mathematical model of a liquid tumor, which aims at overcoming such limitations. The most relevant feature of our framework is the ability to learn the value of patient-specific parameters via a Bayesian update, by exploiting a feedback signal obtained monitoring the tumor burden dynamics of the patient. Our framework employs CasADi, an open-source tool for nonlinear optimization, and guarantees a good and robust numerical estimation of the optimized schedule and a parsimonious use of computational time.

As a case study, we present the application of our framework to Tyrosine Kinase Inhibitor administration in Chronic Myeloid Leukemia (CML), in which we show that our optimized protocols result in a faster decay of CSCs and in a reduction of the overall toxicity.

***Index Terms*—Control theory, Cancer therapy, Bayesian update, Closed-loop optimization**

I. Introduction

The accumulation of clinical data on cancer patients and the concurrent development of efficient computational methods is paving the way to personalized cancer treatments, as it is now possible to employ computational procedures that automatically propose optimized drug scheduling protocols in a reliable and robust way. Such computational strategies can effectively exploit the information extracted from heterogeneous demographic factors of patients and from their ever changing clinical status [1]. In this regard, a combination of

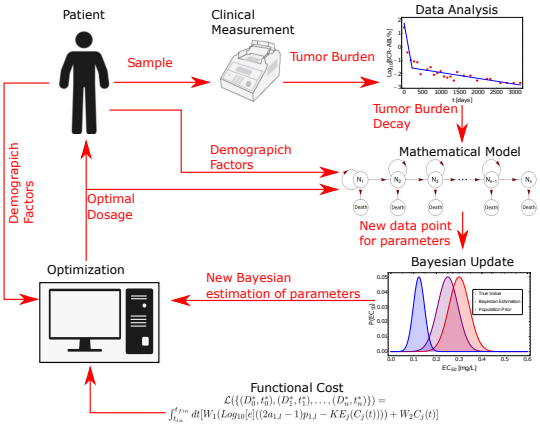


Fig. 1. **Schematic representation of the closed-loop optimization framework.** The procedure followed by our framework can be summarized with the scheme presented in the figure. In detail, given a patient, we employ its demographic factors inside the optimization and mathematical model and we update unknown patient-specific parameters with a Bayesian Update procedure employing clinical data of tumor burden dynamics (i.e., the feedback signal). Then, we optimize the dosage and we administrate the proposed optimized therapeutic strategy. This procedure is then iterated until treatment completion.

978-1-7281-9468-4/20/$31.00 ©2020 IEEE

methods from optimal control theory [2]–[12] techniques for data analysis [13]–[16] and mathematical models of cancer [17], [18] can be used to produce accurate predictive models of the clinical outcome of a given therapy in single cancer patients.

In this work, we introduce a closed-loop optimization framework, which employs CasADi, an open-source tool for nonlinear optimization and algorithmic differentiation [19], with the goal of delivering an optimized personalized drug administration schedule that adapts to the patient response to therapy (see Figure 1 for a schematic representation of the framework).

In brief, pharmacokinetics (PK) models describe the temporal dynamics of the concentration of a given drug in a certain tissue or organ, whereas pharmacodynamics (PD) models depict the efficacy of the drug with respect to distinct concentration values. Our framework uses patient-specific PK models based on demographic factors including age, sex and body weight [20] and it infers the parameters of patient specific PD model using longitudinal experimental data on tumor burden (e.g., the fraction of tumor cells on the total, in liquid tumors) coupled with a hierarchical population dynamics model [13], [21]–[23].

The estimation for these parameters is obtained via a Bayesian update [24] that mimics the continuous update of the distribution of parameters values (our beliefs), that occurs by exploiting new data [25].

Our theoretical scheme can be used to set optimized personalized administration strategies using different paradigms of therapy. In particular, in this work we set a cost that allows us to fine tune the therapy in order to rapidly minimize the tumor burden, while taking into account (i.e., minimizing) the toxicity and *Adverse Events* (AEs) of the therapy. Moreover, by changing the functional cost and without any need for major theoretical modifications, other applications are possible. For example: an adaptive therapy, with the goal to maintain the tumor burden stable and small, can be proposed [19].

Our work constitutes a proof of concept of an adaptive and personalized cancer therapy. In particular, we test it on the specific case of Imatinib administration in patients with Chronic Myeloid Leukemia (CML) and we show the benefit of employing our automated and data-driven framework in terms of increased efficacy of the therapy and reduction of the overall costs and toxicity.

This manuscript is structured as follows: we first introduce the framework, then we describe in detail its application to the CML case study and we finally present the results of simulations over synthetic patients.

II. Theoretical framework

Since in cancer therapy it is often difficult to obtain real-time measurements, straightforward applications of feedback-control theory cannot be employed [3]. For such reason, here we expand the framework presented in [26] with an automated procedure that updates its parameters as soon as new measurements of the patient status are collected. Our framework is composed by the following components.

1) Mathematical models of the disease and of personalized PK/PD.
2) A functional cost that is used to modulate the behavior of the approach.
3) An optimization algorithm.
4) A Bayesian update scheme.

A schematic representation of the workflow can be found in Fig. 1.

In brief, considering a time windows $\Delta t = t_n - t_0$ divided into n arbitrary intervals $\Delta t_i = t_{i+1} - t_i$, $\forall i = 1, \ldots, n-1$, we employ an optimization in the time interval Δt_i using the estimated value of the parameters at t_i. In the scheme, the optimized drug dosage is administrated to the patient until time t_{i+1}. Then, we consider the new measurement taken at t_{i+1} and we update the parameters of the mathematical models. These steps are repeated until treatment completion. More details follow.

A. Mathematical models

The key components of our scheme are the mathematical models used to represent the dynamics of the drug (namely PK/PD models) and the population dynamics model of cancers cells. PK models [27] are mathematical models whose solutions describe the dynamics of the concentration of a substance in a specific tissue of the body. These models are represented as a system of differential equations derived using the law of conservation of mass and assuming that the body is composed by a certain number of macroscopic coupled subsystem (i.e., compartments) [28]. With these assumptions, given a dosage function $D(t)$ (i.e., the control function – see Section II-C– that must be continuous, differentiable and non-negative), and r parameters of a specific drug (i.e., $\{\psi_1, \ldots, \psi_r\}$) the concentration of a drug in the compartments of interest is a continuous and differentiable function ($C(t)$) specified by the following equation:

$$\frac{\partial C(t)}{\partial t} = f(t, D(t); \psi_1, \ldots, \psi_r). \tag{1}$$

A limitation of PK models is that they assume an instantaneous mixing of the drug in a compartment and a perfect transport among them.

In particular, our framework employs population PK models (e.g., [20]) that assume that the drug parameters are a function of k demographic factors (i.e., $\{\nu_1, \ldots, \nu_k\}$) such as age, sex and weight of the patients. Accordingly, Eq. (1) becomes:

$$\frac{\partial C(t)}{\partial t} = f(t, D(t); \psi_1(\nu_1, \ldots, \nu_k), \ldots, \psi_r(\nu_1, \ldots, \nu_k)). \tag{2}$$

Specifically, in population PK models it is assumed that the different concentrations registered in different patients are described by non-linear mixed effect models, and every considered covariate is associated to an induced variance on $C(t)$ via a search algorithm (e.g., maximum likelihood) [29] .

PD describes the biochemical reactions occurring in the body [27], [30] in order to quantify the efficacy of a therapy and its relationship with the drug concentration in the tissue of interest (i.e., $C(t)$) [31]. In general, given s parameters $\{\rho_1, \ldots, \rho_s\}$, the efficacy could be written as a continuous, non-negative and differentiable function, i.e.:

$$E(t) = g(C(t); \rho_1, \ldots, \rho_s). \tag{3}$$

Our framework infers the personalized values of each ρ_i directly from observed data. Together, PK and PD model are called PK/PD models and they are commonly used to define standard dosage guidelines (e.g., [32]).

Our framework employs an additional mathematical model, alongside PK/PD, in order to take into account the response of tumor to the therapy. In recent years, mathematical models of cancer population dynamics have been increasingly employed to study the complexity of cancer, providing valuable insights into tumor mechanisms, as well as accurate quantitative predictions [17], [18].

In particular, hematopoietic cell population dynamics has been widely studied, both in healthy systems [33] and in cancer [13]–[15], [17], [21]–[23].

Some of these models exploit the fact that the cells of the hematopoietic system follow an organized and ordered sequence of discrete differentiation states, which can also be interpreted as a hierarchical structure. Such organization is divided into s non-intersecting compartments or cell types (e.g., *stem cells*, *progenitor cells*, etc.), and every cell in the system is associated, in a unique way, to one of these compartments.

In order to model the cancer population dynamics, our framework employs a system of Ordinary Differential Equations (ODEs) that takes in account the differentiation hierarchy in the hematopoietic system [13]–[15], [17], [21]–[23].

The transitions among the compartments are defined by the underlying biological differentiation process, whose parameters are the following:

- p_i is the division rate of the cells in the i compartments.
- $a_i \in [0, 1]$ is the probability that, when a cell undergoes mitosis, both of its daughters belong to the i compartment; therefore, $1 - a_i$ is the probability of belonging to the $i + 1$ compartment.
- $d_i(E(t))$ is the death rate of the cancer cells in the i compartment. Note that this rate is dependent on the efficacy of the therapy and consequently on the dosage $D(t)$.

In general, for m compartments, we can model the dynamics of the number of cancer cells in the patient (i.e., the *Tumor Burden*) as follows:

$$\begin{aligned}\mathrm{TB}(t) = \\ =h(t, d_1(E(t)), \ldots, d_m(E(t)); p_1, \ldots, p_m, a_1, \ldots, a_m).\end{aligned} \tag{4}$$

B. Functional cost

The *functional cost* is one of the key elements of the optimal control problem. Our theoretical scheme can be used to set optimized personalized administration strategies using different therapy paradigms. In this work, we set a cost based on the assumption that a patient receives maximum benefit by killing as many cancer cells as possible, in the shortest possible time. Thus, this choice allows us to tune the therapy in order to rapidly minimize the tumor burden $\mathrm{TB}(t)$, while taking into account (i.e., minimizing) the therapy's AEs.

To take toxicity into account, we observe that it is directly proportional to the *Area Under the Curve* (AUC) of the concentration function $C(t)$ [34], while to consider the variation of the tumor burden we observe that, if in Eq. (4) $\mathrm{TB}(t)$ is monotonic with respect to variations of the death rate, then maximizing the death rate corresponds to minimizing rapidly the tumor burden. Accordingly, we have Eq. (5) (see top of next page).

In Eq. (5) $r(p_1 \ldots p_m, a_1 \ldots a_m, d_1(E(t)) \ldots d_s(E(t)))$ is a real, continuous and differentiable function that represents the variation of TB(t), which depends on the death rates d_i. Since the cost is a multi-objective function, we have introduced two arbitrary weights W_1 and W_2, which account for the relative relevance of the two distinct components, respectively tumor burden and toxicity.

C. Control Problem and Optimization Algorithm

We define the control problem in the following way: let t_0 and t_n be the starting and the ending time point respectively of the temporal windows in which the therapy is optimized, then our goal is identifying the function $D(t)$ such that the functional cost in Eq. (5) is minimized.

Usually, the control function requires the definition of a lower and upper bound. However, in our framework, these bounds are implicitly regulated by the two weights W_1 and W_2, as long as both of them are different from 0. This is due to the fact that we implicitly assume that the toxicity and the variation of $\mathrm{TB}(t)$ are inversely proportional.

As we aim at defining a framework that can adapt to different biological scenarios represented by different functional forms of ODEs systems and costs (usually non-linear), we here employ a set of heuristics. In particular, we have selected an algorithm that follows the *transform then optimize* paradigm, via multiple shooting [35], and which was implemented in Python by employing the functionalities of CasADi, a package for nonlinear optimization and algorithmic differentiation [19].

D. Bayesian Update scheme

Every time a new measurement is available, our framework employs a recursive scheme for the estimation of the parameters of the mathematical models, which is similar to the approach of Bayesian filters, i.e., predict-observe-filter-predict-observe-filter, etc. [36].

Roughly, for every new measurement, we treat the *posterior probability* (i.e., the Bayesian estimation) computed previously as a *prior* for the current iteration; then we update our knowledge with the new measurement:

$$\mathcal{L}(t,\, C(t,\, D(t))) = \int_{t_0}^{t_n} [W_1 r(p_1 \dots p_m,\, a_1 \dots a_m,\, d_1(E(t)) \dots d_m(E(t))) + W_2 C(t)]\, dt \tag{5}$$

new estimation $\propto$ *previous estimation* $\times$ *likelihood of new data.*

An explicit (analytical) formula can be obtained only by making specific assumptions on the functional form of probability distributions of the parameter values involved in the update. Here, we assume normal distributions.

Let $\mu(t_i)$ for $i = 1, \dots n$ be the average values obtained at time t_i, with $\sigma_{\mu(t_i)}$ is its variance. k is the sample size, $\widetilde{\mu}(t_i)$ is the Bayesian estimation at iteration t_i, with $\widetilde{\sigma}_{\mu(t_i)}$ its variance, and $\widetilde{\mu}(t_0)$ the prior value of the parameter. Then we have that [37]:

$$\widetilde{\mu}(t_i) = \frac{k\widetilde{\sigma}^2_{\mu(t_i)}\mu(t_i) + \sigma^2_{\mu(t_{i-1})}\widetilde{\mu}(t_{i-1})}{k\widetilde{\sigma}^2_{\mu(t_{i-1})} + \sigma^2_{\mu(t_i)}}, \tag{6}$$

$$\widetilde{\sigma}^2_{\mu(t_i)} = \frac{\widetilde{\sigma}^2_{\mu(t_{i-1})}\sigma^2_{\mu(t_i)}}{k\widetilde{\sigma}^2_{\mu(t_{i-1})} + \sigma^2_{\mu(t_i)}}. \tag{7}$$

In our framework we use the distribution of the overall population as prior for the first update.

III. Case Study

The framework above specified is general, and must be adapted to different combination of therapy and disease. Here we present its application to CML and tyrosine kinase inhibitor therapy.

A. Imatinib Patient Specific PK/PD Models

Imatinib is an inhibitor of the BCR-ABL tyrosine kinase. It binds to the inactive form of BCR-ABL even at nanomolar concentration and competes with the ATP for its binding pocket. This interaction hinders the switch of the fusion kinase to the active form leading to the death of the aberrant cells [38].

We here employ the population PK model of intravenous administration of Imatinib [20]. Given, k_a the first order absorption rate, f the bioavailability, $D(t)$ the time-dependent dosage, v the volume of the distribution, CL the clearance, $C(t)$ the concentration in the blood, $\chi_b(t)$ the amount of Imatinib in the blood ($C(t) = \frac{\chi_b(t)}{v}$), then:

$$\frac{d\chi_b(t)}{dt} = +k_a f D(t) - \mathrm{CL} \cdot C(t), \tag{8}$$

Parameters of Eq. (8) are tuned to consider demographic factors such as body weight, age and sex, thus providing patient-specific PK models. We determine the parameters using the equations proposed in [20], i.e.:

$$\begin{aligned} CL = {} & \theta_a \\ & + \theta_1 \frac{BW - \overline{BW}}{\overline{BW}} \\ & + \theta_2 q - \theta_2 (1 - q) \\ & + \theta_3 \frac{AGE - \overline{AGE}}{\overline{AGE}}, \end{aligned} \tag{9}$$

$$v = \theta_b + \theta_4 q - \theta_4 (1 - q), \tag{10}$$

where θ_i, for $i = a,\, b,\, 1,\, 2,\, 3,\, 4$, are constants, BW is the body weight of the patient and $\overline{BW}$ is its population-average, AGE is the age of the patient and $\overline{AGE}$ its population-average and q is a binary variable which takes value 1 for male and 0 for female. The values of such parameters, taken from [20], are given in Tables I and II .

Parameter	Value	Standard Error	Unit of measurement
k_a	0.61	30%	h^{-1}
CL	14.3	7.1%	L/h
v	347	17.9%	L
$\overline{BW}$	70	Na	kg
$\overline{AGE}$	50	Na	$Years$
f	1	Na	–

TABLE I
AVERAGE PARAMETERS OF PK MODEL OF IMATINIB FROM [20].

Parameter	Value
k_a	0.437
θ_a	12.8
θ_b	258
θ_1	12.7
θ_2	0.8
θ_3	-2.1
θ_4	61.0

TABLE II
SUMMARY OF THE DEMOGRAPHIC POPULATION PK PARAMETERS FOR IMATINIB FROM [20].

The PD model used for this case-study is based on the maximum-inhibition effect (E_{max}) [32]:

$$E(C(t)) = \frac{E_{max} \cdot C(t)}{EC_{50} + C(t)}, \tag{11}$$

where $E(t)$ is the efficacy, E_{max} is the maximum efficacy (set to 1), $C(t)$ is the concentration of the drug in the blood, EC_{50} is the concentration of the drug that produces half of maximal effect. In this framework, EC_{50} is the parameter that will be updated at every new measurement exploiting the inference on a patient's data (See Section III-C).

B. CML Mathematical Model

In order to describe the hierarchy and the dynamics of cancer cells in CML, we employ the simplest model, including two compartments: (i) cancer stem cells (CSCs) $l_1(t)$ and (ii) progressively differentiated cancer cells $l_2(t)$ [13]–[15], [21]–[23], [39]. This choice ensures that the ODEs system is identifiable, given the available data on CML tumor burden [26], [40], [41].

Given these hypotheses, we specify the mathematical model for the dynamics of cancer cells (See Eq. (4)) in CML:

$$\begin{array}{rcl} \frac{dl_1(t)}{dt} & = & \lambda l_1(t), \\ \frac{dl_2(t)}{dt} & = & \gamma l_1(t) + \tau l_2(t), \end{array} \tag{12}$$

where:

$$\begin{array}{rcl} \lambda & = & (2a_1 - 1)p_1 - d_1, \\ \gamma & = & 2(1 - a_1)p_1, \\ \tau & = & -d_2. \end{array} \tag{13}$$

This is a typical example of a linear autonomous system, and the analytical solution could be obtained in a recursive way:

$$\begin{array}{rcl} l_1(t) & = & l_1(0)e^{\lambda t}, \\ l_2(t) & = & \frac{e^{\tau t}(\gamma l_1(0) - \lambda l_2(0) + \tau l_2(0)) - l_1(0)\gamma e^{-\lambda t}}{\tau - \lambda}. \end{array} \tag{14}$$

C. Data Analysis

In the follow-up of CML, it is possible to estimate the tumor burden using Q-PCR measurement, which distinguishes between cancer and healthy cells by detecting BCR-ABL mutation. This experiment is non-invasive and low-cost and allows to have longitudinal reliable experimental data of the dynamics of the tumor burden [13]–[16], [26]. The variation of the tumor burden in a single patient is represented by a biphasic exponential, which in log-scale is described by two distinct and intersecting lines [13]–[15].

In detail, we assume that after the first major molecular response, the data accounts for the dynamics of the CSCs subpopulation only [13]–[15]. In this way, it is possible to evaluate the effect of any average concentration of Imatinib therapy directly on the CSCs decay. For every patient, it is possible to estimate via a linear regression β_j, i.e., the measure of the CSCs decay [13]–[16], [26]. Then, given the previous mathematical model, we obtain the following:

$$\beta_j = \log[e][(2a_{1,j} - 1)p_{1,j} - d_{1,j}] = \log[e]\lambda_j, \tag{15}$$

where j is the patient's index and λ_j is the observed net growth rate of the CSCs.

By fixing $a_{1,j}$ and $p_{1,j}$ to patient-specific constant values, we obtain an estimate for patient specific CSCs death rate ($d_{1,j}$). Finally, by supposing a linear relation between average efficacy $\langle E \rangle$ and the CSCs death rate $\langle d_{1,j} \rangle$ [26], [42], we can estimate the personalized parameters of the PD model (i.e., $EC_{50,j}$) for all patients [26], [32], [43] by means of the following relation:

$$EC_{50,j} = \bar{C}_j \left[\frac{KE_{max}}{d_{1,j}} - 1 \right]. \tag{16}$$

Where K is a conversion constant equal to $0.377\,[day^{-1}]$ [26]. Note that in this work, we impose that the maximum efficacy is $E_{max} = 1$, and $\bar{C}_j$ is the time-average drug concentration in patients. $EC_{50,j}$ is the target of our Bayesian update. In particular, we update its patient specific estimate using Eq.s (6) (7) every two simulated q-PCR measurements. Lastly, at t_0 we employ as prior the population average, where $\widetilde{EC}_{50}(t_0) = 0.1234\ [mg/L]$ [32].

D. Functional Cost

Given the mathematical model defined in the previous sections, for this CML case study the cost in Eq. (5) becomes the one shown in Eq. (17) (see top of next page).

Notice that Eq. (17) includes the net CSCs growth rate, i.e., λ in Eq. (14). This choice allows to avoid the estimation of the initial number of CSCs $l_1(0)$ [26]. The ratio $\phi = \frac{W_1}{W_2}$ determines the overall behavior of the optimized solution (see section IV-B).

E. Synthetic Tests

To assess the reliability of our approach, we ran two different extensive synthetic tests.

1) Precision of the Bayesian update scheme: The first set of tests was run to evaluate the precision of the Bayesian update scheme. To this end, we generated 500 pairs of $EC_{50,GT}$ and $\sigma_{EC_{50,GT}}$; for every pair we simulated up to 100 measurements of $EC_{50,GT}$ with a normal error. Then, we evaluated the trend of the relative error (Er) between the Bayesian estimation (i.e., $\widetilde{EC}_{50}(t_i)$) and the ground-truth $EC_{50,GT}$:

$$Er(t_i) = \frac{|EC_{50,GT} - \widetilde{EC}_{50}(t_i)|}{EC_{50,GT}}. \tag{18}$$

Finally, we considered the average relative error (i.e. $\xi(t_i)$) for every iteration.

2) Control scheme performances: To study the performances of our control scheme, we generated 50 synthetic patients. A synthetic patient is represented by a vector of parameters $[AGE,\ BW,\ Sex,\ EC_{50},\ a_1,\ p_1,\ l_1(t_0 = 0)]$. The values for all the parameters, with the exception of Sex, were sampled from a Normal Distribution (see Table III for the specific values of mean and standard deviation for each parameter). The values for Sex were sampled from a Uniform distribution, given that it is a binary variable which takes value 1 for men and 0 for women.

To get the tumor progression data for each simulated patient, we employed an *in-silico* simulation of the disease, given by Eq. (12). Finally, to study a more realistic case we added simulated guassian noise to the obtained measurements of $l_1(t_i)$ with a variance of 5%.

3) Overall Simulation Algorithm: The final layout of the simulation steps is the following.

- Measure the tumor burden $l_1(t_i)$.
- Simulate the time evolution of the tumor burden under the optimized dosage, evaluated at t_i, between t_i and t_{i+2}.
- Measure the tumor burden $l_1(t_{i+1})$ and $l_1(t_{i+2})$.

$$\begin{aligned}\mathcal{L}(t,\, C(t,\, D(t))) &= \int_{t_0}^{t_n} dt\ [W_1 \cdot (\log[e]\lambda(E_j(C^*(t)))) + W_2 C^*(t)] \\ &= \int_{t_0}^{t_n} dt\ [W_1 \cdot (\log[e]((2a_1 - 1)p_1 - K E_j(C^*(t)))) + W_2 C^*(t)]\end{aligned} \tag{17}$$

Parameter	Mean	Standard deviation	Unit of measurament
AGE	50	10	$Years$
BW	70	14	kg
EC_{50}	0.12	0.024	mg/L
a_1	0.87	0.174	–
p_1	0.45	0.09	day^{-1}
$l_1(0)$	10^6	2×10^5	$cell$

TABLE III

PARAMETERS OF THE SYNTHETIC DATASET. WE HAVE GENERATED EACH PATIENT BY SAMPLING THE VALUES FOR EVERY PARAMETER FROM A NORMAL DISTRIBUTION WITH MEAN AND STANDARD DEVIATION SPECIFIED IN THE TABLE. NOTE THAT THE PARAMETER Sex IS NOT INCLUDED IN THE TABLE AS IT IS A BINARY VARIABLE (1 INDICATES MEN AND 0 INDICATES WOMEN) AND ITS VALUES WERE SAMPLED FROM A UNIFORM DISTRIBUTION.

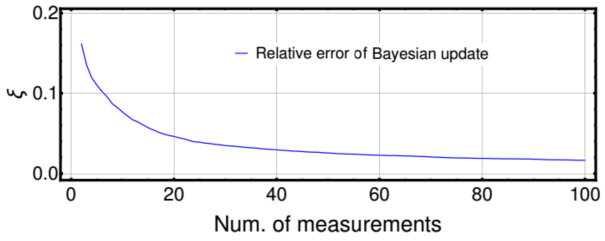


Fig. 2. Trend of the average relative error (ξ) between the Bayesian estimation (i.e., $\widehat{EC_{50}}(t_i)$) and the ground truth $EC_{50,GT}$, with respect to the number of measurements. We observe that our method converges in ≈ 40 iterations to the ground truth value, and it also emerges that after ≈ 30 iterations the error is smaller than 5%.

- Evaluate the slope of the CSCs decay between $l_1(t_i)$ and $l_1(t_{i+2})$.
- Compute EC_{50} at t_{i+2}.
- Update the estimation of EC_{50}.
- Solve the control problem (i.e., find the new optimized personalized dosage).
- Simulate the time evolution of the tumor burden under the new optimized dosage between t_{i+2} and t_{i+4}.
- Restart from the first step.

For all patient, by default, we performed 30 iterations. However, note that the computation was stopped for those patients in which the therapy was not effective (i.e., $l_1 1(t) > 10^7$), or for those where the therapy had eradicated all tumor cells (i.e., $l_1(t) < 1$).

In addition, since different values of parameter Φ highly influence the final outcome, the whole procedure was run using multiple values for $\Phi \in \{50, 55, \ldots, 90\}$.

IV. RESULTS

A. Performance of Bayesian Update

As stated in Section III-E, we first evaluated the precision of the Bayesian update, measuring the population average relative error ($\xi(t_i)$). In Figure 2, one can see that after ≈ 40 measurements there is no substantial improvement (the difference between two consecutive errors is smaller than 1%). More importantly, after 30 iterations the error is smaller than 5%. This result proves that our approach to the inference of the parameters gives good estimates for EC_{50} with a relative small number of measurements, as it converges sufficiently rapidly to the ground truth values. This demonstrates that our framework can be employed to obtain parameters estimates in a scenario without real-time measurements.

B. Performance of close-loop optimization scheme on CML therapy design

In Figure 3, we present the performance assessment of the framework by comparing the optimized therapy provided by our framework with the standard Imatinib therapy (400 [mg] per day, simulated as in [26]). In Figure 3-A, we compare the average decay of CSCs for each single patient obtained with our control framework with that obtained by simulating standard Imanitinb dosage. To calculate the values displayed in the boxplot, we computed the difference between optimized and standard therapy; negative values indicate that the optimized dosage leads to a faster decay of CSCs. Interestingly, one can notice that for certain values of Φ (from 70 to 90) the absolute value of the decay has increased. In particular, for $\Phi = 80$ the median improvement of the decay is $-0.018[\text{day}]^{-1}$.

Besides such improvements, in Figure 3-B we compare the AUC obtained through the optimized protocol with that obtained with standard therapy. Positive values in the plot indicate higher toxicity for 400 [mg] per day of Imatinib and we see that the median AUC difference is smaller than 0 for every Φ, with the exception of 85 and 90. Importantly, one can notice that for $\Phi = 50$ the median decay of the CSCs is similar to one of standard therapy, but the AUC is significantly smaller for almost all patients. This means that with the dosages proposed by our framework, patients receive high benefits in term of AEs.

Our results prove that a personalized and optimized therapy adapts to inter-patient variability and it is viable also when some of the patient specific parameters (e.g., EC_{50}) are not known at the beginning of the therapy, as they are inferred exploiting the longitudinally of data during the drug administration follow-up.

The greatest advantage of the administration protocol proposed by our framework is not in the final outcome of the therapy, but it lies in the significant reduction in therapy toxicity. Then, by following optimized protocols the quality of life for patients can be improved. In addition, an optimized drug schedule implies a long-term lower drug dosage. This leads to lower cost for the healthcare system and thus it determines a wider accessibility to therapies from patients

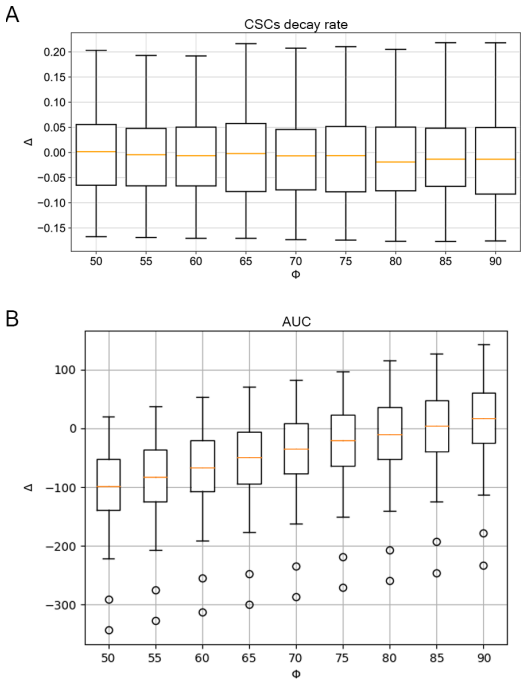


Fig. 3. Performance assessment of our framework. **A**: Δ between the average decay of CSCs for each single patient obtained with our framework and that obtained by simulating standard Imanitinb dosage. Negative values indicate that the optimized dosage has lead to a faster decay of CSCs. Tests were carried out considering different values of Φ, which are all displayed in the boxplot (x-axis). **B**: we compare the AUC obtained through the optimized protocol with that obtained with standard therapy. Positive values in the plot indicate higher toxicity for 400 [mg] per day of Imatinib. Also in this case, performances are presented grouped by the value of Φ used in the tests. From both A and B we observe that, together with the improvement of the CSCs decay rate, our frameworks leads to a substantial decrease in the toxicity of therapy. Indeed, median AUC values are always negative with the exception of $\Phi = 85$ and $\Phi = 90$.

[44]–[46].

V. Conclusions

In this work we propose a framework for the optimization of personalized liquid tumors treatment. Our scheme employs a population PK model and allows us to deliver a dosage that is consistent with the inter-patient variability due to demographic covariates. In addition, it automatically adapts its parameters to the response to the therapy, via a Bayesian update procedure. We assessed the performance of the procedure by means of synthetic simulations. First of all we have shown that our data analysis procedure gives good results, in term of convergence to the ground truth value. This demonstrates that our approach can be effectively employed to estimate the patient's specific parameters during therapy, in a realistic scenario that lacks real-time measurements. Furthermore, our framework improves the current standard therapy in terms of tumor burden reduction and striking improvements are also observed with respect to AUC, meaning that this procedure may lead to a substantial reduction of both the probability of AEs and overall costs of therapy. This might impact to quality of life and the long-term survival of cancer patients subject to a pharmacological therapy.

We also remark that our framework might be further developed, to take advantage of the growing availability of different data types, in order to deliver a more accurate and robust inference, which is able to better capture the biological complexity of pathological states. In particular, data collected from multiple omics measurements are increasingly available at decreasing costs and might be exploited to characterize inter- and intra-tumor heterogeneity [47]–[50], possibly allowing to explicitly consider multiple cancer sub-populations with different drug sensitivity [51].

In addition, the cost function included in our framework can be modified in order to model different therapy designs. For example, it would be possible to design a cost function that assumes that the goal of therapy is not to eradicate the tumor burden, but to maintain it at a constant small dimension, that is by cronicizing the disease, e.g., via evolutionary adaptive therapy [51]).

Nevertheless, the road to automatized personalized cancer treatments remains impervious. For instance, the significant intrinsic variability of cancer (sub)population dynamics, which is due to the high levels of inter- and intra-tumor heterogeneity observed in most cancer (sub)types [52], might limit the generality of the results and the predictions of our framework [53], [54]. On the one hand, improvements in our ability to model complex phenomena involved in cancer evolution might be essential to overcome such limitations. For example, phenomena such as intra-tumor spatial heterogeneity [55] might be included within the modeling framework, possibly by exploiting the resolution of high-resolution measurements such as single-cell sequencing data [56] or spatial transcriptomic data [57]. On the other hand, however, more complex models clearly require the estimation of an increasing number of parameters, and this might often been unfeasible (i.e., identifiability issues). Hence, the trade-off between model expressivity and parameter identifiability should be wisely considered when defining any modeling framework, according to the specific experimental scenario and the related research questions.

Code and data availability

1The source code used to replicate all the analyses is available at this link: `https://github.com/BIMIB-DISCo/closedLoop-CT4TD`.

Acknowledgments

This work was partially supported by the Elixir Italian Chapter and the SysBioNet project, a Ministero dell'Istruzione, dell'Università e della Ricerca initiative for the Italian Roadmap of European Strategy Forum on Research Infrastructures. Partial support was also provided Support was also provided by the CRUK/AIRC Accelerator Award #22790, "Single-cell Cancer Evolution in the Clinic".

REFERENCES

[1] R. Salgado, H. Moore, J. W. Martens, T. Lively, S. Malik, U. McDermott, S. Michiels, J. A. Moscow, S. Tejpar, T. McKee, and et al., "Steps forward for cancer precision medicine," *Nature Reviews Drug Discovery*, vol. 17, no. 1, pp. 1–2, 2018.

[2] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1.

[3] J. M. Bailey and W. M. Haddad, "Drug dosing control in clinical pharmacology," *IEEE Control Systems*, vol. 25, no. 2, pp. 35–51, 2005.

[4] S. Lenhart and J. T. Workman, *Optimal control applied to biological models*. Crc Press, 2007.

[5] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[6] W. M. Haddad, T. Hayakawa, and J. M. Bailey, "Adaptive control for nonlinear compartmental dynamical systems with applications to clinical pharmacology," *Systems & Control Letters*, vol. 55, no. 1, pp. 62–70, 2006.

[7] G. M. Steil, "Algorithms for a closed-loop artificial pancreas: the case for proportional-integral-derivative control," *Journal of diabetes science and technology*, vol. 7, no. 6, pp. 1621–1631, 2013.

[8] J. Shi, O. Alagoz, F. S. Erenay, and Q. Su, "A survey of optimization models on cancer chemotherapy treatment planning," *Annals of Operations Research*, vol. 221, no. 1, pp. 331–356, 2014.

[9] M. Fuentes-Garí, E. Velliou, R. Misener, E. Pefani, M. Rende, N. Panoskaltsis, A. Mantalaris, and E. N. Pistikopoulos, "A systematic framework for the design, simulation and optimization of personalized healthcare: making and healing blood," *Computers & Chemical Engineering*, vol. 81, pp. 80–93, 2015.

[10] D. Jayachandran, A. E. Rundell, R. E. Hannemann, T. A. Vik, and D. Ramkrishna, "Optimal chemotherapy for leukemia: a model-based strategy for individualized treatment," *PloS one*, vol. 9, no. 10, p. e109623, 2014.

[11] I. Naşcu, A. Krieger, C. M. Ionescu, and E. N. Pistikopoulos, "Advanced model-based control studies for the induction and maintenance of intravenous anaesthesia," *IEEE Transactions on biomedical engineering*, vol. 62, no. 3, pp. 832–841, 2015.

[12] N. Babaei and M. U. Salamci, "Personalized drug administration for cancer treatment using model reference adaptive control," *Journal of theoretical biology*, vol. 371, pp. 24–44, 2015.

[13] F. Michor, T. P. Hughes, Y. Iwasa, S. Branford, N. P. Shah, C. L. Sawyers, and M. A. Nowak, "Dynamics of chronic myeloid leukaemia," *Nature*, vol. 435, no. 7046, p. 1267, 2005.

[14] M. Tang, M. Gonen, A. Quintas-Cardama, J. Cortes, H. Kantarjian, C. Field, T. P. Hughes, S. Branford, and F. Michor, "Dynamics of chronic myeloid leukemia response to long-term targeted therapy reveal treatment effects on leukemic stem cells," *Blood*, vol. 118, no. 6, pp. 1622–1631, 2011.

[15] A. Olshen, M. Tang, J. Cortes, M. Gonen, T. Hughes, S. Branford, A. Quintás-Cardama, and F. Michor, "Dynamics of chronic myeloid leukemia response to dasatinib, nilotinib, and high-dose imatinib," *haematologica*, pp. haematol–2013, 2014.

[16] A. Rainero, F. Angaroni, A. Conti, C. Pirrone, G. Micheloni, L. Tararà, G. Millefanti, E. Maserati, R. Valli, O. Spinelli, and et al., "gdna qpcr is statistically more reliable than mrna analysis in detecting leukemic cells to monitor cml," *Cell death & disease*, vol. 9, no. 3, p. 349, 2018.

[17] P. M. Altrock, L. L. Liu, and F. Michor, "The mathematics of cancer: integrating quantitative models," *Nature Reviews Cancer*, vol. 15, no. 12, p. 730, 2015.

[18] K. H. Khan, D. Cunningham, B. Werner, G. Vlachogiannis, I. Spiteri, T. Heide, J. F. Mateos, A. Vatsiou, A. Lampis, M. D. Damavandi *et al.*, "Longitudinal liquid biopsy and mathematical modeling of clonal evolution forecast time to treatment failure in the prospect-c phase ii colorectal cancer clinical trial," *Cancer discovery*, vol. 8, no. 10, pp. 1270–1285, 2018.

[19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[20] N. Widmer, L. Decosterd, C. Csajka, S. Leyvraz, M. Duchosal, A. Rosselet, B. Rochat, C. Eap, H. Henry, J. Biollaz, and et al., "Population pharmacokinetics of imatinib and the role of α1-acid glycoprotein," *British journal of clinical pharmacology*, vol. 62, no. 1, pp. 97–112, 2006.

[21] T. Stiehl and A. Marciniak-Czochra, "Mathematical modeling of leukemogenesis and cancer stem cell dynamics," *Mathematical Modelling of Natural Phenomena*, vol. 7, no. 1, pp. 166–202, 2012.

[22] B. Werner, J. G. Scott, A. Sottoriva, A. R. Anderson, A. Traulsen, and P. M. Altrock, "The cancer stem cell fraction in hierarchically organized tumors can be estimated using mathematical modeling and patient-specific treatment trajectories," *Cancer research*, vol. 76, no. 7, pp. 1705–1713, 2016.

[23] T. Stiehl, A. D. Ho, and A. Marciniak-Czochra, "Mathematical modeling of the impact of cytokine response of acute myeloid leukemia cells on patient prognosis," *Scientific reports*, vol. 8, no. 1, p. 2809, 2018.

[24] J.-Y. Jaffray, "Bayesian updating and belief functions," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 5, pp. 1144–1152, 1992.

[25] K. Popper, *The logic of scientific discovery*. Routledge, 2005.

[26] F. Angaroni, A. Graudenzi, M. Rossignolo, D. Maspero, T. Calarco, R. Piazza, S. Montangero, and M. Antoniotti, "An optimal control framework for the automated design of personalized cancer treatments," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 523, 2020.

[27] P. G. Welling, *Pharmacokinetics: processes, mathematics, and applications*. Amer Chemical Society, 1997.

[28] H. Schwilden, "A general method for calculating the dosage scheme in linear pharmacokinetics," *European Journal of Clinical Pharmacology*, vol. 20, no. 5, pp. 379–386, 1981.

[29] M. Lavielle and F. Mentré, "Estimation of population pharmacokinetic parameters of saquinavir in hiv patients with the monolix software," *Journal of pharmacokinetics and pharmacodynamics*, vol. 34, no. 2, pp. 229–249, 2007.

[30] M. Rowland, T. N. Tozer, H. Derendorf, and G. Hochhaus, *Clinical pharmacokinetics and pharmacodynamics: concepts and applications*. Wolters Kluwer Health/Lippincott William & Wilkins Philadelphia, PA, 2011.

[31] S. Goutelle, M. Maurin, F. Rougier, X. Barbaut, L. Bourguignon, M. Ducher, and P. Maire, "The hill equation: a review of its capabilities in pharmacological modelling," *Fundamental & clinical pharmacology*, vol. 22, no. 6, pp. 633–648, 2008.

[32] B. Peng, P. Lloyd, and H. Schran, "Clinical pharmacokinetics of imatinib," *Clinical pharmacokinetics*, vol. 44, no. 9, pp. 879–894, 2005.

[33] A. Marciniak-Czochra and T. Stiehl, "Mathematical models of hematopoietic reconstitution after stem cell transplantation," in *Model Based Parameter Estimation*. Springer, 2013, pp. 191–206.

[34] F. J. Miller, P. M. Schlosser, and D. B. Janszen, "Haber's rule: a special case in a family of curves relating concentration and duration of exposure to a fixed level of response for a given endpoint," *Toxicology*, vol. 149, no. 1, pp. 21–34, 2000.

[35] D. G. Hull, *Optimal control theory for applications*. Springer Science & Business Media, 2013.

[36] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[37] S. M. Lynch, *Introduction to applied Bayesian statistics and estimation for social scientists*. Springer Science & Business Media, 2007.

[38] C. B. Gambacorti-Passerini, R. H. Gunby, R. Piazza, A. Galietta, R. Rostagno, and L. Scapozza, "Molecular mechanisms of resistance to imatinib in philadelphia-chromosome-positive leukaemias," *The lancet oncology*, vol. 4, no. 2, pp. 75–85, 2003.

[39] D. Wodarz, N. Garg, N. L. Komarova, O. Benjamini, M. J. Keating, W. G. Wierda, H. Kantarjian, D. James, S. O'Brien, and J. A. Burger, "Kinetics of cll cells in tissues and blood during therapy with the btk inhibitor ibrutinib," *Blood*, vol. 123, no. 26, pp. 4132–4135, 2014.

[40] M. P. Saccomani, S. Audoly, G. Bellu, and L. D'Angiò, "Examples of testing global identifiability of biological and biomedical models with the daisy software," *Computers in Biology and Medicine*, vol. 40, no. 4, pp. 402–407, 2010.

[41] H. Hong, A. Ovchinnikov, G. Pogudin, and C. Yap, "Sian: software for structural identifiability analysis of ode models," *Bioinformatics*, vol. 35, no. 16, pp. 2873–2874, 2019.

[42] C. Gambacorti-Passerini, P. Le Coutre, L. Mologni, M. Fanelli, C. Bertazzoli, E. Marchesi, M. Di Nicola, A. Biondi, G. M. Corneo, and et al., "Inhibition of the abl kinase activity blocks the proliferation of bcr/abl+ leukemic cells and induces apoptosis," *Blood Cells, Molecules, and Diseases*, vol. 23, no. 3, pp. 380–394, 1997.

[43] M. T. Weigel, L. Dahmke, C. Schem, D. O. Bauerschlag, K. Weber, P. Niehoff, M. Bauer, A. Strauss, W. Jonat, N. Maass, and et al., "In vitro

effects of imatinib mesylate on radiosensitivity and chemosensitivity of breast cancer cells," *BMC cancer*, vol. 10, no. 1, p. 412, 2010.
[44] T. Fojo and C. Grady, "How much is life worth: cetuximab, non–small cell lung cancer, and the 440 billion question," *Journal of the National Cancer Institute*, vol. 101, no. 15, pp. 1044–1048, 2009.
[45] E. in Chronic Myeloid Leukemia and et al., "The price of drugs for chronic myeloid leukemia (cml) is a reflection of the unsustainable prices of cancer drugs: from the perspective of a large group of cml experts," *Blood*, vol. 121, no. 22, pp. 4439–4442, 2013.
[46] A. Gomez-de León, D. Gómez-Almaguer, G. J. Ruiz-Delgado, and G. J. Ruiz-Arguelles, "Insights into the management of chronic myeloid leukemia in resource-poor settings: a mexican perspective," *Expert review of hematology*, vol. 10, no. 9, pp. 809–819, 2017.
[47] D. Ramazzotti, G. Caravagna, L. Olde L, A. Graudenzi, I. Korsunsky, G. Mauri, M. Antoniotti, and B. Mishra, "Capri: efficient inference of cancer progression models from cross-sectional data," *Bioinformatics*, vol. 31, no. 18, pp. 3016–3026, 2015.
[48] G. Caravagna, A. Graudenzi, D. Ramazzotti, R. Sanz-Pamplona, L. De Sano, G. Mauri, V. Moreno, M. Antoniotti, and B. Mishra, "Algorithmic methods to infer the evolutionary trajectories in cancer progression," *Proceedings of the National Academy of Sciences*, vol. 113, no. 28, pp. E4025–E4034, 2016.
[49] A. Graudenzi, D. Maspero, M. Di Filippo, M. Gnugnoli, C. Isella, G. Mauri, E. Medico, M. Antoniotti, and C. Damiani, "Integration of transcriptomic data and metabolic networks in cancer samples reveals highly significant prognostic power," *Journal of biomedical informatics*, vol. 87, pp. 37–49, 2018.
[50] C. Damiani, L. Rovida, D. Maspero, I. Sala, L. Rosato, M. Di Filippo, D. Pescini, A. Graudenzi, M. Antoniotti, and G. Mauri, "Marea4galaxy: Metabolic reaction enrichment analysis and visualization of rna-seq data within galaxy," *Computational and Structural Biotechnology Journal*, vol. 18, p. 993, 2020.
[51] J. West, L. You, J. Zhang, R. A. Gatenby, J. S. Brown, P. K. Newton, and A. R. Anderson, "Towards multi-drug adaptive therapy," *Cancer Research*, 2020.
[52] I. Dagogo-Jack and A. T. Shaw, "Tumour heterogeneity and resistance to cancer therapies," *Nature reviews Clinical oncology*, vol. 15, no. 2, p. 81, 2018.
[53] P. Embrechts, C. Kluppelberg, and T. Mikosch, "Modelling extremal events," *British actuarial journal*, vol. 5, no. 2, pp. 465–465, 1999.
[54] P. Cirillo and N. N. Taleb, "Tail risk of contagious diseases," *Nature Physics*, pp. 1–8, 2020.
[55] D. Fusco, M. Gralka, J. Kayser, A. Anderson, and O. Hallatschek, "Excess of mutational jackpot events in expanding populations revealed by spatial luria–delbrück experiments," *Nature communications*, vol. 7, p. 12760, 2016.
[56] X. Zhang, S. L. Marjani, Z. Hu, S. M. Weissman, X. Pan, and S. Wu, "Single-cell sequencing for precise cancer research: progress and prospects," *Cancer Research*, vol. 76, no. 6, pp. 1305–1312, 2016.
[57] P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss *et al.*, "Visualization and analysis of gene expression in tissue sections by spatial transcriptomics," *Science*, vol. 353, no. 6294, pp. 78–82, 2016.

A.3 EvoTraceR: an R package to analyse Amplicon Sequence Variants from the EvoBC kit

Understanding what principles control the dissemination of primary tumor cells to distant metastatic sites (i.e., seeding patterns) is still an open question [51]. The main challenge that needs to be addressed is to assess whether each metastasis originates from one tumor cell, or there is evidence for polyclonal seeding patterns that suggest cooperation between distant metastatic sites.

To address this question, the Nowak Lab at Weill Cornell Medicine, New York, USA, is working on a CRISPR/Cas9 barcode platform that enables to model metastatic prostate cancer in vivo. It consists of a barcode designed to accumulate edits during tumor evolution, that correspond to either short insertions or deletions, i.e., indels. In fact, the barcode is a nucleotide sequence that can be divided in 10 target sites, that are characterised by different probabilities of being affected by indels.

This original unmutatetd barcode sequence is injected in mouse models, and while the tumor evolves and spreads to metastatic sites, it accumulates mutations due to the combination of the barcode with CRISPR/Cas9 that generates edits (specifically indels). At the moment we cannot provide additional details on the barcode technology, as this work is currently ongoing and will soon be submitted for publication.

At the end of the experiment, Amplicon Sequencing is performed on samples taken from the primary tumor and metastatic sites. Amplicon Sequencing is a specific technology that is able to capture and sequence specific genomic regions, and thus can be employed to measure the barcode sequences (either mutated or not) present in each sample.

Given the specific barcode design, it was necessary to build a computational pipeline that takes the output of a bulk Amplicon Sequencing experiments and performs (i) the detection of the unique mutated sequences present in each sample (defined as Amplicon Sequence Variants (ASVs)), (ii) aligns each mutated sequence to the un-mutated barcode to detect indels and (iii) uses the obtained sequences with the corresponding mutations to build a phylogenetic tree that described the evolutionary history of the samples. The steps are describes below.

FAST files processing The output of the amplicon sequencing experiment consists of two FASTQ files containing paired-end reads. For each mouse the fastq files were demultiplexed according to the sample, which can be either a metastatic site or the primary tumor. The files were processed to extract all ASVs present in each sample with a standard bioinformatics pipeline, that is presented in Figure A.1A. First of all Trimmomatic [35] trims the adapters and removes low quality bases, and Flash [27] merges the paired-end reads. After this preliminary step we obtain merged reads, and we make a preliminary quantification of the identical sequences, by computing their

frequency.

Amplicon Sequence Variants Processing After the preliminary filters described above, we perform multiple steps to take into account possible sequencing errors and identify the final pool of ASVs to employ in the analyses:

- **Hamming distance**: We start by pooling together the reads characterized by Hamming distance equals or lower than 2. To do so, we first group reads with identical length and then we perform clustering and pooling of the counts using the UMIClusterer class from UMI-tools package [75]. This package is designed to pool together UMIs in a single-cell experiment, and it implements multiple methods to perform this task. We employed the network-based method that employs a directed graph, where each node corresponds to an ASV and edges are created in the following way: for each pair of nodes (A, B), there is an edge from A to B if the Hamming distance is equal or lower than a threshold (in our case we set it to 2) and if $|A| \geq (2 * |B| - 1)$, where $|A|$ and $|B|$ refer to the corresponding sequence counts. Then, each connected component in the graph is treated as an ASV group, where the sequence with the highest number of counts is chosen as the representative and counts from every group member are pooled together.

- **Alignment and merging**: After this pooling, we align every sequence to the original non-marked barcode, using the PairwiseAlignment function from the R package Biostrings. For this step we employed the parameters which were tested and selected in [118]. After alignment, we analyze all mutations and we discard indels that happen too far from any target site to have actually been caused by Cas9. In details, each indel that doesn't span any target site and whose start and end are more than 3 bp distant from any cut site is discarded. Given that Cas9 is responsible for deletions and insertions, we want to neglect substitutions, so we pool together the counts of those sequences that are identical in terms of indels, without taking into consideration their substitutions.

- **Flanking filtering** Finally, to remove possible contamination artifacts we leverage a property of the barcode: from its design we know that the last 10 nucleotides (right flanking sequence) should not be mutated by Cas9. Additionally, also the first 5 nucleotides can be used to identify barcode sequences from the pool. However, they lie in proximity to the first target site which is characterized by the highest probability of being cut by Cas9, and we observed that they may also be affected by indels. Thus, for the analysis we exploit the last 10 nucleotides to perform the contamination filtering step, discarding all sequences whose right flanking does not match the non-marked barcode.

In Figure A.1C, we present an example of output of our pipeline. In particular, the input samples correspond to columns and are namely Pimary tumor (Pri.), Metastatic site 01 (Met 01) and Metastatic site 02 (Met. 02). The rows of the same table correspond to the ASVs detected in each sample, and each element in the table corresponds to the number of sequences detected in each sample.

Phylogenetic tree reconstruction Once all ASVs have been identified, also quantifying their presence in each sample (Figure A.1C), we reconstruct the phylogenetic tree representing the accumulation of mutations. To perform this inference, we build a binary mutation matrix, where each row corresponds to an ASV and each column corresponds to a mutation (Figure A.1B). Given the input binary matrix, to reconstruct the phylogeny we use the greedy approach implemented in the suite Cassiopeia [144]. This algorithm proceeds iteratively by splitting sequences in two groups based on the most common mutation present in the current set and keeps recursively applying the same procedure until a set is composed by only one sequence. Finally, we obtain a tree describing the order to accumulation of mutations (Figure A.1D).
This pipeline was implemented in an R package `EvoTraceR`, and is currently being exploited to analyse multiple samples and formulate experimental hypotheses regarding patterns of tumor evolution, that will be discussed in detail in the final manuscript that is currently in preparation.
The code is available at `https://github.com/Nowak-Lab/EvoTraceR_pipeline`.

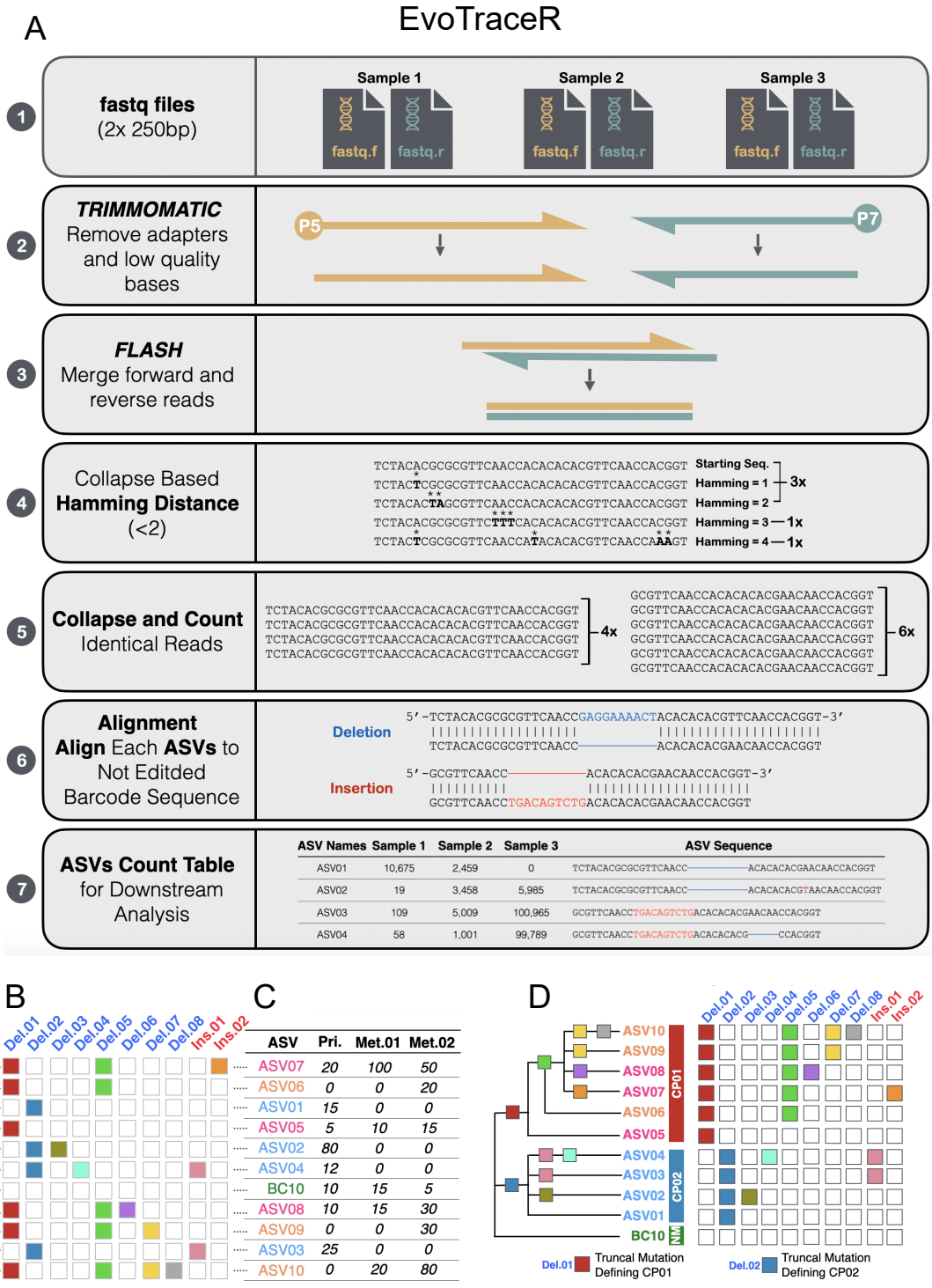


Figure A.1: Representation of the EvoTraceR pipeline. A: Schematic overview of the main steps employed for ASV detection. B: Example of a mutation matrix exploited to reconstruct phylogenetic trees. C: Example of the count table returned by the preprocessing steps of EvoTraceR. D: example of a tree describing the prder of accumulation of mutations in the detected samples.

Bibliography

[1] F. Crick. "Central dogma of molecular biology". In: *Nature* 227.5258 (1970), pp. 561–563 (cit. on p. 19).

[2] P. C. Nowell. "The Clonal Evolution of Tumor Cell Populations: Acquired genetic lability permits stepwise selection of variant sublines and underlies tumor progression." In: *Science* 194.4260 (1976), pp. 23–28 (cit. on p. 4).

[3] P. C. Nowell. "The Clonal Evolution of Tumor Cell Populations". In: *Science* (Oct. 1, 1976). DOI: 10.1126/science.959840 (cit. on p. 52).

[4] G. Schwarz. "Estimating the dimension of a model". In: *The annals of statistics* (1978), pp. 461–464 (cit. on p. 87).

[5] J. Felsenstein. "Alternative Methods of Phylogenetic Inference and Their Interrelationship". In: *Systematic Biology* 28.1 (Mar. 1, 1979), pp. 49–62. DOI: 10.1093/sysbio/28.1.49 (cit. on p. 52).

[6] J. Felsenstein. "Phylogenies from Molecular Sequences: Inference and Reliability". In: *Annual Review of Genetics* 22.1 (1988), pp. 521–565. DOI: 10.1146/annurev.ge.22.120188.002513. pmid: 3071258 (cit. on p. 52).

[7] R. Sager. "Expression genetics in cancer: shifting the focus from DNA to RNA". In: *Proceedings of the National Academy of Sciences* 94.3 (1997), pp. 952–955 (cit. on p. 28).

[8] H. Akaike. "Information theory and an extension of the maximum likelihood principle". In: *Selected papers of hirotugu akaike* (1998), pp. 199–213 (cit. on p. 87).

[9] M. I. Jordan et al. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233 (cit. on p. 86).

[10] C. Biernacki, G. Celeux, and G. Govaert. "Assessing a mixture model for clustering with the integrated completed likelihood". In: *IEEE transactions on pattern analysis and machine intelligence* 22.7 (2000), pp. 719–725 (cit. on p. 87).

[11] T. e. a. Hawkins. "Initial sequencing and analysis of the human genome". In: *nature* 409.6822 (2001), pp. 860–921 (cit. on p. 19).

[12] G. Orphanides and D. Reinberg. "A unified theory of gene expression". In: *Cell* 108.4 (2002), pp. 439–451 (cit. on pp. 19, 21).

[13] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning.* Vol. 4. Springer, 2006 (cit. on p. 86).

[14] V. D. Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008 (cit. on pp. 25, 102).

[15] W. Filipowicz, S. N. Bhattacharyya, and N. Sonenberg. "Mechanisms of post-transcriptional regulation by microRNAs: are the answers in sight?" In: *Nature reviews genetics* 9.2 (2008), pp. 102–114 (cit. on p. 80).

[16] S. Richards et al. "Regulation of B-cell entry into the cell cycle". In: *Immunological reviews* 224.1 (2008), pp. 183–200 (cit. on p. 94).

[17] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008) (cit. on p. 102).

[18] N. Altman. "Batches and blocks, sample pools and subsamples in the design and analysis of gene expression studies". In: *Batch effects and noise in microarray experiments: sources and solutions* (2009), pp. 33–50 (cit. on p. 4).

[19] M. Gerstung et al. "Quantifying Cancer Progression with Conjunctive Bayesian Networks". In: *Bioinformatics* 25.21 (Nov. 1, 2009), pp. 2809–2815. DOI: 10.1093/bioinformatics/btp505 (cit. on p. 53).

[20] P. J. Park. "ChIP-seq: Advantages and Challenges of a Maturing Technology". In: *Nature Reviews. Genetics* 10.10 (Oct. 2009), pp. 669–680. DOI: 10.1038/nrg2641. pmid: 19736561 (cit. on p. 17).

[21] F. Tang et al. "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature methods* 6.5 (2009), pp. 377–382 (cit. on p. 22).

[22] Z. Wang, M. Gerstein, and M. Snyder. "RNA-Seq: A Revolutionary Tool for Transcriptomics". In: *Nature reviews. Genetics* 10.1 (Jan. 2009), pp. 57–63. DOI: 10.1038/nrg2484. pmid: 19015660 (cit. on p. 18).

[23] Z. Wang, M. Gerstein, and M. Snyder. "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature reviews genetics* 10.1 (2009), pp. 57–63 (cit. on p. 27).

[24] J. M. Mullaney et al. "Small insertions and deletions (INDELs) in human genomes". In: *Human molecular genetics* 19.R2 (2010), R131–R136 (cit. on p. 20).

[25] P. Van Loo et al. "Allele-specific copy number analysis of tumors". In: *Proceedings of the National Academy of Sciences* 107.39 (2010), pp. 16910–16915 (cit. on p. 23).

[26] S. Islam et al. “Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq”. In: *Genome research* 21.7 (2011), pp. 1160–1167 (cit. on pp. 18, 22).

[27] T. Magoč and S. L. Salzberg. “FLASH: fast length adjustment of short reads to improve genome assemblies”. In: *Bioinformatics* 27.21 (2011), pp. 2957–2963 (cit. on p. 147).

[28] M. J. Garnett et al. “Systematic identification of genomic markers of drug sensitivity in cancer cells”. In: *Nature* 483.7391 (2012), pp. 570–575 (cit. on p. 91).

[29] T. Kivioja et al. “Counting absolute numbers of molecules using unique molecular identifiers”. In: *Nature methods* 9.1 (2012), pp. 72–74 (cit. on p. 29).

[30] S. Behjati and P. S. Tarpey. “What is next generation sequencing?” In: *Archives of Disease in Childhood-Education and Practice* 98.6 (2013), pp. 236–238 (cit. on p. 17).

[31] R. Ranganath, S. Gerrish, and D. M. Blei. “Black box variational inference”. In: *arXiv preprint arXiv:1401.0118* (2013) (cit. on pp. 84, 86).

[32] A. Sottoriva et al. “Intratumor heterogeneity in human glioblastoma reflects cancer evolutionary dynamics”. In: *Proceedings of the National Academy of Sciences* 110.10 (2013), pp. 4009–4014 (cit. on p. 4).

[33] T. I. Zack et al. “Pan-cancer patterns of somatic copy number alteration”. In: *Nature genetics* 45.10 (2013), pp. 1134–1140 (cit. on p. 20).

[34] T. Baslan and J. Hicks. “Single cell sequencing approaches for complex biological systems”. In: *Current opinion in genetics & development* 26 (2014), pp. 59–65 (cit. on p. 4).

[35] A. M. Bolger, M. Lohse, and B. Usadel. “Trimmomatic: a flexible trimmer for Illumina sequence data”. In: *Bioinformatics* 30.15 (2014), pp. 2114–2120 (cit. on p. 147).

[36] J. Foo and F. Michor. “Evolution of Acquired Resistance to Anti-Cancer Therapy”. In: *Journal of Theoretical Biology* 355 (Aug. 21, 2014), pp. 10–20. DOI: 10.1016/j.jtbi.2014.02.025 (cit. on p. 53).

[37] D. A. Jaitin et al. “Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types”. In: *Science* 343.6172 (2014), pp. 776–779 (cit. on p. 28).

[38] A. P. Patel et al. “Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma”. In: *Science* 344.6190 (2014), pp. 1396–1401 (cit. on pp. 26, 70, 79, 80).

[39] A. P. Patel et al. “Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma”. In: *Science* 344.6190 (2014), pp. 1396–1401. DOI: 10.1126/science.1254257. eprint: https://www.science.org/doi/pdf/10.1126/science.1254257. URL: https://www.science.org/doi/abs/10.1126/science.1254257 (cit. on p. 28).

[40] S. Picelli et al. “Full-length RNA-seq from single cells using Smart-seq2”. In: *Nature protocols* 9.1 (2014), pp. 171–181 (cit. on p. 28).

[41] A. Roth et al. “PyClone: Statistical Inference of Clonal Population Structure in Cancer”. In: *Nature Methods* 11.4 (4 Apr. 2014), pp. 396–398. DOI: 10.1038/nmeth.2883 (cit. on p. 22).

[42] R. F. Schwarz et al. “Phylogenetic quantification of intra-tumour heterogeneity”. In: *PLoS computational biology* 10.4 (2014), e1003535 (cit. on p. 53).

[43] E. L. Van Dijk et al. “Ten years of next-generation sequencing technology”. In: *Trends in genetics* 30.9 (2014), pp. 418–426 (cit. on p. 17).

[44] H. Zare et al. “Inferring clonal composition from multiple sections of a breast cancer”. In: *PLoS computational biology* 10.7 (2014), e1003703 (cit. on p. 53).

[45] J. D. Buenrostro et al. “ATAC-seq: a method for assaying chromatin accessibility genome-wide”. In: *Current protocols in molecular biology* 109.1 (2015), pp. 21–29 (cit. on pp. 4, 18, 21, 24).

[46] D. A. Cusanovich et al. “Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing”. In: *Science* 348.6237 (2015), pp. 910–914 (cit. on pp. 4, 21).

[47] G. Escaramís, E. Docampo, and R. Rabionet. “A decade of structural variants: description, history and methods to detect structural variation”. In: *Briefings in Functional Genomics* 14.5 (Apr. 2015), pp. 305–314. ISSN: 2041-2649. DOI: 10.1093/bfgp/elv014. eprint: https://academic.oup.com/bfg/article-pdf/14/5/305/5037790/elv014.pdf. URL: https://doi.org/10.1093/bfgp/elv014 (cit. on p. 20).

[48] F. Favero et al. “Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data”. In: *Annals of Oncology* 26.1 (2015), pp. 64–70 (cit. on pp. 23, 81).

[49] G. Finak et al. “MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data”. In: *Genome biology* 16.1 (2015), pp. 1–13 (cit. on p. 6).

[50] J. Guinney et al. “The consensus molecular subtypes of colorectal cancer”. In: *Nature medicine* 21.11 (2015), pp. 1350–1356 (cit. on p. 102).

[51] G. Gundem et al. “The evolutionary history of lethal metastatic prostate cancer”. In: *Nature* 520.7547 (2015), pp. 353–357 (cit. on p. 147).

[52] L. Haghverdi, F. Buettner, and F. J. Theis. “Diffusion maps for high-dimensional single-cell analysis of differentiation data”. In: *Bioinformatics* 31.18 (2015), pp. 2989–2998 (cit. on p. 6).

[53] M. El-Kebir et al. “Reconstruction of Clonal Trees and Tumor Composition from Multi-Sample Sequencing Data”. In: *Bioinformatics* 31.12 (June 15, 2015), pp. i62–i70. DOI: 10.1093/bioinformatics/btv261 (cit. on pp. 53, 54, 105).

[54] I. C. Macaulay et al. “G&T-seq: parallel sequencing of single-cell genomes and transcriptomes”. In: *Nature methods* 12.6 (2015), pp. 519–522 (cit. on p. 25).

[55] E. Z. Macosko et al. “Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets”. In: *Cell* 161.5 (2015), pp. 1202–1214 (cit. on p. 28).

[56] N. McGranahan and C. Swanton. “Biological and therapeutic impact of intratumor heterogeneity in cancer evolution”. In: *Cancer cell* 27.1 (2015), pp. 15–26 (cit. on p. 20).

[57] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. “Variational inference: a review for statisticians. arXiv”. In: *arXiv preprint arXiv:1601.00670* (2016) (cit. on p. 84).

[58] W. F. Doolittle and T. D. P. Brunet. “What Is the Tree of Life?” In: *PLOS Genetics* 12.4 (Apr. 14, 2016), e1005912. DOI: 10.1371/journal.pgen.1005912 (cit. on p. 52).

[59] C. Gawad, W. Koh, and S. R. Quake. “Single-cell genome sequencing: current state of the science”. In: *Nature Reviews Genetics* 17.3 (2016), pp. 175–188 (cit. on pp. 21, 22).

[60] T. Hashimshony et al. “CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq”. In: *Genome biology* 17.1 (2016), pp. 1–7 (cit. on p. 29).

[61] C. G. Hubert et al. “A three-dimensional organoid culture system derived from human glioblastomas recapitulates the hypoxic gradients and cancer stem cell heterogeneity of tumors found in vivobrain cancer stem cell organoids”. In: *Cancer research* 76.8 (2016), pp. 2465–2477 (cit. on p. 4).

[62] L. A. Hug et al. “A New View of the Tree of Life”. In: *Nature Microbiology* 1.5 (5 Apr. 11, 2016), pp. 1–6. DOI: 10.1038/nmicrobiol.2016.48 (cit. on p. 52).

[63] K. Jahn, J. Kuipers, and N. Beerenwinkel. “Tree Inference for Single-Cell Data”. In: *Genome Biology* 17.1 (May 5, 2016), p. 86. DOI: 10.1186/s13059-016-0936-x (cit. on pp. 23, 54).

[64] E. Jang, S. Gu, and B. Poole. "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144* (2016) (cit. on pp. 86, 119).

[65] M. Sajitz-Hermstein et al. "iReMet-Flux: Constraint-Based Approach for Integrating Relative Metabolite Levels into a Stoichiometric Metabolic Models". In: *Bioinformatics (Oxford, England)* 32.17 (2016), pp. i755–i762 (cit. on p. 123).

[66] E. Talevich et al. "CNVkit: genome-wide copy number detection and visualization from targeted DNA sequencing". In: *PLoS computational biology* 12.4 (2016), e1004873 (cit. on p. 81).

[67] M. D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific data* 3.1 (2016), pp. 1–9 (cit. on p. 121).

[68] G. C. L. analysts: Aguet François 1 Brown Andrew A. 2 3 4 Castel Stephane E. 5 6 Davis Joe R. 7 8 He Yuan 9 Jo Brian 10 Mohammadi Pejman 5 6 Park YoSon 11 Parsana Princy 12 Segrè Ayellet V. 1 Strober Benjamin J. 9 Zappala Zachary 7 8 et al. "Genetic effects on gene expression across human tissues". In: *Nature* 550.7675 (2017), pp. 204–213 (cit. on p. 22).

[69] C. Damiani et al. "Linking alterations in metabolic fluxes with shifts in metabolite levels by means of kinetic modeling". In: *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry: 11th Italian Workshop, WIVACE 2016, Fisciano, Italy, October 4-6, 2016, Revised Selected Papers 11*. Springer. 2017, pp. 138–148 (cit. on p. 123).

[70] A. Davis, R. Gao, and N. Navin. "Tumor Evolution: Linear, Branching, Neutral or Punctuated?" In: *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*. Evolutionary Principles - Heterogeneity in Cancer? 1867.2 (Apr. 1, 2017), pp. 151–161. DOI: 10.1016/j.bbcan.2017.01.003 (cit. on p. 53).

[71] M. El-Kebir et al. "Complexity and algorithms for copy-number evolution problems". In: *Algorithms for Molecular Biology* 12.1 (2017), pp. 1–11 (cit. on p. 53).

[72] A. Regev et al. "Science forum: the human cell atlas". In: *elife* 6 (2017), e27041 (cit. on p. 28).

[73] A. N. Schep et al. "chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data". In: *Nature methods* 14.10 (2017), pp. 975–978 (cit. on p. 21).

[74] V. A. Schneider et al. "Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly". In: *Genome research* 27.5 (2017), pp. 849–864 (cit. on p. 19).

[75] T. Smith, A. Heger, and I. Sudbery. "UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy". In: *Genome research* 27.3 (2017), pp. 491–499 (cit. on p. 148).

[76] A. S. Venteicher et al. “Decoupling genetics, lineages, and microenvironment in IDH-mutant gliomas by single-cell RNA-seq”. In: *Science* 355.6332 (2017), eaai8478 (cit. on p. 28).

[77] J. D. Welch, A. J. Hartemink, and J. F. Prins. “MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics”. In: *Genome biology* 18.1 (2017), pp. 1–19 (cit. on p. 26).

[78] H. Zahn et al. “Scalable whole-genome single-cell library preparation without preamplification”. In: *Nature methods* 14.2 (2017), pp. 167–173 (cit. on pp. 4, 21, 23).

[79] L. Zappia, B. Phipson, and A. Oshlack. “Splatter: simulation of single-cell RNA sequencing data”. In: *Genome biology* 18.1 (2017), p. 174 (cit. on p. 31).

[80] G. X. Zheng et al. “Massively parallel digital transcriptional profiling of single cells”. In: *Nature communications* 8.1 (2017), pp. 1–12 (cit. on pp. 4, 22, 28).

[81] C. Ziegenhain et al. “Comparative analysis of single-cell RNA sequencing methods”. In: *Molecular cell* 65.4 (2017), pp. 631–643 (cit. on p. 29).

[82] A. A. AlJanahi, M. Danielsen, and C. E. Dunbar. “An introduction to the analysis of single-cell RNA-sequencing data”. In: *Molecular Therapy-Methods & Clinical Development* 10 (2018), pp. 189–196 (cit. on p. 100).

[83] T. S. Andrews and M. Hemberg. “False signals induced by single-cell imputation”. In: *F1000Research* 7 (2018) (cit. on pp. 30, 31).

[84] R. Argelaguet et al. “Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets”. In: *Molecular systems biology* 14.6 (2018), e8124 (cit. on pp. 69, 79).

[85] E. Azizi et al. “Single-cell map of diverse immune phenotypes in the breast tumor microenvironment”. In: *Cell* 174.5 (2018), pp. 1293–1308 (cit. on p. 28).

[86] C. G. de Boer and A. Regev. “BROCKMAN: deciphering variance in epigenomic regulators by k-mer factorization”. In: *BMC bioinformatics* 19.1 (2018), pp. 1–13 (cit. on p. 24).

[87] G. Caravagna et al. “Detecting repeated cancer evolution from multi-region tumor sequencing data”. In: *Nature methods* 15.9 (2018), pp. 707–714 (cit. on pp. 3, 119, 121).

[88] D. A. Cusanovich et al. “A Single-Cell Atlas of In Vivo Mammalian Chromatin Accessibility”. In: *Cell* 174.5 (2018), 1309–1324.e18. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2018.06.052. URL: https://www.sciencedirect.com/science/article/pii/S0092867418308559 (cit. on pp. 19, 24).

[89] J. Fan et al. “Linking transcriptional and genetic tumor heterogeneity through allele analysis of single-cell RNA-seq data”. In: *Genome research* 28.8 (2018), pp. 1217–1227 (cit. on pp. 26, 70, 79, 80).

[90] S. Freytag et al. “Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data [version 2; peer review: 3 approved]”. In: *F1000Research* 7 (2018) (cit. on p. 25).

[91] M. Huang et al. “SAVER: gene expression recovery for single-cell RNA sequencing”. In: *Nature methods* 15.7 (2018), pp. 539–542 (cit. on p. 31).

[92] B. Hwang, J. H. Lee, and D. Bang. “Single-cell RNA sequencing technologies and bioinformatics pipelines”. In: *Experimental & molecular medicine* 50.8 (2018), pp. 1–14 (cit. on p. 29).

[93] C. M. Koch et al. “A beginner’s guide to analysis of RNA sequencing data”. In: *American journal of respiratory cell and molecular biology* 59.2 (2018), pp. 145–157 (cit. on pp. 22, 27).

[94] G. La Manno et al. “RNA Velocity of Single Cells”. In: *Nature* 560.7719 (7719 Aug. 2018), pp. 494–498. DOI: 10.1038/s41586-018-0414-6 (cit. on p. 25).

[95] R. Lopez et al. “Deep generative modeling for single-cell transcriptomics”. In: *Nature methods* 15.12 (2018), pp. 1053–1058 (cit. on p. 32).

[96] L. McInnes, J. Healy, and J. Melville. “Umap: Uniform manifold approximation and projection for dimension reduction”. In: *arXiv preprint arXiv:1802.03426* (2018) (cit. on pp. 94, 102).

[97] Z. Miao et al. “DEsingle for detecting three types of differential expression in single-cell RNA-seq data”. In: *Bioinformatics* 34.18 (2018), pp. 3223–3224 (cit. on p. 6).

[98] E. Papalexi and R. Satija. “Single-cell RNA sequencing to explore immune cell heterogeneity”. In: *Nature Reviews Immunology* 18.1 (2018), pp. 35–45 (cit. on p. 28).

[99] B. E. Slatko, A. F. Gardner, and F. M. Ausubel. “Overview of next-generation sequencing technologies”. In: *Current protocols in molecular biology* 122.1 (2018), e59 (cit. on p. 17).

[100] C. Soneson and M. D. Robinson. “Bias, robustness and scalability in single-cell differential expression analysis”. In: *Nature methods* 15.4 (2018), pp. 255–261 (cit. on p. 25).

[101] L. Valihrach, P. Androvic, and M. Kubista. “Platforms for single-cell collection and analysis”. In: *International Journal of Molecular Sciences* 19.3 (2018), p. 807 (cit. on p. 29).

[102] D. Van Dijk et al. "Recovering gene interactions from single-cell data using data diffusion". In: *Cell* 174.3 (2018), pp. 716–729 (cit. on p. 31).

[103] G. Vlachogiannis et al. "Patient-derived organoids model treatment response of metastatic gastrointestinal cancers". In: *Science* 359.6378 (2018), pp. 920–926 (cit. on p. 4).

[104] F. A. Wolf, P. Angerer, and F. J. Theis. "SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis". In: *Genome Biology* 19.1 (Feb. 6, 2018), p. 15. DOI: 10.1186/s13059-017-1382-0 (cit. on pp. 24, 102, 121).

[105] H. Yang et al. *Patient-derived organoids: A promising model for personalized cancer treatment*. 2018 (cit. on p. 4).

[106] S. Zaccaria et al. "Phylogenetic copy-number factorization of multiple tumor samples". In: *Journal of Computational Biology* 25.7 (2018), pp. 689–708 (cit. on p. 53).

[107] L. Zhang and S. Zhang. "Comparison of computational methods for imputing single-cell RNA-sequencing data". In: *IEEE/ACM transactions on computational biology and bioinformatics* 17.2 (2018), pp. 376–389 (cit. on pp. 30, 31).

[108] D. Aran et al. "Reference-Based Analysis of Lung Single-Cell Sequencing Reveals a Transitional Profibrotic Macrophage". In: *Nature Immunology* 20.2 (2 Feb. 2019), pp. 163–172. DOI: 10.1038/s41590-018-0276-y (cit. on p. 25).

[109] G. Baruzzo, I. Patuzzi, and B. Di Camillo. "SPARSim single cell: a count data simulator for scRNA-seq data". In: *Bioinformatics* 36.5 (Oct. 2019), pp. 1468–1475. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz752. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/5/1468/32793843/btz752_supplementary_data.pdf. URL: https://doi.org/10.1093/bioinformatics/btz752 (cit. on p. 88).

[110] E. Bingham et al. "Pyro: Deep Universal Probabilistic Programming". In: *J. Mach. Learn. Res.* 20 (2019), 28:1–28:6. URL: http://jmlr.org/papers/v20/18-403.html (cit. on pp. 80, 86, 87).

[111] K. R. Campbell et al. "clonealign: statistical integration of independent single-cell RNA and DNA sequencing data from human cancers". In: *Genome biology* 20.1 (2019), pp. 1–12 (cit. on pp. 15, 26, 70, 79, 81, 105, 114).

[112] H. Chen et al. "Assessment of computational methods for the analysis of single-cell ATAC-seq data". In: *Genome biology* 20.1 (2019), pp. 1–25 (cit. on pp. 19, 24).

[113] R. Diaz-Uriarte and C. Vasallo. "Every Which Way? On Predicting Tumor Evolution Using Cancer Progression Models". In: *PLOS Computational Biology* 15.8 (Aug. 2, 2019), e1007246. DOI: 10.1371/journal.pcbi.1007246 (cit. on p. 53).

[114] S.-R. Hosseini et al. “Estimating the Predictability of Cancer Evolution”. In: *Bioinformatics* 35.14 (July 15, 2019), pp. i389–i397. DOI: 10.1093/bioinformatics/btz332 (cit. on p. 53).

[115] V. Y. Kiselev, T. S. Andrews, and M. Hemberg. “Challenges in unsupervised clustering of single-cell RNA-seq data”. In: *Nature Reviews Genetics* 20.5 (2019), pp. 273–282 (cit. on p. 25).

[116] I. Korsunsky et al. “Fast, sensitive and accurate integration of single-cell data with Harmony”. In: *Nature methods* 16.12 (2019), pp. 1289–1296 (cit. on p. 102).

[117] A. Kulkarni et al. “Beyond bulk: a review of single cell transcriptomics methodologies and applications”. In: *Current opinion in biotechnology* 58 (2019), pp. 129–136 (cit. on p. 4).

[118] K. Labun et al. “Accurate analysis of genuine CRISPR editing events with ampliCan”. In: *Genome research* 29.5 (2019), pp. 843–847 (cit. on p. 148).

[119] E. Laks et al. “Clonal decomposition and DNA replication states defined by scaled single-cell genome sequencing”. In: *Cell* 179.5 (2019), pp. 1207–1221 (cit. on pp. 21, 23, 100).

[120] M. D. Luecken and F. J. Theis. “Current Best Practices in Single-Cell RNA-seq Analysis: A Tutorial”. In: *Molecular Systems Biology* 15.6 (2019), e8746. DOI: 10.15252/msb.20188746. eprint: https://www.embopress.org/doi/pdf/10.15252/msb.20188746 (cit. on pp. 24, 32, 100, 107).

[121] T. Ma and A. Zhang. “Integrate multi-omics data with biological interaction networks using Multi-view Factorization AutoEncoder (MAE)”. In: *BMC genomics* 20 (2019), pp. 1–11 (cit. on pp. 69, 79).

[122] A. S. Nam et al. “Somatic mutations and cell identity linked by Genotyping of Transcriptomes”. In: *Nature* 571.7765 (2019), pp. 355–360 (cit. on p. 79).

[123] D. Ramazzotti et al. “Learning Mutational Graphs of Individual Tumour Evolution from Single-Cell and Multi-Region Sequencing Data”. In: *BMC Bioinformatics* 20.1 (Apr. 25, 2019), p. 210. DOI: 10.1186/s12859-019-2795-4 (cit. on p. 23).

[124] A. T. Satpathy et al. “Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion”. In: *Nature biotechnology* 37.8 (2019), pp. 925–936 (cit. on pp. 88, 90, 92).

[125] M. Setty et al. “Characterization of Cell Fate Probabilities in Single-Cell Data with Palantir”. In: *Nature Biotechnology* 37.4 (4 Apr. 2019), pp. 451–460. DOI: 10.1038/s41587-019-0068-4 (cit. on p. 25).

[126] T. Stuart et al. "Comprehensive integration of single-cell data". In: *Cell* 177.7 (2019), pp. 1888–1902 (cit. on p. 102).

[127] V. A. Traag, L. Waltman, and N. J. Van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific reports* 9.1 (2019), pp. 1–12 (cit. on pp. 24, 25, 102).

[128] F. Wagner, D. Barkley, and I. Yanai. "Accurate denoising of single-cell RNA-Seq data using unbiased principal component analysis". In: *BioRxiv* (2019), p. 655365 (cit. on p. 31).

[129] J. Wang et al. "Data denoising with transfer learning in single-cell transcriptomics". In: *Nature methods* 16.9 (2019), pp. 875–878 (cit. on pp. 31, 32).

[130] Y. Wang et al. "iTALK: an R package to characterize and illustrate intercellular communication". In: *BioRxiv* (2019), p. 507871 (cit. on p. 6).

[131] D. Weighill et al. "Data integration in poplar:'omics layers and integration strategies". In: *Frontiers in Genetics* 10 (2019), p. 874 (cit. on p. 4).

[132] J. D. Welch et al. "Single-cell multi-omic integration compares and contrasts features of brain cell identity". In: *Cell* 177.7 (2019), pp. 1873–1887 (cit. on p. 102).

[133] K. E. Yost et al. "Clonal replacement of tumor-specific T cells following PD-1 blockade". In: *Nature medicine* 25.8 (2019), pp. 1251–1259 (cit. on pp. 88, 90, 92).

[134] X. Zhang et al. "Comparative analysis of droplet-based ultra-high-throughput single-cell RNA-seq systems". In: *Molecular cell* 73.1 (2019), pp. 130–142 (cit. on p. 28).

[135] A. Acar et al. "Exploiting evolutionary steering to induce collateral drug sensitivity in cancer". In: *Nature communications* 11.1 (2020), p. 1923 (cit. on p. 79).

[136] R. Argelaguet et al. "MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data". In: *Genome biology* 21.1 (2020), pp. 1–17 (cit. on pp. 69, 79).

[137] V. Bergen et al. "Generalizing RNA Velocity to Transient Cell States through Dynamical Modeling". In: *Nature Biotechnology* 38.12 (12 Dec. 2020), pp. 1408–1414. DOI: 10.1038/s41587-020-0591-3 (cit. on p. 25).

[138] G. Caravagna et al. "Subclonal Reconstruction of Tumors by Using Machine Learning and Population Genetics". In: *Nature Genetics* 52.9 (9 Sept. 2020), pp. 898–907. DOI: 10.1038/s41588-020-0675-5 (cit. on pp. 22, 52, 53).

[139] H. Chen et al. “Comprehensive assessment of computational algorithms in predicting cancer driver mutations”. In: *Genome biology* 21.1 (2020), pp. 1–17 (cit. on p. 6).

[140] S. Christensen et al. “Detecting evolutionary patterns of cancers using consensus trees”. In: *Bioinformatics* 36.Supplement_2 (2020), pp. i684–i691 (cit. on p. 121).

[141] M. Efremova et al. “CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes”. In: *Nature protocols* 15.4 (2020), pp. 1484–1506 (cit. on p. 6).

[142] W. Hou et al. “A systematic evaluation of single-cell RNA-sequencing imputation methods”. In: *Genome biology* 21 (2020), pp. 1–30 (cit. on pp. 30, 31).

[143] Z. Hu et al. “Multi-Cancer Analysis of Clonality and the Timing of Systemic Spread in Paired Primary Tumors and Metastases”. In: *Nature Genetics* 52.7 (7 July 2020), pp. 701–708. DOI: 10.1038/s41588-020-0628-z (cit. on p. 53).

[144] M. G. Jones et al. “Inference of single-cell phylogenies from lineage tracing data using Cassiopeia”. In: *Genome biology* 21.1 (2020), pp. 1–27 (cit. on p. 149).

[145] D. Lähnemann et al. “Eleven Grand Challenges in Single-Cell Data Science”. In: *Genome Biology* 21.1 (Feb. 7, 2020), p. 31. DOI: 10.1186/s13059-020-1926-6 (cit. on p. 25).

[146] H.-O. Lee et al. “Lineage-dependent gene expression programs influence the immune landscape of colorectal cancer”. In: *Nature Genetics* 52.6 (2020), pp. 594–603 (cit. on p. 103).

[147] B. Lim, Y. Lin, and N. Navin. “Advancing cancer research and medicine with single-cell genomics”. In: *Cancer cell* 37.4 (2020), pp. 456–470 (cit. on p. 79).

[148] X. F. Mallory et al. “Methods for copy number aberration detection from single-cell DNA-sequencing data”. In: *Genome biology* 21.1 (2020), pp. 1–22 (cit. on p. 24).

[149] L. Patruno et al. “A review of computational strategies for denoising and imputation of single-cell transcriptomic data”. In: *Briefings in Bioinformatics* 22.4 (Oct. 2020). bbaa222. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa222. eprint: https://academic.oup.com/bib/article-pdf/22/4/bbaa222/39136487/bbaa222.pdf. URL: https://doi.org/10.1093/bib/bbaa222 (cit. on pp. 32, 118).

[150] A. Serin Harmanci, A. O. Harmanci, and X. Zhou. “CaSpER identifies and visualizes CNV events by integrative analysis of single-cell or bulk RNA-sequencing data”. In: *Nature communications* 11.1 (2020), pp. 1–16 (cit. on pp. 26, 70, 79).

[151] J. D. Silverman et al. "Naught all zeros in sequence count data are the same". In: *Computational and structural biotechnology journal* 18 (2020), pp. 2789–2798 (cit. on pp. 31, 32).

[152] V. Svensson. "Droplet scRNA-seq is not zero-inflated". In: *Nature Biotechnology* 38.2 (2020), pp. 147–150 (cit. on pp. 31, 32).

[153] H. T. N. Tran et al. "A benchmark of batch-effect correction methods for single-cell RNA sequencing data". In: *Genome biology* 21 (2020), pp. 1–32 (cit. on p. 102).

[154] G. A. Van der Auwera and B. D. O'Connor. *Genomics in the cloud: using Docker, GATK, and WDL in Terra*. O'Reilly Media, 2020 (cit. on p. 81).

[155] C. Wang et al. "Integrative analyses of single-cell transcriptome and regulome using MAESTRO". In: *Genome biology* 21.1 (2020), pp. 1–28 (cit. on pp. 6, 26).

[156] F. Yan et al. "From reads to insight: a hitchhiker's guide to ATAC-seq data analysis". In: *Genome biology* 21.1 (2020), pp. 1–16 (cit. on pp. 21, 101).

[157] S. Abramov et al. "Landscape of allele-specific transcription factor binding in the human genome". In: *Nature communications* 12.1 (2021), p. 2751 (cit. on p. 80).

[158] F. Angaroni et al. "PMCE: Efficient Inference of Expressive Models of Cancer Evolution with High Prognostic Power". In: *Bioinformatics* (Oct. 14, 2021), btab717. DOI: 10.1093/bioinformatics/btab717 (cit. on p. 53).

[159] R. Argelaguet et al. "Computational principles and challenges in single-cell data integration". In: *Nature biotechnology* 39.10 (2021), pp. 1202–1215 (cit. on pp. 6, 9, 25, 79).

[160] K. Belhocine, L. DeMare, and O. Habern. "Single-Cell Multiomics: Simultaneous Epigenetic and Transcriptional Profiling: 10x Genomics shares experimental planning and sample preparation tips for the Chromium Single Cell Multiome ATAC+ Gene Expression system". In: *Genetic Engineering & Biotechnology News* 41.1 (2021), pp. 66–68 (cit. on pp. 19, 25, 79, 94).

[161] S. Ciccolella et al. "Inferring Cancer Progression from Single-Cell Sequencing While Allowing Mutation Losses". In: *Bioinformatics* 37.3 (Feb. 1, 2021), pp. 326–333. DOI: 10.1093/bioinformatics/btaa722 (cit. on p. 23).

[162] J. Diaz-Colunga and R. Diaz-Uriarte. "Conditional Prediction of Consecutive Tumor Evolution Using Cancer Progression Models: What Genotype Comes Next?" In: *PLOS Computational Biology* 17.12 (Dec. 21, 2021), e1009055. DOI: 10.1371/journal.pcbi.1009055 (cit. on p. 53).

[163] X. Fan et al. "SMOOTH-seq: single-cell genome sequencing of human cells on a third-generation sequencing platform". In: *Genome biology* 22.1 (2021), pp. 1–19 (cit. on p. 21).

[164] R. Fang et al. "Comprehensive analysis of single cell ATAC-seq data with SnapATAC". In: *Nature communications* 12.1 (2021), pp. 1–15 (cit. on pp. 21, 24).

[165] R. Gao et al. "Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes". In: *Nature biotechnology* 39.5 (2021), pp. 599–608 (cit. on pp. 26, 70, 79, 80, 89).

[166] P. Guilhamon et al. "Single-cell chromatin accessibility profiling of glioblastoma identifies an invasive cancer stem cell population associated with lower survival". In: *Elife* 10 (2021), e64090 (cit. on p. 80).

[167] Y. Hao et al. "Integrated Analysis of Multimodal Single-Cell Data". In: *Cell* 184.13 (June 24, 2021), 3573–3587.e29. DOI: 10.1016/j.cell.2021.04.048. pmid: 34062119 (cit. on pp. 24, 94, 102, 121).

[168] L. Harbers et al. "Somatic copy number alterations in human cancers: An analysis of publicly available data from the cancer genome atlas". In: *Frontiers in oncology* (2021), p. 2877 (cit. on p. 20).

[169] T. Heide et al. "The co-evolution of the genome and epigenome in colorectal cancer". In: *bioRxiv* (2021) (cit. on p. 4).

[170] S. Jin et al. "Inference and analysis of cell-cell communication using CellChat". In: *Nature communications* 12.1 (2021), p. 1088 (cit. on p. 6).

[171] B. Kaminow, D. Yunusov, and A. Dobin. "STARsolo: accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data". In: *Biorxiv* (2021) (cit. on p. 99).

[172] Z. Navidi, L. Zhang, and B. Wang. "simATAC: a single-cell ATAC-seq simulation framework". In: *Genome biology* 22.1 (2021), pp. 1–16 (cit. on p. 88).

[173] P. Nieto et al. "A Single-Cell Tumor Immune Atlas for Precision Oncology". In: *Genome Research* (Sept. 21, 2021). DOI: 10.1101/gr.273300.120. pmid: 34548323 (cit. on p. 25).

[174] A. Nikolic et al. "Copy-scAT: Deconvoluting single-cell chromatin accessibility of genetic subclones in cancer". In: *Science advances* 7.42 (2021), eabg6045 (cit. on pp. 26, 80).

[175] N. M. Novikov et al. "Mutational Drivers of Cancer Cell Migration and Invasion". In: *British Journal of Cancer* 124.1 (1 Jan. 2021), pp. 102–114. DOI: 10.1038/s41416-020-01149-0 (cit. on p. 53).

[176] S. K. Rehman et al. "Colorectal cancer cells enter a diapause-like DTP state to survive chemotherapy". In: *Cell* 184.1 (2021), pp. 226–242 (cit. on p. 104).

[177] C. D. Robles-Espinoza et al. "Allele-specific expression: Applications in cancer and technical considerations". In: *Current opinion in genetics & development* 66 (2021), pp. 10–19 (cit. on p. 80).

[178] S. Salehi et al. "Clonal Fitness Inferred from Time-Series Modelling of Single-Cell Cancer Genomes". In: *Nature* 595.7868 (7868 July 2021), pp. 585–590. DOI: 10.1038/s41586-021-03648-3 (cit. on p. 53).

[179] A. Sarkar and M. Stephens. "Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis". In: *Nature genetics* 53.6 (2021), pp. 770–777 (cit. on p. 88).

[180] J. W. Squair et al. "Confronting false discoveries in single-cell differential expression". In: *Nature communications* 12.1 (2021), pp. 1–15 (cit. on p. 25).

[181] T. Stuart et al. "Single-cell chromatin state analysis with Signac". In: *Nature methods* 18.11 (2021), pp. 1333–1341 (cit. on pp. 94, 105).

[182] S. Taavitsainen et al. "Single-cell ATAC and RNA sequencing reveal pre-existing and persistent cells associated with prostate cancer relapse". In: *Nature communications* 12.1 (2021), p. 5307 (cit. on pp. 91, 95).

[183] C.-Y. Wu et al. "Integrative single-cell analysis of allele-specific copy number alterations and chromatin accessibility in cancer". In: *Nature biotechnology* 39.10 (2021), pp. 1259–1269 (cit. on pp. 26, 80).

[184] S. Zaccaria and B. J. Raphael. "Characterizing allele-and haplotype-specific copy numbers in single cells with CHISEL". In: *Nature biotechnology* 39.2 (2021), pp. 207–214 (cit. on p. 24).

[185] X. Zhu et al. "Cancer Evolution: A Means by Which Tumors Evade Treatment". In: *Biomedicine & Pharmacotherapy* 133 (Jan. 1, 2021), p. 111016. DOI: 10.1016/j.biopha.2020.111016 (cit. on p. 53).

[186] F. Angaroni et al. "J-SPACE: a Julia package for the simulation of spatial models of cancer evolution and of sequencing experiments". In: *BMC bioinformatics* 23.1 (2022), p. 269 (cit. on p. 118).

[187] R. S. Brüning et al. "Comparative analysis of common alignment tools for single-cell RNA sequencing". In: *GigaScience* 11 (Jan. 2022). giac001. ISSN: 2047-217X. DOI: 10.1093/gigascience/giac001. eprint: https://academic.oup.com/gigascience/article-pdf/doi/10.1093/gigascience/giac001/42297447/giac001.pdf. URL: https://doi.org/10.1093/gigascience/giac001 (cit. on p. 99).

[188] S. Choudhary and R. Satija. "Comparison and evaluation of statistical error models for scRNA-seq". In: *Genome biology* 23.1 (2022), p. 27 (cit. on p. 88).

[189] J.-H. Du, Z. Cai, and K. Roeder. "Robust probabilistic modeling for single-cell multimodal mosaic integration and imputation via scVAEIT". In: *Proceedings of the National Academy of Sciences* 119.49 (2022), e2214414119 (cit. on p. 69).

[190] T. Gao et al. "Haplotype-aware analysis of somatic copy number variations from single-cell transcriptomes". In: *Nature Biotechnology* (2022), pp. 1–10 (cit. on p. 80).

[191] D. Hanahan. "Hallmarks of cancer: new dimensions". In: *Cancer discovery* 12.1 (2022), pp. 31–46 (cit. on pp. 4, 79).

[192] X. Hou et al. "Opportunities and challenges of patient-derived models in cancer research: patient-derived xenografts, patient-derived organoid and patient-derived cells". In: *World Journal of Surgical Oncology* 20.1 (2022), pp. 1–9 (cit. on pp. 18, 79).

[193] M. Lotfollahi, A. Litinetskaya, and F. J. Theis. "Multigrate: single-cell multi-omic data integration". In: *BioRxiv* (2022), pp. 2022–03 (cit. on pp. 69, 70).

[194] S. Milite et al. "A Bayesian method to cluster single-cell RNA sequencing data using copy number alterations". In: *Bioinformatics* 38.9 (2022), pp. 2512–2518 (cit. on pp. 71, 79, 81, 89, 119).

[195] S. Nurk et al. "The complete sequence of a human genome". In: *Science* 376.6588 (2022), pp. 44–53 (cit. on p. 19).

[196] D. Ramazzotti et al. "LACE: Inference of cancer evolution models from longitudinal single-cell sequencing data". In: *Journal of Computational Science* 58 (2022), p. 101523 (cit. on pp. 23, 54, 88, 105, 106).

[197] C. Wang et al. "Network-based integration of multi-omics data for clinical outcome prediction in neuroblastoma". In: *Scientific Reports* 12.1 (2022), p. 15425 (cit. on p. 6).