Development of bioinformatics tools for single-cell transcriptomics applied to the search for signatures of symmetric versus asymmetric division mode in neural progenitors

Nathalie Lehmann

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Institut de Biologie de l'Ecole normale supérieure

Development of bioinformatics tools for single-cell transcriptomics applied to the search for signatures of symmetric versus asymmetric division mode in neural progenitors

Soutenue par

Nathalie Lehmann

Le 7 Décembre 2021

École doctorale n°515

Complexité du vivant

Spécialité

Génomique

Composition du jury :

Denis PUTHIER TAGC, Université d'Aix-Marseille	*Rapporteur*
Marie SEMON LBMC, ENS de Lyon	*Rapportrice*
Chunlong CHEN Programme de réplication et instabilité du génome, Institut Curie	*Président, représentant PSL*
Fabienne PITUELLO UMR5547, CBI-Toulouse	*Examinatrice*
Morgane THOMAS-CHOLLIER Ecole normale supérieure	*Directrice*
Xavier MORIN Ecole normale supérieure	*Co-directeur*
Evelyne FISCHER Ecole normale supérieure	*Invitée*

1794
ENS

En mémoire de William.

"Voir en l'autre un bonheur
Voir en l'autre son coeur.
Cette flamme dont on veut s'approcher,
De brûler nous aveugle la capacité.
L'émerveillement que génère un tel brasier,
Peut pousser certains à abandonner,
Mais dans la ferveur d'un amour inconditionné,
On trouve toujours le pied qui nous fera avancer."

William Lehmann

Remerciements

J'aimerais tout d'abord remercier les membres du jury, en commençant par les rapporteurs, Denis Puthier et Marie Sémon. Vos remarques ont été particulièrement utiles dans les dernières semaines de préparation de ma soutenance de thèse. Merci également aux examinateurs, Fabienne Pituello et Chunlong Chen. Merci à tous d'avoir donné de votre temps afin d'évaluer mon travail.

J'aimerais également remercier les membres de mon comité de thèse pour leurs conseils avisés lors de plusieurs étapes clés de ma thèse: Marie-Agnès Dilliès, Adrien Six et Rachel Golub.

Cela va sans dire que mon doctorat n'aurait pas été le même sans tous ces moments partagés avec mes collègues, que ce soit au sein des équipes des plateformes de génomique et d'informatique, l'équipe DYOGEN et celle de Xavier Morin, sans oublier bien sûr l'équipe administrative qui m'a souvent facilité la tâche lorsque je me perdais parmi les méandres de l'administration française. Un merci tout particulier à tous les membres de l'équipe CSB (actuels et passés), pour votre soutien, votre amitié et votre patience (et oui, les débuts n'étaient pas faciles !). Merci également à l'équipe de SincellTE de m'avoir laissé l'opportunité d'enseigner parmi vous, et à tous ces supers moments vécus ensemble à Roscoff. Merci à vous tous de m'avoir offert l'opportunité de grandir et d'apprendre à vos côtés.

Un immense merci également à mes directeurs de thèse Morgane Thomas-Chollier, Xavier Morin et Evelyne Fischer. Xavier et Evelyne, merci pour toutes ces discussions passionnantes et pour vos remarques toujours pertinentes. Morgane, je ne saurai jamais comment te remercier suffisamment pour ton soutien inconditionnel. Merci d'avoir été un tel modèle, à la fois en

tant que femme et comme scientifique. Merci également d'avoir toujours su ménager "la chèvre et le chou" pour m'offrir un cadre d'apprentissage à la fois bienveillant et exigeant.

Merci à tous mes amis, pour votre amour, votre patience, votre soutien indéfectible et pour toutes ces fois où j'ai dû décliner vos offres, car j'avais encore du boulot à terminer. J'espère pouvoir vite me rattraper !

Il n'y a pas assez de mots pour dire à quel point je suis reconnaissante envers ma famille, à commencer par mes parents. On dit souvent qu'il y a un facteur chance dans la réussite de quelqu'un, et je ne fais pas exception: je me demande comment j'aurais pu écrire toute une thèse en anglais sans la classe européenne et les cours de français exigeants au lycée Sainte Marie, les nombreux voyages à l'étranger et que sais-je encore. Maman, merci d'être mon roc, d'être toujours prête à tout donner pour nous. A feu mon père, merci de m'avoir appris que tout était possible, que les seules limites qui existent sont celles que l'on se fixe. Merci également de m'avoir transmis sa passion pour la médecine et la connaissance: "croire savoir c'est l'ignorance, savoir c'est la science". Merci à ma sœur, Claire, d'être toujours là pour moi, de me faire rire même dans les moments difficiles et pour notre éternelle complicité. Merci à feu mon petit frère, William, à qui je dédie cette thèse. Je n'ai certainement pas son talent de poète, mais j'espère quand même avoir pu lui faire honneur en écrivant cette thèse. Merci à ma grand-mère, Micheline, pour avoir été un si bel exemple et m'avoir poussé à forger un esprit critique et une curiosité inépuisable. Merci aussi pour toutes ces discussions endiablées du dimanche après-midi sur la médecine, les sciences et les droits des femmes (sic!). Merci enfin à mes tantes, mon oncle et mes cousins pour leur soutien et leur amour.

En especial, me gustaría dar las gracias a la mejor de todas: mi esposa Dania. Gracias por estar siempre a mi lado, apoyándome y acompañándome en este camino de luces y sombras. Sabes que sin ti no hubiese llegado tan lejos, así que ya sabes, este doctorado es también un poco tuyo. Espero que podamos alcanzar juntas muchos más objetivos y aquí estaré yo para seguir ayudándote a conseguir tus sueños como tú lo has hecho para mí.

Abstract

In recent years, single-cell RNA-seq (scRNA-seq) has fostered the understanding of complex processes (e.g. cell differentiation, tumorigenesis) and their underlying cell heterogeneity at a remarkable high resolution. This approach constitutes powerful means to characterise new cell subtypes and to define their corresponding gene signatures. Although these novel technologies have become widely democratised over the past three years, the analysis of scRNA-seq data remains a challenge due to data sparsity, particularly for organisms whose genomic annotations are partial.

During my PhD, I observed that the chick *Gallus gallus* (galGal6) genomic annotations are often incomplete, thus resulting in a loss of a large number of sequencing reads for the sole reason that they cannot be assigned to any gene. This phenomenon can lead to a substantial loss of information and strongly affect the biological analysis and interpretation of the scRNA-seq data. I investigated how an enriched annotation affects the biological results and conclusions from these analyses. In this respect, we developed a novel approach based on the re-annotation of the genome with scRNA-seq data (10x Genomics short reads) and long reads bulk RNA-seq (Oxford Nanopore Technologies) from the same cell types in the chicken embryo. I developed an open-source pipeline written in Nextflow, scAnnotatiONT, which supports this approach.

This computational biology project capitalises on a tight collaboration with the experimental team of Xavier Morin (IBENS). The main biological focus is the search for signatures of symmetric versus asymmetric division mode in neural progenitors. In order to identify the key transcriptional switches that occur during the neurogenic transition of vertebrate neural progenitors,

scRNA-seq data from chicken embryos (generated by our collaborators) and mice (public data) were analyzed. To this end, I have implemented bioanalysis approaches dedicated to the search for gene signatures.

Finally, I set up an analysis pipeline to automate the processing of scRNA-seq data. This has allowed us to guarantee stable and reproducible analyses, based on an open-source tool with an optimal computing environment.

Résumé

Ces dernières années, l'émergence des approches transcriptomiques en cellules uniques (ou scRNA-seq) a favorisé la caractérisation de l'hétérogénéité tissulaire et la compréhension de processus complexes (e.g. différenciation cellulaire, tumorigenèse) avec une précision inégalée. Ces approches ont ainsi permis la découverte de nouveaux sous-types cellulaires jusque-là indiscernables, pour lesquels une signature d'expression génique a pu être établie. Bien que ces nouvelles technologies se soient largement démocratisées au cours de ces trois dernières années, le caractère sporadique des données scRNA-seq rend leur analyse complexe, en particulier pour les organismes dont les annotations sont lacunaires.

Au cours ma thèse, j'ai observé que les annotations génomiques du poulet *Gallus gallus* (galGal6) sont lacunaires, ce qui engendre la perte d'un grand nombre de lectures de séquençage qui ne peuvent être assignées à aucun gène. Cet effet peut mener à une perte d'information conséquente et affecter fortement l'analyse et l'interprétation biologique des données scRNA-seq. J'ai cherché à étudier et évaluer à quel point une annotation améliorée affecte les résultats biologiques et les conclusions issues de ces analyses. Dans ce but, nous proposons une nouvelle approche basée sur la ré-annotation du génome à partir de données scRNA-seq (lectures courtes de 10x Genomics) et de RNA-seq bulk en lectures longues (Oxford Nanopore Technologies), issues des mêmes types cellulaires. J'ai développé un pipeline open-source écrit en Nextflow, scAnnotatiONT, qui accompagne cette approche.

Ce projet de biologie computationnelle s'appuie sur une étroite collaboration avec l'équipe expérimentale de Xavier Morin (IBENS). Le principal objectif biologique est la recherche de signatures de mode de division symétrique

et asymétrique au sein de progéniteurs neuronaux. Afin d'identifier les principaux changements transcriptionnels qui se produisent pendant la transition neurogénique des progéniteurs neuronaux chez les vertébrés, des données scRNA-seq issues d'embryons de poulet (générées par nos collaborateurs) et de souris (données publiques) ont été analysées. Dans ce but, j'ai mis en place des approches de bioanalyse dédiées à la recherche de signatures géniques.

Enfin, j'ai mis en place un pipeline d'analyse afin d'automatiser le traitement des données scRNA-seq. Cela nous a permis de garantir des analyses stables dans le temps et reproductibles, basées sur un outil open-source et doté d'un environnement de calcul optimal.

Summary

List of Figures

Acronyms

BAM Binary Alignment Map. 45
BCL Binary Base Call. 44
bp Base Pairs. 24

CDS Coding DNA Sequence. 45

DE Differentially Expressed. 53
DR Dimension Reduction. 55

ENCODE ENCyclopedia Of DNA Elements. 65
ESTs Expressed Sequence Tags. 23

FACS Fluorescence-Activated Cell Sorting. 34

GFF3 General Feature Format (version 3). 45
GTF Gene Transfer Format. 45

HCA Human Cell Atlas. 38
HGP Human Genome Project. 21
HPC High Performance Computing. 49
HVG Highly Variable Genes. 53

ICA Independent Component Analysis. 55

KNN K-Nearest Neighbours. 58

LR Long-reads. 25

MT mitochondrial. 68

NCBI National Center for Biotechnology Information. 65
NGS Next Generation Sequencing. 21

ONT Oxford Nanopore Technologies. 25

PacBio Pacific Biosciences. 25
PAM Partitioning Around Medoids. 121
PCA Principal Component Analysis. 55
PCR Polymerase Chain Reaction. 24

QC Quality Control. 50

SAM Sequence Alignment Map. 45
SBS Sequencing By Synthesis. 24
scRNA-seq Single-cell RNA-seq. 20
SMRT Single Molecule Real Time Technology. 25
SR Short-reads. 24

t-SNE t-Distributed Stochastic Neighbor Embedding. 55
T.I. Trajectory Inference. 59
TP Time-Point. 115

UMAP Uniform Manifold Approximation and Projection. 55
UMI Unique Molecular Identifier. 36
UTR UnTranslated Region. 45

Chapter 1

Introduction

If there is one particular achievement that constitutes a turning point in Biology, this is the completion of the human genome sequence in 2003. Two years ahead of schedule, the International Human Genome Sequencing Consortium announced to the world they fulfilled their goal assembling millions of sequencing reads into a readable version of the human genome after 13 years of relentless efforts [1]. That's more than 3 billion base pairs (bp) to decipher and analyse. At that time, this single consensus reference genome required both a demanding collaborative work of thousands of researchers and an enormous amount of money (almost 1$ per bp) [2]. In this context, needless to say that personal genomics would not even be an option. However, thanks to intense technological enhancements, the sequencing technologies cost is on an ever decreasing cost curve [3] while their throughput continuously grows [4]. As a result, nowadays, sequencing projects have become the Swiss army knife of today's research in Biology. For each project, a growing quantity of data needs to be stored and analysed. It has even been estimated that genomics would gather more data than Youtube and Twitter altogether by 2025 [5].

Yet, how do we extract meaningful biological knowledge from all these data in a reasonable amount of time ? That's where computational approaches come in. The history of bioinformatics began early, much before every lab was equipped with desktop computers. The term itself appeared

in the early 1970s and it would describe the "study of informatic processes in biotic systems", whereby informatic processes were described as the study of the capacity to gather, process, store, and use information [6]. With the advent of computer science, it naturally evolved into what we now know as the development and application of computational tools dedicated to biological data. It is also striking to see how computational and technological development both widens the possibilities and defines the limits of our understanding of biological data [7]. With the advent of novel sequencing protocols these recent years, biological data have become even more massive, complex and noisier than before. On one side, this growth is the promise of a deeper understanding of biological systems, but on the other side it has its share of challenges. We will go over a few examples in this manuscript.

This thesis focuses on the development and implementation of bioinformatics tools for single-cell transcriptomics (a.k.a. scRNA-seq), with a highlight on genomics annotation issues when analysing scRNA-seq data from poorly-annotated species. All the analyses are applied to the search of transcriptomics signatures of symmetric and asymmetric divisions in neural progenitors.

The present introductory chapter is structured as follows :

- I will first provide a brief overview of sequencing. Its historical development, aims and standard processing will be presented. I will also describe differences, advantages and limits of short-reads RNA-seq and long-reads RNA-seq since these are essential concepts to understand our work on genome annotation in Chapter 4.

- Next, I will outline some important work in single-cell transcriptomics that laid down the foundations for the ongoing single-cell revolution. I will also explain the basics on how to process and analyse scRNA-seq data with state-of-the-art approaches.

- Finally, I will illustrate some key concepts on genome annotation, including the way public references are built and their impact on RNA-seq analyses results.

1.1 An overview of transcriptomics and sequencing technologies

These recent years, sequencing technologies facilitated full-scale approaches such as genomics, proteomics or transcriptomics. In this thesis, we will only focus on the latter one. Transcriptomics comes from the association of the terms *transcripts* and the suffix *-omics* that refers to the broad collection and characterization of biological molecules [8]. Thus, transcriptomics is defined as the study of an organism's transcriptome, the sum of all its RNA molecules.

Most of the sequencing methods were first designed for genomic DNA sequencing. They were adapted for RNA sequencing in a second phase by adding a reverse transcription step, where RNA is transcribed into complementary DNA (cDNA) [9]. Very few sequencing methods actually sequence RNA fragments [10, 11]. Thus, the history of transcriptomics is closely linked to the history of DNA sequencing technologies.

In order to consider transcriptomics in its technological and historical context, I will present here a brief overview of sequencing evolution over time and outline some technological considerations for transcriptomics approaches. Sequencing technologies are commonly separated into three generations: the first generation refers to the capacity of sequencing DNA fragments, the second generation to the capability to massively sequence small DNA fragments, and the third generation the possibility to generate much longer pieces of DNA, without the need for fragmentation. Next generation sequencing (NGS) refers to the latter two. The three generations of sequencing technologies are summarized in Figure 1.1.

1.1.1 Before NGS

1.1.1.1 First-generation sequencing

At the time of the Human Genome Project (HGP), sequencing approaches heavily relied on Sanger technology, the first sequencing technology ever implemented [12]. It was developed by the two time Nobel Laureate Frederick

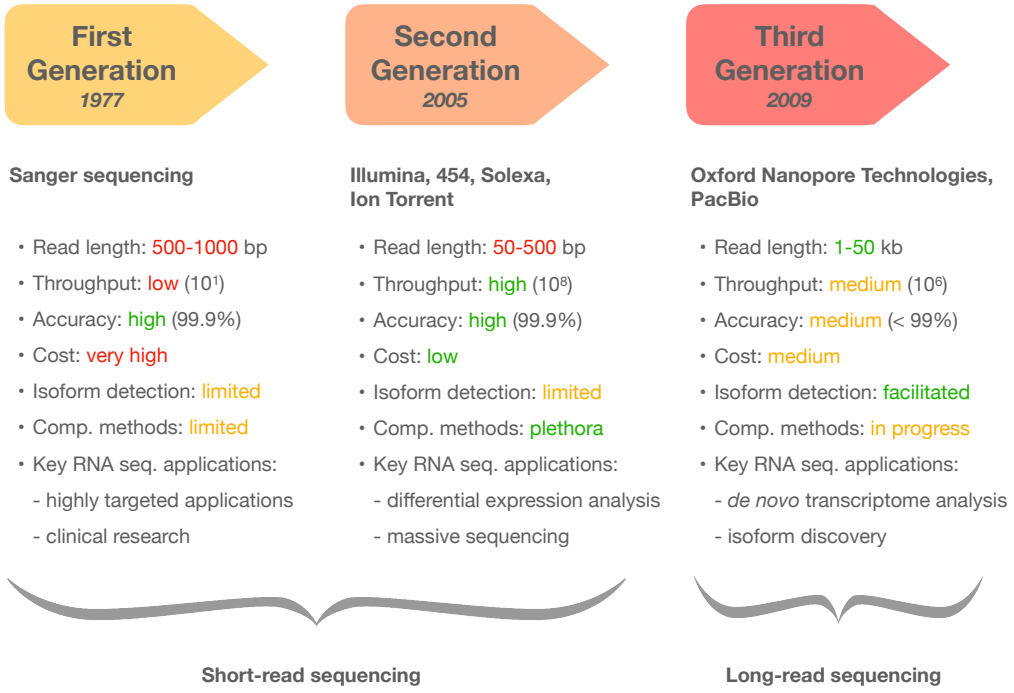


Figure 1.1 The evolution of sequencing technologies. This scheme highlights the main characteristics of the three generations of sequencing. The years in italics refer to the starting point of each generation. The second generation is best represented by Illumina sequencers, but 454, Solexa and Ion Torrent are historical sequencers. The read lengths are averages over all the sequencing reads for each generation. Ranges of absolute read lengths, specially for third-generation sequencing, are actually wider (see 1.1.3). Throughput refers to the mean order of magnitude of sequencing reads per experiment. Accuracy refers to the read accuracy, or the probability that the identified sequence is correct. One can notice that the accuracy per read and the throughput of 3rd generation is lower than 2nd generation. These values are the current standard, but they evolve rapidly.

Sanger and his colleagues in 1977. The concept behind Sanger sequencing is to generate every possible DNA fragments that differ in length by one base, up to the full length of the target DNA. Subsequently, each fragment can be separated by molecular weight (via electrophoresis), which will further allow the identification of each base (by locating the last base of each fragment through the addition of radioactive or fluorescent markers). While being the most popular sequencing method over thirty years, its application is limited to low-throughput experiments due to high cost and low efficiency [13].

Initial efforts to sequence RNA transcripts on a large scale started in the early 1990's with the advent of expressed sequence tags (ESTs) [14, 15]. ESTs are collections of short sequence reads (200–800 bp) originated from cDNA molecules that are individually cloned and Sanger-sequenced. Although this approach is low-throughput, it has been widely used to discover novel genes and enhance genome annotations for twenty years [16].

1.1.1.2 Other transcriptomics approaches before NGS

In 1995, besides sequencing approaches, the arrival of hybridization-based approaches with microarrays [17] radically changed the prospects of transcriptomics projects [18]. Microarrays' approach is based on the hybridization of DNA fragments with millions of microscopic DNA spots attached to a solid surface (*e.g.* microchips). One crucial difference between sequencing approaches available at that time and microarrays is the volume of transcripts that can be handled. When Sanger allowed the detection of a few genes, in the same time microarrays enabled the identification of thousands of genes. Above all, both the identification and quantification of RNA transcripts became possible in a single experiment. This is the beginning of gene expression profiling approaches, whereby the expression levels of thousands of genes are simultaneously assessed. However, the major drawback of microarrays is that they are based on a set of predefined known genes, preventing the discovery of novel or rare genes [19].

This restriction has been automatically lifted with the advent of next-generation sequencing approaches (or NGS) in the mid 2000s [20]. NGS specifically refers to untargeted massively parallel sequencing. It is best represented by the RNA-seq technology, whose outstanding success led transcriptomics to be mostly defined by RNA-seq. It opened the way to unbiased transcriptome-wide approaches, that quickly became a standard in research laboratories since it could offer unprecedented discovery power to detect novel or rare transcripts. These recent years, RNA-seq have evolved into two distinct categories: second-generation (short-reads) and third-generation (long-reads) sequencing technologies. I will detail both in the next paragraphs.

1.1.2 Second-generation sequencing

As mentioned above, second-generation sequencing most commonly refers to short-read (SR) RNA-seq. If the technology emerged more than a decade after the first microarrays, its applications and large endorsement did not take long. The first studies which mention high throughput sequencing were released in 2006 [21], although the term RNA-seq appeared a little bit later [22, 23].

In a typical RNA-seq protocol, a population of RNA is first reversed transcribed into cDNA. Each cDNA molecule is then fragmented into smaller pieces, typically fragments of 50-500 bp. Then, to make sequencing priming possible, sequencing adaptors are attached to one (*single-end*) or both ends (*paired-end*) of each molecule. To ensure sufficient amount of cDNA to be sequenced, all molecules are then copied into multiple pieces through polymerase chain reaction (PCR) [24]. Then, in case of Illumina sequencing (about 90% of the overall sequencing data in the world[1]), the library is sequenced via sequencing by synthesis (SBS) [25]. This technique relies on fluorescently labeled dNTPs that are progressively integrated into a DNA template strand, which allows to detect each of the bases of the original sequence. Finally, the resulting sequence "reads" can be analysed with bioinformatics approaches.

In terms of transcriptomics, the main advantage of RNA-seq over microarrays is that it removes the need to have previous knowledge of the RNA transcripts to detect. This opened the way to the discovery of numerous previously unknown genes in a wide range of fields of Biology from oncology [26] to evolution [27]. Compared to previous methods, it is also much more cost-effective [28]. As RNA-seq became broadly adopted, a wide range of analysis tools and pipelines have been developed for RNA-seq analyses [29].

Despite its undeniable success in transcriptome analysis, standard RNA-seq suffers from limitations due to its restricted read length [30]. With the short-read approach (based on fragmented cDNA), isoforms must be reconstructed with computational methods thanks to the identification of reads

[1] https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf

that overlap two exons, as shown in Figure 1.2-c. Tools for transcriptome reconstruction have been extensively developed and used by the community, but still show limitations in a large number of cases [31]. For example, it fails at identifying complex or rare isoforms, and reconstructing particularly long transcripts is impossible: a single RNA-seq short-read will not allow to associate exons that are more than 1 kb apart [9]. It is also a major source of concern when there is no reference genome or when its quality is insufficient.

1.1.3 Third-generation sequencing

The NGS third-generation refers to long-read approaches. Unlike short-read, long-read (LR) sequencing technology provides the ability to identify and quantify RNA transcripts from end to end (*i.e.* from the 3' polyA tail to the 5' cap). While standard RNA-seq generates reads of up to 500 bp, long-read sequencing technologies routinely generate reads of 10 kb [33]. By its unprecedented capacity in deciphering complex and highly accurate isoform-level transcriptomes, recent studies estimate long-read will eventually become the new standard for transcriptome analysis, as short-read RNA-seq did a decade ago [9].

Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT) are the two market leaders of long-read sequencing technologies. While they rely on very distinct principles, their technologies have proved equally successful and are applied to an increasing number of studies [34]. PacBio sequencers, the first ones to introduce long-read approaches in 2008, are based on the single molecule real time technology (SMRT) [35, 36]. The addition of each single nucleotide, by a polymerase, emits fluorescence that is recorded and associated to a specific base. With this approach, the longevity of the polymerase defines read length limits (from 250 bases to 50 kb). More recently, in 2014, ONT introduced a technology where read length is mostly constrained by the molecular weight of cDNA fragments [37]. Longest read lengths are provided by ONT sequencing, from 500 bp to the all-time record of 2.3 Mb [38]. Each fragment goes through a biological nanopore enabling the quantification of ionic current fluctuations. These variations differ according

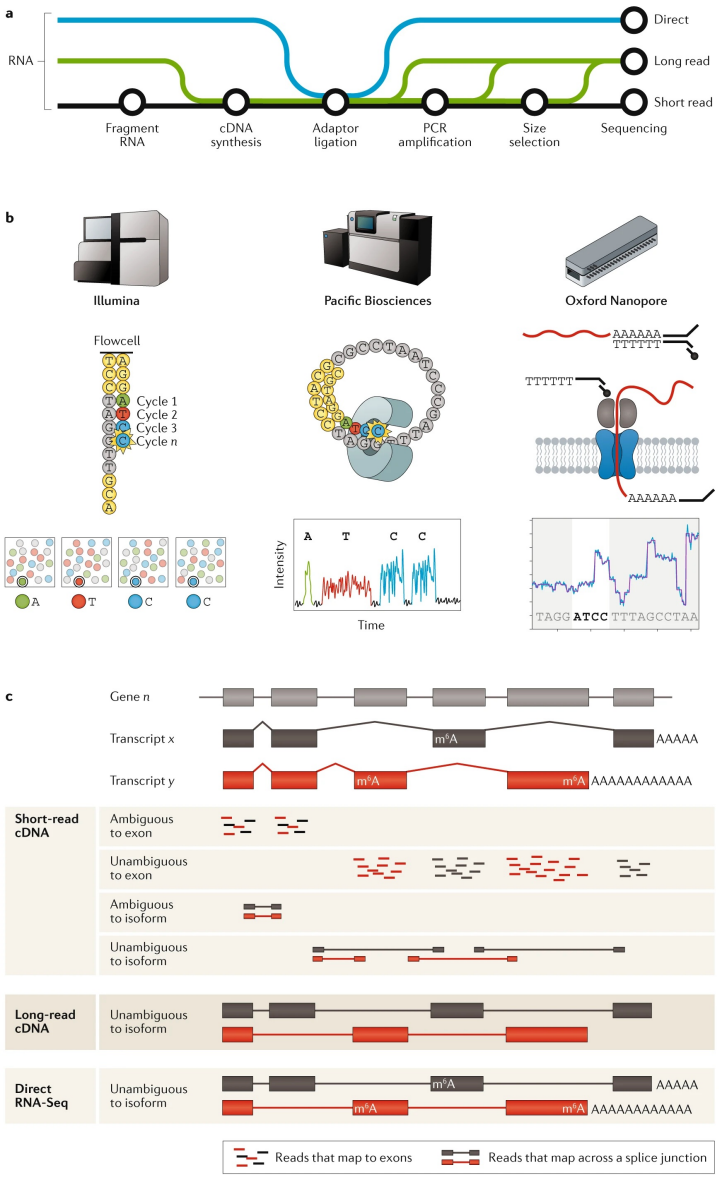


Figure 1.2 Short-read and long-read RNA-seq technologies a) Overview of the shared steps between the three main RNA-seq library preparation protocols. Short-read sequencing is represented in black, long-read cDNA in green and long-read direct RNA in blue. All methods involves an adaptor ligation step. b) Each protocol is represented by its emblematic machine (*e.g.* Illumina's NextSeq, PacBio's RSII and ONT's MinION), followed by a schematic representation of its corresponding sequencing approach. Illumina technology is based on sequencing by synthesis (left panel), PacBio workflow depends on fluorescently labelled nucleotides (middle panel), and ONT approach relies on current variation whenever a base passes through a biological nanopore (right panel). c) Computational analysis differ depending on each protocol. A gene n is represented by two isoforms: transcripts x and y. With short-read, isoform detection can be compromised by reads that map ambiguously. Long-read cDNA methods can generate full-length isoform reads, thus unveil complex isoforms. Direct RNA-seq enables a thorough analysis of isoforms, including full-length isoforms characterization, epitranscriptomic modification (*e.g.* N6-methyladenosine or m6A) and polyA tail detection. Figure from *Stark et al., 2019* [32]

to which base is going through the pore. It results in a unique pattern of electrical fluctuations which is used to establish the corresponding sequence of bases.

LR technologies mostly suffer from lower sequencing quality (thus low *accuracy*), that intense development have managed to tackle these last years [39]. As a result, 3rd generation read accuracy is evolving extremely fast. Latest studies have estimated accuracy of PacBio and ONT to be around 99% [40] and 95% respectively [41]. To improve data quality, LR are frequently used in combination with SR in what is called *hybrid* approaches [42]. It refers to a fine combination of short-read (for their high accuracy and high throughput) and long-read (high isoform precision) approaches, and was precisely the object of first LR major studies [43, 44]. Early studies also applied LR technologies to a small selected set of highly complex transcripts impossible to decipher with short-read approaches [45]. LR also undergoes a reduced throughput compared to SR approaches but this has never prevented LR to be used to perform whole transcriptomes with long-read alone since 2013 [46, 47]. More recently, the possibility to sequence RNA directly with ONT sequencers (without the cDNA reverse transcription step) opened up new opportunities and challenges [11, 48, 49]. It has been shown that the systematic use of reverse transcription in traditional RNA-seq approaches introduces biases [50] and entails loss of information such as polyA tail length and RNA base modifications (Figure 1.2-c) [32]. These recent developments are a great opportunity to bypass these limitations [51].

While third-generation sequencing costs decrease, long-read RNA-seq is quickly becoming the state-of-the-art to:

- Improve genome reference annotations;
- Perform RNA-seq on organisms whose genome is left unannotated;
- Facilitate transcript isoform identification and quantification.

Together with read accuracy and throughput that are continuously improving, LR sequencing is thought to eventually replace standard RNA-seq for bulk approaches in a near future [52]. However, there is a considerable need to develop tailored analysis tools, since LR approaches differ substantially

from SR approaches. Initiatives such as https://long-read-tools.org/ [53] reference emerging LR dedicated tools. As of 17 September 2021, the database catalogues 555 tools across 32 categories. The most represented category is precisely "error correction and polishing" (over 20% of all the tools).

1.2 The single-cell transcriptomics breakthrough

Human bodies are estimated to contain around 38 trillions cells [54], that are often claimed to be divided into 210 cell types [55]. What makes the specificity of each of these cells stands in the unique set of genes that they express, now routinely detected through transcriptomics. When RNA-seq started to establish itself as the cutting-edge approach to study cells transcriptome, it was restricted to the analysis of mixture of cells [56–58]. RNA-seq projects were population-based studies, what is commonly called *bulk* analyses. The prospect of unravelling the transcriptome of individual cells quickly emerged, but was put on hold due to technical limitations [20]. *Tang et al., 2009* broke down some barriers in 2009 when applying RNA-seq to a unique cell for the first time [59]. Single-cell RNA-seq (or *scRNA-seq*) was soon to become the novel forefront strategy to do transcriptomics.

In this section, we will first go through a quick overview of a scRNA-seq experiment steps, then highlight some of the elements of single-cell protocols that were key to drive these approaches to an increasing adoption in the scientific community, and finish with some examples of significant achievements and remaining challenges in single-cell transcriptomics studies.

1.2.1 General overview of a single-cell experiment

As described in the previous section (see 1.1), bulk RNA-seq studies enabled major discoveries in numerous sub-domains of biology. However, the main drawback of bulk analyses stands in the fact that genes expressions are averaged across the cells of a sample. Although this is not necessarily an issue to compare different tissues or conditions (*e.g.* treatment VS control), it is a restricting factor when a finer resolution is needed (*e.g.* understanding cells differences in a developing embryo or identifying rare cell populations). Even if similar cells are selected with meticulous care, tissues are rarely homogeneous [60, 61]. Unlike with bulk approaches, single-cell RNA-seq provides a unique expression profile for each single cell within a sample. Therefore, scRNA-seq appears as the most appropriate approach to answer questions that require cellular resolution.

The following two sections will present in details scRNA-seq analyses, but I will first provide an overview of the main steps involved in all scRNA-seq experiments (also summarized in Figure 1.3).

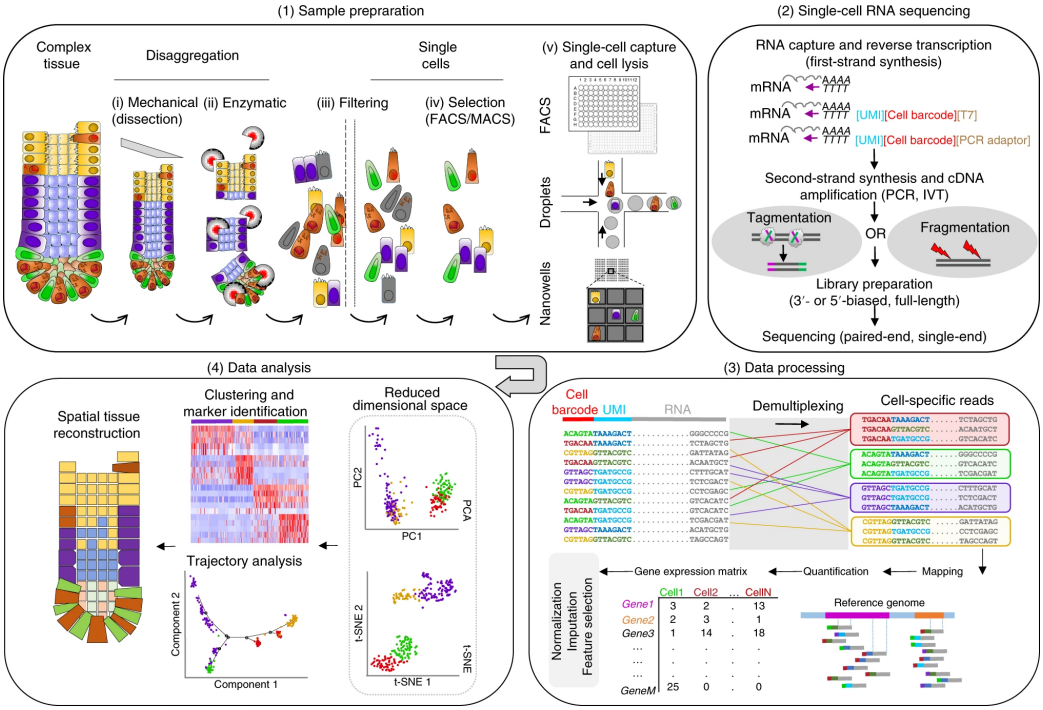


Figure 1.3 The single-cell RNA sequencing experiment steps. Image from *Lafzi et al., 2018* [62]

Broadly speaking, a scRNA-seq experiment can be divided into five parts (steps 1 to 3 are experimental, while steps 4 and 5 are computational):

1. **Sample preparation:** this part is intended to prepare cell samples before going into partitioning. It is a critical step to i) ensure cells will be of sufficient quality and ii) there will be enough viable cells for the analyses that will follow. The design of its protocol depends almost exclusively on the wet-lab knowledge we have on the cell type of interest, which is out of the scope of this thesis. I can however recommend *Lafzi et al., 2018* [62] that is a recent and excellent review that covers essential guidelines for this step.

2. **Single-cell dissociation:** physically partitioning the cells into indi-

vidual reaction containers is the core task of scRNA-seq protocols. It consists mainly into cell capture and lysis. Cells may be separated inside wells (e.g. Smart-Seq2 [63]) or oil droplets (e.g. Drop-seq [64]). We detail some of the key elements to take into considerations for this step in the following section (see 1.2.2).

3. **Library preparation and sequencing:** this step covers successively RNA capture, reverse transcription, cDNA amplification, library preparation and finally sequencing strictly speaking (see previous section 1.1.2 for more details on sequencing technologies). It outputs biological cDNA fragments as sequencing reads in a machine-readable text format, enabling the quantification of gene products in the following step.

4. **Primary analyses:** this part covers all the steps to generate, starting from the raw sequencing reads, a count matrix in which each row is a gene and each column is a cell. It can also be referred to as *data preprocessing*. The key phases are: reads quality checking and filtering, genome mapping and expression quantification. The detailed process is described in section 1.3.1.

5. **Secondary analyses:** they define the data analyses *per se*, from which biological conclusions can be extracted and novel hypotheses formulated. Data analyses are tailored to the biological question, which allows an infinite combination of tools and approaches to apply to. Generally, these analyses are separated into i) data cleaning (*e.g.* quality control, normalisation), ii) cell assignment (e.g. clustering, pseudotime reconstruction) and iii) gene identification (e.g. differential expression) [65]. It is detailed in section 1.3.2.

1.2.2 Single-cell transcriptomics protocols

When *Tang et al.* performed the first RNA-seq experiment on a single-cell, it was in fact not the first time a team attempted to identify and quantify gene expression within a cell. In 1992, *Eberwine et al.* also measured gene

expression at the cellular scale with an innovative protocol based on *in vivo* reverse transcription and *in vitro* transcription [66]. At the time, this study shed light on the novel idea that cells morphologically similar might have different patterns of expression. As innovative this approach might have been, it only applied to a handful of pre-selected genes and was cumbersome to implement. Several other similar studies emerged in the following decade, with simpler protocols (*e.g.* PCR-based amplification [67]) and broader scale of application (*e.g.* microarrays allowed transcriptome-wide studies [68, 69]). What changed with *Tang et al.* approach is the ability to recover a cell's RNA transcripts in an untargeted and unbiased way (unlike microarrays), and submit them to massively parallel sequencing.

Once the possibility to study transcriptome-wide single cells have been demonstrated, came the time of the question of the amount of cells to process. Until then, cells were meticulously selected and limited to a low number. *Guo et al., 2010* pointed out the interest and feasibility of sequencing several hundreds of cells [70]. To this end, they demonstrated that various cell types could be identified and compared on the basis of their transcripts, without the need for an upstream cell selection step (even though their analysis focused only on 48 genes). The following year, *Islam et al., 2011* took a lead in the field by combining both approaches (transcriptome-wide analysis applied to a large number of cells simultaneously) and developed the first method to access the entire transcriptomes of a multitude of single cells, reaching 85 cells with an overall of 13,879 detected genes [60].

Since then, over fifty [72] highly multiplexed protocols for single-cell RNA-seq have flourished with a few standing out, such as Smart-seq2 [63], Drop-seq [64] or 10x Genomics Chromium [73]. The choice of the strategy used for cell isolation underlies the *throughput* of the experiment (*i.e.* the number of cells to isolate). Each protocol has its own strengths, weaknesses and biases, which will require specific adjustments at the steps of data analyses.

With more sensitive and accurate methods, both the gene detection rate [74–76] and the number of processed cells [71] have significantly increased in recent years. Routine studies now include thousands of genes within 1k to 10k of single cells. Massive studies yield 100k cells (see section 1.2.3). On

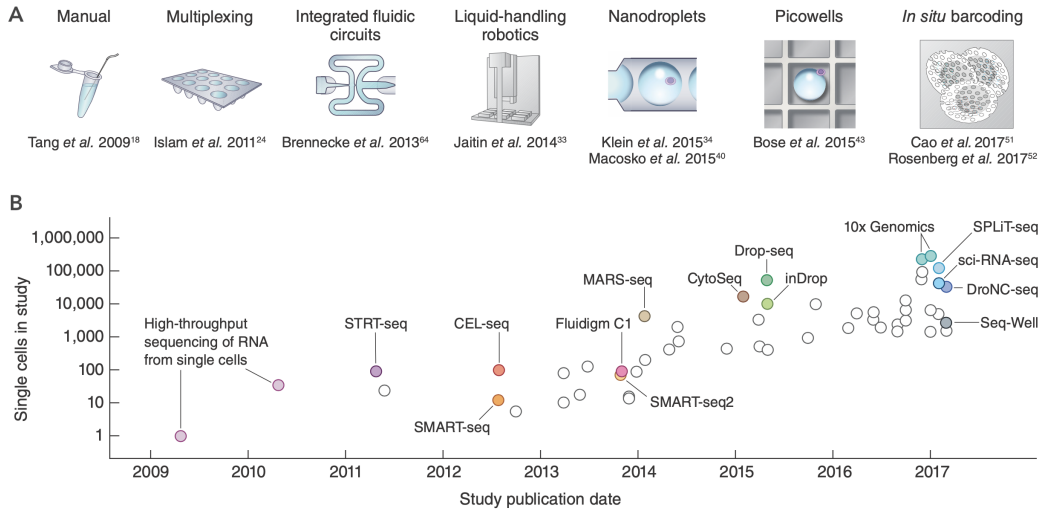


Figure 1.4 Time evolution of scRNA-seq experiments. A) Main technologies that enabled the quick growth of scRNA-seq. B) Throughput observed in major scRNA-seq publications, ordered by publication date. Image from *Svensson et al., 2018* [71]

top of that, two papers from *Cao et al., (2019 and 2020)* even went up to respectively 2 and 4 millions of cells [77, 78]. Figure 1.4 highlights the exponential growth of single-cell experiments protocols from 2009 to 2017.

Single-cell approaches would certainly not have been such a breakthrough without innovative protocols that allowed the exponential scaling of scRNA-seq. We focus here on two key elements: i) the ability to separate cells in a non-destructive and efficient manner and ii) the capacity to massively capture and sequence RNA in an unbiased way.

1.2.2.1 Automatic cell isolation

Pioneer single-cell protocols relied on manual separation of cells in individual tubes [71]. This approach is necessarily limiting as soon as one wishes to process hundreds of cells simultaneously. In order to increase the throughput of the experiment, an automated cell isolation technique is mandatory. Particular attention must also be brought to cells integrity: reaching a higher throughput should not be obtained at the cost of cells quality.

Depending on how the cells are captured and isolated, prevailing protocols for single-cell partitioning are usually separated into three distinct approaches: microwell plates, microfluidics chips and microfluidics droplet methods. Key features of each of these approaches are summarized in Figure 1.5.

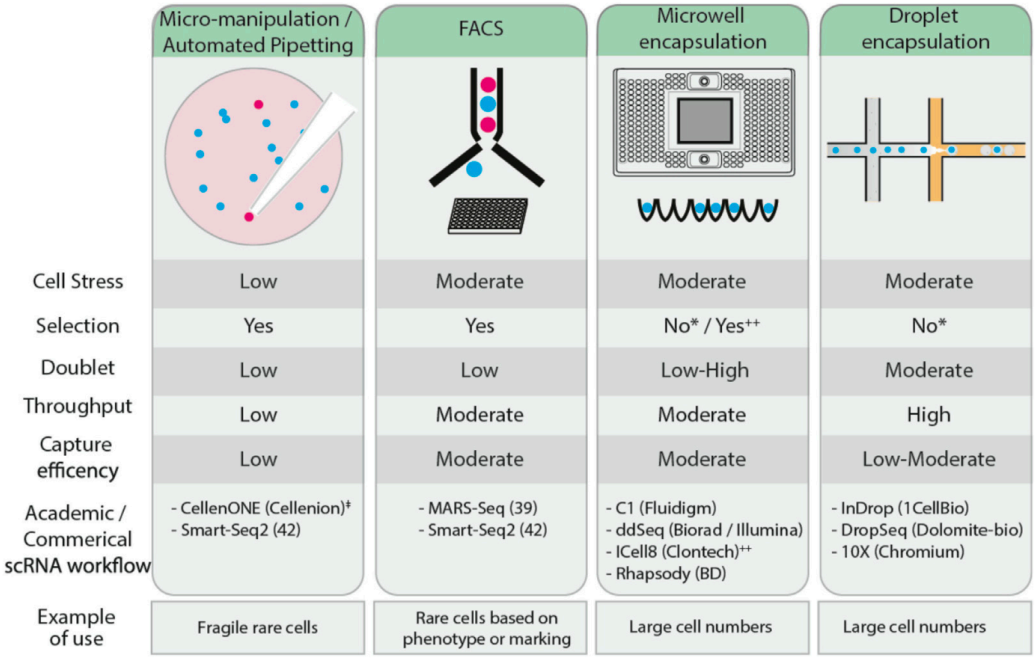


Figure 1.5 Comparison of scRNA-seq protocols with their key features. Comparison of the key features of the four main approaches to isolate cells. *Doublet* refers to the tendency of the technology to poorly isolate cells (some cells are thus mixed together, which is generally referred to as doublets). Image adapted from *Nguyen et al., 2018* [79]

Microwell plates This approach is the lowest throughput. It is the method used in original studies such as *Tang et al.*. It is now best represented by Smart-seq2 [63] and the more recent Smart-seq3 [80] protocols. With this setting, cell separation is mostly based on fluorescence-activated cell sorting (FACS), though manual pipetting or microdissection are still possible [62]. The main drawback of well-based methods is that they require considerable time, effort and money per processed cell. This is the method of choice for precious samples of rare cells [71]. Its appreciable advantages include the

possibility to add an extra layer of information on each cell, such as macro details (size, morphology, etc.), or visually discard doublets or damaged cells.

Microfluidics chips This technology comes from an original idea of the Fluidigm company when they introduced the C1 system in 2013, the first automatised commercial protocol dedicated to single cell isolation [71]. What is innovative with this approach is the use of microfluidics to deliver precise tiny volumes (e.g. cells) into nano-reaction chambers. In terms of throughput, it is a better alternative than plate-based protocols. However, it performs poorly in terms of cell capture as only around 10% of the cells from the original sample are processed, a major disadvantage for populations of rare or delicate cells.

Microfluidics droplets This method is the highest throughput and also the most recent as it appeared in 2017. Due to commercialisation efforts of the 10x Genomics Chromium, an all-in-one solution for automatic single-cell isolation, this approach has become extremely popular. It is based on nanoliter droplet emulsions. Within the microfluidics channels of a chip, each cell is captured inside a nanoliter-sized oil droplet. Reverse transcription reagents and beads containing barcoded oligonucleotides (necessary for library preparation) are also encapsulated in the droplet. Because each bead contains a unique identifier, all the transcripts of a cell can be identified with the same *cell barcode*. This is key to allow for multiplexing: when droplets are lysed, RNA transcripts can be mixed and sequenced together without losing the information of the cell they originated from. It allows to process tens of thousands of single cells in parallel within a 1-day workflow [81]. The cell capture rates vary highly depending on the chosen platform. For 10x Genomics, it usually lies between 50% to 65% [73, 82].

1.2.2.2 Broad and unbiased transcript quantification

The capacity to capture any given RNA from the cell, and process it up to the quantification step is called the *sensitivity*. To ensure high sensitivity, one needs to take two key factors into consideration: i) amplification biases

and ii) transcript quantification methods. The sensitivity differs significantly for each scRNA-seq protocol.

Amplification biases With standard RNA-seq protocols, about 0.1 to 1 µg is needed to properly detect RNA [71]. In terms of RNA quantity, this represents millions of cells since the amount of RNA in a single cell ranges from 1 to 50 pg (depending on the cell type, size and state) [83], especially since less than 5% of it is mRNA. This first challenge can be overcome with PCR amplification, which exponentially creates copies of each transcript. Yet, PCR amplification is known to cause amplification biases: due to differences between the length of RNA transcripts, the amplification rate is not identical for all the transcripts [84]. It thus causes disparities between real RNA quantification and estimated quantification post-amplification, some might be over-estimated. With droplet-based protocols, it is possible to tackle this issue by associating a *Unique Molecular Identifiers* (UMI) to each transcript. UMIs are random nucleotide sequences which are usually 10-12 bases long. It is generally considered long enough to ensure their uniqueness (4^{10} - 4^{12} possibilities). Each polyT primer (which binds to mature mRNA molecules) is associated to a UMI together with a unique cell barcode. During hybridization step, each transcript gets its own unique combination of UMI and cell barcode. It then enables the unbiased counting of transcripts [85], as PCR duplicates can be removed during the computational analyses steps. It is worth mentioning that UMI processing requires specific processing steps that handles UMI and the possibility that they contain sequencing errors (see section 2.2) [86].

Transcript quantification method In the above paragraph, we mentioned that UMIs are available only for droplet-based protocols. The reason for this stems from the type of RNA capture method, that has a direct impact on transcript coverage and quantification. In fact, two types of transcript quantification exist: full-length and tag-based (a.k.a. 3'-end-biased / 5'-end-biased). The former aims at spanning the whole transcript, while the latter preferentially captures just one end of the transcript (5' or 3' end).

Full-length protocols are specific to plate-based protocols (such as Smart-seq2) and are similar to bulk RNA quantification methods. RNA transcripts are fragmented into multiple short reads, and the full original transcript can be reconstructed with computational methods (see 1.1). This approach thus benefits from a detailed picture (*isoform resolution*) for the quantification. They are also known to be more sensitive than tag-based protocols: the mean number of genes detected in cells can be two-fold higher [75].

On the contrary, tag-based protocols are limited to *gene quantification*, as only one end is sequenced. In this respect, isoform identification is jeopardized, making it difficult to differentiate reads that come from a transcript or another. This quantification approach is nonetheless the most popular among scRNA-seq protocols for the reasons outlined above. Note also that it is restricted to droplet-based protocols.

Thus, low amounts of starting material in single-cell protocols is a promise for a deeper understanding of cellular tissues, but this comes at the cost of more uncertainty and variability in the estimates of transcriptional states than bulk (see 5.2.1) [87]. Moreover, the transcriptomes analysed from the single-cells are just a sample of all RNA transcripts physically present in the cell: i) the majority of protocols use primers that bind to polyA tails, thereby capturing mostly mRNA and some non-coding RNAs, while missing some mRNAs that would not have a polyA tail and ii) there are high discrepancies in RNA capture depending on the chosen protocol. With plate-based protocols, we estimate than 30% to 40% of transcripts are captured [80]. This number drops to 5% to 20% (*i.e.* for the most recent approaches) with droplet-based protocols [76]. While first single-cell studies focused on atypically large cells (10 to 100 fold larger than an average cell) [60], the most recent high-throughput protocols can deal with most of cell sizes and shapes [76].

The choice of the protocol for a given study mostly depends on:

- **Cost:** lower-throughput methods being between 100 to 1000 times more expensive than high-throughput methods (in price per cell) [75, 88].

- **The biological question:** is it more critical to get a deep understanding of a few selected cells (low-throughput / high sensitivity) or to grasp a more general picture of cell diversity in a sample (high-throughput / low sensitivity) ? It is a matter of finding the right balance between sensitivity and the amount of cells to process.

All the analyses in this thesis rely only on high-throughput, droplet-based, data (produced with the 10x Genomics Chromium technology).

1.2.3 Recent achievements in biology

Single-cell technologies have been widely used in all fields of biology such as immunology [89], embryology [90, 91] or oncology [92]. Great achievements in the single-cell field are plentiful [93], and far be it from me to enumerate them all. However, I wish to mention a few key applications that already had an impact on both fundamental research and medicine.

- **Developmental biology:** Because embracing how individual cells differentiate into complex and exquisitely organized tissues is a fundamental question in Biology, developmental biology is undoubtedly one of the fields that most benefited from scRNA-seq since its beginning [94, 95]. Most recent studies rely on an astonishing amount of data (which would not have been possible just a few years ago) to build comprehensive models of mouse gastrulation from whole-embryos [96], study developmental cell fate decisions in zebrafish [97], or profile the human developing spinal cord transcriptome [98], to name just a few. Beyond simply providing novel or improved cell catalogues, these studies question some basic notions, such as the very definition of cell types [55], or the traditional models associated with cell differentiation [96]. *Ton et al., 2020* provides a comprehensive review of the significant contribution of scRNA-seq to the developmental biology field [99].

- **Cell atlases:** A great amount of studies aim at providing exhaustive catalogues of cells. The most popular ones are probably the Human Cell Atlas (HCA) [100], the Fly Cell Atlas [101] and Tabula Muris

(mouse atlas) [102]. The drosophila was the first organism that benefited a cell atlas initiative [103]. The HCA, which started in 2016 and is led by Sarah Teichmann (Wellcome Sanger Institute) and Aviv Regev (Broad Institute), is a massive collaborative project, gathering over 2000 members over the world. It aims at annotating and referencing every cell types in the healthy body, across time from development to adulthood. It has been thought as a real "Google map" of the human body [104]. Throughout this type of effort, high-resolution single-cell transcriptomics data adds ground-breaking and highly valuable information into fundamental knowledge of human tissues.

- **Health and disease:** Before single-cell analyses, cell type identification in immunology was mostly based on a limited set of surface markers and morphological characteristics. Single-cell enabled the field to move beyond this discrete classification and tempered cell type definition by joining a unique transcriptomics signature to each cell type [105]. It also allowed the discovery of major (but rare) cell types (e.g. in blood [106]). In cancer, single-cell analyses provide a deeper understanding of tumor heterogeneity, evolution in time and how cells transient from a state to another [107]. Finally, in the context of the global pandemic, single-cell analyses have been harnessed to quickly comprehend and tackle COVID-19 infections [104, 108, 109] and its key role is widely recognized [110]. The COVID-19 single-cell atlases have been quickly implemented thanks to the HCA community [111]. As a matter of fact, more than 25% of the COVID-19 related scientific papers mention single-cell analyses as well[2].

[2]Results of the research on Google Scholar "("single-cell RNA-seq" OR "scRNA-seq" OR "single-cell analyses") AND intitle:("COVID-19" OR "SARS-COV-2")" gave 4520 results. Results for intitle:("COVID-19" OR "SARS-COV-2") outputs 16800 papers. Results retrieved the 25th August 2021.

1.3 Analysing single-cell RNA-seq data

Single-cell RNA-seq data analyses involve a multitude of interdependent steps that constitutes the overall *in silico* pipeline. Due to data complexity, its heterogeneity, the bewildering evolution of technologies and the vast diversity of questions that can be address with scRNA-seq, a great variety of tools and options are available. As of 12 August 2021, 1027 tools have been catalogued in the highly comprehensive website scrna-tools.org [65], 765 of which were added within the last 3 years[3] (Figure 1.7).

This continuously growing number both reflects the significant demand for scRNA-seq analysis and dedicated tools, but also emerging issues that come with novel technological developments. Considering the colossal task to reference and compare all the possibilities is out of scope of this PhD thesis, I will here give the keys to globally understand scRNA-seq data analyses process, and introduce how to select tools and approaches depending on the biological questions.

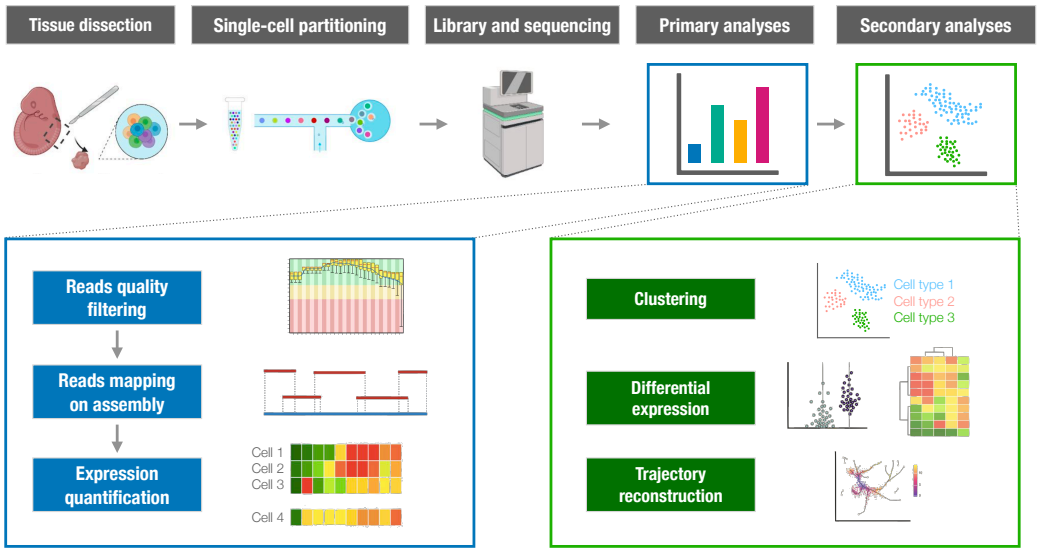


Figure 1.6 Schematic representation of scRNA-seq analyses. Primary analyses (in blue) include all the steps from the raw sequencing reads up to the generation of the count matrix. Secondary analyses (in green) encompass all the steps more directly aiming at revealing biological insights. My thesis focuses on both types of analyses.

[3] As of 10 August 2018, 262 tools were referenced in scrna-tools.org.

These analyses are generally separated into two main parts: primary analyses (a.k.a. *pre-processing*) and secondary analyses (Figure 1.6). Even though secondary analyses often overshadow pre-processing parts because they reveal biological insights, a major part of this thesis tackles primary analyses issues, and highlight how the pre-processing can have a major impact on the secondary analyses. In this respect, I will introduce here each step of scRNA-seq analysis and will give more explanations in their corresponding chapters: Chapter 2 for primary analyses and Chapter 3 for secondary analyses.

1.3.1 Primary analyses: from reads to count matrix

The purpose of primary analyses is to construct a scRNA-seq count (or *expression*) matrix from the sequencing reads. In a count matrix, genes are commonly defined as rows, cells as columns, and the values are the estimated gene expression levels. Primary analysis process broadly includes:

1. **Demultiplexing and barcode processing:** this step mostly consists of handling and processing raw sequencing files (*FASTQ* files);
2. **Alignment :** mapping each sequence read to a reference genome (or, in some cases, transcriptome);
3. **Feature assignment:** assigning reads to identified features (which are commonly genes or transcripts) according to a reference annotation;
4. **Quantification:** quantifying gene (or transcripts) expression levels within each cell.

A general overview of a pre-processing pipeline is shown in Figure 1.8. The accuracy and successful completion of each of these steps is of the utmost importance to ensure good-quality and reliable biological results since all the downstream analyses will depend on the resulting count matrix [113]. In addition, scRNA-seq pre-processing requires high-performance computing capabilities in order to process the millions (or even billions) of sequencing reads generated by scRNA-seq protocols. Many of the steps described below

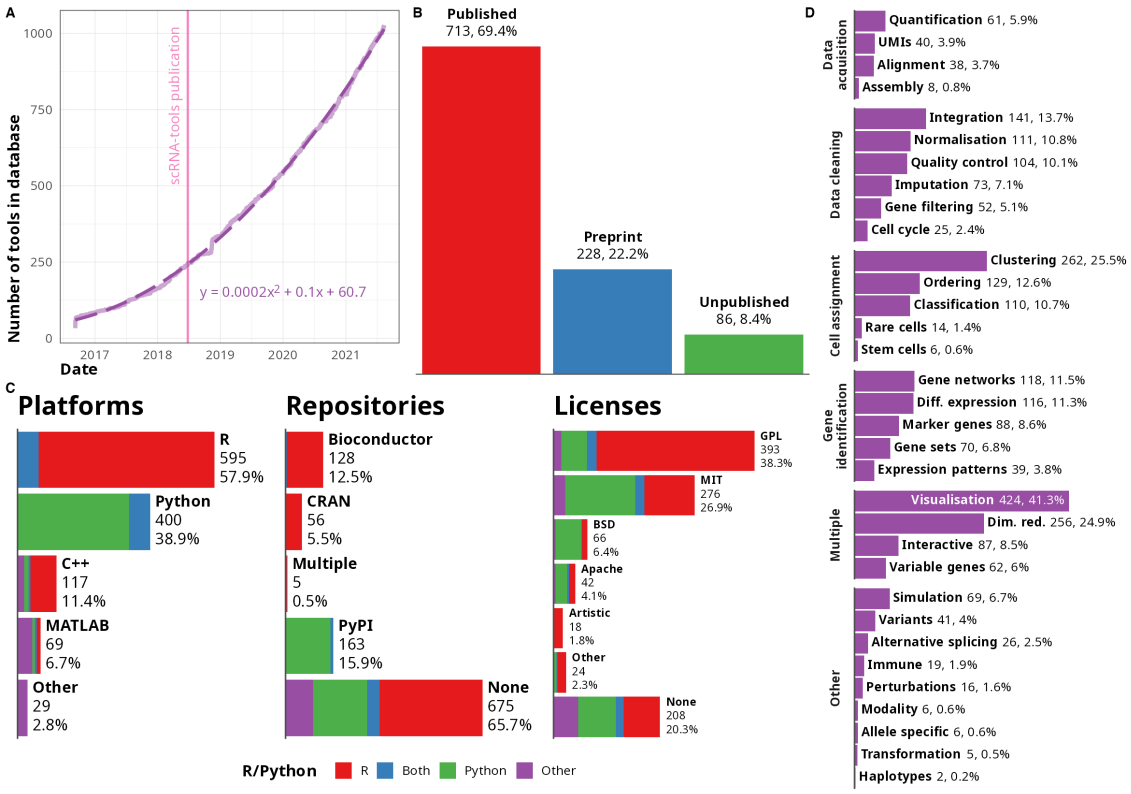


Figure 1.7 Overview of tools catalogued in the scRNA-tools database. A) Time evolution of the number of tools catalogued in the scRNA-tools database. We can observe here that the dotted line follows a quadratic fit. B) Distribution by publication status of each of the tools included in the database. Over two thirds of these tools are published (713 tools, representing 69.4% of all tools). C) Distribution of platforms, software repositories and software licenses among the tools of the database. The colours represent the programming languages of these tools (a vast majority being R or Python). D) Bar plots showing the distribution of tools among the main scRNA-seq analyses categories. Image from *Zappia et al., 2021* [112]

are shared with standard bulk RNA-seq pipelines, but it will be explicitly mentioned wherever changes are notable.

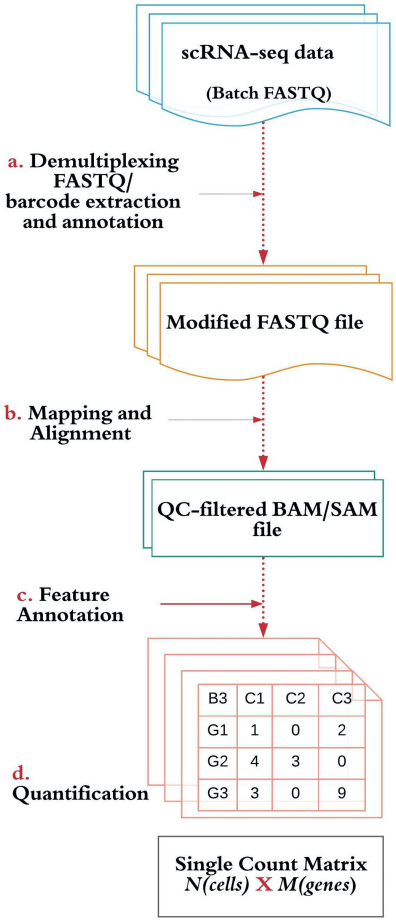


Figure 1.8 A standard scRNA-seq pre-processing pipeline. The different steps are identified with small letters. Typical file formats are written within colored boxes. (a) The first step of the pre-processing pipeline usually consists of FASTQ files demultiplexing. (b) The resulting FASTQ files are used as input for genome mapping. (c) Reads are then assigned to a feature (usually genes) thanks to a gene annotation file. Unassigned or inappropriately assigned reads are filtered out. (d) Then, correctly assigned reads are quantified. It thus enables to determine the number of genes associated to each cell barcode. The result is stored in a count matrix. Genes are commonly defined as rows, cells as columns, and the values are the estimated expression levels. Image adapted from *Nayak et al., 2021* [114]

1.3.1.1 Demultiplexing and barcode processing

The starting point of the scRNA-seq *in silico* pipeline is the conversion of raw sequencing data to human readable files, known as FASTQ files. Illumina RNA-seq sequencers traditionally output files in the binary base call (BCL) format. The first step is thus to convert BCL to FASTQ format. This is usually done in parallel with *FASTQ demultiplexing*, which consists of splitting the reads into distinct files (depending on the sample index). Both the conversion to FASTQ files and demultiplexing is carried out by the Illumina proprietary tool *bcl2fastq*[4]. A FASTQ file is a text-based file format that stores the raw sequence and its corresponding quality scores. For a single scRNA-seq experiment there are often several dozens of FASTQ files (see Chapter 2).

The next step is to assign all sequencing reads to the cells they originally belonged to. This is called *cell barcode demultiplexing*. Each scRNA-seq protocol requires a specific cell barcode demultiplexing method due to the differences in barcodes structure, processing and FASTQ files organization.

Finally, the overall quality of each FASTQ file can be estimated with standard bulk RNA-seq tools like FASTQC [115]. If some reads quality is too poor (*e.g.* the probability that a given sequence is erroneous is too high), they may be filtered out at this stage.

1.3.1.2 Alignment

After FASTQ processing and quality checking, the sequences need to be located on a reference assembly (which could be either a genome or a transcriptome), assuming reads are similar enough to some part of this reference to be associated with. To this aim, the alignment step consists in scoring reads sequence similarity at multiple positions on the genome or transcriptome. RNA-seq mapping is admittedly one of the most challenging tasks for it requires to handle millions of reads in a time-efficient manner, despite taking into account sequencing errors, repetitive elements and biological dif-

[4] https://support.illumina.com/sequencing/sequencing_software/bcl2fastq-conversion-software.html

ferences between the reference assembly and the genome of the studied cells [116]. Moreover, the choice of an alignment tool rather than another might critically change the output of bulk RNA-seq data analyses [117, 118], which is even more true with scRNA-seq analyses [119].

As shown in Figure 1.8-b, this step outputs SAM files or its binary version, BAM files. In SAM/BAM format, each line defines a read. It contains information such as the complete nucleotide sequence, mapping position on the genome and strand to which the alignment was estimated [120]. At this stage, reads are either tagged *mapped* or *unmapped*, depending on their alignment status. Some reads may map to multiple locations, which are defined as *multi-mapped*.

In scRNA-seq, genome alignment is usually the method of choice, since it is more accurate than transcriptome mapping [121]. Aligners originally designed for bulk RNA-seq are commonly-used for scRNA-seq, with STAR [122] (or its recent single-cell version STARsolo [113]) among the most popular. These aligners are both fast and able to handle splice junctions.

1.3.1.3 Feature assignment and quantification

The next goal is to assign reads to known genomic features, and estimate the expression levels of each of them. Read alignment results alone are meaningless as long as they have not been associated to some known genomic features (which are genes for most scRNA-seq analyses). Feature assignment (a.k.a. *feature annotation*) precisely refers to connecting each read to a feature depending on its mapping location, as schematized in Figure 1.9. Reads that map in a region where multiple genes are annotated are defined as *ambiguously assigned reads*. Quantification refers to counting the total number of reads that fall in a given region, which is also called the *read coverage*. Popular quantification tools are featureCounts [123] and htseq-count [124].

Feature assignment is carried out using a reference annotation file. The regular file formats for any reference annotation are GTF and GFF3. They are also referred to as *gene models*. These files gather the coordinates of all the regions of interest of a genome (*e.g.* exon, CDS, transcript, 3'UTR, etc.).

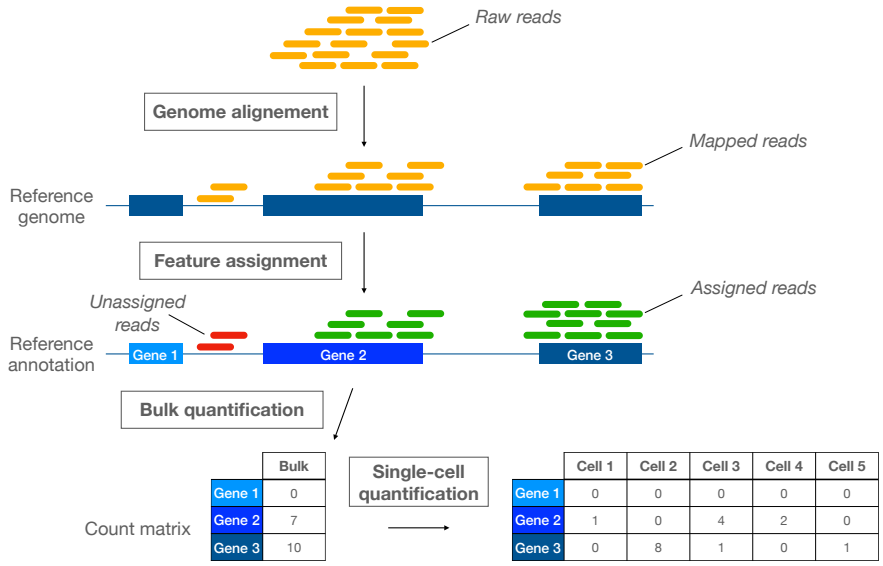


Figure 1.9 Scheme of read processing during scRNA-seq primary analyses. Raw reads (in orange) are first aligned to a reference genome (in blue). Their position is defined by the genome region with which they share the highest similarity. Some reads can be left unmapped if no such region could be identified (not shown). It is followed by the feature annotation step (a.k.a. feature assignment). The mapped reads are thus classified into assigned reads (in green) if they could map to some known feature (*e.g.* a gene) or unassigned reads (in red) if not (*e.g.* intergenic region). Unassigned reads are usually filtered out at this stage. Assigned reads are clustered depending on the feature (*e.g.* gene) they are associated to. All clusters are then quantified, and a total read count number is attributed to each feature. These results are stored in the expression matrix. In this illustration, we see that no reads could be assigned to *Gene 1*, thus leading to a zero-count in the expression matrix. *Gene 2* and *Gene 3* have been counted 7 and 10 times respectively (bulk counts). For single-cell analyses, the reads then need to be splitted depending on the cell they belong to (not shown). This results in a single-cell expression matrix.

The choice of a gene model can have tremendous impact on the quantification step, thus on the resulting count matrix. Details on gene models and their role in RNA-seq analyses is covered in more details in section 1.4.

For bulk data analysis, feature assignment and quantification are performed simultaneously. However, most scRNA-seq protocols (*e.g.* droplet-based protocols) require a supplementary step which consists of separating the count of each gene within each cell, as shown in Figure 1.9, and handling

UMI counts potential errors. Standard quantification estimates the overall number of gene within all the dataset, thus missing the cell information. Tools such as UMI-tools [86], zUMIs [125] or umis [74] are designed to handle this additional task.

1.3.1.4 All-in-one scRNA-seq pipelines

To fulfill the requirements of scRNA-seq data primary analyses, a dozen of pre-processing pipelines have been developed these last three years. While well-based protocols can accommodate most bulk RNA-seq pipelines, droplet-based protocols impose extra processing to handle UMIs and assign cell barcodes, as mentioned above. The choice of the pipeline depends on multiple criteria, such as:

- The experimental protocol used to generate the data;
- Mean read throughput;
- Total number of cells;
- Computational resources available;
- The quality of the references (*e.g.* when there is a possibility between aligning the reads on the genome or on the transcriptome - see Chapter 2);

Some pipelines are designed for one particular protocol, such as Drop-seq-tools [64] (dedicated to the eponymous data) or CellRanger which is a fully automated tool and proprietary solution specifically designed for 10x Genomics datasets. Other end-to-end pipelines, such as dropEst [126], scPipe [127], zUMIs [125] or Alevin [128] tend to simplify and harmonize this process. Pipelines based on tools such as umis [74] or UMI-tools [86] offer more flexible options as they can be included into a user-defined pipeline.

A comprehensive comparison of multiple scRNA-seq pre-processing pipelines have been performed only this year by *Gao et al., 2021* [129]. As shown in Figure 1.10, their performances in terms of run time and computing resources differ significantly. *Gao et al.* also demonstrates that these pipelines produce important variability in the count matrix content and data analysis quality,

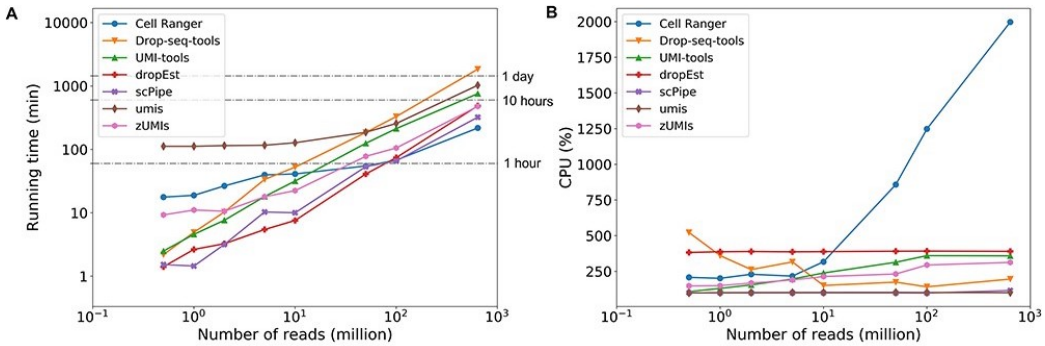


Figure 1.10 Computational performance comparison of 7 scRNA-seq pre-processing pipelines. These performances have been calculated with a standard 10x Genomics dataset of 10K cells (which comprise 640 million reads). All pre-processing steps but the alignment have been performed. A) Total run time. We can observe that the run time increases with the number of reads for all pipelines. In terms of run time, some tools might be recommended for smaller datasets (*e.g* dropEst, Drop-seq-tools, UMI-tools) while others would be more efficient for bigger datasets (*e.g* CellRanger, zUMIs). B) Maximum CPU usage. CellRanger demonstrates very high level of CPU usage beyond 10 million reads. Except for this latter, CPU usage is rather stable, with UMI-tools and zUMIs showing a slight increase and Drop-seq-tools decreasing with the number of reads to process. Image adapted from *Gao et al., 2021* [129]

thus potentially affecting the biological conclusions.

As mentioned above, all the scRNA-seq datasets we used for this thesis are produced with the 10xGenomics single-cell technology, which can be pre-processed with the proprietary tool CellRanger. However, at the beginning of my PhD thesis, we struggled at identifying the inner steps of this software as it was, and still is, a "black box", and the number of parameters we could adjust was unsatisfactory (specially in case of doubts regarding the data quality). Nevertheless, CellRanger have benefited from major developments within these last years (*e.g.* cell barcode and UMI correction), thereby becoming a standard in the single-cell community, despite its heavy computational requirements and lack of access to source code.

1.3.2 Secondary analyses: retrieving biological signal

Secondary analyses start once scRNA-seq raw data have been processed and summarized in the expression matrix. It describes all the steps that aim at ensuring that the data quality is sufficient (*data cleaning*) and answering biological questions (*downstream analyses*). While all preceding steps required dedicated computing infrastructures (*high performance computing*, or HPC) and followed each other in a linear manner, secondary analyses, on the contrary, are neither necessarily linear, nor they impose the use of HPC. The computing requirements are highly dependent on the chosen tools and steps to process, and may differ from one dataset to another. One of the most challenging tasks here is undoubtedly to select the right tools and approaches that best tackle the biological questions, while accounting for all the biases inherent to scRNA-seq (see section 5.2.1). Secondary analysis process broadly includes:

- **Data cleaning**: In order to exclude low-quality cells and potential outliers, the expression matrix first undergoes diverse quality control and filtering steps (this step may also be called *data processing*). It also includes normalization, data correction, feature selection and dimensionality reduction to prepare the data for downstream analysis in the best possible way.
- **Cell assignment**: It aims at aggregating cells into groups that share a similar expression profile. This is commonly achieved through clustering (unsupervised), classification (supervised) or trajectory reconstruction (time ordering of the cells).
- **Gene identification**: Its purpose is to identify the genes that best describe and differentiate the previously defined groups. It mainly includes differential expression and signature extraction.

In this section, I will briefly outline all the above-mentioned steps. A couple of them will be further described in more details in Chapter 3 (such as data correction).

1.3.2.1 Cleaning the expression matrix

Quality control Quality control (or QC) is critical to identify and filter out individual cells that might be of insufficient quality for downstream analyses. It is often performed gradually, since determining a cell's quality *a priori* is challenging and highly depends on downstream analysis assessment (*e.g.* clustering and differential analysis performance). Thus, permissive QC thresholds allow first to investigate data quality in a comprehensive manner, and more stringent thresholds are beneficial to specifically isolate high-quality populations [130]. Low-quality cells, if not removed, can lead to misleading results (*e.g.* some may form a cluster of their own which could be mistaken with an unknown cell type) or could contribute to add non-negligible noise to the data (which is already noisy by definition, see 5.2.1) [131]. Poor-quality cells include:

- Dying or damaged cells;
- Multiple cells that have been processed together, resulting in *doublets* or *multiplets* (instead of *singlets*) [132];
- *Empty droplets* or *empty wells*, which describe a mix of ambient mRNAs that have been sequenced but do not originate from a single cell [133, 134].

Cell quality is defined by a combination of multiple criteria, the most widely used being the number of reads or UMI counts per cell, the number of individual genes per cell, the percentage of mitochondrial genes within each cell and the probability of a cell to define a doublet or an empty drop [130]. Assessment of these criteria allows to define adequate cutoff values to isolate good-quality from poor-quality cells (Figure 1.11). However, these thresholds require rather arbitrary settings and do not necessarily reflect the entire landscape of low quality cells [131]. For example, the higher the percentage of mitochondrial genes, the larger is the probability that the cell is dying or damaged. Yet, this threshold varies depending on the cell types and species [135], and as we shall see below, on annotation quality and assignment strategy. Some packages attempt to remedy these deficiencies by automating the quality filtering process by relying on machine learning data-driven

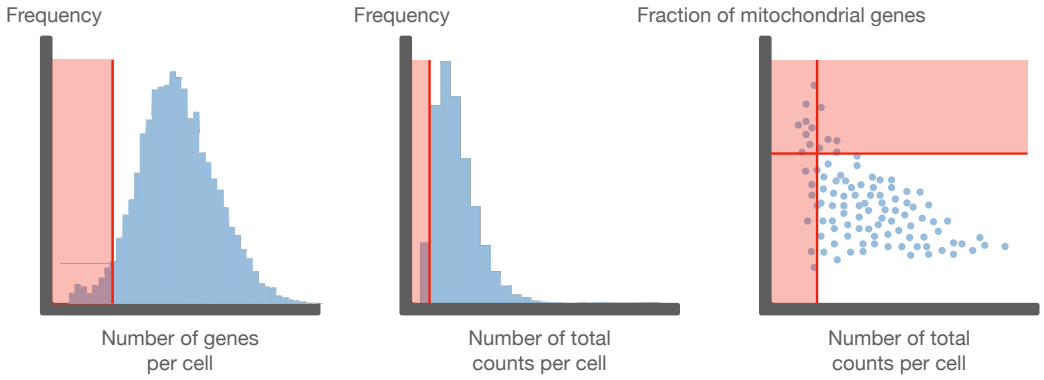


Figure 1.11 Illustration of standard scRNA-seq QC metrics plots. From left to right, the plots show respectively the distribution of the number of genes detected per cell, the distribution of the total counts (UMI or reads) per cell, and the fraction of mitochondrial genes versus the total counts per cell. The cells that fall into the red zones are filtered out, once a threshold (red line) has been set. The thresholds are usually chosen based on a combination of i) the expected distributions, ii) the knowledge of the biological data (*e.g.* mitochondrial genes proportion may vary widely between two datasets) and iii) other levels of QC metrics that complement these plots and steer decision making. The two histograms are extracted from *Luecken et al., 2019* [130].

approaches, such as the cellity package [131] or miQC [136]. Other packages offer a more exploratory approach, mostly based on visual inspection of the data, such as Seurat [137] or scater [138].

Doublets are commonly separated into two categories: homotypic (formed by transcriptionally similar cells) and heterotypic (formed by transcriptionally distinct cells) [139]. Heterotypic doublets are usually easier to detect due to their distinct gene expression profile. Moreover, some of the doublets can be reliably detected by depicting cells with an unexpectedly high number of counts or unique genes. Nevertheless, these criteria may also be representative of rare cell types or states, and other doublets may have a particular expression profile that will go unnoticed. Thus, more sophisticated approaches to detect doublets have been developed, such as DoubletDecon [140], Doublet Finder [141] or Scrublet [142]. These approaches rely on algorithms that mix the expression profiles of two randomly selected cells, in order to artificially create doublets. They then compute a score to assess the similarity between any given cell and the artificial doublets. Nine of these methods have been

reviewed recently in *Xi and Li, 2021* [132].

Feature selection Among the (ten of) thousands of genes that are routinely retrieved in a scRNA-seq experiment, many are uninformative (*e.g.* housekeeping genes) or too lowly expressed to be meaningful (*e.g.* genes detected just once within a couple of cells among the thousands available in the dataset). Moreover, the more genes there are in a dataset, the more there are dimensions to be handled. Reducing the number of dimensions, which is otherwise called *dimensionality*, highly contributes to improving computing speed. Thus, feature selection is key to reduce noise and ease the computational burden on downstream analysis [130].

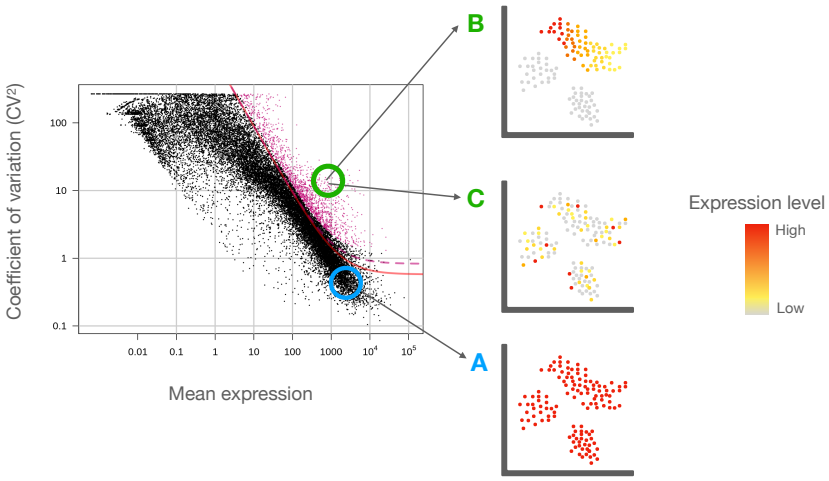


Figure 1.12 Feature selection of highly variable genes. Example of a feature selection approach, by determining the relationship between the squared coefficient of variation and the mean expression of each gene (left-most plot). Black dots represent the non-variable genes that will be filtered out of downstream analyses (example gene in blue). Pink dots are the selected HVGs (examples in green). A) The gene represented here is highly and evenly expressed in every cell. It is thus little informative, unless to be used as a positive control. B) A specific cell population is described by the medium-to-high expression of this gene. It is thus differentially expressed, and should be kept for downstream analyses. C) This gene does not seem to be representative of a clear biological pattern, but its variability among all cells enforce its preservation for downstream analyses. It might be representative of a hidden effect, and more investigation is needed. Left-most plot is extracted from M3Drop vignette https://www.bioconductor.org/packages/release/bioc/vignettes/M3Drop/inst/doc/M3Drop_Vignette.pdf.

Feature selection aims at reducing the initial gene set to a subset of representative genes (typically between 1000 and 5000) [130]. It has been demonstrated that the way they are selected and the number of selected genes can have a significant effect on downstream analyses [143, 144]. One of the most popular approaches is to select *Highly Variable Genes* (HVG), that reflect the variability within a given dataset [145] (Figure 1.12). Diverse methods have been developed to select HVGs. A popular one is based on the estimation of the ratio between variance and mean, as implemented in Scanpy [146] or Seurat [137]. Seven HVG approaches are reviewed in *Yip et al., 2019* [147]. Alternative feature selection approaches, such as M3Drop [148], rely on identifying genes with an unexpectedly high number of zeros (a.k.a. high *dropout-rate*), since it has been shown that dropout-rates are interrelated with gene expression level [149, 150]. *Sheng and Li, 2021* recently reviewed in a comprehensive manner 17 feature selection approaches [151].

Normalization Similarly to bulk approaches, data normalization is essential to ensure samples (or cells) and genes are comparable with each other. However, due to counts variability and instability between cells (see 5.2.1), scRNA-seq normalization is much more challenging. Thus, single-cell normalization first and foremost needs to take into account that two identical cells might exhibit different sequencing depths. The purpose of library size normalization is precisely to scale count data to correct for differences in gene expression counts in between cells [152]. Historically, the first normalization methods applied to scRNA-seq were mere adaptations of bulk techniques [153], such as Trimmed Mean of M-values (TMM) [154] or DESeq/DESeq2 [155, 156]. Moreover, these techniques are based on the assumption that most of the genes are not *differentially expressed* (DE) between samples, which might compromise single-cell analyses in which the variability and sparsity are substantial. Thus, in order to account for the many biases inherent to single-cell, the need for specific normalization techniques quickly emerged [157]. These effects are summarized in Figure 1.13).

Depending on the scRNA-seq protocol used to generate the data, single-cell normalization needs distinct adjustments. For plate-based protocols

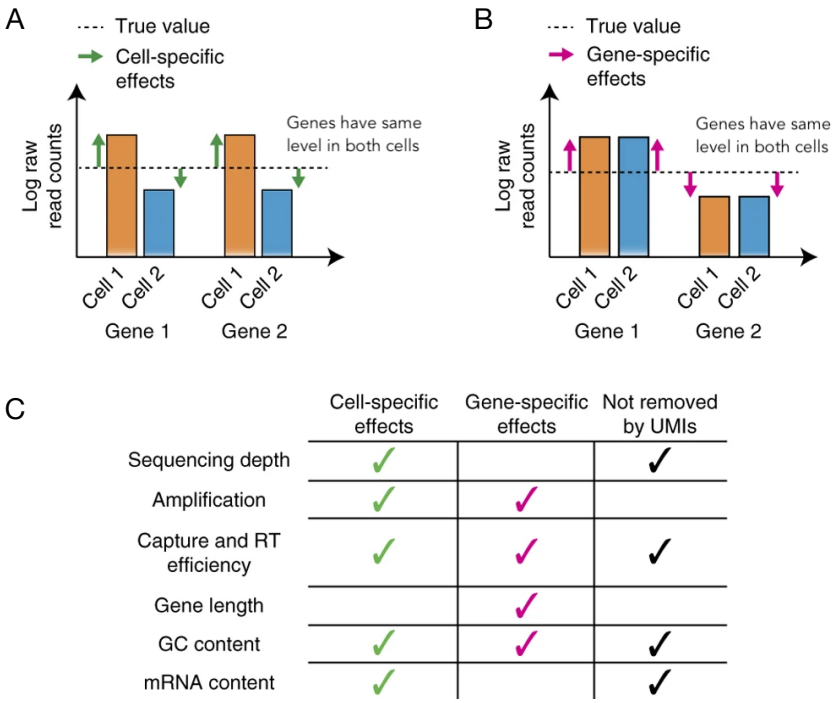


Figure 1.13 Illustration of cell and gene specific effects in scRNA-seq. A) and B) show two cells that each express two genes at equivalent levels. A) Schematic illustration of cell-specific effects. B) Schematic illustration of gene-specific effects. C) Summary of the main gene or cell specific effects. Last column indicates if they are removed by UMIs or not. Image adapted from *Vallejos et al., 2017* [153].

(full-length mRNA sequencing), correcting for gene length is critical, as is the case with bulk RNA-seq. Early full-length normalization methods also included external synthetic RNA sequences, or *spike-ins* [145, 158, 159]. Knowing in advance their precise amount allows to infer changes in expression of endogenous genes that would be solely due to technical variability [160]. More recently, approaches such as ISnorm [161], suggest to use internal spike-ins (*e.g.* small fraction of constantly expressed genes) as a more accessible and straightforward way to quantify variability and normalize scRNA-seq datasets accordingly.

For UMI-based protocols, it is meaningless to account for gene-length, since reads originate specifically from the transcripts extremities [162]. However, due to higher sparsity and instability, many specific approaches have

been developed, scran [157] being one of the most popular. The scran approach relies on pooling transcriptionally-alike cells in order to generate pseudo-bulk scaling factors among these small sets of cells. This approach allows to overcome the great number of dropouts when considering cells as a unique unit. It has been shown that the scaling factors estimates are more accurate when applying this strategy [153, 157]. Several comprehensive benchmarks and reviews compare the performances of single-cell dedicated normalization techniques [153, 163, 164]. Correcting for biological effects (*e.g.* cell-cycle, gender) or other technical biases (*e.g.* batch effect) is further detailed in Chapter 3.

Dimensionality reduction and visualization In order to properly visualize a scRNA-seq dataset on a 2D representation, the dimensions of the count matrix need to be reduced and optimized through dimensionality reduction (DR) approaches. It aims at finding the best possible representation of the dataset, by capturing the underlying structure within the minimum number of dimensions [165]. It is particularly suited for scRNA-seq data analyses, since it has been shown that scRNA-seq is inherently low-dimensional (*e.g.* most of the relevant information is contained in the first dimensions of the dataset) [166].

Ten DR approaches have recently been reviewed and applied to scRNA-seq in *Xiang et al., 2021* [168]. They can be divided into three categories: linear (*e.g.* PCA [169], ICA [170]), nonlinear (*e.g.* t-SNE [171], UMAP [172], diffusion maps [173]) and machine learning-based methods (*e.g.* SIMLR [174], ZIFA [150], SPRING [175]). The most popular are t-SNE and UMAP, since nonlinear techniques are best suited to avoid populations overlapping [167]. They both efficiently reveal local data structure, however, contrary to t-SNE, UMAP is able to preserve inter-cluster relationships (a.k.a. global data structure), as shown in Figure 1.14. Moreover, UMAP can easily scale to large datasets, making it the current most convenient tool for exploratory data visualization [130, 168].

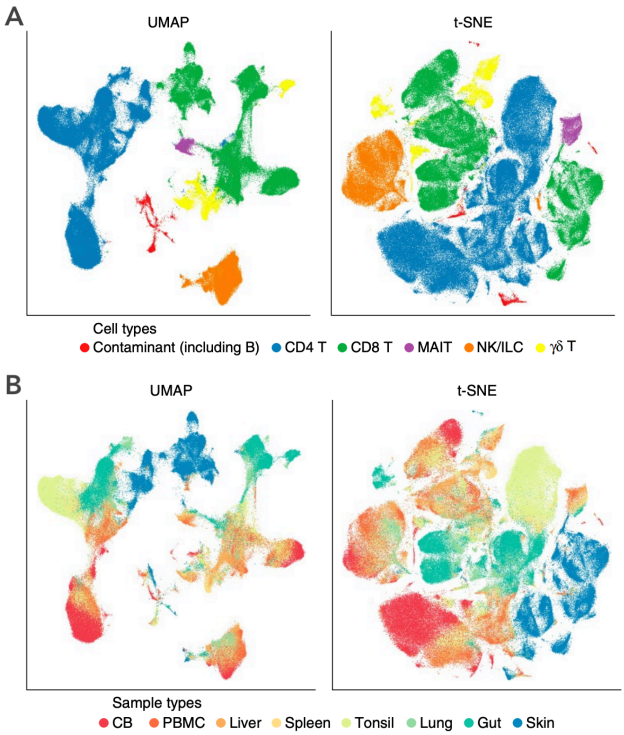


Figure 1.14 Dimensionality reduction and visualization with t-SNE versus UMAP. Cells are colored according to A) cell type and B) tissue of origin. In this specific example, we can observe that UMAP is more efficient to separate cell populations by cell type, while t-SNE is more likely to split cells depending on their tissue of origin. Image adapted from *Becht et al., 2019* [167].

1.3.2.2 Cell assignment

Clustering Grouping cells into clusters of similar expression is one of the key steps of scRNA-seq data analyses. It often stands as a first intermediary result [130]. Clustering mostly aims at assigning an identity to each cell. While clustering in itself is a vast research field in statistics [176], over a hundred approaches have been specifically developed for scRNA-seq data in order to account for the technical and biological challenges inherent to single-cell approaches [112]. Similarity between cells is determined with *distance-based metrics* (such as Euclidean distances) or *correlation-based metrics* (such as Pearson's correlation) [177].

Clustering techniques can be divided into three broad approaches:

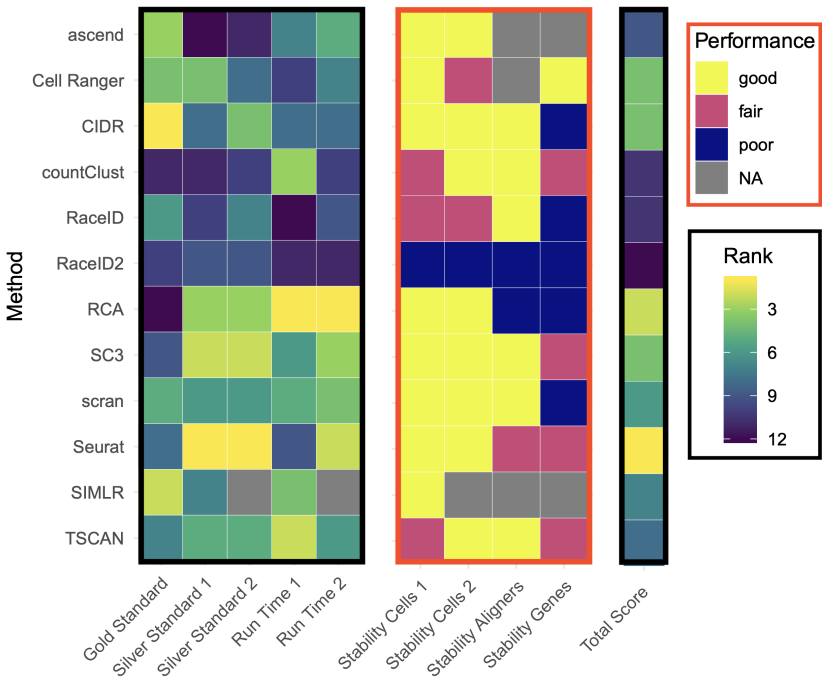


Figure 1.15 Summarized performances of twelve scRNA-seq clustering techniques. Several performance metrics have been specifically established to compare clustering approaches (see *Freytag et al., 2018* for details on these metrics). Overall, Seurat demonstrates the best results, while RaceID2 is the one that performs the worst. Image adapted from *Freytag et al., 2018* [178].

- Partitional methods;
- Hierarchical methods;
- Graph partitioning methods (a.k.a. *community detection* methods).

Under this definition, partitional and hierarchical methods mainly focus on a unique modality (*e.g.* distance matrix), while community detection also relies on a network structure (*e.g.* cells are the nodes of the graph and their relationships are the edges) [179]. Several popular approaches have been reviewed and benchmarked by *Freytag et al., 2018* (Figure 1.15) [178].

Partitional approaches, among which the popular k-means algorithm, have been first applied to scRNA-seq datasets. However, results greatly vary depending on the similarity metrics used [130]. Since correlation-based distances are invariant to data scaling (contrary to distance-based metrics), it has recently been shown that correlation-based metrics might improve

clustering, especially for droplet-based data [177].

Hierarchical clustering allows to establish a hierarchy among clusters. It can be performed with a *top-down* approach (a single cluster is sequentially divided into smaller groups) or a *bottom-up* approach (each individual cell identifies as a cluster, and these clusters are then sequentially merged as larger groups). Even though hierarchical clustering has largely been applied to scRNA-seq data, it is limited to small datasets or requires a specific selection of the cells, since it computationally scales poorly with large datasets [180].

Community detection methods rely on a graph obtained with a *k-Nearest Neighbours* approach (KNN graph). Each cell is thus connected to its k most alike neighbours. Since the search space is greatly reduced (only neighbouring cells are compared), this approach is often more computationally-efficient than partitional and hierarchical methods [180]. Graph-based approaches are best represented by the Louvain algorithm [181] as implemented in the Seurat [137] and Scanpy [146] workflows. Several benchmarks have demonstrated that this approach is the best suited for scRNA-seq datasets [182, 183].

Classification When comprehensive references (such as cell atlases) are available, automatic cell classification offers a powerful alternative to clustering. Cell classification takes advantage of previous knowledge and prevents manual annotation or user-defined cell types. Tools such as CaSTLe [184], scmap [185] and scPred [186] are among the most popular ones. Although classification tools emerged later than the other scRNA-seq analyses categories, many methods have since been developed. Some of them have recently been reviewed in *Abdelaal et al., 2019* and *Zhao et al., 2020* [187, 188].

The two major drawbacks of such approaches lies in the fact that it totally prevents novel cell type discovery, and the cell type identification is as good as its annotated reference. These approaches are thus predominantly recommended for well-studied cell types and species. *Lin et al., 2021* (scAL) [189] and *Ranjan et al., 2021* (scConsensus) [190] suggest combining both classification and unsupervised clustering to improve cell type identification.

Trajectory inference Considering that scRNA-seq offers an unprecedented view on an infinite rainbow of cellular states, grouping cells into discrete states might appear as too restrictive in some cases. Specifically, *trajectory inference* (a.k.a. *pseudo-temporal ordering*) allows to study continuous processes such as cell differentiation and progression along lineages, which is particularly relevant in developmental biology [191]. Introduced by *Trapnell et al., 2014* with their tool Monocle [192], this approach aims at inferring optimal paths by minimizing transcriptional changes between cells. Starting from a root cell, neighbouring cells are ordered on a fictional temporal axis (a.k.a. *pseudo-time*) that serves as a proxy for developmental time (Figure 1.16) [193].

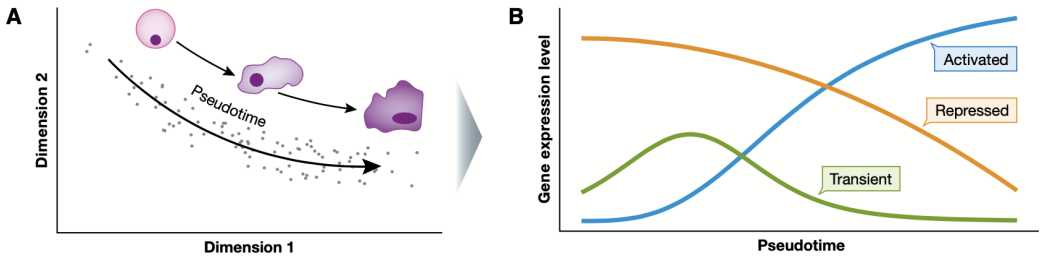


Figure 1.16 Pseudo-temporal ordering is a proxy for developmental time. A) Once the transcriptional changes between successive cell pairs are identified and ordered, a seemingly homogeneous group of cells can be turned into a continuum along a pseudo-temporal axis. B) Significant changes in expression along the pseudo-temporal axis can be imputed to some specific genes. Image from *Griffiths et al., 2018* [194].

Much like clustering approaches, trajectory inference (T.I.) benefited these last years from a noticeable rise in methodological developments. Over a hundred tools specifically dedicated to T.I. have been catalogued in scrna-tools.org as of 15 September 2021. Some 45 of them have been benchmarked in the highly comprehensive study of *Saelens et al., 2019* [195] (Figure 1.17). They mostly differ by the trajectory topologies they infer (*e.g.* linear, tree-like, cyclic...), and there is no single approach that fit all topologies. Authors advise to test multiple approaches, with at least one with free topology in order to take informed decisions before assigning a topology to a dataset.

Trajectory inference algorithms can be divided into two broad strategies: DR-based methods (such as Monocle [192], Wishbone [196] or Slinghsot

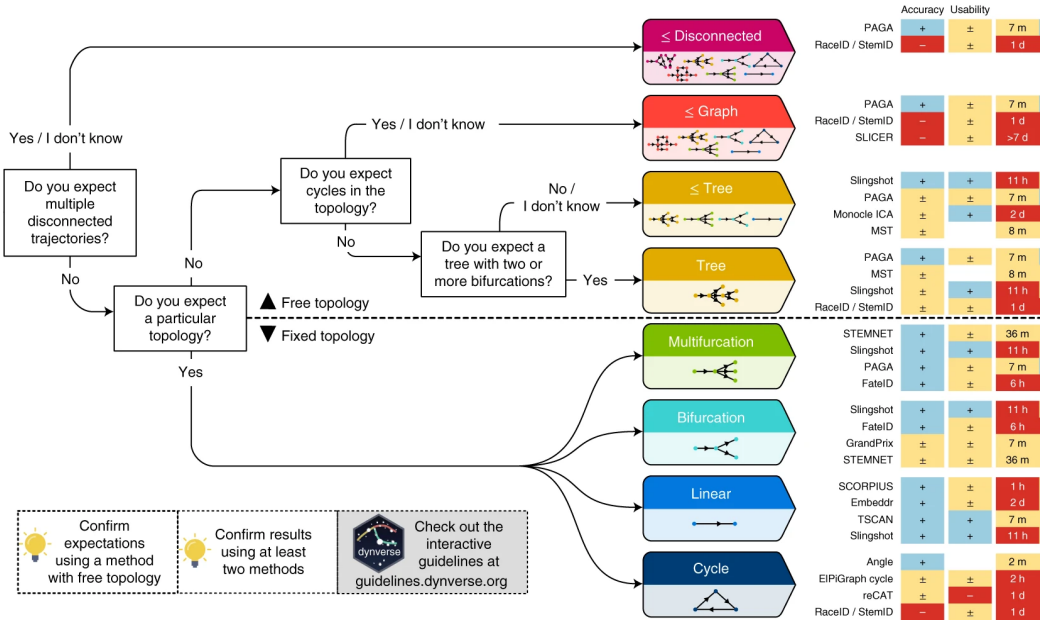


Figure 1.17 Decision graph and performance comparison of single-cell trajectory inference methods. Here are practical guidelines to choose the adequate T.I. approach, proposed by *Saelens et al., 2019*. The choice of a T.I. method depends above all on the user's previous knowledge of the system topology (*e.g. disconnected*, *graph*, *tree*, etc. - see colored boxes). Image adapted from *Saelens et al., 2019* [195].

[197]) and clustering-based methods (such as PAGA [198], Palantir [199] or TSCAN [200]). The former ones primarily rely on a reduced-dimensionality space before inferring connection between cells (such as the construction of a minimum spanning tree, as is the case for Monocle and Slinghsot). The latter ones first cluster the cells in a low-dimensional space before building a network which aims at connecting clusters centers [201].

The major challenges in inferring cells trajectories are to handle sparsity and the lack of synchronization among cells. Indeed, cellular processes are rarely totally synchronized at the cellular level. On top of that, the need to rely on previous knowledge to choose a topology makes the choice between all T.I. approaches a difficult task. The dynverse toolkit[5] offers a unique set of guidelines and embedded tools to perform and compare T.I. on any dataset with multiple methods [195]. More recently, a promising and innova-

[5] https://dynverse.org/

tive methodological development suggests to add directionality to single-cell trajectories thanks to RNA velocities (estimated by disentangling unspliced and spliced mRNAs) [202, 203]. This approach aims at facilitating the study of developmental lineages and cellular dynamics.

1.3.2.3 Gene identification

While the previous section covers a set of analyses that are specific to single-cell, gene identification is a common task to both bulk and single-cell approaches. It thus benefits from an already well-documented and largely investigated field [204]. It aims at identifying distinct expression profiles and the corresponding genes that are the key drivers of cellular heterogeneity. There are many ways to investigate gene-level diversity. They are usually divided into *differential expression* (DE) analyses, gene set and pathways analyses, and *gene regulatory networks*. Since it is the most widely used in scRNA-seq, only the former one will be addressed in this section.

The most popular approach in scRNA-seq is to compare cell clusters between themselves (a.k.a. *pseudo-bulk* approach). Although applied to scRNA-seq in the first instance, bulk differential expression approaches are not designed to handle the high technical noise specific to single-cell. Yet, a couple of robust bulk-dedicated methods are still applied to scRNA-seq datasets since they outperform (or perform at least as well) as single-cell dedicated approaches [205, 206]. This is particularly the case for DESeq2 [156] and edgeR [154]. Their performance can be further improved when accounting for gene-weights to better model single-cell data (tools such as ZINB-WaVE [207] are particularly recommended for this aim). A plethora of methods have been specifically developed for scRNA-seq DE analyses (thus taking into account single-cell biases), with a few standing out such as MAST [208], scDD [209] and SCDE [149]. A significant number has been comprehensively reviewed in *Soneson et al., 2018* (Figure 1.18) [205].

Aside from cluster-based DE analyses, T.I. analyses paved the way to deciphering the key drivers of changes in gene expression. The goal of trajectory-based DE methods is to identify the genes whose expression evolve

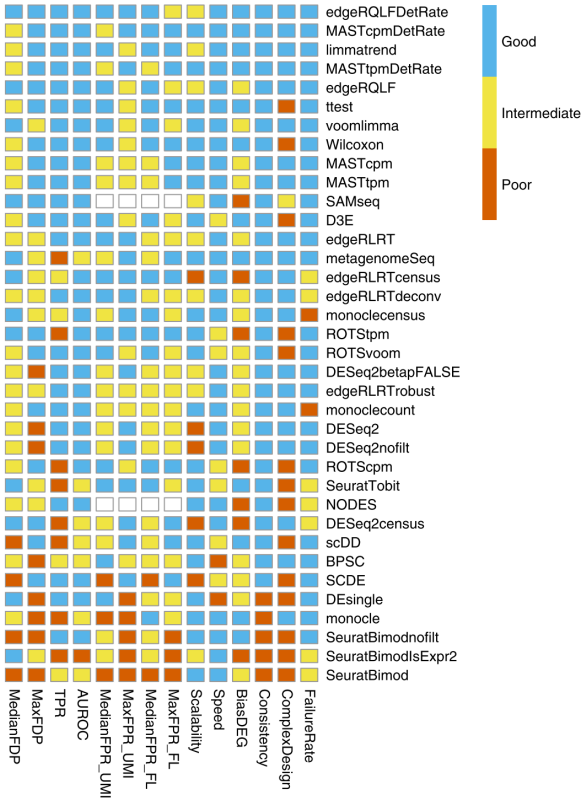


Figure 1.18 Performance comparison of 36 differential expression approaches. These DE approaches are ordered (from top to bottom) by their average performance according to multiple criteria. See online methods for details on evaluation criteria and cutoff. Image from *Soneson et al., 2018* [205].

according to the pseudo-time axis, instead of testing the differences between two groups. Monocle's BEAM [192] and tradeSeq [211] are among the most popular.

DE approaches mostly suffer from large discrepancies between the resulting gene lists, which consequently lead to many false-positive DE genes (Figure 1.19) [212]. Extracting meaningful information out of these gene lists is often a complex and time-consuming task, and must be carried out in close collaboration with experts of the considered biological domain.

Among all this profusion, a handful of tools stood out and became inescapable in the single-cell community. This is particularly true for "all-in-

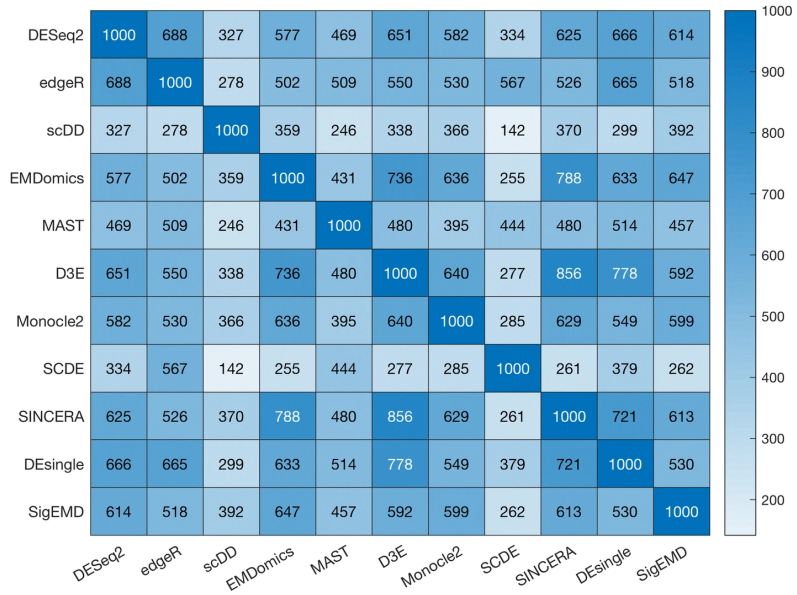


Figure 1.19 Pairwise comparison of the top 1000 DE genes identified by 11 popular DE approaches. The numbers reflects the amount of DE genes shared between various DE approaches. For each method, the top 1000 DE genes have been selected after being ranked by their adjusted p-values or FDRs. Common DE genes between pairs of tools oscillate between 142 (scDD versus SCDE) and 856 (D3E versus SINCERA). This comparison highlights a lack of consensus and agreement between all these approaches. Image from *Wang et al., 2019* [210].

one" toolkits such as Seurat, Monocle or scanpy. However these tools are rarely self-sufficient due to the necessity to adapt the analysis and tools to each single-cell analysis. On the one hand, this abundance led to an increased accuracy in the interpretation and deeper understanding of single-cell datasets. On the other hand, it also complicated the tasks of selecting and combining tools to answer specific questions. A main challenge is thus to select a range of adequate tools to apply in custom analyses. Although these last years have benefited from a certain harmonization in scRNA-seq data processing, there are still developments required to adapt our analysis pipelines to each data type (scRNA-seq protocol, organism, cell types...) and biological aims. Hence, no plug and play solution exists, and a good knowledge of the pros and cons of each tool coupled to a solid understanding of the constraints of the biological system under study are required to design the best *ad hoc* pipeline.

1.4 Reference genome assembly and annotation in RNA-seq analyses

As mentioned in 1.3.1, RNA-seq analyses heavily rely on what I will refer to as "biological references". These biological references encompass the reference genome assembly and the reference annotation. These references are decisive parameters in the generation of the count matrix. Any read that could not be mapped to the reference genome or assigned to a gene catalogued in the reference annotation is simply not taken into account, and thus constitutes a source of data loss. But how are these biological references produced ? What biases may they bring into the analyses ? Actually, these points are rarely questioned.

This section is intended to make readers aware of the inherent biases related to these biological references and the potential effects it can have on RNA-seq studies, with an emphasis on reference annotations. I will first describe the main characteristics of these biological references, and then describe the impact of the choice of the annotation on RNA-seq analyses.

1.4.1 Biological references

1.4.1.1 Reference genomes assemblies

Although this is not a major focus in this thesis, reference genomes usually go hand in hand with annotations, therefore I chose to take advantage of this section to say a few words about them. A reference genome (a.k.a. *genome assembly*) defines a representative example of the DNA sequence of a given species. Not all species have a reference genome, but whenever it is available, it plays a key role in RNA-seq projects by conferring coordinates to each sequencing read (*e.g.* location start and end on the genome).

Even though technological progress have been made since the first human genome sequence, the construction of an assembly remains a tedious and continuous process for higher eukaryotes. Due to the presence of sequence patterns that are tough to assemble (such as repeated elements), most assemblies are filled with gaps. With this respect, long-read approaches are

more and more used in order to improve assemblies completion [213, 214]. For example, the human genome was still incomplete until very recently. In May 2021, *Nurk et al.* announced they succeeded in achieving a gapless assembly (reference T2T-CHM13), which is thus the first complete sequence of a human genome, 20 years after the first draft [215]. This means that all previous analyses relied on an evolving and incomplete version of the genome [216].

1.4.1.2 Reference annotations

The process of delineating genes along the genome arose simultaneously with the first complete bacterial genome: in 1995, the DNA sequence of *Haemophilus influenzae* was provided with an annotation of 1742 protein-coding genes [217]. Since then, the definition of genome annotations has not changed: it is described as the process of identifying and placing all known landmarks into the genome [218]. To some extent, it also describes the process of coupling a gene with its functions (or potential functions), which rather refers to functional annotations. This thesis mainly focuses on genomic annotations, but functional annotations will be briefly mentioned further in Chapter 3.

As *Stein, 2001* stated, "*the value of the genome is only as good its annotation; it is the annotation that bridges the gap from the sequence to the biology of the organism.*" [218]. Therefore, the quality and accuracy of reference annotations have brought particular attention over the years. Just like assemblies, annotations evolve rapidly and are regularly completed with novel data. There are two main approaches to annotate a genome: manually or automatically (from sequencing data). There are huge collective efforts to provide high-quality annotations. Most reference annotations thus originate from consortia of researchers, which rely on specific and often distinct pipelines, such as Ensembl [219, 220], NCBI/RefSeq [221] or GENCODE [222]. This latter is part of the ENCODE (ENCyclopedia Of DNA Elements) project, which aims at developing a comprehensive map of functional elements in the human genome [223]. While manual annotation (HAVANA) is the cornerstone of GENCODE annotation process, automated gene predic-

tion is also used to complement manual curation [222]. The diverse ways to annotate a genome will be presented in more details in Chapter 4.

1.4.2 Impact of the annotation in RNA-seq analyses

1.4.2.1 In bulk

The choice of an annotation might have a dramatic impact on RNA-seq analyses [224]. However, this topic remains very little explored. To my knowledge, there are only two published studies that directly question this impact, that dates back from 2013 (*Wu et al., 2013*) [225] and 2015 (*Zhao et al., 2015*) [226]. More recently, two preprints target this impact with a focus on quantification (*Chisanga et al., 2021*) [227] and differential expression (*Hamaguchi et al., 2021*) [228]. Furthermore, all of these four papers solely focus on human bulk data. Other studies explore the impact of each step of the RNA-seq workflow, including the comparison of different annotations, such as *Simoneau et al., 2020* [229] in bulk (human). As an illustration of how reference annotations might differ, I compared the overlaps between the three main human annotations (Figure 1.20).

The difficulty about choosing a gene model lies in the very complexity of defining quality standards. No gene model is perfect and there is no ground truth available to be compared to. For example, in order to compare three reference annotations, *Hamaguchi et al., 2021* define the "mappability" criteria, a metric of the complexity of gene annotation, as the fraction of reads that originate from a transcript (in the dataset) that truly ends up mapping to the original transcript (annotated in the reference) [228]. *Wu et al., 2013*, in turn, define the complexity of a genome annotation in terms of the number of features (*e.g.* genes, isoforms, exons) it contains [225].

All of the above-mentioned papers reached two identical conclusions:

- The selection of a gene annotation over another results in discrepancies in gene expression quantification, which therefore propagates in downstream analyses such as differential expression;
- The more an annotation is rich and complex, the more it negatively impacts RNA-seq analyses. *Hamaguchi et al.* thus suggest to exclude

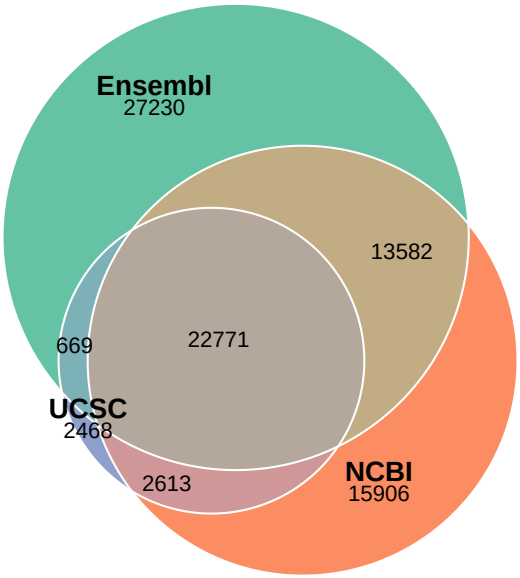


Figure 1.20 Venn diagram of the overlap between three human reference annotations. Comparison of the overlaps and intersections among three reference annotations from Ensembl (in green), NCBI/RefSeq (in orange) and UCSC (in blue) from the latest human reference hg38 (downloaded from https://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/genes/ the 15th September 2021.). An overlap is defined as follows: if a gene in reference A spans at least 50% of a gene in reference B, and vice versa. It only reports overlaps if the genes are on the same strand. It shows that 22771 genes are shared between all three annotations. Ensembl represents the largest annotation, with 27230 unique genes, while NCBI and UCSC gather respectively 15906 and 2468 unique genes. It must be noted that I have deliberately chosen to take unfiltered annotations for this comparison. They may therefore include both coding and non coding genes, and pseudogenes, which explains the high figures (specially for Ensembl).

> unnecessary gene models from reference annotations [228]. In the same vein, *Zhao et al., 2015* and *Wu et al., 2013* recommend to use a low-complexity annotation if the aim is to provide a robust and reproducible analysis, while a higher-complexity should be preferred if the emphasis is put on characterizing novel transcriptional or regulatory mechanisms [225, 226].

This latter point is mainly attributable to the increase of multi-mapped reads in the presence of a high-complexity annotation. The vast majority of RNA-seq analysis pipelines discard the multi-mapped reads in the early

steps. However, there is a rising debate on whether or not the multi-mapped reads should be kept, and on the multiple ways to handle them [230].

1.4.2.2 In single-cells

In single-cell analysis, a preprint from *Brüning et al., 2021* questioned the impact of using various popular pre-processing workflows on downstream analyses, with a small section on the differences between two annotations: one unfiltered (which includes the complete Ensembl GTF), and the other one filtered (as recommended by 10x Genomics, which exclusively includes protein coding genes and lncRNA) [119]. To my knowledge, there is no other paper that directly addresses this issue within a benchmark. However, it solely focuses on already well-annotated species (mouse and human).

Among all the scRNA-seq workflows available, only Alevin keeps the multi-mapped reads [128]. It processes them by equally dividing the counts to all potential mapping positions. *Brüning et al., 2021* precisely showed that multi-mapped reads have a major impact on scRNA-seq analysis [119]. Interestingly, they are the first to demonstrate that using a high-complexity annotation reduces the fraction of mitochondrial (MT) genes estimated in each cell, a key feature to evaluate cell quality in scRNA-seq. They assume that the reduced MT-genes content could be explained by the reads increased tendency to map to several locations in the context of high-complexity annotations. On the contrary, a lower-complexity annotation would thus overestimate MT-gene expression. They suggest that future mapping tools should prevent multi-mapping by considering the likelihood of a gene to be expressed in a given cell type (which would lead to a kind of cell-type specific gene annotation).

Nevertheless, it is not clear how reference annotations differences affect RNA-seq when applied to poorly-annotated species (*e.g.* more or less any species other than human or mouse). For example, a study on Zebrafish from last year (*Lawson et al., 2020*) stands out as an exception, as it compares the output of bulk RNA-seq with two reference annotations (Ensembl and RefSeq) on one side, and estimates the gain of an improved annotation

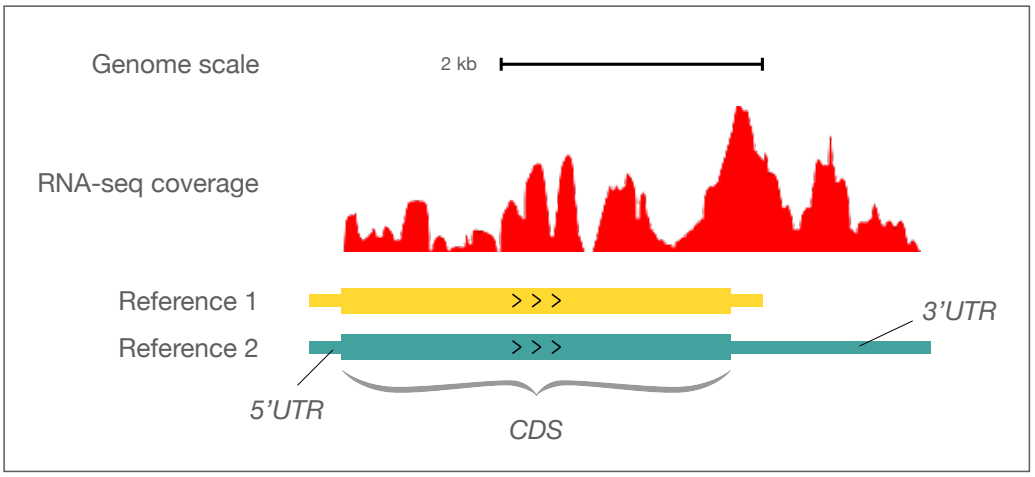


Figure 1.21 Illustration of a genome data viewer with diverging gene models. The RNA-seq coverage track, in red, represent the sum of all the sequencing reads that fall into a given window. This track is extracted from *Zhao et al., 2015*, Figure 6 (bulk signal) [226]. The gene model from Reference 1 (in yellow) has a much shorter 3'UTR than the gene model from Reference 2 (in green). All the rightmost RNA-seq signal will be lost if the quantification is carried out with Reference 1.

on scRNA-seq analysis on the other side [231]. Unlike previously-mentioned papers, *Lawson et al.* conclude that the discrepancies observed in RNA-seq analyses are mostly due to both the length variations of the 3'UTRs, and the absence of thousands of genes in each annotation. To illustrate the impact of different gene models to quantify RNA-seq reads, I made a schematic representation of a genome browser with diverging gene models in Figure 1.21. If incomplete 3'UTR gene models have been shown to have an impact in bulk RNA-seq analysis, it has barely been a subject of investigation in scRNA-seq. What is more, the widely-used 10x Genomics scRNA-seq technology specifically targets 3'UTR; we can thus already hypothesise that incomplete 3'UTR may impact the quantification of scRNA-seq signal.

Therefore, similarly to bulk data, an accurate and appropriate reference annotation is paramount to rigorously quantifying scRNA-seq data. It is thus essential to appreciate how and to what extent a given annotation impacts scRNA-seq analyses, specially when working with poorly-annotated species.

1.5 PhD overview and aims

This computational biology project capitalizes on a tight collaboration with the experimental team of Xavier Morin (IBENS), in the context of the ANR project "SYMASYM". This project aims at identifying molecular mechanisms that control whether a neural progenitor cell will enter a symmetric or asymmetric mode of division (see Chapter 3 for more details). In the context of this project, scRNA-seq datasets were produced with the 10x Genomics platform, from chicken cervical spinal progenitors at 66 hours of embryonic development. However, due to issues we encountered with the chicken genome annotation that greatly impeded our scRNA-seq analyses (see Chapter 4), we decided to set up a re-annotation analysis pipeline dedicated to scRNA-seq data. Actually, the chicken genome annotation is poorly annotated compared to the annotation of other model organisms like human, mouse or drosophila, that have benefited from large collaborative efforts such as ENCODE [223].

In this regard, I aimed to address the following general questions:

- How much an incomplete reference annotation affects scRNA-seq data analysis in a poorly-annotated species, and how much information can be recovered with a project-specific annotation ?
- How to ensure scalability, reproducibility and traceability of scRNA-seq 10x Genomics analyses ?

Once the scRNA-seq data are reliably processed, it becomes possible to proceed to the downstream analyses, that aim at answering biological questions. In the context of the ANR SYMASYM project, we are interested in the mechanisms controlling the neurons production rate from neuroepithelial progenitors, within the vertebrate central nervous system. The progenitor pool is first amplified via proliferative symmetrical divisions (which produce two progenitors, thereafter called SYM divisions), progressively switches to neurogenic asymmetric divisions producing a progenitor and a committed progeny (ASYM), and finally to symmetrical terminal divisions producing two differentiating neurons (TERM). Much remains to be discovered about the molecular and cellular mechanisms underlying the decision to enter an

asymmetric division, and its execution. The main biological focus is thus the search for transcriptomics signatures of SYM versus ASYM division mode in neuronal progenitors, using scRNA-seq datasets from chicken and mouse.

For this part, my aim is to address these questions:

- Can we identify changes in expression that reflect the progressive transition from the SYM to the ASYM state out of scRNA-seq data ?
- If so, what are most differentially-expressed genes between SYM and ASYM populations ?

Altogether, my thesis aims at evaluating, developing and applying bioinformatics approaches for scRNA-seq to study the neurogenic transition in vertebrate neural progenitors. My results are organised as follows:

- **Chapter 2:** Development of an automated pipeline dedicated to the pre-processing of 10x Genomics scRNA-seq data, in a scalable and reproducible manner.
- **Chapter 3:** Application of scRNA-seq analysis approaches to various neural progenitors datasets (mouse and chicken), in order to isolate and uncover differences in gene expression between SYM and ASYM populations.
- **Chapter 4:** Design of an hybrid approach to process scRNA-seq from poorly-annotated species, by building a project-specific annotation based on long-read transcripts, the scRNA-seq signal and the reference annotation.

Chapter 2

Eoulsan 2: automated scRNA-seq pre-processing pipeline

In this chapter, I present a manuscript introducing the workflow manager *Eoulsan 2*. It has been specifically designed to process huge amounts of sequencing data and support their analyses in an efficient and reproducible manner. First developed by the IBENS genomic platform and introduced in 2012 [232], it has since benefited from numerous improvements, in particular to handle more recent data types such as scRNA-seq and bulk long-read RNA-seq. During my PhD thesis, I developed the pipeline dedicated to 10x Genomics scRNA-seq analyses.

Following a few words on our motivations to build such a pipeline, I will introduce some additional concepts on 10x Genomics data pre-processing, necessary to understand my work. This chapter will end with results of the pre-processing of the chicken scRNA-seq dataset produced for the SYMASYM project.

2.1 Motivation

As mentioned in 1.3.1, at the beginning of my PhD thesis we noticed that using CellRanger was limiting, especially in case of low-quality data. This is explained by the fact i) that the number of parameters was, and still is, limited to the strict minimum (*e.g.* no choice on the parameters of either mapping or assignment options, or on the way to identify the "true" cells), ii) there was no access to the code at the time, and the documentation provided only very brief description of the tools and parameters used (now the code is on GitHub https://github.com/10XGenomics/cellranger), iii) it is impossible to re-run just a part of the pre-processing workflow (often needed, especially in case of errors or doubts on the data), and iv) it lacks of detailed intermediary and regular quality checks. In addition, it is computationally heavy and time-consuming.

For all these reasons, we decided to develop an open-source and flexible approach to process scRNA-seq datasets based on the Eoulsan workflow manager (available to the community on GitHub: https://github.com/GenomicParisCentre/eoulsan). This software has been built according to state-of-the-art requirements.

We now routinely use both solutions, since Eoulsan provides many more quality checks to detect potential problems from the library preparation up to the count matrix generation. Furthermore, it is much more flexible in terms of parameter settings. It is also appropriate to include novel developments. On the other hand, CellRanger outputs allow a more straightforward comparison with data from other labs, and provides a proprietary application *Loupe Browser*[1] that enables biologist collaborators to visualize and start exploring their data immediately at the end of the workflow (without any data checking or cleaning though).

[1] https://www.10xgenomics.com/products/loupe-browser

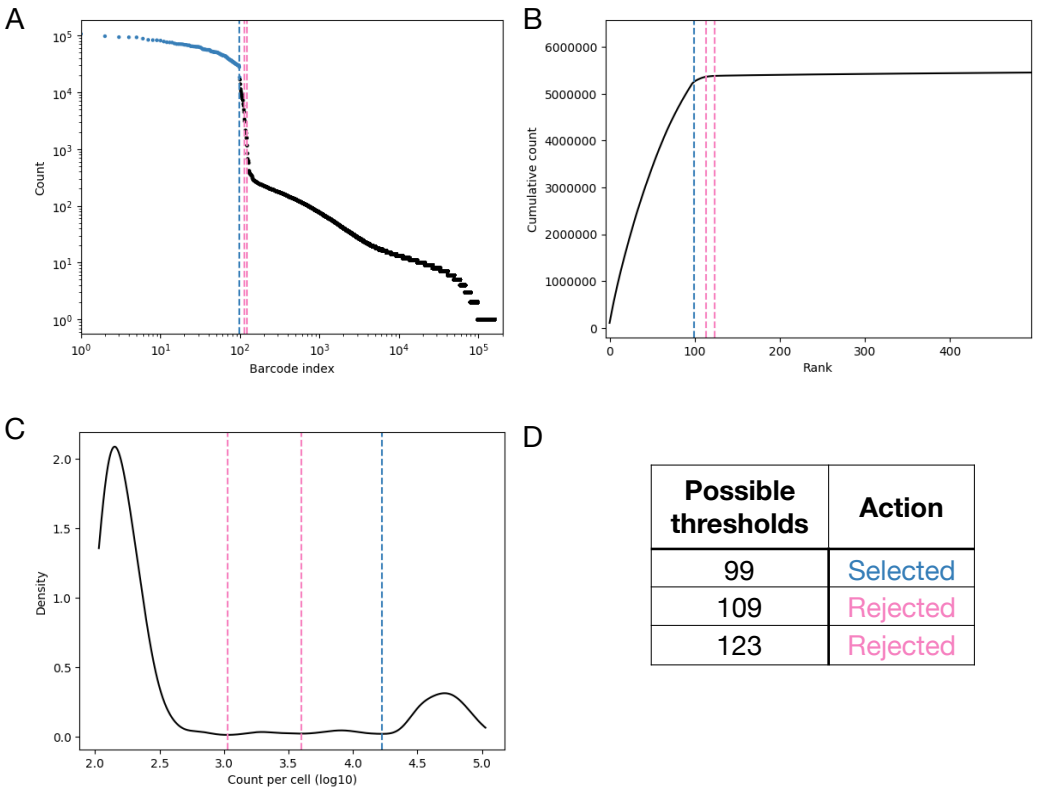
2.2 Methodological background

2.2.1 Identifying true cells

One strength of droplet-based scRNA-seq approaches is their ability to process a great number of cells in an automated manner. In order to recover and identify these cells during the pre-processing steps, each droplet has a unique cell barcode that will be added to the transcripts (see 1.2.2). The amount of droplets for a given experiment, and thus cell barcodes, is several orders of magnitude higher than the actual amount of input cells. This results in an excess in cell barcodes that do not refer to real cells. Therefore, one essential step of the droplet-based scRNA-seq workflow is i) to depict how many cells have effectively been processed and ii) isolate the reads that originate from these cells to pursue with the analyses.

A popular approach to identify if a barcode might refer to a "true" cell is called the knee method (Figure 2.1). Over the years, several flavors of the knee method have been developed. It was initially described in the very first study based on Drop-seq data by *Macosko et al., 2015* [64]. They suggested relying on the cumulative frequency plot, which is expected to exhibit an inflection point that matches the boundary between real cells and background barcodes (Fig. 2.1-B). This was the method implemented in CellRanger up to the 3.0 version[2] (which was released in 2019). However, this approach does not handle the case where some "false" barcodes would be due to sequencing errors. UMI-tools provides an error-proof, network-based approach, initially developed to handle UMI errors, that has been adapted to identify cells whitelist [86]. Figure 2.1 shows the output plots of *UMI-tools whitelist*, that are used to visually assess if a threshold has been properly estimated.

[2] https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/algorithms/overview



Possible thresholds	Action
99	Selected
109	Rejected
123	Rejected

Figure 2.1 Summary plots of UMI-tools whitelist output. The dotted lines represent the possible thresholds tested (values are specified in D). A) Barcode rank plot. It shows the total number of UMI counts detected in each "cell" (or barcode). X axis represents a ranked list of cell barcodes, ordered by the number of counts (from high to low). The part in blue highlights the barcodes identified as "true" cells, while the part in black represent the barcodes identified as background. B) Cumulative frequency plot. C) Density plot according to the total number of counts per cell (in log10). D) Possible thresholds tested by UMI-counts whitelist. It selected 99 as the best possible threshold. These plots were produced during the analysis of a toy dataset of 100 cells (public dataset downloaded from 10x Genomics website http://cf.10xgenomics.com/samples/cell-exp/1.3.0/hgmm_100/hgmm_100_fastqs.tar).

It should be noted that, since its 3.0 version, CellRanger cell whitelist identification relies on an algorithm introduced by *Lun et al., 2019* with their dedicated tool EmptyDrops [133]. It mainly relies on the comparison of expression profiles between top cells (the ones with the higher UMI counts) and lower cells (the ones that most probably belong to the background).

2.2.2 Cell barcodes and UMI processing

2.2.2.1 Barcodes assignment to each single read

The number of FASTQ files for an experiment differs depending on the sequencing settings. For a 10x Genomics run, read details are spread within three files:

- **Illumina barcodes:** used only during the FASTQ demultiplexing step;
- **Reads 1 (R1)**: contains the cell barcodes and the UMIs;
- **Reads 2 (R2)**: contains the sequencing reads *per se*.

Figure 2.2 illustrates this 10x Genomics-specific setting.

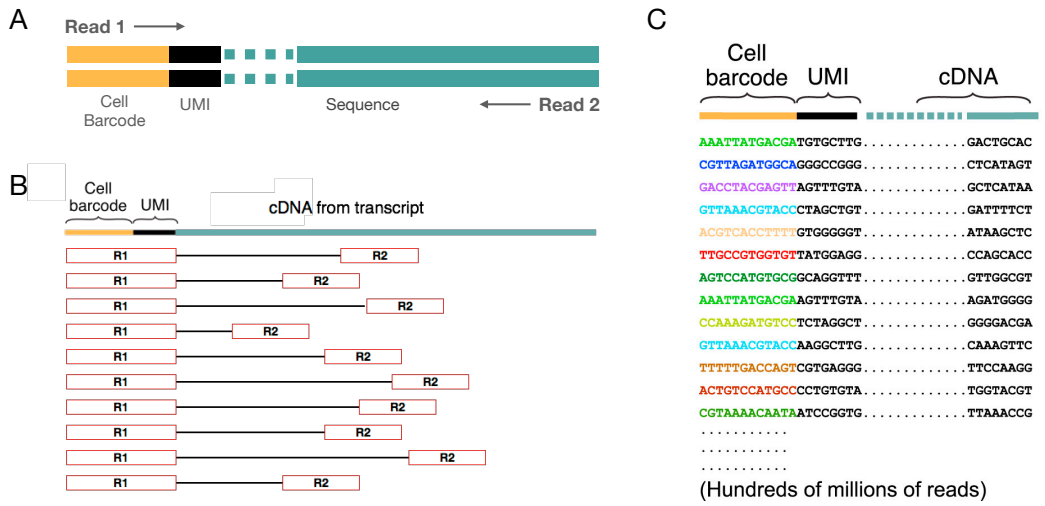


Figure 2.2 Illustrations of 10x Genomics cell barcode and UMI settings. A) Illustration of a 10x Genomics library fragment. 10x Genomics used paired-end sequencing. R1 is on the forward strand while R2 is on the reverse strand (Illumina index not shown). B) Schematic representation of the distribution of the R2 along the transcripts. C) Schematic representation of the reads sequences. Each color represents a unique cell. Images adapted from *Macosko et al.* [64] and http://data-science-sequencing.github.io/Win2018/lectures/lecture16/.

Moreover, each flow cell is attributed with 4 different Illumina indexes. This results in at least 12 FASTQ files ($4 * 3$) for each 10x Genomics experiment. In order to simplify file processing, all R1/R2 reads are usually merged together, resulting in just 2 files to handle. A standard practice is

then to add the R1 details (barcode and UMI) to the read name in R2 files. These steps can be handled with *UMI-tools extract* [86].

2.2.2.2 Handling sequencing errors

The emergence of UMIs have largely resolved PCR-amplification biases and they are now extensively used in scRNA-seq. However UMIs, as much as cell barcodes, still require correction for potential errors, and low-quality filtering. UMI errors handling is still an active research field [233]. A simple and straightforward method to account for PCR or sequencing errors (as implemented in CellRanger) is to rely on the Hamming distance between similar UMIs belonging to the same cell (threshold set to 1)[3]. Nevertheless, other types of errors can occur, such as mapping errors (a single read / UMI is assigned to multiple transcripts or to the wrong transcript) or collision errors (different mRNAs sharing the same UMI). More sophisticated approaches that would correct for all these types of errors are thus recommended for UMI-based datasets. The network-based approach implemented in UMI-tools precisely allows to account for both the relative frequency of similar UMIs and the number of mismatches to depict potential errors [86].

[3] https://kb.10xgenomics.com/hc/en-us/articles/115003133812-How-does-cellranger-count-process-and-fil

2.3 Eoulsan 2: an efficient workflow manager for reproducible single-cell and long-read transcriptomics analyses

2.3.1 Personal contribution

In the following work, I started by designing the 10x Genomics pipeline which I then developed and integrated into Eoulsan. I wrote and tested the Docker images corresponding to each newly integrated tool. I have also documented the 10x Genomics workflow and compared its performance with CellRanger. Lastly, I participated in the revisions of the manuscript, as well as in the design and generation of the figures. The following manuscript have been deposited in BioRxiv: https://www.biorxiv.org/content/10.1101/2021.10.13.464219v1.

2.3.2 Manuscript

Eoulsan 2: an efficient workflow manager for reproducible bulk, long-read and single-cell transcriptomics analyses

Nathalie Lehmann[2,§], Sandrine Perrin[1,§] , Claire Wallon[1], Xavier Bauquet[1], Vivien Deshaies[1], Cyril Firmo[1], Runxin Du[1], Charlotte Berthelier[1], Céline Hernandez[2], Cédric Michaud[2], Denis Thieffry[2], Stéphane Le Crom[1,4], Morgane Thomas-Chollier[1,2,3], Laurent Jourdren[1,*]

1- Genomics core facility, Institut de Biologie de l'ENS (IBENS), Département de biologie, École normale supérieure, CNRS, INSERM, Université PSL, 75005 Paris, France

2- Computational System Biology team, Institut de Biologie de l'ENS (IBENS), Département de biologie, École normale supérieure, CNRS, INSERM, Université PSL, 75005 Paris, France

3- Institut Universitaire de France

4- Sorbonne Université, CNRS, Institut de Biologie Paris-Seine (IBPS), Laboratory of Computational and Quantitative Biology (LCQB), F-75005, Paris, France

§ Co-first authors

* Corresponding Author: jourdren@bio.ens.psl.eu

ABSTRACT

Motivation: Core sequencing facilities produce huge amounts of sequencing data that need to be analysed with automated workflows to ensure reproducibility and traceability. Eoulsan is a versatile open-source workflow engine meeting the needs of core facilities, by automating the analysis of a large number of samples. Its core design separates the description of the workflow from the actual commands to be run. This originality simplifies its usage as the user does not need to handle code, while ensuring reproducibility. Eoulsan was initially developed for bulk RNA-seq data, but the transcriptomics applications have recently widened with the advent of long-read sequencing and single-cell technologies, calling for the development of new workflows.
Result: We present Eoulsan 2, a major update that (i) enhances the workflow manager itself, (ii) facilitates the development of new modules, and (iii) expands its applications to long reads RNA-seq (Oxford Nanopore Technologies) and scRNA-seq (Smart-seq2 and 10x Genomics).

The workflow manager has been rewritten, with support for execution on a larger choice of computational infrastructure (workstations, Hadoop clusters, and various job schedulers for cluster usage). Eoulsan now facilitates the development of new modules, by reusing wrappers developed for the Galaxy platform, with support for container images (Docker or Singularity) packaging tools to execute. Finally, Eoulsan natively integrates novel modules for bulk RNA-seq, as well as others specifically designed for processing long read RNA-seq and scRNA-seq. Eoulsan 2 is distributed with ready-to-use workflows and companion tutorials.

Availability and implementation: Eoulsan is implemented in Java, supported on Linux systems and distributed under the LGPL and CeCILL-C licenses at: http://outils.genomique.biologie.ens.fr/eoulsan/. The source code and sample workflows are available on GitHub: https://github.com/GenomicParisCentre/eoulsan. A GitHub repository for modules using the Galaxy tool XML syntax is further provided at: https://github.com/GenomicParisCentre/eoulsan-tools

Contact: eoulsan@bio.ens.psl.eu

Background

For the last fifteen years, technological advances in sequencing devices have resulted in a dramatic increase in read throughput. Furthermore, the rise of long-read sequencing with the third generation sequencers from Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT) enables the sequencing of much longer fragments. Altogether, the panel of High Throughput Sequencing (HTS) applications is very large, with a current democratization of single-cell approaches.

A common issue with HTS data, is to ensure reproducibility and traceability of the bioinformatics analyses performed on sequencing results, regardless of the application (research, medical diagnostics or forensics). Analysis pipelines encapsulate a series of sequential steps, for which a variety of individual tools must be connected. Such pipelines can be written as simple scripts, or with more elaborate workflow management systems. Sequencing core facilities are particularly in need of reliable automated pipelines that do not require manual intervention, while ensuring proper traceability.

Since its inception, the aim of Eoulsan (Jourdren et al. 2012) is to provide an efficient, stable and reliable workflow engine supporting HTS analyses, targeting bioinformatician end-users, and especially core facilities that analyze several dozens of projects each year. To do so, Eoulsan's core design encodes the analysis workflow within an XML flat file, rather than embedding it within the code of a program. Another core feature is the proper separation

between the experimental design (samples and associated metadata) and the analysis workflow itself. This model allows to reuse and standardize workflows across many experiments. Moreover, Eoulsan avoids file manipulation issues, by using file aliases linked to external repositories (e.g., for genomes, features annotations or mapper genome indexes). It automatically checks the input data on startup, and prevents output files overriding. Eoulsan is distributed with state-of-the-art individual tools (e.g., FASTQC, MultiQC, DESeq2) embedded in reusable modules, and offers alternative choices for many of the steps (e.g., mapping can be performed with either BWA, Bowtie1, Bowtie2, STAR, GMAP, GSNAP or MiniMap2).

Eoulsan supports diverse computing infrastructures to parallelize and distribute computation - from workstations to large clusters - thereby ensuring its efficiency and versatility. Our software was one of the very first bioinformatic tools able to run on an Hadoop cluster. The aim of Hadoop is to apply algorithms where data is physically stored, rather than sending data to computing nodes hosting the algorithms. In genomics, sequencing data are often stored in quite huge files. On a typical scientific cluster, transferring data to the computing nodes is thus problematic, with a risk of staturating access to network file systems, resulting in using a lower number of nodes. A Hadoop cluster solves this input/output (I/O) issue, enabling the use of the maximum available number of nodes. Hadoop thus remains particularly suited for genomics applications.

Several workflow engines have been developed specifically for genomics (recently reviewed in (Wratten, Wilm, and Göke 2021)). Among them, the most widely used are Galaxy (Afgan et al. 2018), Snakemake (Mölder et al. 2021) and Nextflow (Di Tommaso et al. 2017). Eoulsan's main principle is its low-code approach, which is closer to Galaxy's design. Its strength also lies in its installation ease, the stability of the code, and continuous maintenance and evolution for the last 10 years.

Here, we present Eoulsan 2, a major update of our workflow engine software with many new features and enhancements. This new version facilitates the development of new modules, includes module containerization, and extends its execution environments with support for several job schedulers. Novel modules have also been added to use state-of-the-art third-party tools. Eoulsan 2 supports transcriptomics applications, including both short and long reads bulk differential analyses, and single-cell RNA-seq workflows for Smart-seq2 and 10x Genomics. Moreover, we provide several ready-to-use workflows and tutorials for common analyses to help users to start with our workflow engine, accessible from the GitHub page of the project [https://github.com/GenomicParisCentre/eoulsan].

IMPLEMENTATION

Eoulsan is a free software published under the GNU LGPL and CeCCIL-C licenses. A Java Runtime Environment under Linux is the only requirement of this tool. Its unique user interface is the command line. Eoulsan can be run under three modes: local (workstation), Hadoop cluster or "standard" cluster (SLURM, TORQUE, HT-Condor and PBSPro job schedulers are supported). In cluster mode, users can define memory and processor count requirements for each task. Moreover, merger and splitter steps can be added to the workflow to better scale data processing all over the cluster and speedup computation.

Only two text files describing the pipeline need to be provided by the user to the input of the workflow engine: (i) the experimental design and (ii) the workflow definition (Figure 1). The experimental design of an Eoulsan analysis is stored in a text file (see example in Figure 2). For Eoulsan 2, the design file format has been enhanced to handle complex designs for DESeq2 differential gene analysis (Love, Huber, and Anders 2014), such as multiple comparisons and DESeq2 design formulas. The workflow steps and theirs parameters are listed in a companion XML file, separated from the code, ensuring flexibility and traceability (see example on the GitHub project page : https://raw.githubusercontent.com/wiki/GenomicParisCentre/eoulsan/files/workflow-rnaseq.xml). Altogether, these two files allow to quickly resume large analyses upon trouble-shooting, and guarantees reproducibility.

In the first Eoulsan release, the bundled RNA-seq workflow was hardcoded in the software, thus not allowing to easily add new steps. In Eoulsan2, we have rewritten the workflow engine, making it versatile for any bioinformatics workflow to be implemented.

In Eoulsan, each step of the workflow is a module, corresponding to encapsulated third-party tools (Figure 1). In addition to previous modules, Eoulsan2 is shipped with established tools such as DESeq2 (Love, Huber, and Anders 2014), HTSeq-count (Anders, Pyl, and Huber 2015), sam2bam [https://github.com/samtools/htsjdk], FastQC [https://www.bioinformatics.babraham.ac.uk/projects/fastqc/], MultiQC (Ewels et al. 2016), Trimmomatic (Bolger, Lohse, and Usadel 2014)...). External modules can be added through Java plugins. In an effort to simplify the development of a new module, Eoulsan 2 is now able to handle modules defined with the Galaxy tool XML syntax. Numerous modules have been adapted using this syntax (e.g., Cutadapt (Martin 2011), featureCounts (Liao, Smyth, and Shi 2014), UMI-Tools (Smith, Heger, and Sudbery 2017), …), and are easily accessible on a dedicated GitHub repository [https://github.com/GenomicParisCentre/eoulsan-tools]. It is thus easy to adapt an existing program from Galaxy to Eoulsan.

To ensure reproducibility of results over multiple computers, Docker [https://www.docker.com/] has streamlined container management. Eoulsan can rely on Docker or Singularity [https://sylabs.io/singularity/] containers for both Galaxy Tool XML and Java modules (e.g., DESeq2). Each module thus runs separately in its own container. This approach simplifies the deployment of the modules and avoids dependency management and complex installation procedures.

Since the first version of our tool, the number of available modules has dramatically increased (10 modules initially vs 33 modules, among which 14 are built with the Galaxy tool syntax and available on our specific GitHub repository). To ensure consistency of the code over versions, we have deployed a functional test system based on Jenkins [https://www.jenkins.io/] continuous integration software. This system checks every week whether changes in Eoulsan source code has led to result changes in more than 190 reference analyses (2 days of analyses). The tests cover all modules and Galaxy tools, mapping programs, as well as local and Hadoop execution modes.

With unit tests for Java source code, functional tests and container management, Eoulsan 2 ensures a very strong confidence in reproducibility of user results.

RESULTS

While the first version of Eoulsan was designed only for bulk RNA-seq differential analysis workflows, Eoulsan 2 is shipped with ready-to-use workflows for various types of transcriptomics analyses:

- bulk RNA-seq Illumina differential analysis
- Oxford Nanopore long reads RNA-seq
- scRNA-seq with Smart-seq2 protocol
- scRNA-seq with 10x Genomics protocol

Ready-to-use workflows, along with tutorials detailing their usage, are accessible on the GitHub page of Eoulsan [https://github.com/GenomicParisCentre/eoulsan].

Bulk RNA-seq Illumina differential analysis

In its first version, Eoulsan supported four read mappers (BWA, Bowtie, SOAP2, GSNAP). In Eoulsan 2, the RNA-seq workflow has been greatly improved with new integrated mappers (STAR, Bowtie2, GMAP/GSNAP), while SOAP2 is no longer supported. We have developed a fast implementation of the HTSeq-count algorithm in Java, which is included in Eoulsan 2. It also supports complex designs with DESeq2.

Bulk Oxford Nanopore long reads RNA-seq differential analysis

The Oxford Nanopore long-reads RNA-seq workflow uses Minimap2 (Li 2018) to align sequences over the reference genome. The workflow is otherwise very similar to the Illumina workflow, with the differential analysis performed with DESeq2. In all the steps (except for mapping), the workflow uses the same modules as the Illumina workflow, with a tuning of the parameters to adapt to the long-reads. The steps with tuned parameters are : filter raw FASTQ files (filterreads module), remove unmap and multimatches SAM entries (filtersam module) and count alignments with HTSeq-count (expression module). Even though the workflow currently relies on the same tools as used for the Illumina workflow, we are expecting to update the workflow once the LRGASP challenge reaches its conclusions [https://www.gencodegenes.org/pages/LRGASP/]. This systematic evaluation of different methods for transcript computational identification and quantification will constitute a reference to help us identify the most relevant tools, and integrate them as new modules in Eoulsan.

Single-cell RNA-seq pre-processing analyses

To answer user growing interest for single-cell transcriptomics data, we developed scRNA-seq workflows for both Smart-seq2 and 10x Genomics technologies. These workflows focus on the pre-processing steps, from the raw reads up to the generation of the count matrix. These workflows are detailed in Figure 3. These first steps are crucial to ensure the quality of the pre-processed data before the downstream analyses.

The Smart-seq2 workflow is derived from the bulk RNA-seq Illumina workflow, as the steps are quite similar, with the mapping performed by STAR. The main difference is that data for each cell is stored in a separate FASTQ file. A typical experiment thus produces hundreds of FASTQ files. The fact that Eoulsan uses Hadoop calculation clusters makes this tool well-suited for parallelized processes, hence handling this issue.

The 10x Genomics workflow reuses some existing modules of Eoulsan (e.g., mapping with STAR), but has required the development of novel modules to take into account the cellular barcodes and Unique Molecular Identifiers (UMIs), intrinsic to this technology. The 10x Genomics protocol uses cell barcodes (16bp) to identify each cell. Many barcodes do not correspond to real cells, which means "valid" barcodes corresponding to real cells must be inferred. For these steps, we rely on the third-party tool UMI-Tools (Smith, Heger, and Sudbery 2017). By default, UMI-tools whitelist detects these valid barcodes with the knee method, by retaining the top most abundant barcodes (umiwhitelist module).

Following the identification of these « valid » barcodes, the next step is to filter out the reads that do not match with these barcodes (umiextract module). Once the mapping has been performed, reads must be assigned to the genes they most probably originated from. This step is handled with the featureCounts module (Liao, Smyth, and Shi 2014), which outputs a bulk count matrix. Finally, the single-cells counts are isolated with the umicount module, thus producing a single-cell expression matrix. It should be noted that only the reads mapping to the same strand as the annotated genes are taken into account, since the 10x Genomics protocol is stranded.

Downstream analyses are generally directly performed in R. Eoulsan facilitates this transition to post-processing steps with R, by generating a RDS file containing a SingleCellExperiment Bioconductor object. It encapsulates the count matrix, along with cells and gene annotations, as a ready-to-use object for downstream analyses. Another possibility is to output a CellRanger formatted count matrix with the MatrixToCellRangerMatrix module. This option allows users to perform downstream analyses in R with functions that take as input a CellRanger matrix. Altogether, Eoulsan 2 ensures users don't have to change their downstream analyses pipeline without further changes, by directly providing the count matrix in the correct format.

Comparison with similar tools

In terms of Workflow Management Systems, Nextflow and Snakemake are currently the most common engines for a usage at the command line, and Galaxy for a usage through a web browser interface. Eoulsan's originality lies in its internal design as a low-code workflow manager. Indeed, the workflows use XML files without any code. The modules are actually an abstraction layer over the actual tools, thereby hiding the complete syntax of all their parameters. This organisation enables to maintain the code of the modules separately from the workflow itself. This choice is particularly suited in the context of a core facility platform that analyses dozens of projects each year. Each project is thus associated with a reproducible workflow file, independent of the command lines that are actually run. Portability of the code is comparable to the above-mentioned workflow engines, as many modules are containerized with Docker or Singularity, and we take advantage of systems developed by Galaxy (Galaxy Tools XML syntax). In addition, the code consistency in Eoulsan is tested with functional tests using a Jenkins server on a weekly basis.

Our 10x Genomics scRNA-seq workflow can be compared to the CellRanger program, developed by the same company. CellRanger performs the pre-processing steps, and provides a user-ended HTML report already including some downstream analyses (clustering, t-SNE

visualisation). The Quality Check (QC) values are limited in CellRanger, while Eoulsan provides FASTQC and MultiQC reports that enables a deeper interpretation of the quality of the data. Eoulsan relies on state-of-the-art tools such as UMI-Tools, which enables more versatility than CellRanger. CellRanger's lack of time efficiency and high requirement for memory usage has previously been underlined (Gao et al. 2021). Our workflow is 2-3 times faster with limited resource usage for similar results (R2=0.998 for UMI count per cell) (Figure 4).

Conclusions

Our workflow engine Eoulsan 2 aims at facilitating high throughput sequencing analysis for bioinformaticians, in particular for transcriptomics applications. Eoulsan handles the most resource-expensive parts of analyses, and it can conveniently be deployed on any cluster, as well as on workstations. Result reproducibility has always been one of our major concerns in Eoulsan development. This is why Eoulsan source code is extensively tested through unit tests, as well as a huge number of functional tests. In addition, since Docker and Singularity container systems have emerged, they have been utilized for Eoulsan modules external dependencies. To ease module creation, we introduced in Eoulsan 2 the support for the Galaxy Tool XML files, and a comprehensive documentation for developers. Moreover, several ready-to-use workflows for both short and long reads RNA-seq, Smart-seq2 and 10x Genomics scRNA-seq are available, along with tutorials. With all these enhancements since its first version ten years ago, and a strong foundation for reproducibility and scalability, Eoulsan is particularly suited for core facilities wishing to implement a long-term stable solution for managing their transcriptomics workflows.

Availability and requirements

Project name: Eoulsan

Project home page:

https://www.outils.genomique.biologie.ens.fr/eoulsan/: Binary downloads and reference documentation

https://github.com/GenomicParisCentre/eoulsan: Source code and workflow wiki

https://github.com/GenomicParisCentre/eoulsan-tools: Additionnal modules

Operating system(s): Unix

Programming language: Java

License: GNU LGPL and CeCCIL-C licenses

Any restrictions to use by non-academics: none

Acknowledgements

We thank Hugo Varet for helping with DESeq2 usage, Sophie Lemoine for intensive application testing and feedbacks, Pierre-Marie Chiaroni for the initial work on modules of the ChIP-seq pipeline, Aurelien Birer for help with two Galaxy tools, Geoffray Brelurut for helping with the Smart-seq pipeline, Hatim El Jazouli for testing the 10x single-cell pipeline, and Bpipe authors for the job scheduler submission scripts. We also thank the IBENS informatics and bioinformatics core facilities for helping with testing Eoulsan on the cluster, as well as the TGCC platform for testing on their cluster.

Contributions

N.L developed, tested, documented and compared the 10x Genomics workflow. S. P developed the Galaxy tools integration and functional test system. C.W ported HTSeq-count to Java and Hadoop. X.B implemented the new design file format and support for DESeq2. V.D rewritten DESeq 1 support. C.F created sam2fastq, bam2sam modules and enhanced XLSX/ODS export system. R.D integrated many read filters using Trimmomatic. C.B improved design file checks for DESeq2. C.H supervised C.M who developed modules for format interconversion bed/bigbed/bam/bigwig and the trimadapt module. D.T. was involved in student supervision and provided ideas for the general development of Eoulsan. S.L.C. and M.T.C. were involved in proposing and planning the addition of new features, and were involved in the supervision of students and engineers. L.J. developed the first version of Eoulsan, rewrote the workflow engine, supervised and coordinated the addition of new modules and features, and ensured the long-term maintenance of the project L.J. and M.T.C. drafted the manuscript, with input from N.L, S.L.C and D.T.

Funding

This work was supported by the France Génomique national infrastructure, funded as part of the "Investissements d'Avenir" program managed by the Agence Nationale de la Recherche (contract ANR-10-INBS-09). S.P. was also supported by France Génomique. Agence Nationale de la Recherche supported N.L. (contract ANR-14-CE11-0006-01 and ANR-16-CE15-0024) and C.H. (contract ANR-13-EPIG-0001). M.T.C. is supported by the Institut Universitaire de France.

Competing interests. The authors declare no competing or financial interests.

References

Afgan, Enis, Dannon Baker, Bérénice Batut, Marius van den Beek, Dave Bouvier, Martin Cech, John Chilton, et al. 2018. "The Galaxy Platform for Accessible, Reproducible and Collaborative Biomedical Analyses: 2018 Update." *Nucleic Acids Research* 46 (W1): W537–44.

Anders, Simon, Paul Theodor Pyl, and Wolfgang Huber. 2015. "HTSeq--a Python Framework to Work with High-Throughput Sequencing Data." *Bioinformatics* 31 (2): 166–69.

Bolger, Anthony M., Marc Lohse, and Bjoern Usadel. 2014. "Trimmomatic: A Flexible Trimmer for Illumina Sequence Data." *Bioinformatics* 30 (15): 2114–20.

Di Tommaso, Paolo, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. "Nextflow Enables Reproducible Computational Workflows." *Nature Biotechnology* 35 (4): 316–19.

Ewels, Philip, Måns Magnusson, Sverker Lundin, and Max Käller. 2016. "MultiQC: Summarize Analysis Results for Multiple Tools and Samples in a Single Report." *Bioinformatics* 32 (19): 3047–48.

Gao, Mingxuan, Mingyi Ling, Xinwei Tang, Shun Wang, Xu Xiao, Ying Qiao, Wenxian Yang, and Rongshan Yu. 2021. "Comparison of High-Throughput Single-Cell RNA Sequencing Data Processing Pipelines." *Briefings in Bioinformatics* 22 (3). https://doi.org/10.1093/bib/bbaa116.

Jourdren, Laurent, Maria Bernard, Marie-Agnès Dillies, and Stéphane Le Crom. 2012. "Eoulsan: A Cloud Computing-Based Framework Facilitating High Throughput Sequencing Analyses." *Bioinformatics* 28 (11): 1542–43.

Liao, Yang, Gordon K. Smyth, and Wei Shi. 2014. "featureCounts: An Efficient General Purpose Program for Assigning Sequence Reads to Genomic Features." *Bioinformatics* 30 (7): 923–30.

Li, Heng. 2018. "Minimap2: Pairwise Alignment for Nucleotide Sequences." *Bioinformatics* 34 (18): 3094–3100.

Love, Michael I., Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15 (12): 550.

Martin, Marcel. 2011. "Cutadapt Removes Adapter Sequences from High-Throughput Sequencing Reads." *EMBnet.journal* 17 (1): 10–12.

Mölder, Felix, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, et al. 2021. "Sustainable Data Analysis with Snakemake." *F1000Research* 10 (January): 33.

Smith, Tom, Andreas Heger, and Ian Sudbery. 2017. "UMI-Tools: Modeling Sequencing Errors in Unique Molecular Identifiers to Improve Quantification Accuracy." *Genome Research* 27 (3): 491–99.

Wratten, Laura, Andreas Wilm, and Jonathan Göke. 2021. "Reproducible, Scalable, and Shareable Analysis Pipelines with Bioinformatics Workflow Managers." *Nature Methods*, September. https://doi.org/10.1038/s41592-021-01254-9.

FIGURES

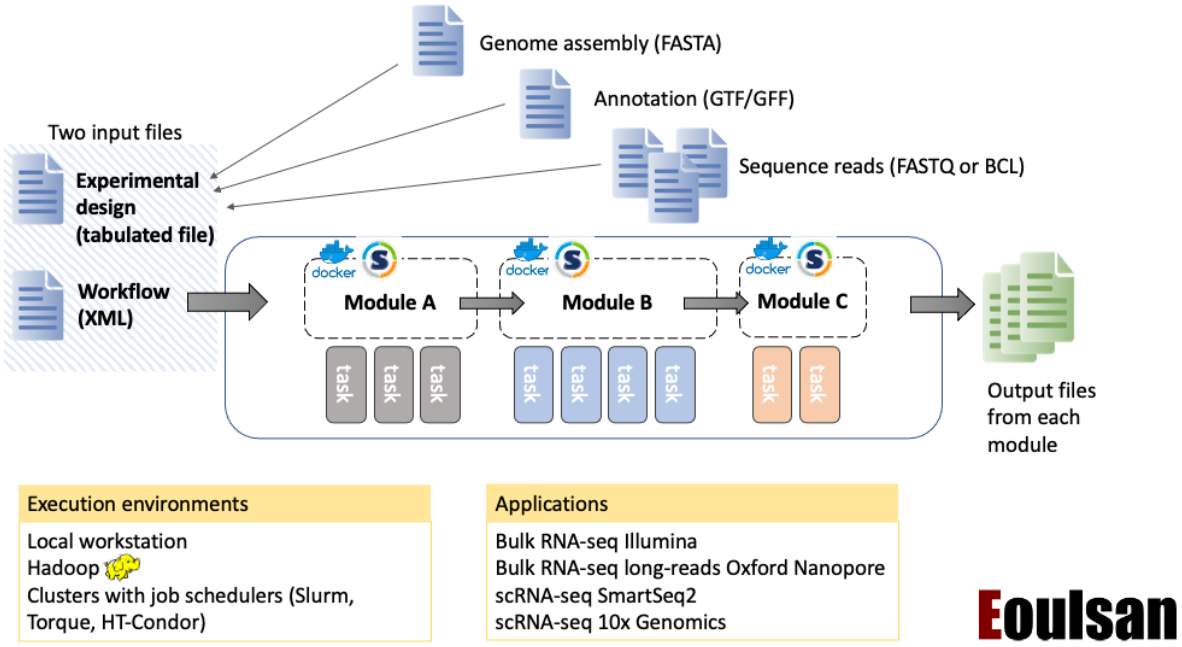


Figure 1 : Overview of Eoulsan 2 workflow management system

Eoulsan is a workflow engine designed for high-throughput genomics analyses, with a focus on transcriptomics. The workflow is described as an XML file, rather than code, and each step corresponds to a reusable independent module. Each module may internally run several tasks with third-party programs. Installation of these dependencies is facilitated by containerization with Docker or Singularity. Implementation of a new module is facilitated by the use of Galaxy Tools XML (not shown). The experimental design file specifies the metadata of the samples, as well as their status (controls), and links to larger files that may be stored elsewhere in the filesystem, such as the genome assembly and annotation. The workflow can be executed either on a local workstation or on a cluster, which enables parallelization of the tasks when possible. Ready-to-use workflows are available for four applications.

```
[Header]
DesignFormatVersion=2
GenomeFile=mm10ens91.fasta.bz2
GffFile=mm10ens91.gff.bz2
GtfFile=mm10ens91.gtf.bz2
AdditionalAnnotationFile=mm10ens91.tsv.bz2
[Experiments]
Exp.exp1.name=demo-mouse-rnaseq

[Columns]
SampleId                                SampleName  Reads                              Date        FastqFormat   RepTechGroup  Exp.exp1.Condition  Exp.exp1.Reference
20130267                                KO1         2013_0267_S1_L001_R1_001.fastq.bz2 2015-10-04  fastq-sanger  KO1           KO                  0
20130268                                KO2         2013_0268_S2_L001_R1_001.fastq.bz2 2015-10-04  fastq-sanger  KO2           KO                  0
20130269                                KO3         2013_0269_S3_L001_R1_001.fastq.bz2 2015-10-04  fastq-sanger  KO3           KO                  0
20130270                                WT1         2013_0270_S4_L001_R1_001.fastq.bz2 2015-10-04  fastq-sanger  WT1           WT                  1
20130271                                WT2         2013_0271_S5_L001_R1_001.fastq.bz2 2015-10-04  fastq-sanger  WT2           WT                  1
20130272                                WT3         2013_0272_S6_L001_R1_001.fastq.bz2 2015-10-04  fastq-sanger  WT3           WT                  1
```

Figure 2 : Example of a design file for Eoulsan

The header section contains general information about the design and the project. This includes the genome file that will serve for the mapping, and the annotation. The "Experiments" section specifies details on the experimental design. It can include multiple experiments and comparisons that should be made for the differential analyses. The "Columns" section provides details on each sample, including the information about technical and biological replicates. The file can be found on the GitHub project for Eoulsan at this URL: https://raw.githubusercontent.com/wiki/GenomicParisCentre/eoulsan/files/design-rnaseq.txt

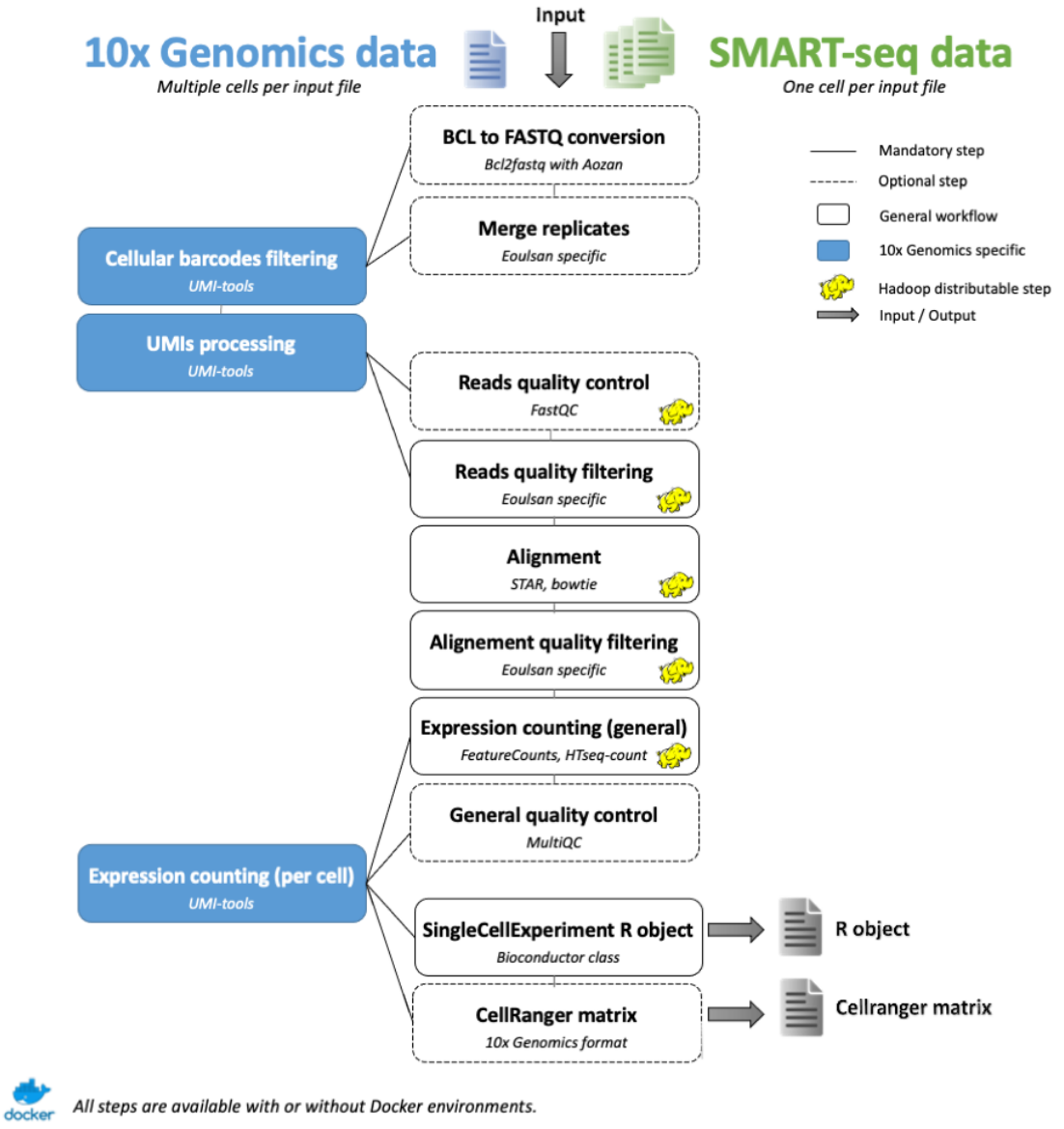
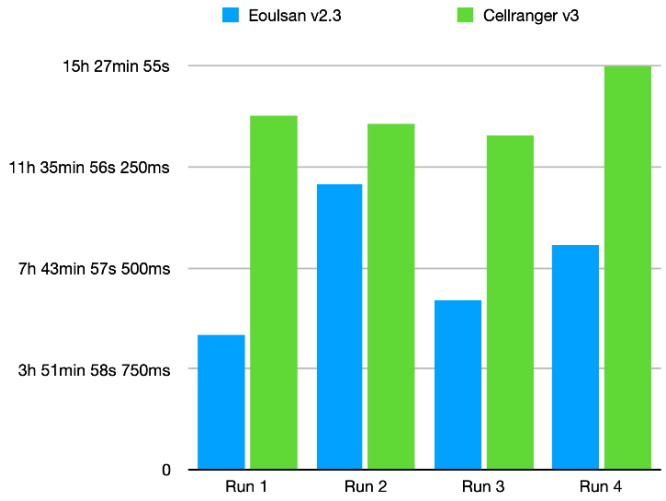


Figure 3 : Single-cell RNA-seq workflows in Eoulsan 2

Eoulsan 2 supports two workflows for scRNA-seq data processing: Smart-seq2 and 10x Genomics. Some modules were specifically developed to support this type of data. While the Smart-seq2 workflow is quite similar to a bulk RNA-seq workflow, the 10x Genomics workflow has several specific modules to treat cell barcodes and Unique Molecular Identifiers (UMIs).

A



B

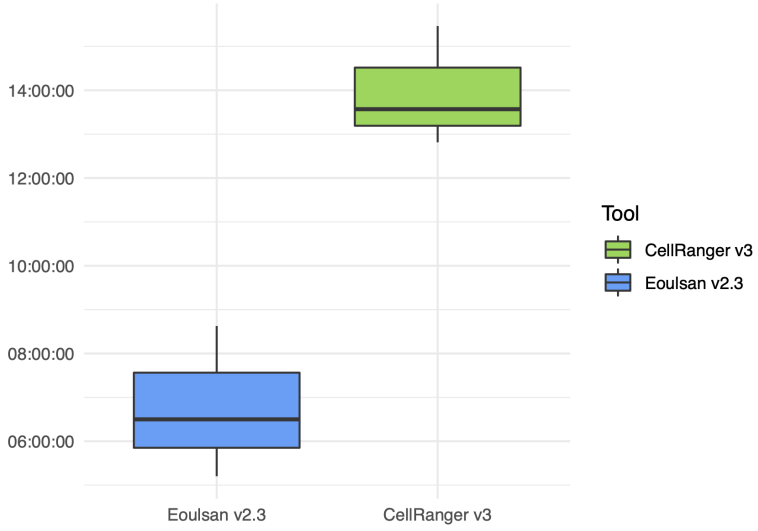


Figure 4 : Comparison of computing efficiency between Eoulsan and CellRanger

The comparison was performed using Eoulsan v2.3 and CellRanger v3, on a dataset of 5000 PBMC cells generated with the 10x Genomics technology (v3 Chemistry) downloaded from 10x Genomics website: [https://www.10xgenomics.com/resources/datasets/5-k-peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-v-3-chemistry-3-1-standard-3-0-2]. **A**. Barplot of each of the 4 individual runs. For comparison purposes with CellRanger that uses a pre-computed index, we ran Eoulsan with a pre-computed STAR index, except for run 2. Note that for run 2, Eoulsan is faster than CellRanger, even when adding the step of building the STAR index. **B**. Boxplot showing the execution time (Y axis) for each program, summarizing the three runs using a pre-computed index.

2.4 Pre-processing of the SYMASYM dataset

The overall 10x Genomics scRNA-seq workflow steps are schematized in Figure 3 of the above manuscript and detailed in the results section *Single-cell RNA-seq pre-processing analyses*. I describe below the results obtained with the SYMASYM dataset, produced by Xavier Morin's team. The full workflow used for its analysis is shown in Annexes A.2 (it provides a full description of the steps and parameters settings). I also compare these results with the output of CellRanger.

The SYMASYM dataset has been produced from chicken cervical spinal progenitors at 66 hours of embryonic development. For this dataset, 5000 cells were originally loaded into the 10x Genomics cell partitioning system (Chromium). Due to low cell capture rates (as mentioned in 1.2.2), we expected around 50%[4] of the cells to be captured.

We used the NCBI reference genome GCF_000002315.5_GRCg6a_genomic.fna and a corrected annotation ref_GRCg6a_top_level.corrected.gtf to process this dataset. By corrected I mean that I designated the mitochondrial genes as "exons" in the GTF file. In the original file, they are described as "CDS" features, which precludes them from being used in the gene assignment step, and results in misclassification and errors in identifying and filtering cells (since the percentage of mitochondrial genes associated with a cellular barcode is used as a key parameter to keep or exclude "healthy" and "unhealthy" cells).

2.4.1 With Eoulsan

Executing commands The full workflow has been executed with the following two commands:

```
# Automatically create design file
eoulsan createdesign -p data/fastq/*
    data/genome/GCF_000002315.5_GRCg6a_genomic.fna
    data/annotation/ref_GRCg6a_top_level.corrected.gtf
```

[4] https://www.10xgenomics.com/wp-content/uploads/2016/04/10x_Single_Cell_App_Note.pdf

```
# Execute full workflow
eoulsan -conf conf exec workflow_10xGenomics.xml design.txt
```

Listing 2.1 Commands to execute Eoulsan workflow

The first command outputs the design file (see Annexes A.1). I chose here to process each lane separately, in order to facilitate computing parallelization and depict if a lane appears as of deficient quality. It thus results in $4 * 2$ FASTQ files to handle (4 *R2* and their corresponding 4 *R1*, see below). The outputs are merged in a latter phase, at the end of the workflow.

Reads quality checking and filtering To evaluate the reads quality, we rely on the FASTQC [115] module (integrated within Eoulsan). Then, we filter out low-quality reads with the Eoulsan built-in module, thanks to mean Phred scores (see Annexes A.2 for details on Phred scores). We set the mean read quality to a minimum threshold of 30. Figure 2.3 shows the reads quality summarized by MultiQC [234], and Figure 2.4 highlights a sample of the output produced by MultiQC on top of the results of FASTQC (other FASTQC results are less relevant for scRNA-seq data), before filtering. Finally, Figure 2.5 reveals reads quality summary after quality filtering.

Sample Name	% Dups	% GC	Length	M Seqs
S1_R1_001	34.1%	47%	28 bp	135.1
S1_R2_001	60.3%	49%	55 bp	135.1
S2_R1_001	31.9%	47%	28 bp	124.1
S2_R2_001	61.3%	49%	55 bp	124.1
S3_R1_001	33.0%	47%	28 bp	129.1
S3_R2_001	62.0%	49%	55 bp	129.1
S4_R1_001	35.4%	47%	28 bp	145.3
S4_R2_001	62.4%	49%	55 bp	145.3

Figure 2.3 SYMASYM reads quality summary statistics before filtering. Key characteristics of each FASTQ file. There are 2 files per lane (each lane is identified with S1, S2, S3 or S4). Duplicated reads are automatically removed later on with UMIs.

We can see here that all the FASTQ files turn up to be of good quality. Before the filtering there were 533 millions of reads. Afterwards, we can observe that the number of reads dropped down to 470 million, that is to say 88% of the reads are identified as good quality.

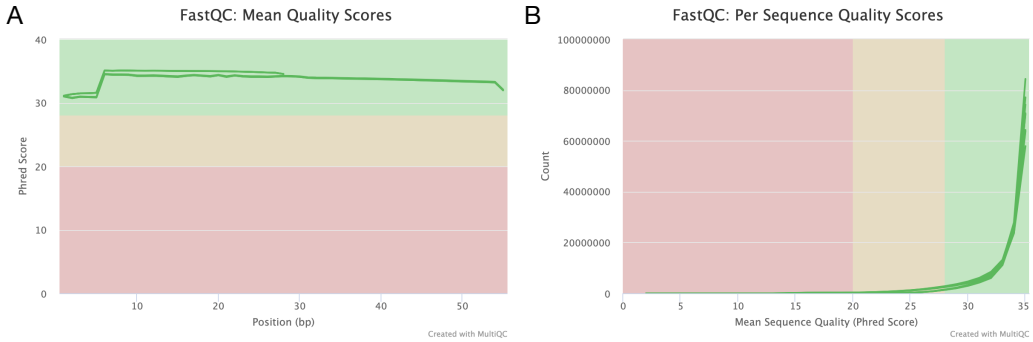
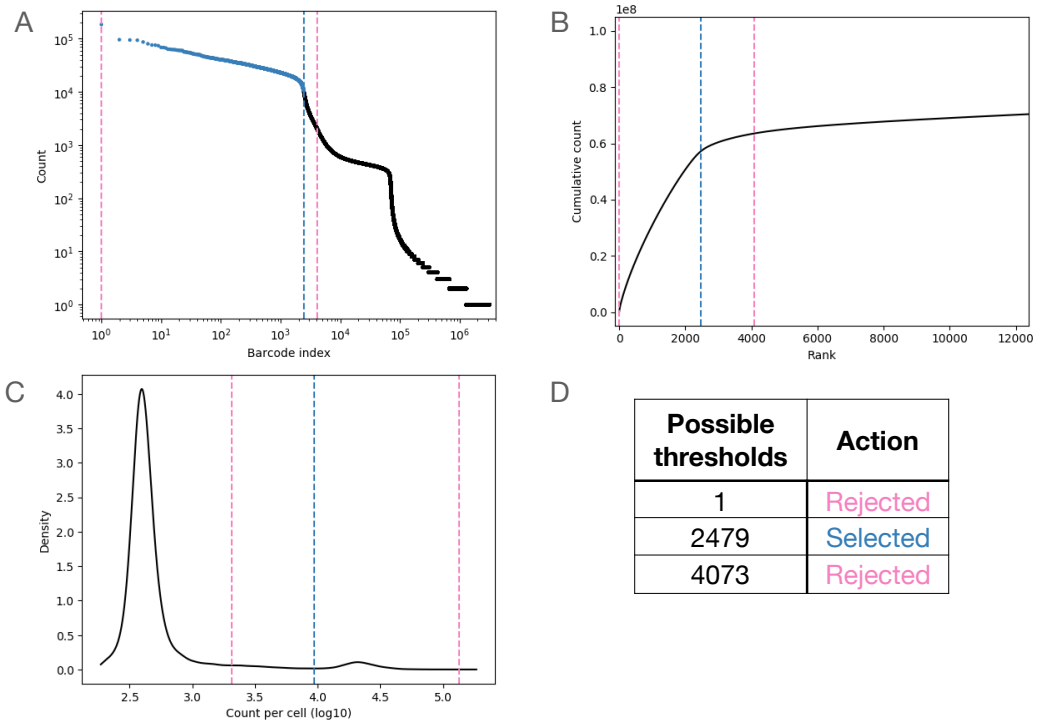


Figure 2.4 SYMASYM results of reads quality checking. A) Mean quality scores of all reads according to base position. Each green line represents a single FASTQ file. They are difficult to distinguish as they all overlap. B) Number of reads for each mean quality score in each of the FASTQ files. Colors represent low (red), medium (yellow) and high Phred scores (green).

Sample Name	% Dups	% GC	Length	M Seqs
step2filterreads_output_reads_S1R1001_file0	32.2%	47%	28 bp	118.6
step2filterreads_output_reads_S1R1001_file1	71.8%	48%	55 bp	118.6
step2filterreads_output_reads_S2R1001_file0	30.3%	47%	28 bp	109.1
step2filterreads_output_reads_S2R1001_file1	70.9%	48%	55 bp	109.1
step2filterreads_output_reads_S3R1001_file0	30.8%	47%	28 bp	114.2
step2filterreads_output_reads_S3R1001_file1	71.1%	49%	55 bp	114.2
step2filterreads_output_reads_S4R1001_file0	33.8%	47%	28 bp	128.2
step2filterreads_output_reads_S4R1001_file1	72.3%	49%	55 bp	128.2

Figure 2.5 SYMASYM reads quality summary statistics after filtering. Same plot as 2.3, after quality checking and filtering.

Cell whitelist Cell barcodes corresponding to "true" cells are identified with *umiwhitelist* Eoulsan module, which is based on UMI-tools whitelist [86]. Figure 2.6 shows the summary plots produced by this module on SYMASYM data. From the tested threshold values, we can see that the first one (threshold value of 1) is due to an outlier which has twice as many counts as its closest cells in terms of total counts (Figure 2.6-A). The corrected threshold of 2479 seems appropriate according to both the knee position in both barcode rank plot (Figure 2.6-A) and the cumulative frequency plot, and the expected number of cells that was around 2500 (input of 5000 cells, with an approximated capture rate of 50%). On the contrary, the 4073 threshold seems too far from the knee, so we can assume it was correct to reject it.



Possible thresholds	Action
1	Rejected
2479	Selected
4073	Rejected

Figure 2.6 Eoulsan cells whitelist summary plots. See 2.1 for a description of these plots. UMI-tools whitelist here selected 2479 as the best possible threshold, out of 2500 expected cells.

Mapping and gene assignment For the mapping and gene assignment steps, we used STAR [122] and featureCounts [123] Eoulsan modules. Figure 2.7 highlights the outputs of both steps, summarized by MultiQC.

The very low number of unmapped reads (0.1% in each lane) shows that the chicken *galGal6* genome assembly quality is enough not to impact on read loss. However, we can observe that nearly 8% of the reads are multi-mapped (7.7% multiple loci, 0.2% too many loci - all lanes have the same values). A multi-mapping inclusive approach would thus allow to recover about 17 millions of reads. For the rest of the analyses, we retained only the uniquely aligned reads (92% of all reads).

As for the gene assignment values, we observe that 68.2% of the reads that passed previous filters (*e.g.* uniquely mapped) are assigned to a gene

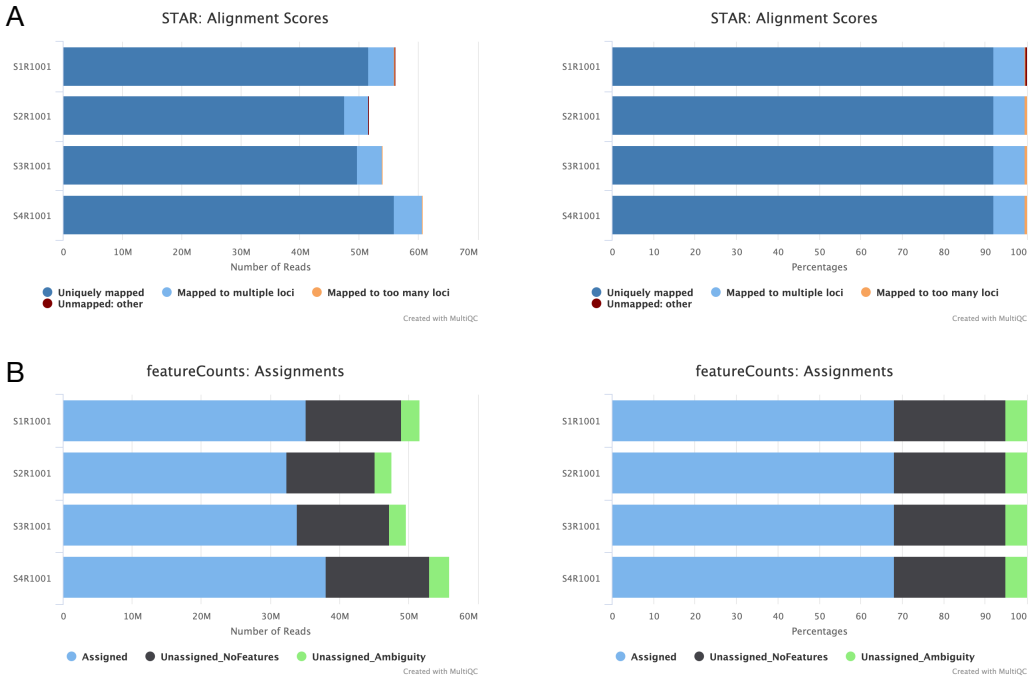


Figure 2.7 SYMASYM mapping and gene assignment outputs summary plots. A) Alignment scores of the reads (separated by lane), represented in terms of total counts (left plot) and percentage (right plot). The uniquely mapped reads are represented in dark blue, and the multi-mapped reads are in light blue (< 10 loci) or orange ($>= 10$ loci). B) Gene assignment values of the reads (separated by lane), represented in terms of total counts (left plot) and percentage (right plot). The uniquely assigned reads are shown in blue. The unassigned reads are splitted into unassigned due to the absence of known feature (in black) and unassigned due to an ambiguity (read overlapping 2 known features on the same strand, in green). Notice here the large proportion of unassigned reads (in black).

annotated in the reference. Depending on the considered lane, that makes a total of 32.4 (S2) to 38.1 (S4) millions of assigned reads. Most of the reads lost at this stage are due to the lack of feature in the reference annotation (26.7% of all reads). The rest of the reads, 5.1%, are lost because of an ambiguity.

We can see here that despite small differences in reads lane repartition, there is no differences in terms of percentage, neither for the mapping nor for the gene assignment.

2.4.2 Comparison with CellRanger v3.0.1

At the end of the workflow, CellRanger provides a summary report of some key quality values. Figure 2.8 shows a screenshot of the SYMASYM CellRanger report. Among others features, we can notice from this report that CellRanger detected 7285 cells.

Regarding computing times, CellRanger successfully ran the analyses in 15H51min, while Eoulsan reported a runtime of 5H34min. This significant difference could be explained by the fact that CellRanger processed 7285 cells whereas Eoulsan processed three times less cells (which means as much less reads to be processed).

Because Eoulsan and CellRanger both rely on STAR for the mapping step, the lower number of reads that map confidently to the genome with CellRanger (70.7%, or 375 millions of reads) may be:

- Indicative of too stringent parameters for the mapping steps (which are hard coded in the pipeline for all versions < 4.0, thus cannot be changed by the user[5]) - moreover, it is key to note that the mapping with CellRanger relies on both the genome and transcriptome ;
- Inflated by the low-quality reads (that does not seem to be filtered by CellRanger);
- The reads belonging to the next to 5000 background "cells".

We checked this last assumption by forcing the number of cells to be processed by CellRanger to 2700 (*e.g.* the top 2700 in terms of total counts). In this configuration, the percentage of reads confidently mapped to the genome remained the same (70.5%, data not shown). Considering that Eoulsan could still map 432 millions of reads that passed the quality filtering (92% of 470 million), we could assume that CellRanger poorer results in terms of mapping are due to unsuitable mapping settings and inflated by low-quality reads.

[5]https://kb.10xgenomics.com/hc/en-us/articles/360003877352-How-can-I-modify-the-STAR-alignment-parameters-in-Cell-Ranger-

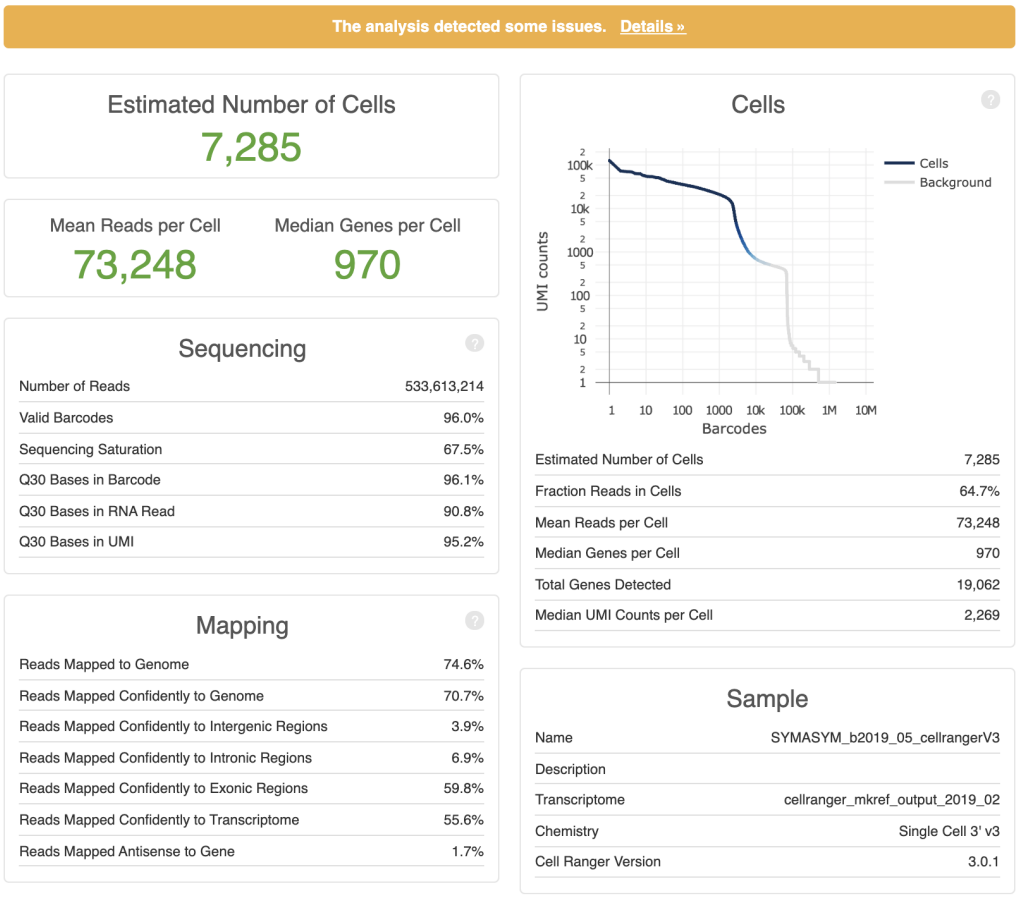


Figure 2.8 Screenshot of SYMASYM CellRanger report. The plot on the right is equivalent to the Barcode rank plot produced by Eoulsan. The main difference with Eoulsan is the number of barcodes detected as "true" cells.

Moreover, transcriptome mapping is not recommended unless a high-quality annotation is available (see 5.2.3).

Regarding the gene assignment, it is a bit more complex to compare directly the proportion of reads that are assigned to a feature, since the gene assignment step in CellRanger is carried out simultaneously to the mapping. The indicated 55.6% reads mapping confidently to the transcriptome are the reads that are taken into account for UMI counting.

We can also observe that the number of median genes per cell is very low, certainly due to the nearly 5000 cells that were incorrectly assigned as cells. Eoulsan does not output these types of values (mean reads per cell, me-

dian genes per cell), since these values are highly sensitive to data cleaning. We thus prefer to compute them in a second time, during secondary analyses.

In conclusion, SYMASYM raw data are of sufficient quality to be used for secondary analyses. The only concerning value is the 68,2% of reads assigned to genes, which seems quite low considering the protocol targets transcripts, and made us hypothesise a possible lack in the gene annotation. If the data were processed with CellRanger, we should have filtered out manually the nearly 5000 incorrectly assigned cells, or re-run the whole pipeline with a parameter forcing the number of cells to be kept. Both CellRanger and Eoulsan provide in depth summary reports, however the modular design of Eoulsan allows for a much more flexible and responsive pipeline.

Chapter 3

Analyses of neural progenitors

In this chapter, I will present the results of the analyses I performed on a mouse dataset, obtained from neural progenitors. Due to genome annotation issues we encountered with the chick data, we decided to also investigate a publicly-available mouse embryo spinal cord scRNA-seq dataset from *Delile et al., 2019* [235]. The biological questions are expected to be valid in both model organisms.

I will first introduce some key concepts on the biological background, following with a few words on how to handle noise and confounding effects in scRNA-seq (methodological background). I will then present a selection of results we obtained out of my reanalysis of *Delile et al.* dataset. I will emphasize the approaches I selected in order to extract the most meaningful signal possible to tackle the biological questions.

3.1 Biological background

The vertebrate central nervous system (CNS) is a complex assembly of thousands of cell types organized in an exquisite manner to form functional neural circuits [236]. This amazing diversity of neuronal and glial cell types originates from a limited pool of neuroepithelial progenitors. Precise coordination between proliferation and differentiation is paramount to produce the correct amount of cells "at the right place at the right time" [237]. The progenitor pool is first amplified via proliferative symmetrical divisions (which produce two progenitors, thereafter called SYM divisions), progressively switches to neurogenic asymmetric divisions producing a progenitor and a committed progeny (ASYM), and finalise the differentiation with symmetrical terminal divisions producing two differentiating neurons (TERM) [238].

Asymmetric cell division is a fundamental mechanism to generate cell diversity. Much remains to be discovered about the molecular and cellular mechanisms underlying the decision to enter an asymmetric division, and its execution. In the context of the SYMASYM project, our objective is to characterize the key regulatory circuits inducing this transition by exploring transcriptomic changes that occur before and during this switch (Figure 3.1). The main biological focus is thus the search for transcriptomic signatures of SYM versus ASYM division modes in neuronal progenitors. It must be noted that the definition of both population signatures is not straightforward due to their coexistence in space and time.

Depending on the stage and regions in the developing CNS, the production of differentiating neurons may rely on either *direct* neurogenesis, whereby ASYM divisions from progenitors will produce a self-renewing progenitor and a post-mitotic differentiating neuron, or *indirect* neurogenesis, in which the "more committed" daughter cell is a progenitor which differs from the mother cell at the morphological and molecular levels and harbors a reduced proliferation potential. These intermediate progenitors are particularly abundant in the neocortex of mammals, where they accumulate basally and increase the neuronal output per surface unit. Several different subtypes have been described; their emergence and diversification during evolution is thought to

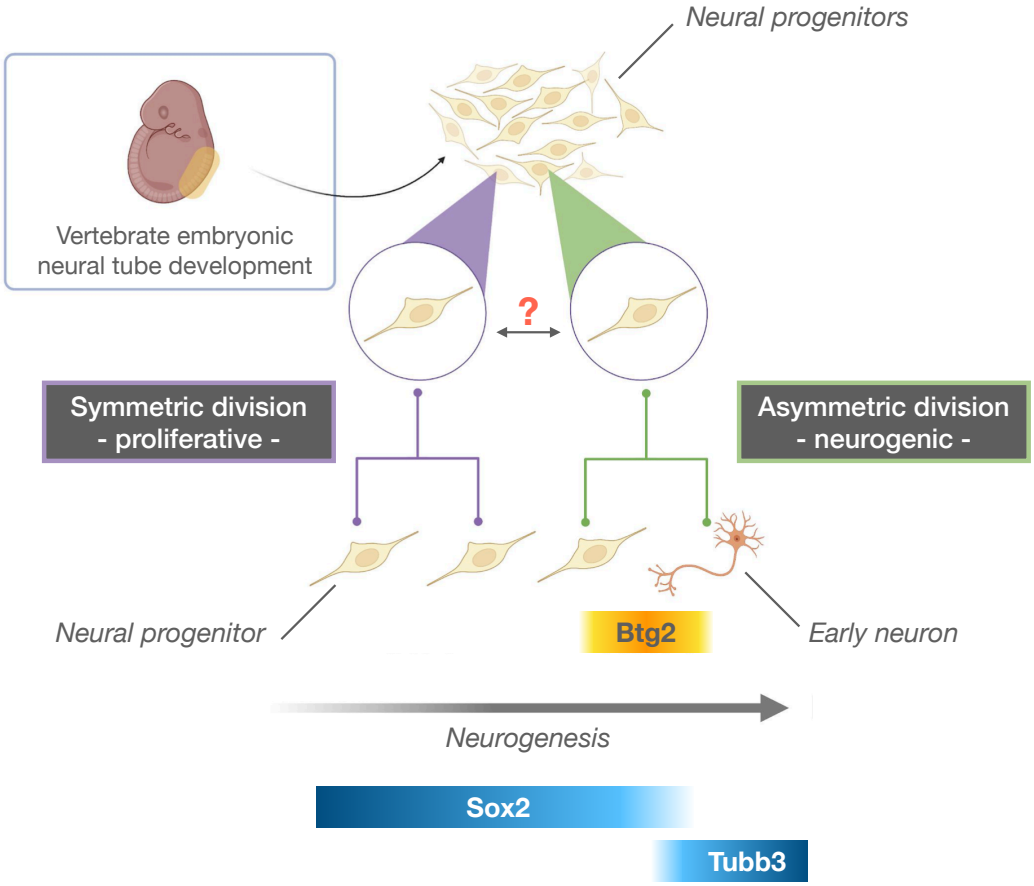


Figure 3.1 Biological context summary. In the context of neurogenesis, dividing neural progenitors may undergo either a symmetric division (a.k.a proliferative phase - SYM), or an asymmetric division (a.k.a. neurogenic phase - ASYM). As the neurogenesis progresses, the population of ASYM progenitors rises while the population of SYM progenitors declines. We aim at deciphering the transcriptional changes that drive the switch between SYM and ASYM populations. Key markers for this study are illustrated: in blue, the pan-progenitor marker Sox2 is expected to be detected in all progenitor cells, and the neuronal marker Tubb3 is expected to match early neuron emergence. In yellow, the transcription factor Btg2 is expected to specifically highlight ASYM progenitors (see below).

be a major driver of the extraordinary expansion and diversification of the mammalian neocortex, and in particular of primates.

While the evolutionary history and the biology of intermediate progenitors constitute fascinating biological questions, from a practical standpoint their diversity adds complexity in analysing the SYM/ASYM question. It is however expected that the fundamental mechanisms that control SYM versus ASYM modes of division in the CNS predate the appearance of intermediate progenitors and will be to some extent conserved at the molecular and cellular levels between structures that rely solely on direct neurogenesis, such as the spinal cord, and more complex brain regions that have acquired indirect modes of neurogenesis. For this reason, the Morin group decided to tackle the question of asymmetric division in the early embryonic spinal cord, where "only" two types of cells, apical progenitors in the ventricular zone, and post-mitotic cells in the mantle zone, are found at early neurogenic stages, thereby reducing the number of confounding factors (this dichotomy is of course a simplification, since complex patterning mechanisms of the progenitor and neuron populations are also present in the spinal cord, as detailed below in 3.1.2)

In order to identify the key transcriptional switches that occur during the neurogenic transition of vertebrate neural progenitors, we had initially planned to analyse the chick dataset mentioned in the previous chapters. Faced with the difficulties of the poor annotation of the chicken genome, we decided to reanalyze a public scRNA-seq dataset of mouse embryo at comparable developmental stages from *Delile et al., 2019* [235]. In the following sections, I will introduce some fundamental concepts on asymmetric cell division in the vertebrate CNS and highlight neural progenitors diversity.

3.1.1 Asymmetric cell divisions in the vertebrate CNS

Asymmetric cell division refers to the process by which a parent cell divides into two dissimilar daughter cells that differ both in types and functions [239]. Asymmetric divisions may be induced by either *extrinsic* signals (distinct environmental signals received by the sister cells) or *intrinsic* signals (Figure

3.2). In the latter case, the mother cell originally contains unique *fate determinants* (which may be a protein, an RNA, or even an organelle) that are geared towards each of the daughter cells, before cell division [240]. Since the following sections deal exclusively with this mode of division, it will be simply referred to as asymmetric division for simplicity.

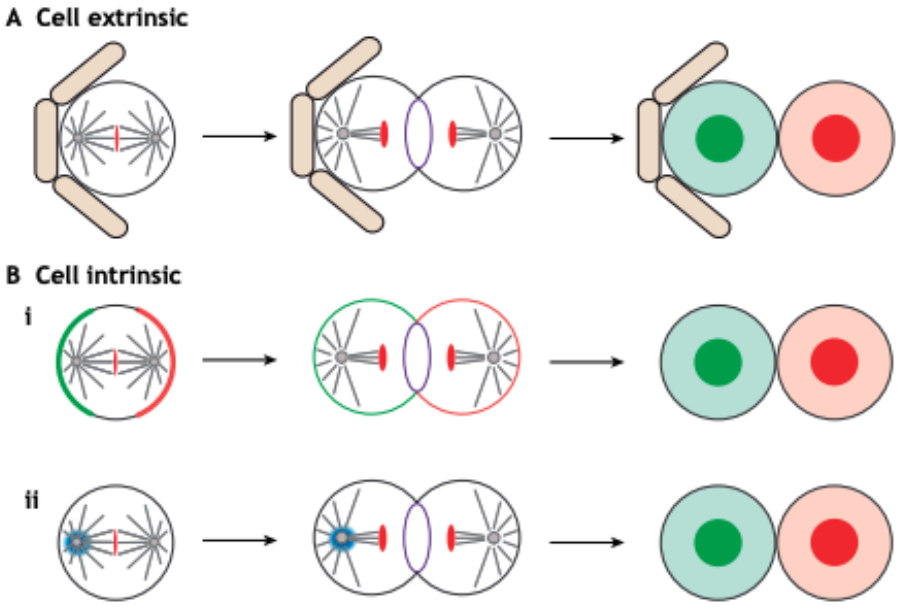


Figure 3.2 Basic concepts of asymmetry and biased segregation. A) Asymmetric cell division can be induced by cell extrinsic cues that are provided by the local niche. B) Asymmetric cell division can also be induced by cell intrinsic mechanisms. Asymmetric distribution of fate determinants (*e.g.* RNA, proteins) relies on i) cell polarity or ii) intrinsic asymmetry of the mitotic spindle. Other modalities exist, see *Sunchu and Cabernard, 2020* [241] from which this figure is adapted.

Asymmetric division is a key source of cell diversity, starting from prokaryotes up to highly complex multi-cellular organisms [239]. Historically, the first molecular characterizations of asymmetric division mechanisms were performed in invertebrate models, relying on "obligate" modes of cell division for the development of specific structures with invariant lineages (*e.g.* the first division of the *C. elegans* zygote, or the sensory organs of *D. melanogaster*). In addition to being amenable to large scale genetic screens, the invariant nature of these lineages allowed for the unambiguous characterization of phe-

notypes that could be explained by defects in asymmetric fate choices. These studies established the conceptual framework defining an intrinsic mode of asymmetric division: the *polarization* of the mother cell before division (*e.g.* fate determinants are unequally localized in the mother cell) lead to the unequal transmission to the daughter cells at the time of mitosis.

After mitosis, it will be differentially active between the sister cells and therefore dictate a different fate. In order to achieve asymmetry, some form of polarity must also be present in the mother cell to drive the unequal distribution of the fate determinants. Depending on the cell types and developmental contexts, the *polarity* element differs: examples of asymmetries fate determinant localization depending on canonical apico-basal or planar polarity abound, in which cases a key factor is coordination of the mitotic spindle with these axes of polarity to ensure asymmetric partitioning during mitosis. More recently, it has been shown that the built-in asymmetry in the mitotic spindle can be used to recruit fate determinants asymmetrically to its poles. While the fundamental concepts and mechanisms initially deciphered in invertebrate models have since been translated to more complex systems, it also emerged that the conservation is not complete, and that substantial diversity exists between species (extensively reviewed in *Sunchu and Cabernard, 2020* [241]).

Contrary to historical models, where invariant lineages and short developmental times allowed to link molecular and cellular asymmetries in dividing cells to fate decisions in the progeny, the study of asymmetric divisions in higher eukaryotes (and particularly in vertebrates) faces two key challenges: first, due to more complex tissue organization, longer cell cycles, and generally longer developmental and differentiation processes, it is difficult to monitor both cell division and daughter cells outcome with a cellular resolution.

In addition, in the developing vertebrate CNS, neural progenitors are usually divided into proliferative (*self-renewal*, a.k.a symmetrically dividing) and neurogenic progenitors (see Figure 3.1). They coexist in a highly regulated manner in space and time during development, as maintenance of proliferative progenitors is essential for prolonged growth, while neurogenic

progenitors producing differentiating neurons ensure that a functional nervous system emerges. Hence, at any given time point, the proportion of SYM/ASYM divisions differs between the developing CNS region. Furthermore, within the same region, these proportions also vary over time (Figure 3.3). This complex regulation lays the foundations for CNS organization in vertebrates, and in particular, is considered as a major player in the evolution of the neocortex [242].

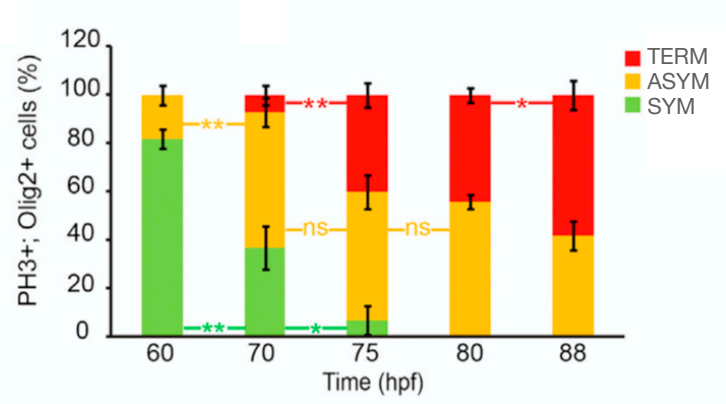


Figure 3.3 Differences between SYM and ASYM populations at various stages of development. Proportion of the diverse progenitor types at 5 different stages of development, as revealed by the differential expression of Sox2 and Btg2 transcription factors in dividing (PH3+) motoneuron progenitors ($Olig2^+$) in the chick spinal cord. We can observe that the ratio of SYM to ASYM populations decreases over time. Image adapted from *Saade et al., 2013* [243].

It has been shown that this heterogeneity is driven by varying degrees of extrinsic (tissue level) and intrinsic (molecular level) signals [244]. At the tissue level, signalling molecules such as BMP4 and SHH play a key role to maintain a balanced ratio between proliferation and differentiation in the vertebrate spinal cord [243, 245]. At the molecular level, the mechanisms that control whether a progenitor cell will enter a symmetric or asymmetric mode of division are assumed to rely on modifications of the cells transcriptional profiles. Some studies suggest that the transition of a cell from a symmetric to an asymmetric mode of division is irreversible and occur at successive stages of the neural developmental program. For all these reasons, it remains difficult to assign an identity (SYM or ASYM) to a progenitor at any

given time during development. Based on its dynamics of expression during the neurogenic transition, *Iacopetti et al., 1999* proposed that the Btg2 (a.k.a. Tis21) transcription factor is specifically expressed in ASYM progenitors [246], as illustrated in Figure 3.1 (in yellow). Consistent with this, functional studies that modify the switch from proliferation to neurogenesis specifically also modulate Btg2 expression (as shown in both mouse [247] and chick [243]).

Yet, In the absence of clonal analyses exploring the fate of daughter cells born from the division of Btg2^{+} or Btg2^{-} progenitors, the question remains open whether Btg2 is a *bona fide* marker of ASYM and TERM divisions, or simply a general marker of progression in the neurogenic process, whose expression dynamics broadly correlates with an increasing probability that any progenitor will undergo a neurogenic division. In addition, while Btg2 is described as an anti-proliferative factor, modulating its expression in embryonic and postnatal stages only marginally affects neurogenesis, suggesting Btg2 itself is not a key regulator of the mode of division. In any case, it can be expected that genes that are instrumental for deciding the mode of division are likely to harbor an expression dynamics that correlates with that of Btg2 in bulk or scRNASeq datasets. Along this line, *Arai et al., 2011* have shown in a pilot differential screen that a number of genes are differentially expressed between Btg2-GFP^{+} and Btg2-GFP^{-} progenitors, although these candidates were not tested functionally for a role in the mode of division [248].

Besides the expression of specific genes, another parameter that has been associated with neurogenesis is the regulation of cell cycle dynamics. There is a general consensus that the duration of the cell cycle increases as the neural tube develops [249], despite a great heterogeneity of the cell cycle length of neural progenitors at a given time point. For example, early studies using time lapse imaging of chicken neural tube slice culture showed that progenitor cell cycle length ranges from 9h to 28h [250]. A recent study using reporters of the different phases of the cycle established that this variability is distributed over all cell cycle phases, with heterogeneity in the G1 phase representing the main contributor of this phenomenon. Remarkably, G1 duration appears

to increase from one cell generation to the next, in parallel to the neurogenic potential [251]. Functional studies using drugs or genetic means to modulate the duration of specific phases of the cell cycle also modulate the neurogenic rate (reviewed in [252]). Based on these observations, it has been suggested that the duration of its different phases may directly impact the mode of division (SYM vs ASYM vs TERM), although few studies have actually investigated the phenotypes with a cellular resolution to analyse the fate of pairs of sister cells [253]. Interestingly, several studies have demonstrated that cell cycle regulators may act as transcriptional regulators of cell fate, independently of their role in cell cycle progression, but whether they act at the level of the mode of division remains to be elucidated.

3.1.2 Neural progenitors are highly diversified

Besides their division mode, neural progenitors also widely differ in terms of morphology and neurogenic potential. In the neural tube, progenitors are precisely organized by domains along the dorso-ventral (DV) axis (Figure 3.4). According to their localization, progenitors will thus acquire distinct characteristics (Figure 3.4-B shows the combinatorial expression of known markers for each domain). This phenomenon defines *patterning*. Each domain will give rise to different neuron subtypes, and has its own temporality. Ventral domains (p0, p1, p2, pMN, p3) tend to initiate neurogenesis earlier than dorsal domains (dp1-6). This ensures the progressive formation of all neuron subtypes. When neurogenesis is over, the progenitors remaining in the neural tube will then switch to producing glial cells [254]. Therefore, the immense heterogeneity in cell types within the CNS is also driven by dorso-ventral *patterning*.

The simultaneous presence of these intrinsically different progenitors which evolve at different paces towards neurogenesis, in addition to the strong effect of the cell cycle, lead to take into account multiple layers of complexity when analyzing scRNA-seq data. Each of these factors need to be considered in order to lessen their impact on scRNA-seq analyzes.

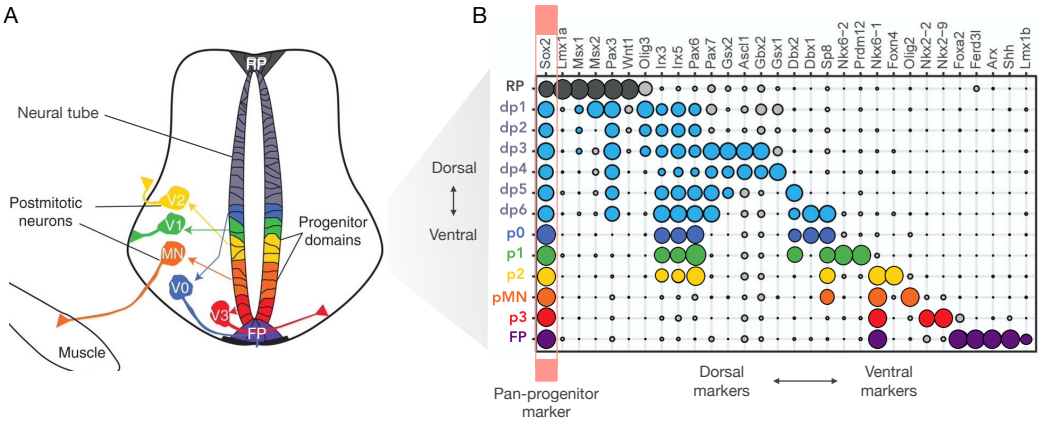


Figure 3.4 Illustration of progenitors diversity in the developing spinal cord. A) Spatial organization of progenitor domains in the neural tube (see B for domain names). Each neuron subtype originates from a distinct progenitor domain. Only neurons derived from ventral progenitors are shown. RP: roof plate. V0-V3: interneurons. MN: motor neuron. FP: floor plate. Image adapted from *Ribes and Briscoe, 2009* [255]. B) Bubble chart showing the combinatorial expression of known markers within neural progenitors, along the DV axis. *Delile et al., 2019* relied on these markers to identify each of the DV domains in their scRNA-seq analyses. The markers are organized from the most dorsal markers (left) to the most ventral (right), except for Sox2 which is a pan-progenitor marker. Image adapted from *Delile et al., 2019* [235].

3.1.3 Neural progenitors in scRNA-seq studies

Regarding single-cell approaches, several groups recently produced and analyzed neural progenitors scRNA-seq dataset. Indeed, *Delile et al., 2019* (Briscoe lab) took the lead by publishing an atlas of spinal cord gene expression [235]. This massive study, conducted as a cell atlas (33754 cells spread over 5 time-points) focuses on both progenitors and neurons diversity. Earlier this year (April 2021), the same laboratory published a preprint describing the human developing spinal cord (just as massive as the mouse study, with a total of 71219 cells over 4 time-points) [98]. This same month, *Scott et al., 2021* released a preprint focused on zebrafish spinal cord pMN progenitors (6489 cells, 3 time-points) [256]. Beyond studies targeting the neural tube, we can cite *Moreau et al., 2021* who analyzed the progenitors diversity and induced neuronal fate acquisition in the mouse cerebral cortex (4225 cells at a single time-point) [257]. It must be noted that all these studies rely on 10x

Genomics technology.

It should be noted that for the most part, these studies aimed at describing the diversity of progenitor and neuronal subtypes in specific regions of the developing CNS (cortex, striatum, spinal cord) or focused on gene changes that accompany the neural differentiation process. Although some of these datasets span several time points and should contain progenitors in both amplification and neurogenic phases, the specific question of the mode of division of progenitors (SYM vs ASYM) was not the focus of the analyses and has not been directly addressed.

3.2 Methodological background

3.2.1 Data correction

In the section 1.3.2, we emphasized on the importance of cleaning scRNA-seq data in order to properly pursue with the analyses. Even after normalization, the count matrix often contains unwanted sources of variation (a.k.a. *confounding factors*), such as cell cycle or batch effects. They may be technical or biological effects. Data correction precisely aims at taking into account these factors. In this regard, two main approaches can be considered: completely removing these effects or, alternatively, correcting for these effects to lessen their impact on downstream analyses. The decision of which approach should be adopted depends on the biological questions and the intended downstream analyses [130]. Accounting for technical effects such as batch effects is not a major issue in this thesis so we will not cover this topic here. I can however recommend *Chazarra-Gil et al., 2021* and *Tran et al., 2020* who recently reviewed dozens of batch-effect correction methods for scRNA-seq [258, 259].

As for biological effects, the most ubiquitous and popular data correction relates to the cell cycle effects. A simple linear regression against the cell cycle is often sufficient to remove these effects. For example, this is the approach chosen by Seurat [260] and Scanpy [137] developers. With this method, a *cell cycle score* for each phase (G1, G2/M and S - G0 cannot be clearly distinguishable) is computed for each cell, and the cell is then assigned a phase

depending on its highest score. The same approach can be used to remove any other biological effect (*e.g.* mitochondrial gene expression) provided that a list of marker genes is available. More advanced approaches may constitute interesting alternatives, such as scLVM [261], f-scLVM [262] or ccRemover [263] which rely on complex mixture models.

However, the lack of benchmark testing these approaches, and their different underlying assumptions (*e.g.* scLVM assumes that the cell cycle genes expression is similar among all cell types) complicate the task to compare and chose an appropriate approach. Moreover, removing a given biological signal may impact or hide other meaningful signals due to their interdependence. For example, for proliferative cell populations it is advised not to remove the whole cell cycle effect, but to specifically differentiate cycling cells (G2/M, S) and non-cycling cells (G1/G0). On the contrary, totally removing the cell cycle effect may improve trajectory inference [264]. Lastly, most recent approaches (such as Peco, Revelio or Tricycle) suggest to go beyond discretized cell cycle phases by rather modelling cell cycle as a continuous process by relying on periodic functions [265–267]. These approaches allow to precisely situate cells on the cell cycle continuum. Tricycle (available as a preprint since April 2021) even go further by being the first tool proposing an universal approach to infer cell cycle that is able to accurately infer cell cycle, independently of cell types or scRNA-seq protocols [267].

3.3 Analyses of mouse neural progenitors

3.3.1 Data preparation

The *Delile et al., 2019* single-cell atlas dataset gathers 12 cervicothoracic samples from mouse embryos at 5 successive time-points (from E9.5 to E13.5) [235]. The count matrix provided by the authors is not filtered. We downloaded it from https://www.ebi.ac.uk/arrayexpress/files/E-MTAB-7320/E-MTAB-7320.processed.1.zip. We thus decided to process to our own data cleaning, starting from the count matrix.

Data quality checking and filtering The dataset contains in total 12 replicates: 2 replicates for each of the E9.5, E10.5, E11.5 embryos, and 3 replicates for E12.3 and E13.5 (since embryos grow in size and cell number). Among them, 7 are females and 5 are males (we have been able to recover this information based on the female specific Xist expression). In total, 41025 cells were detected. The overall number of cells vary from 6672 (E11.5) to 10450 (E13.5). Figure 3.5-A shows the distribution of the total number of genes per cell, within each of the time-points (TP). Depending on the observed distributions, we then applied filtering with the following thresholds (see Notebooks[1] for more details):

- Filtering by mitochondrial (MT) genes: we excluded cells with a MT proportion larger than 8% (339 cells);
- Filtering by hemoglobin (Hb) genes (to exclude blood cells): we excluded cells with a Hb proportion larger than 0.3% (961 cells);
- Filtering by total number of UMI: in order to remove outliers, we removed the cells below the 0.5^{th} percentile and the ones above the 99.9^{th} percentile, for each TP independently. We repeated this operation with the total number of genes.

In the end, we recovered 39343 cells. With regard to gene filtering, we kept genes that are expressed at least once in at least 3 cells. The number of unique genes in the dataset thus dropped down from 21889 to 20082.

Normalization and dimension reduction We then log-normalized the data with Seurat function *NormalizeData*. Due to confounding factors that we observed in the dataset (see Figure 3.5-D), we then regressed out the confounding factors due to cell cycle and gender with Seurat function *ScaleData*. Regarding cell cycle, we isolated cells into "cycling" (G2/M and S) and "non-cycling" (G1), in order to preserve differences in cell cycle between proliferating and non-proliferating cells. We then performed dimensional reduction on the scaled data, and obtained a 2D representation of the dataset

[1] https://github.com/LehmannN/Mouse-progenitors-reanalysis

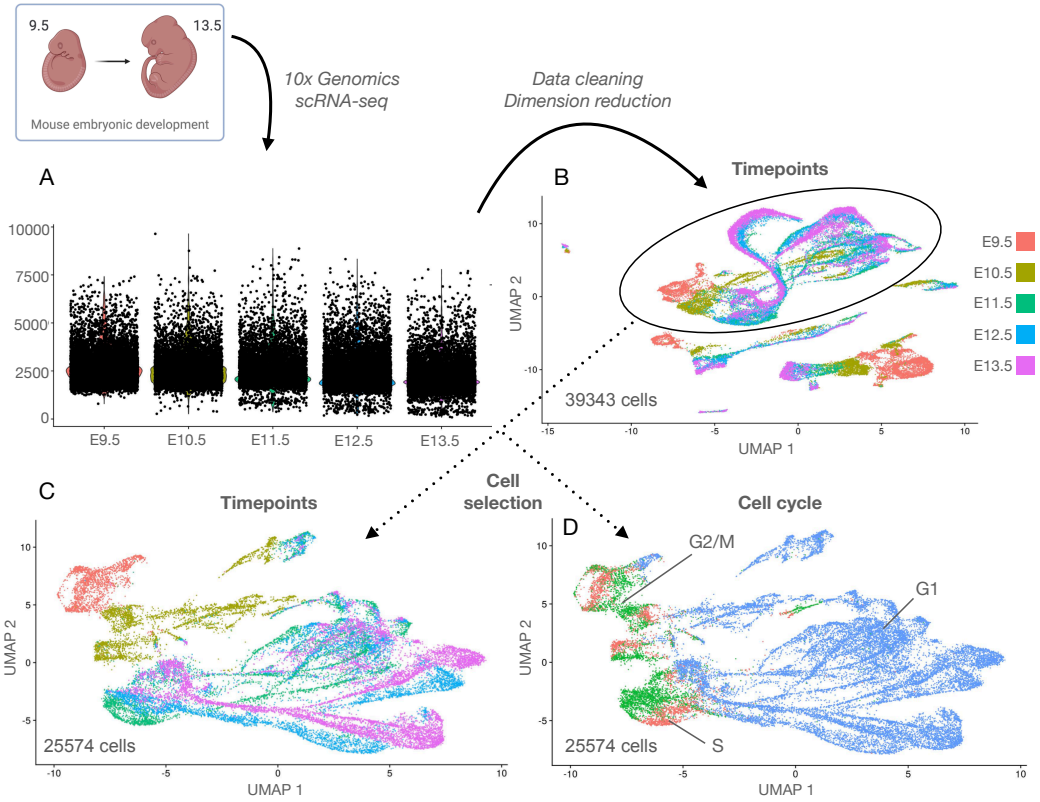


Figure 3.5 First exploration of *Delile et al., 2019* dataset. A) Distribution of the number of unique genes detected per cell, for each time-point. Each dot is a cell. B) UMAP of quality filtered dataset. There are 39343 cells overall. Colors represent the various time-points. C) UMAP of the dataset after selection of progenitors and neurons. There are 25574 cells overall. Same colors as in B. D) Same UMAP as in C. Colors here represent the cell cycle phase: S (red), G2/M (green), G1 (blue).

with PCA (not shown) and UMAP plots (Figure 3.5-B to D). We kept the first 30 components (or dimensionalities) of the datasets, after consulting the percentage of variance explained by each component.

Identifying cell types We then performed graph-based clustering with Louvain algorithm, as implemented in Seurat. Based on the expression of some known markers (described in *Delile et al.*), we removed all cells that were neither progenitors nor neurons, such as mesoderm or neural crest (Figure 3.5-C). At this stage, there were 25574 remaining cells.

All of the above mentioned steps were performed using R packages dedicated to single-cell analyses : Seurat v3 [137] and scater [138]. To ensure reproducibility, all code and resulting figures are stored in a bookdown (Rmarkdown) documents, shared with the experimental team in the form of HTML pages. These notebooks are also shared on GitHub, under the following link: https://github.com/LehmannN/Mouse-progenitors-reanalysis. To enable the experimental team to further explore manually the dataset, I also :

- designed a dedicated shiny application: https://symasym.shinyapps.io/app_web_v3;
- set up a SPRING server (SPRING is a tool dedicated to visualize scRNA-seq data, which is more interactive than shiny): https://kleintools.hms.harvard.edu/tools/springViewer_1_6_dev.html?client_datasets/NeuralProgenitors2/ [175].

3.3.2 Identifying the population of interest

During this first exploration, we realized that the clusters were driven by multiple factors: patterning (axis formation), time, cell cycle and embryo gender. Even when selecting just one of all the time point, all of these effects were still strong, hiding other signals. Thus, in order to extract meaningful information to answer our biological questions, we had to define a specific filtering and denoising strategy.

With this aim, our first objective was to specifically extract the progenitor populations (SYM and ASYM) and thus filter out neurons. Even though *Delile et al.* provide metadata with already computed cell classification, we chose to set up a different kind of cell classification. In fact, *Delile et al.* classification system is designed with the aim to build a comprehensive cell atlas. They impute cell types based on a two-level classification system: first cells are isolated into main types (*e.g.* progenitor or neuron), and then a second round of classification allow to assign each cell to a subtype (*e.g.* pMN, p0, p1). The main drawback of the first step of this classification (cell types) is that it is based on a very limited number of markers: progenitors are

defined by Sox2 expression and neurons by Tubb3 expression only. Undefined cells are assigned to the cell type they are the most similar with. On the contrary, the second-time classification (cell subtypes) are defined based on a combination of markers, as shown in Figure 3.4.

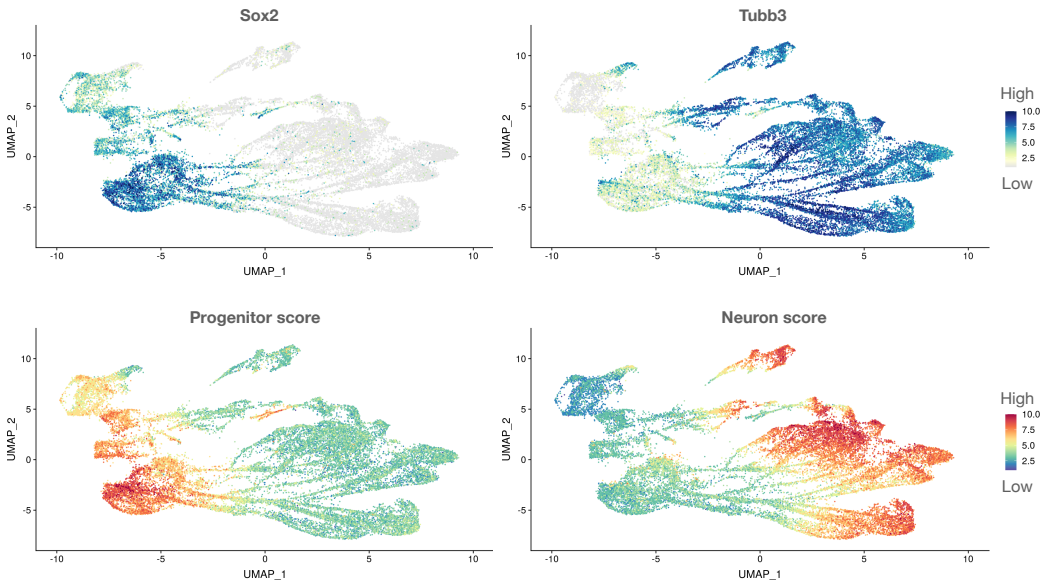


Figure 3.6 Scoring system to define progenitor and neuron populations. The two top UMAPs show the expression of Sox2 (progenitor marker) and Tubb3 (neuron marker). *Delile et al.* progenitor / neuron classification is based solely on these two markers. We defined a progenitor score and a neuron score based on a combination of about twenty markers (bottom UMAPs). Progenitor score markers: Sox2, Notch1, Rrm2, Hmgb2, Cenpa, Ube2c, Hes5, Fabp7. Neuron score markers: Tubb3, Stmn2, Nova1, Snrpn, Pcsk1n, Meg3, Rtn1, Stmn3, Mllt11, Mapt, Ina

Then, in order to specifically select the progenitors and separate them from the earliest neurons, we would rather rely on a scoring system, allowing us to be more permissive than a strict classification. We thus defined progenitor (P) and neuron (N) signatures scores with the Seurat *AddModuleScore* function. We set these scores based on both known and novel markers (originating from differential analysis we performed in the first dataset exploration), adding up to twenty markers (see Figure 3.6). These markers were defined in close collaboration with the experimental team.

We finally applied a k-mean clustering and defined thresholds to extract

the populations of interest. Ultimately, we ended up with 7441 cells of interest (see Figure 3.7). We kept the cells with the highest P / lowest N scores (clusters: 2, 3 and 4).

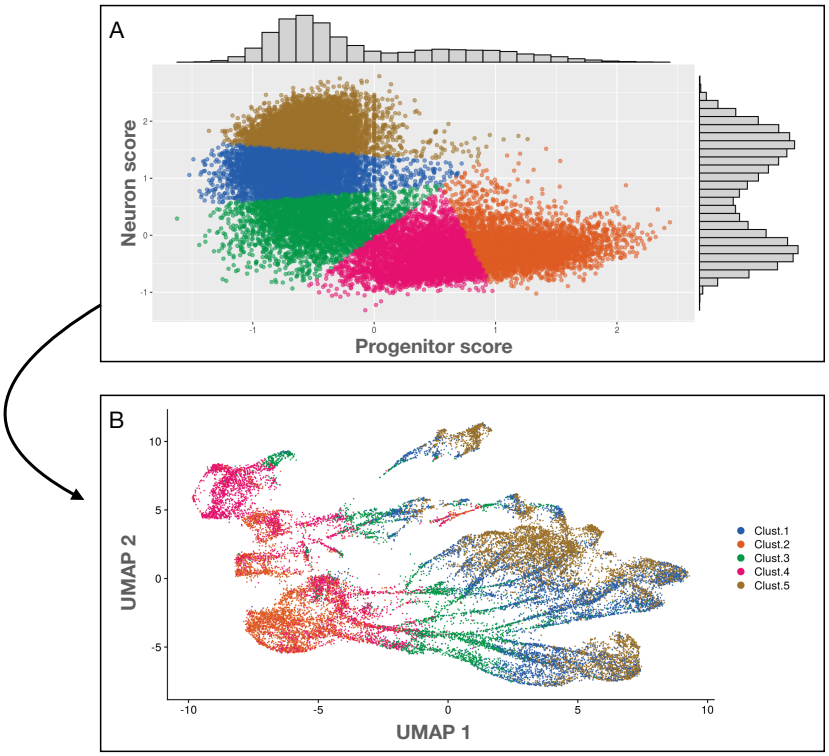


Figure 3.7 Clustering based on our scoring system. A) Scatter plot and its corresponding marginal histograms of neuron score versus progenitor score. Each dot is a cell. Each color correspond to a cluster. In the following analyses, we kept the clusters with the highest ratio of P/N : cluster 2 (orange), cluster 4 (pink), cluster 3 (green). B) Same UMAP as in 3.6. The colors represent the clusters as defined in A.

Up to this point, we have refined the data with the idea of getting a cleaner and clearer output to identify the SYM and ASYM progenitors. The clustering results (with Louvain method for community detection) and differential expression (with a negative binomial test, as recommended for UMI-based datasets) showed us that this was not enough to directly differentiate the two SYM/ASYM populations, since the populations were still mainly separated by the patterning signal on the DV axis, even though we could now identify a Btg2$^+$ cluster. Resulting UMAPs are shown in Figure 3.8.

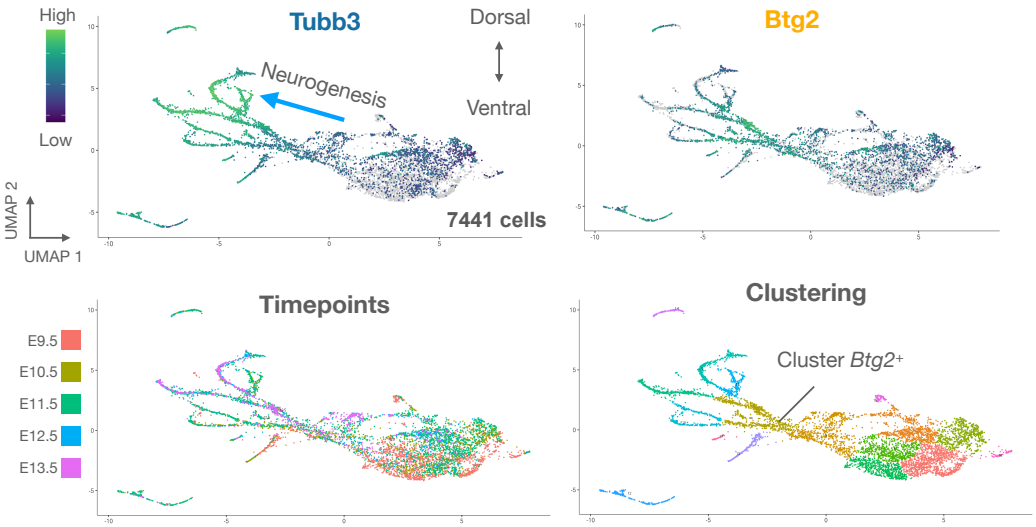


Figure 3.8 UMAP of the filtered dataset. We show here the resulting UMAP after filtering out the clusters with highest N scores (see Figure 3.7). There are 7441 remaining cells. Neurogenesis direction and DV axis are shown in the top left UMAP.

We thus decided to set a "denoising strategy" to find other sources of variation in the data. For this aim, we defined two different approaches:

- Pseudotime analysis ran separately on each of the subpopulations ;
- Analysis based on the binarized expression of Btg2, thus distributed into 2 groups: no expression and expression detected. This analysis was also run separately on each of the subpopulations. The overall results were then reassembled as described below.

3.3.3 Marker gene detection with pseudotime analysis

In order to identify the transcriptional changes in the transcriptome of the subpopulations of interest, we here relied on the pre-computed pseudotime scores of *Delile et al.* since they took the same approach we were interested in (pseudotime performed separately on each of the subpopulation, and then reassembled). To this aim, we will show here the results only on the pMN population for the sake of simplicity (247 cells). In all the following analyses, we thus relied on the pMN classification as described in *Delile et al.*.

	P-value	Number of cells	Cluster
Tmsb4x	2.07253422062957e-118	247	2
Ppp1r14a	7.23996189475735e-35	70	2
Hes6	3.10481203492362e-26	101	2
Neurod4	1.11126923184985e-24	49	2
Btg2	7.64878271591928e-18	99	2
Tubb3	5.4689979630698e-15	88	2
Neurod1	1.30703322382605e-11	18	2
H19	1.85578588574275e-10	114	2
Ppib	4.27785740239142e-10	218	2
Cotl1	9.66106108910397e-10	174	2
Eif3f	1.08987463216096e-08	227	2
Ebf2	8.41630773517702e-08	13	2
Ypel3	2.1632785413925e-07	73	2
Crmp1	1.08295121510633e-06	52	2
Selenow	5.82568229815042e-06	225	2
Rps9	8.24767004505995e-06	247	2
Rpl37a	1.77052876597124e-05	247	2
Atp6v1g1	3.06610800008248e-05	222	2
Stmn2	3.5198880880549e-05	45	2
Btg1	4.81729980674056e-05	142	2

Figure 3.9 Top 20 genes identified in the Btg2$^+$ pMN population (pseudotime approach). This table shows the top 20 differentially expressed genes of cluster 2 (see Figure 3.10), ordered by lowest p-values. We filtered out the genes detected in less than 10 cells.

Up to this point we performed differential expression between clusters of cells. With the aim of identifying genes whose expression varies over time, we applied here a different strategy. We relied on the trajectory inference dedicated tool, Monocle 2 [192], which provides a *differentialGeneTest* function especially designed to find differentially expressed genes as a function of time. These genes were then clustered with hierarchical clustering with the aim of ordering them all along the pseudotime axis. Then, we clustered the cells by similar expression profiles on the pseudotime axis with the partitioning around medoids algorithm (a.k.a. PAM). We chose the PAM algorithm in this situation, since it is more robust to outliers than a simple k-mean clustering. Figure 3.10 shows the resulting heatmap. From this heatmap, we can observe that Btg2 is detected in cluster 2. Thus, we also highlight here the top 20 most differentially expressed genes detected in cluster 2 in Figure 3.9.

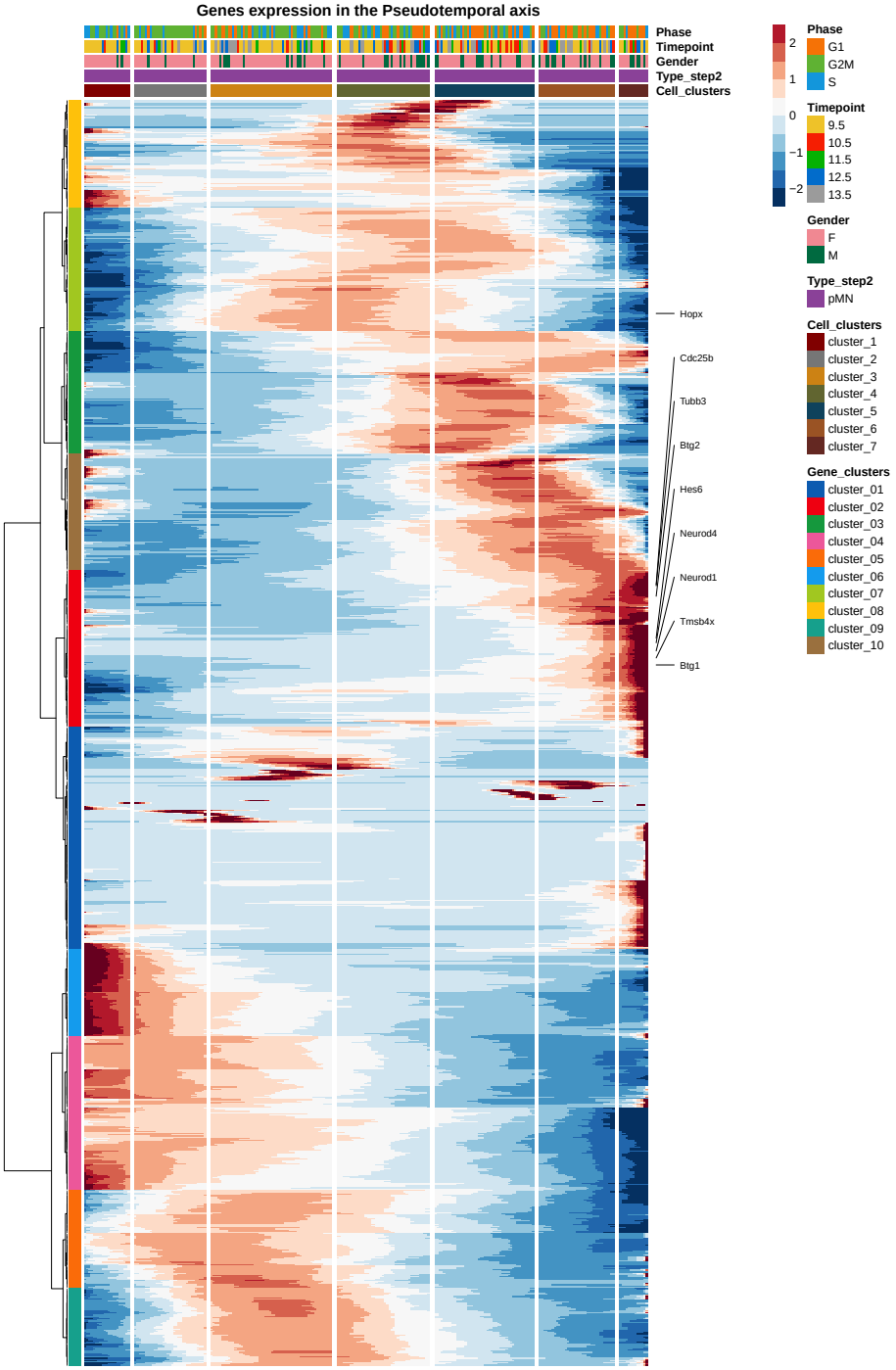


Figure 3.10 Gene expression along the pseudotime axis. Heatmap of the 1088 genes whose expression have been identified as variable along the pseudotime axis. Vertical legend (Gene_clusters) represent the genes with similar patterns. The expression of some marker genes is highlighted. Horizontal legends represent: i) the cell cycle phase, ii) timepoint, iii) embryo gender, iv) cell subtype (only pMN here), v) clustering on the cells (Cell_clusters). The blue/red color gradient represent the value of the Z-scores.

3.3.4 Signature extraction through isolation of DV domains

As a last approach, we tested an analysis based on the discretized expression of Btg2. We first isolated each of the 13 subpopulations of progenitors as described in *Delile et al.* (see Figure 3.4). For each of these populations, cells are classified depending on their expression of Btg2: no expression (group 0), expression detected (group 1). We then performed DE analysis with a negative binomial test, between each of these groups. Finally, we gathered all the subpopulations analyses and looked for the genes that are the most shared between the corresponding groups. Figure 3.11 highlights the DE genes found in the Btg2$^+$ populations, identified with this strategy.

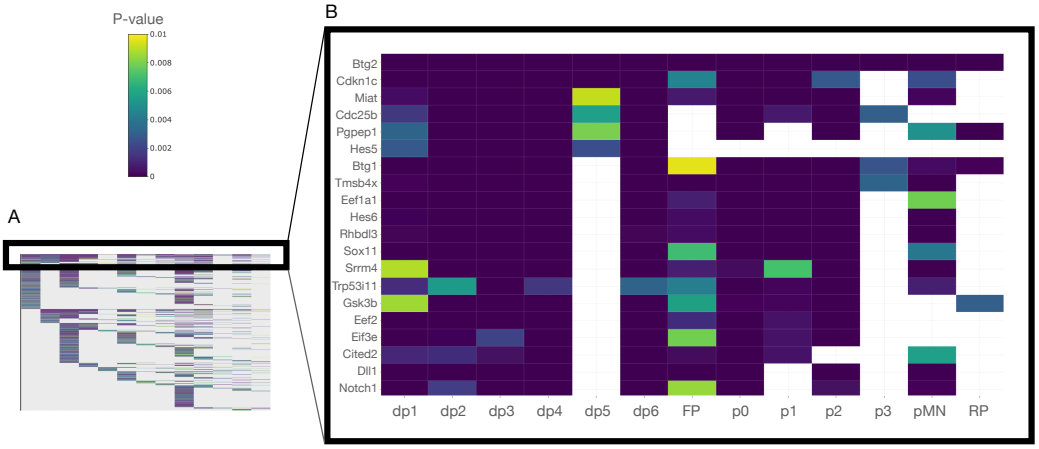


Figure 3.11 Heatmap of the most shared genes between Btg2$^+$ populations. Colors represent the p-values. A) We found 1294 differentially expressed genes in Btg2$^+$ populations, after extracting genes that are shared in a least 2 subpopulations. This massive heatmap is interactive and allows to zoom in and out to select a given population (made with plotly). B) Top 20 genes, extracted from zooming in A.

Finally, we compared the genes identified through the two strategies. Among the top 100 genes for each approach, we found 15 shared genes (apart from Btg2 and Tubb3): Btg1, Tmsb4x, Cdc25b, Hes6, Cdk2ap1, Dpysl4, Nfia, Ypel3, Afap1, Ascl1, Cd24a, Eif3f, Nfib, Ogt, Selenow. The function and biological relevance of these genes (and more from the complete lists) are currently under examination by Xavier Morin's team.

3.4 Conclusion

In conclusion, we performed a custom analysis based on a public atlas dataset, and selected the appropriate tools based on both the data requirements (atlas data might need more preprocessing than custom data) and the biological question. Specifically, we aimed at deciphering the transcriptional changes that drive the switch between SYM and ASYM populations. I adopted two complementary approaches: the first one based on pseudotime analysis, the second one based on the discretized expression of Btg2 among each of the DV domains. These approaches enabled to identify potential genes of interest: a first list of genes extracted from the population of cells suspected to contain ASYM divisions, analysed by grouping their expression profiles along a pseudotime; a second list of genes based on the Btg2 marker, considering that the ASYM division process is similar in all 13 subpopulations (thereby searching for a common signal among the different subpopulations). Altogether the genes in common are: Btg1, Tmsb4x, Cdc25b, Hes6, Cdk2ap1, Dpysl4, Nfia, Ypel3, Afap1, Ascl1, Cd24a, Eif3f, Nfib, Ogt, Selenow. These genes constitute a list of candidates to be further analyzed for their functions. Getting biological insights from this gene list is beyond the scope of my work, and is being handled by the team of Xavier Morin. The first steps are a literature search on each of these genes to get more insights on their function and possible involvement in the neurogenesis. The second step is to study the Gene Ontology annotation of these genes, and see if the associated terms are related to the neurogenesis or embryo development. The team will then decide if some genes are good candidates for experimental validation.

Chapter 4

An improved genome annotation workflow for scRNA-seq

In this chapter, I will highlight the issues we encountered when analyzing the chicken scRNA-seq data and the re-annotation strategy we developed to address this issue. I will first introduce some key concepts on genome re-annotation approaches. I will then present a selection of results we obtained on i) comparing the impact of using different reference annotations, ii) the pipeline we developed, iii) the results we obtained with this novel approach when reanalyzing the scRNA-seq dataset.

4.1 Methodological background

4.1.1 Chicken annotation

The chicken Gallus gallus has long been a model organism, particularly for developmental biology research, as well as an important organism in the agricultural industry. It was thus natural that sequencing this genome became a key project in the early 2000s [268], with the first assembly being released in 2004 by the International Chicken Genome Sequencing Consortium [ref chicken consortium]. Ensembl was directly involved in this consortium, to contribute to building the reference annotation. From this draft assembly, authors predicted between 20000 and 23000 protein-coding genes, with up to 10% of genes substantially truncated or missing. In parallel, efforts to identify genes from experimental data were conducted by transcriptomics sequencing of 20000 cDNAs, describing 12,000 genes [269]. Even including previous cDNA datasets, only 4560 cDNAs corresponded to full-length genes.

Since then, the genome assembly has been improved by collaborative efforts [270], with a last publication in 2017 (assembly version 5) [271]. This assembly benefited from long-read sequencing (PacBio) to refine the assembly of short reads into chromosomes, and attain a higher coverage. The current genome assembly named GRCg6a was released in 2018 by the Genome Reference Consortium, now in charge of improving the chicken assembly https://www.ncbi.nlm.nih.gov/grc/chicken.

Regarding the annotation, three main references are currently available, built independently from each other. The Ensembl annotation pipeline builds gene models by aligning publicly available cDNAs, protein sequences and RNA-seq data on the genome [272]. For this release, seven RNA-seq datasets generated with long reads were also included (see annotation report on Ensembl website : https://www.ensembl.org/info/genome/genebuild/2018_12_chicken_gene_annotation.pdf). The NCBI RefSeq annotation is built with the Eukaryotic Genome Annotation Pipeline https://www.ncbi.nlm.nih.gov/genome/annotation_euk/process/. The annotation report for release 104 of the chicken annotation specifies that chicken long reads from the SRA

database have been included to build this annotation (without further details) https://www.ncbi.nlm.nih.gov/genome/annotation_euk/Gallus_gallus/104/. The long reads RNA-seq datasets were included to facilitate the annotation with full-length transcript sequences ; in chicken, these were generated with the PacBio sequencer (Iso-seq approach) (reviewed in *Burt et al., 2018* [270]). The UCSC genome browser distributes a third reference annotation named refGene. This annotation is based on the realignment of RefSeq genes on the genome assembly, with different programs as used by the NCBI (see UCSC FAQ: http://genome.ucsc.edu/FAQ/FAQgenes.html#ens), and using exclusively NM and NR accessions, ignoring the XM and XR unvalidated gene prediction category of annotations (http://genome.ucsc.edu/cgi-bin/hgTrackUi?db=galGal6&g=refSeqComposite). It is thus expected to be very similar to the NCBI RefSeq annotation for high-quality annotated genes.

In Ensembl v101, 24356 genes are annotated, comprising 16878 protein-coding genes, 7166 non-coding genes and 312 pseudogenes https://www.ensembl.org/Gallus_gallus/Info/Annotation. NCBI RefSeq and UCSC RefGenes comprise 23726 and 6938 genes, respectively. Currently, the chicken annotation is not considered as high-quality as the human or mouse genomes, for which extensive functional genomics datasets have been generated in many tissues and developmental stages, through the ENCODE consortium [223].

4.1.2 Some considerations on annotation files

As mentioned in section 1.3.1, genomic annotations are stored either in GFF3 (*General Feature Format*) or GTF (*Gene Transfer Format*). Both formats contain the same information, which is organized differently though. GTF and GFF3 are tab-delimited plain text files with 9 fields per line, where each line is a feature. Features are organised in a hierarchical manner. Each gene has one or more transcripts, each of which has one or more exons. A transcript may also be coupled with a single CDS. The main difference stands in the way data is organised within each format: GTF is a gene-centric format, that can only handle an implicit hierarchy between features (linear relation-

ships), while GFF3 is a general annotation format, which can support more explicit, multi-level hierarchies, that may be thought of as a directed acyclic graph. In this respect, GFF3 provides directional relationships between features ("parent") and their subfeatures ("child") [273].

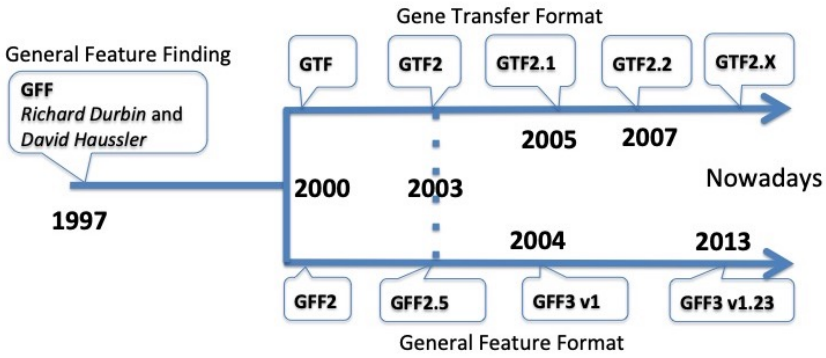


Figure 4.1 The evolution over time of the different annotation formats. Genomic annotation files have changed substantially over the years. Their evolution have resulted in the presence of multiple flavors and versions, but not all the programs stick to the latest versions. The dashed line shows that GTF2 are sometimes described as similar to GFF2.5. Image from https://agat.readthedocs.io/en/latest/gxf.html

Since 1997, where it was first introduced at a Conference on computational gene finding (Isaac Newton Institute, Cambridge, UK), the formats regularly evolved (Figure 4.1). However, while allowing a great flexibility, these formats are often criticized for their lack of standardization. In fact, most programs use annotation that do not comply exactly with the format criteria and incorporate subtle differences[1]. This sometimes complicate the task to share and use annotation formats within different pipelines, since each expect different specifications. Due to these limitations, we can expect these formats to keep evolving in the coming years.

[1] https://agat.readthedocs.io/en/latest/gxf.html

4.2 Contribution

In the following work, I designed and developed the whole re-annotation pipeline. I also performed all the analysis (pre-processing and downstream analysis) of both scRNA-seq and long-read datasets.

4.3 Improving scRNA-seq analysis in poorly-annotated genomes with matching long-read transcriptome

4.3.1 Introduction

A crucial step in the analysis of scRNA-seq data is the generation of a count matrix summarizing the signal detected for all the genes and all the cells. The content of the count matrix is directly dependent on the annotation of the genome, as only signals covering the annotated genes or transcripts are taken into account. Single-cell signal obtained with 10x Genomics technology is limited to the 3' region of the transcripts, which may lead to signal loss, particularly in poorly-annotated genomes. For example, the annotation of the chicken *Gallus gallus* is not yet as complete as human or mouse [274]. In order to assess to which extent such incomplete annotation affects scRNA-seq data analysis, we propose a novel approach to improve scRNA-seq analyses using long-read bulk transcriptome sequencing in matching cell samples.

We produced scRNA-seq data (10x Genomics / Illumina) from chicken cervical spinal progenitors at 66 hours of embryonic development. After quality filtering and alignment to the reference genome assembly (galGal6), up to 40% of the reads were not included in the count matrix. Visualizing the aligned reads in a genome browser revealed that significant signals fell outside of several known genes, and were thus not considered in the count matrix (as in the case of Sox2, a key marker for this study). Yet, the signal was often located in the vicinity of annotated genes. We thus concluded that loss of scRNA-seq signal was due to incomplete gene delineation, in particular

at their 3' extremities.

To address this issue, we generated bulk long-read RNA-seq (ONT technology) from samples matching our scRNA-seq data, in order to delineate the transcripts specific to these cells. ONT was chosen as it enables a sequencing of cDNAs from the 3' end, as for 10x Genomics / Illumina data. We exploited the long-reads data to expand the reference annotations collected from NCBI and Ensembl. We have evaluated various tools enabling the generation of gene annotations from aligned reads, such as StringTie2 [275] or Scallop [276], and selected the most appropriate ones to build our project-specific annotation. The resulting annotation combines the long-read bulk data, the scRNA-seq reads, and the reference annotation. Using this novel annotation, we were able to assign up to 87% of the reads at the genome scale, compared to 60% using only the reference annotation. We are currently evaluating the impact of this hybrid approach on the results of downstream scRNA-seq analyses. This approach could be used to improve scRNA-seq analyses of other poorly-annotated genomes, *i.e.* the majority of available eukaryotic genomes, at a reasonable cost.

All the results presented here are shared on GitHub, under the following links : https://github.com/LehmannN/scAnnotatiONT-paper (analyses) and https://github.com/LehmannN/scAnnotatiONT (pipeline).

4.3.2 Differences between the reference annotations lead to discrepancies in scRNA-seq analyses

When starting with the analysis of the scRNA-seq chick dataset, we first observed with the Ensembl annotation (v101) that only 60% of the mapped reads were assigned to a feature. We thus wondered to which extent the use of the NCBI/RefSeq annotation would improve the gene assignment. In this case, assignment performed even worse (42% of assigned reads). We realized that this was due to an improper naming of some of the genes (mostly mitochondrial genes) in the NCBI GTF file. Manually changing these names lead to a recovery of 68% of assigned reads (as shown in Chapter 2 - Figure

2.7). Faced with these discrepancies, we wondered to which extent the use of a reference annotation over another would impact scRNA-seq analyses. We also wondered if it would be possible to recover more than 68% of the scRNA-seq signal.

For the chicken, three main gene annotations are available from major genome databases (Ensembl, NCBI Refseq, UCSC refGene). To determine if these differences in reference annotations influence the subsequent analysis of scRNA-seq dataset, we processed the assignment step three times, with each one of these annotations. For the sake of simplicity, we downloaded the three corresponding GTF files from UCSC database: https://hgdownload.soe.ucsc.edu/goldenPath/galGal6/bigZips/genes/. The main advantage here is that these annotations share the same system of coordinates and the same annotation files standards, which facilitate their comparison[2]. The Gallus gallus genome assembly GRCg6a/galGal6 was also downloaded from UCSC: https://hgdownload.soe.ucsc.edu/goldenPath/galGal6/bigZips/galGal6.fa.gz.

	Genes	Transcripts	Transcripts per gene	Transcripts mean length	Exons	Exons per transcript	Exon mean length
Ensembl	24356	39288	1.61	2148.47	369120	9.4	228.68
NCBI	23726	62170	2.62	4262.04	796527	12.81	332.66
UCSC	6938	7482	1.06	2047.9	66442	8.88	230.61

Figure 4.2 General statistics between the three reference annotations.

We first calculated raw statistics, obtained with Mikado [277], and compared these values for the three annotations. Each annotation contains 24356, 23726 and 6938 genes (Ensembl, NCBI and UCSC respectively). We can observe that although Ensembl contains only 630 more genes than NCBI, these two references differ widely in terms of transcript number (over 1.5 as more in NCBI than Ensembl) and in transcript mean length (twice as long in NCBI

[2]Downloading the annotation files on each of their corresponding website imposes 1) the use of a conversion system and 2) multiple files manipulations in order to make each file obey the same standards. At the beginning of the project, I chose this approach, I thus participated in completing this GitHub repository, which contains many chromosome/contig name mappings between various databases (UCSC, Ensembl, Gencode): https://github.com/LehmannN/ChromosomeMappings.

annotation). UCSC annotation shows a very low number of genes compared to the other two, as it was expected (see 4.1).

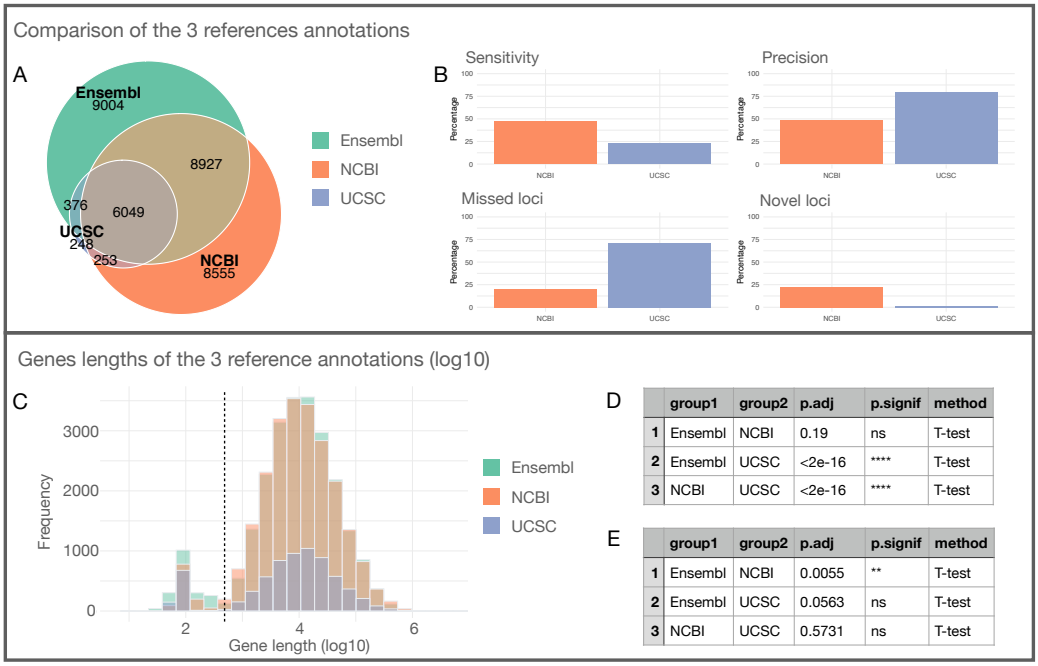


Figure 4.3 Comparison of the 3 reference annotations. In the top panel, we compared the overlaps between the 3 reference annotations. A) Venn diagram of the overlaps between the three chick reference annotations. Overlaps are defined the same way as in Figure 1.20 (see 1.4). B) Comparison of NCBI and UCSC annotations with Ensembl taken as reference (in terms of overlap, not based on the gene ID). These figures were obtained with GffCompare [278]. C) Gene length distribution of each annotation (as histograms). Y axis shows the total number of genes. X axis shows the gene lengths (in log10). It was cut at 10Mb for visualization purposes. Since we observed a bimodal distribution, the black dotted line separates the smallest genes ($<$ 500 nucleotides) from all the other genes ($\geq$ 500 nucleotides). D) Pairwise comparisons of the longest genes ($\geq$ 500 nucleotides) from the three annotations, with paired t-tests. E) Pairwise comparisons of the shortest genes ($\leq$ 500 nucleotides) from the three annotations, with paired t-tests. Coloring scheme is the same for all plots: Ensembl (green), NCBI (orange) and UCSC (blue).

Following these observations, we then investigated i) the overlap between the three references, and ii) their differences in gene lengths. Figure 4.3-A and B show the overlaps in terms of coverage (not based on the gene ID) between the three references. In this case, we can observe that only 6049

genes are in common between all three annotations, covering almost the entire UCSC dataset (Figure 4.3-A). When comparing NCBI to Ensembl, we observe that both sensitivity and precision are equal to 50%, consistent with the overlaps observed between both annotations in the Venn diagram (Figure 4.3-B). Moreover, about 25% of the genes are specific either to Ensembl or NCBI (missed loci and novel loci in 4.3-B).

In terms of gene lengths, we can observe a bimodal distribution in all three cases: a first group of small genes around 100 bp, and a second group of genes (the majority) between 1kb to 100kb (Figures 4.3-C, D and E). We isolated the smallest genes from the rest of the distribution, leading to two distinct normally distributed groups of genes. For both groups, the differences in mean have been evaluated with paired t-tests. We observe that for the smallest genes (Figure 4.3-D), there is no difference between Ensembl and NCBI annotations. On the contrary, the differences are statistically significant between UCSC/NCBI and UCSC/Ensembl annotations. Regarding the longest genes (Figure 4.3-E), there is little (Ensembl/NCBI) or no difference (UCSC/Ensembl and UCSC/NCBI) between the annotations. Moreover, when we take all the values together, the median values are very similar, concordant with the distributions observed in Figure 4.3-C: 9414, 9616 and 9268 (Ensembl, NCBI and UCSC respectively).

In order to estimate the impact of all three annotations on scRNA-seq analysis, we processed the scRNA-seq data following these steps:

1. Pre-processing with Eoulsan (see Chapter 2), with each one of the annotations;
2. Data cleaning and preparation performed globally the same way as the mouse dataset (see Chapter 3);
3. Secondary analyses performed with Seurat Louvain clustering (resolution of 0.5) and differential expression (DE) analysis performed with a negative binomial test, as recommended for UMI-based datasets. To ease comparison, DE genes represent the set of all genes identified as DE (independently of which cluster) within each of the analyses.

All three analyses have been performed the same way (*e.g.* same threshold values) to ease comparison. Regarding the cell filtering, we kept cells in which we detected at least 1000 genes, and removed the cells where the proportion of mitochondrial genes were higher than 20%, due to the observed distribution in this dataset. It must be noted that, in order to stay as objective as possible, we did not alter the GTF files downloaded from UCSC, which means that NCBI (and UCSC) dataset do not include the mitochondrial genes (as mentioned above). Regarding gene filtering, we filtered out the genes that are detected in less than 5 cells. Figure 4.4 highlights the main statistics obtained after processing the scRNA-seq dataset with the three references. We can note that the numbers of genes (mean genes per cell, total number of genes and number of DE genes) are significantly higher with NCBI annotation, despite the loss of all mitochondrial gene counts in the pre-processing steps.

	Original number of cells	Number of cells after filtering	Mean number of genes per cell	Number of genes after filtering	Number of DE genes
Ensembl	2481	2353	4176	15680	4233
NCBI	2481	2416	4994	17192	5473
UCSC	2481	2117	2164	5485	2248

Figure 4.4 Summary table of the statistics obtained after scRNA-seq processing with the 3 annotations.

The dataset corresponds to a population of different cell types, including neurons, progenitors, neural crest and mesoderm (more details in Chapter 4). In order to identify the cell populations of interest, we set four cell type scores based on the combination of markers as described in *Delile et al., 2019* (see also Chapter 3) [235]:

- **Progenitor score**: SOX2, LMX1, MSX1, MSX2, PAX3, WNT1, OLIG3, IRX3, IRX5, PAX6, PAX7, ASCL1, GBX2, GSX1, DBX2, DBX1, SP8, NKX6-2, PRDM12, NKX6-1, FOXN4, OLIG2, NKX2-2, FOXA2, FERD3L.
- **Neuron score**: TUBB3, ARX, SHH, LMX1B, POU4F1, LHX2, BARHL1, BARHL2, ATOH1, FOXD3, LHX1, LHX5, ISL1, TLX3, OTP, LBX1, PAX2, GBX1, BHLHE22, PTF1A, DMRT3, WT1, EVX1, EVX2, PITX2, EN1.

- **Neural crest score**: SOX10, TPM1, LM04, NPR3.
- **Mesoderm score**: FOXC1, FOXC2, TWIST1, TWIST2, MEOX1.

We then performed cell subtypes identification (RP, dp, p, pMN, p3, FP - p referring to the populations p0, p1 and p2), based also on the mouse markers described in *Delile et al., 2019*. For complete details on the analyses, one can refer to the following notebooks: https://github.com/LehmannN/scAnnotatiONT/tree/paper/pipeline_output_references (one notebook for each annotation).

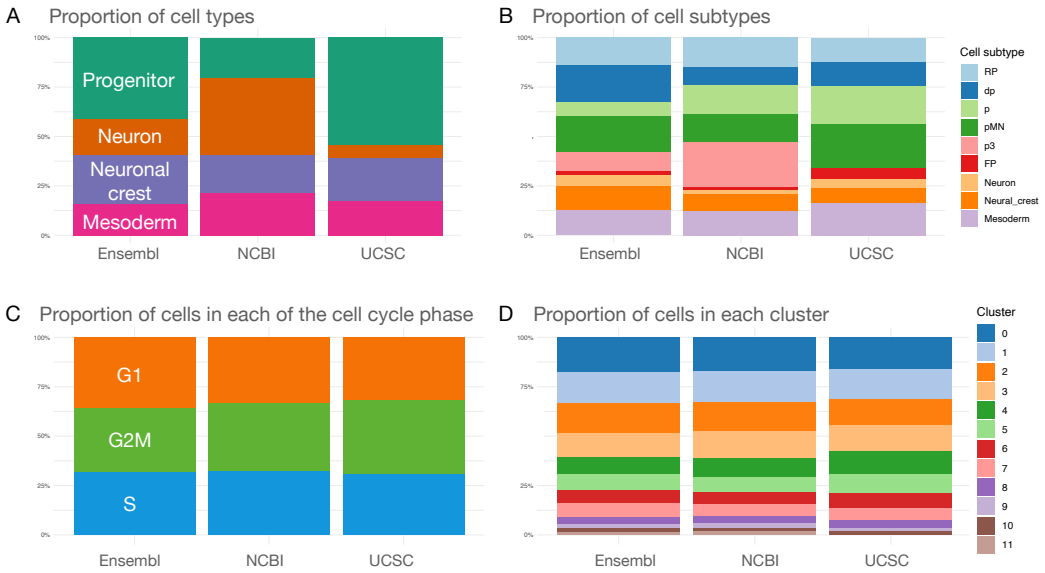


Figure 4.5 Use of different annotations impacts estimation of cell proportions. A) Proportion of cells types for each annotation. B) Proportion of cells subtypes for each annotation. C) Proportion of cells in each phase of the cell cycle. D) Proportion of cells in each cluster.

The differences observed in terms of cell types and subtypes identification between the three annotations are highlighted in Figures 4.5 and 4.6. We can notice that these proportions vary widely between the three annotations. Regarding cell types identification, over 54% of the cells are designated as progenitors with UCSC annotation, while this proportion drops to 41% with Ensembl and 20% with NCBI (Figure 4.5-A). Moreover, none of the markers of the p3 population could be identified with UCSC, thus leading to the loss of this population (Figure 4.5-B). On the contrary, p3 population forms the

major subtype population with NCBI. We also observe that the proportions in both cell cycle phase assignment and in the diverse clusters are consistent between all three annotations 4.5-C and D. Regarding clustering, the main difference is that 11 clusters were found with the UCSC annotation, in contrast with the two others where 12 clusters were found. This probably results from the lower complexity of the UCSC annotation. From these results, we can conclude that the use of different annotations alter both cell type identification and clustering in scRNA-seq analyses.

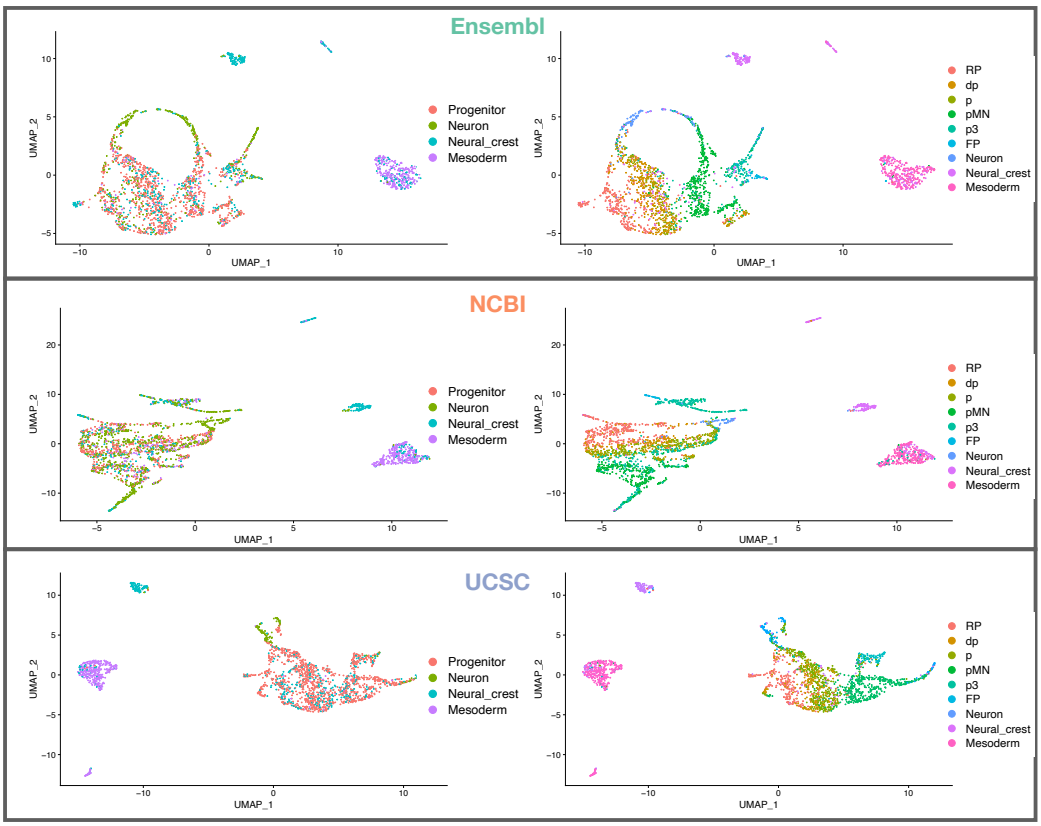


Figure 4.6 UMAPs showing discrepancies in scRNA-seq analyses with the different annotations. The three panels refer to data processed respectively (from top to bottom) with Ensembl, NCBI and UCSC. Apart from differences in the general UMAP structure, we observe differences in cell type identification (*e.g.* almost no neuron in UCSC, much more progenitors in Ensembl than in NCBI).

Finally, we compared the overall DE genes obtained with the three annotations. The number of DE genes are 4233, 5473 and 2248 for Ensembl, NCBI and UCSC respectively (Figure 4.4). Here, we will only focus on the

differences between Ensembl and NCBI annotations. We performed several selections among all DE genes (see below). Figure 4.7 highlights the 1607 DE genes that are mutually exclusive of each annotations (*e.g.* genes identified as DE in NCBI, but not in Ensembl, and *vice versa*). We could identify genes such as SNAI1 (DE only with Ensembl) and POU3F3 (DE only with NCBI) that are both involved in neurogenesis. Moreover, expression levels from each annotation for genes selectively identified only using NCBI or Ensembl showed a poor correlation, with significantly higher expression in the annotation in which the gene was identified as differential. Taken together, these observations suggest underlying differences in chicken Ensembl and NCBI annotations contribute to inconsistencies in differential gene analysis from scRNA-seq data. The same observations was made with UCSC/Ensembl and UCSC/NCBI comparisons (data not shown).

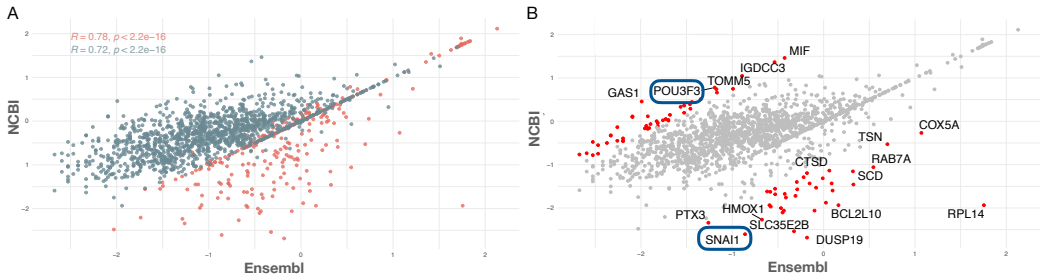


Figure 4.7 Correlation of expression levels between mutually exclusive DE genes. A) Correlation of expression levels between NCBI and Ensembl annotations, on the selection of the mutually exclusive DE genes (log10 counts). Genes that are only DE with NCBI are colored in blue-gray, and the ones specific to Ensembl are in orange. R values indicate Spearman correlation for each annotation. B) Same plot as A, with colors highlighting the genes with the highest count differences between the two annotations (red dots).

4.3.3 3'UTRs poor annotations seem to be the major source of scRNA-seq signal loss

Since we constructed the scRNA-seq libraries from 10x Genomics data, we expected a biased 3'end gene detection. We thus decided to specifically investigate whether the above-mentioned discrepancies were due to differences in the 3'UTR annotations between the three references. To this aim, we first

extracted the 3'UTR annotations from each of the GTF files to calculate general statistics. Surprisingly, we found that more than half of the genes annotated in Ensembl lacked a 3'UTR (only 11768 have a 3'UTR). In comparison, this value drops to 30% (16559) in NCBI and 20% (5531) in UCSC. Moreover, we observed that NCBI 3'UTRs are significantly longer than Ensembl and UCSC's (Figure 4.8). The mean lengths are 696, 1453 and 787 bases, while the medians values are 481, 795 and 457 (Ensembl, NCBI and UCSC respectively).

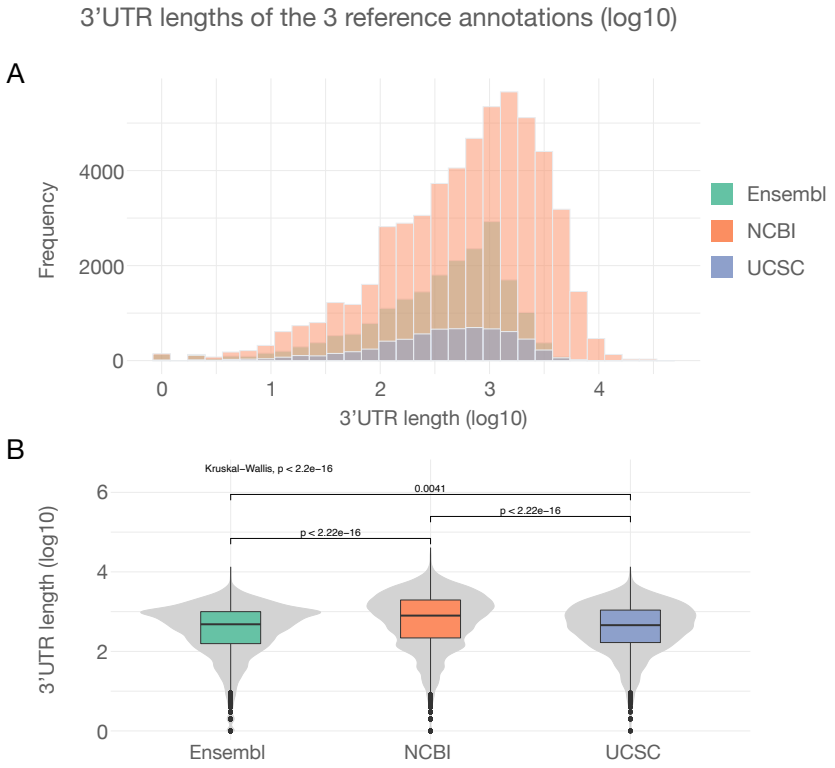


Figure 4.8 Comparison of the 3 annotation in terms of 3'UTR lengths. A) 3'UTR length distribution of each annotation (as histograms - in log10). B) 3'UTR length distribution of each annotation (as boxplots - in log10). The significance of gene distributions between the three annotations have been estimated by a Kruskal-Wallis test (non-parametric test). Pairwise comparisons between annotations have been performed with Wilcoxon paired tests. Coloring scheme is the same for all plots: Ensembl (green), NCBI (orange) and UCSC (blue).

In order to estimate if the differences in 3'UTR amounts and lengths contribute to the discrepancies in our scRNA-seq analyses, we investigated

in the first instance some genes among the ones identified as differentially expressed. Figure 4.9-A shows the correlation of expression levels between genes that are identified as DE in both NCBI and Ensembl (3218 genes in total). The highlighted genes (red dots) are the ones with the highest count differences between the two annotations. We built a UCSC trackhub[3] for the purpose of identifying if this set of genes displayed differences in their 3'UTR annotations. Strikingly, we noticed that all 18 genes had no or shorter 3'UTR in the annotation where they show the lowest counts. In particular, we investigated HES6, that has been identified as a potential gene candidate in the mouse study (see Chapter 3). Figure 4.9-B is a snapshot of our UCSC trackhub at HES6 location. We noticed that NCBI 3'UTR annotation (in blue) is much longer and cover all of the scRNA-seq HES6 signal. In contrast, the Ensembl HES6 annotation does not contain a 3'UTR, which leads to the loss of almost all the HES6 signal. We show the resulting UMAPs of HES6 expression, in the context of the Ensembl (Figure 4.9-C) and NCBI annotations (Figure 4.9-D).

Similar results were found when investigating genes among all the genes shared between the two annotations (10457 in common). Precisely, we considered COTL1 (a microglia-specific marker [279, 280]) and noticed that its NCBI annotation is lacking a 3'UTR, as opposed to the Ensembl annotation (Figure 4.10). Ensembl resulting UMAP of COTL1 expression displays a wide range of levels of expression over all the populations of cells, although lower in Mesoderm (Figure 4.10-C). Expression levels built on NCBI annotation are almost all equal to zero (Figure 4.10-D).

These results thus support our hypothesis that the differences in 3'UTR amounts and lengths play a key role in the discrepancies observed in our scRNA-seq analyses. In order to address this issue, we set a strategy to improve the reference annotations, based on both scRNA-seq data and matching bulk long-reads (ONT). ONT was chosen as it enables a sequencing of cDNAs from the 3' end, as for 10x Genomics / Illumina data. We thus exploited the

[3] https://genome-euro.ucsc.edu/cgi-bin/hgTracks?db=galGal6&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chr1%3A136260546%2D136266566&hgsid=276668722_AACY0aH4H2dam8fdwESNlBfTR6an

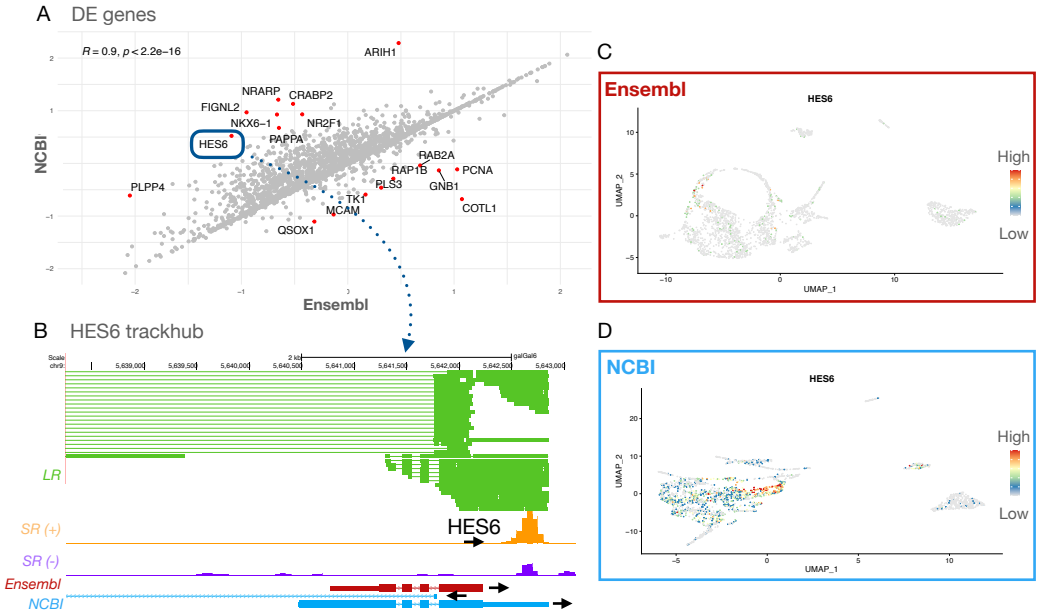


Figure 4.9 Incomplete 3'UTR annotation of HES6 leads to discrepancies in scRNA-seq analysis. A) Correlation of expression levels between NCBI and Ensembl annotations, on the selection of genes that are DE with both annotations (log10 counts). Genes with the highest count differences between the two annotations are indicated by red dots. R value indicates the Spearman correlation. B) UCSC trackhub of HES6. The tracks represent: LR reads (green), SR forward reads coverage (yellow), SR reverse reads coverage (purple), Ensembl v101 annotation (red), NCBI annotation (blue). The LR protocol is unstranded, we are thus unable to isolate forwards from reverse reads. Orientation of the reference transcripts and HES6 associated reads are indicated with the black arrows. C) UMAP of the chick dataset generated with Ensembl annotation. The colors represent the level of expression of HES6. We notice that HES6 seems to be almost completely absent from the dataset. D) UMAP of the chick dataset generated with NCBI annotation. Color gradient is the same as in C. We notice here that HES6 expression is spread among the progenitor population.

long-reads data to expand the reference annotations. These data have been generated by the team of Xavier Morin, as part of the SYMASYM project, on the same type of cell samples than the ones used for scRNA-seq analyses. I performed the analysis of this data with Eoulsan ONT specific workflow. I also assessed the raw data quality with the dedicated tool ToulligQC[4], that is also developed at IBENS. Results of these analysis are shown in the Annexes

[4]https://github.com/GenomicParisCentre/toulligQC

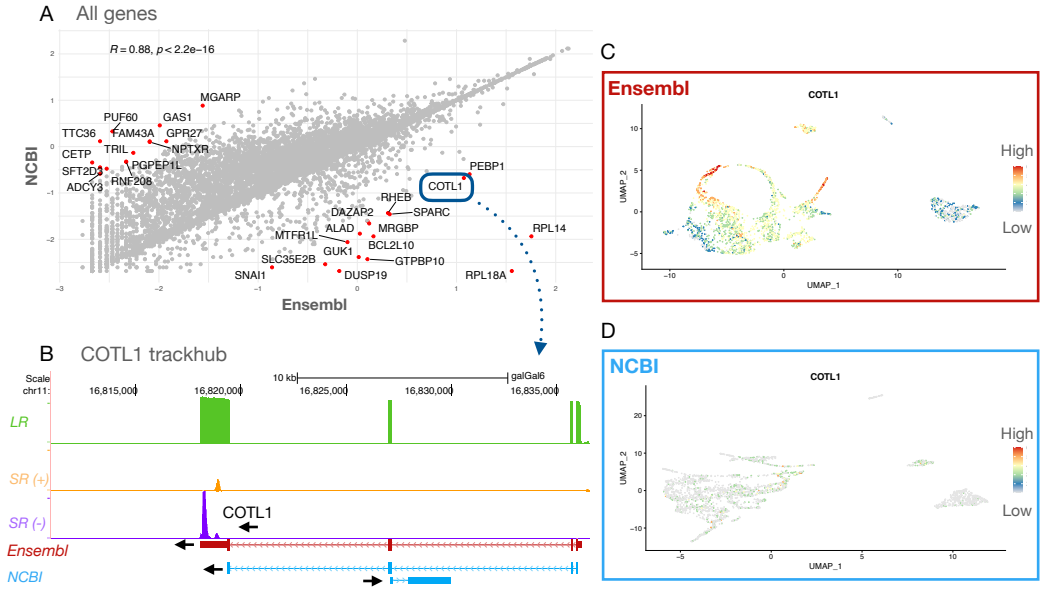


Figure 4.10 Incomplete 3'UTR annotation of COTL1 leads to discrepancies in scRNA-seq analysis. A) Same plot as Figure 4.9-A, except that we represent here all the genes that are detected with both Ensembl and NCBI annotations. B) UCSC trackhub at COTL1 location. See plot description in 4.9-B (the only difference is that, here, LR track is shown as coverage - not raw reads - due to too many reads in this position). C) Same plot as Figure 4.9-C, with COTL1 level of expression instead of HES6. C) Same plot as Figure 4.9-D, with COTL1 level of expression instead of HES6.

B.

4.3.4 A pipeline to improve annotation for 3' biased scRNA-seq data

In order to build a dedicated re-annotation pipeline, I have first evaluated the two top reference-based tools dedicated to generate an annotation from aligned reads: Stringtie2 [275] and Scallop [276]. Moreover, they are both able to handle long-reads (LR) as well as short-reads (SR). I have also tested the recent update of Scallop (Scallop-LR) more specific to long-reads, but it performed poorly on ONT reads since it was first designed for PacBio data (data not shown) [281]. I have integrated this approach into an open-source and reusable bioinformatics pipeline built with Nextflow [282], that we called

scAnnotatiONT. We chose Nextflow since it is best suitable to handle the rapid integration of multiple software packages and conflicts that may occur between them. To ensure reproducibility, all the following analyses have been performed in a single Conda environment. Figure 4.11 shows an overview of the scAnnotatiONT pipeline, and the tools implemented in the pipeline are shown Figure 4.12.

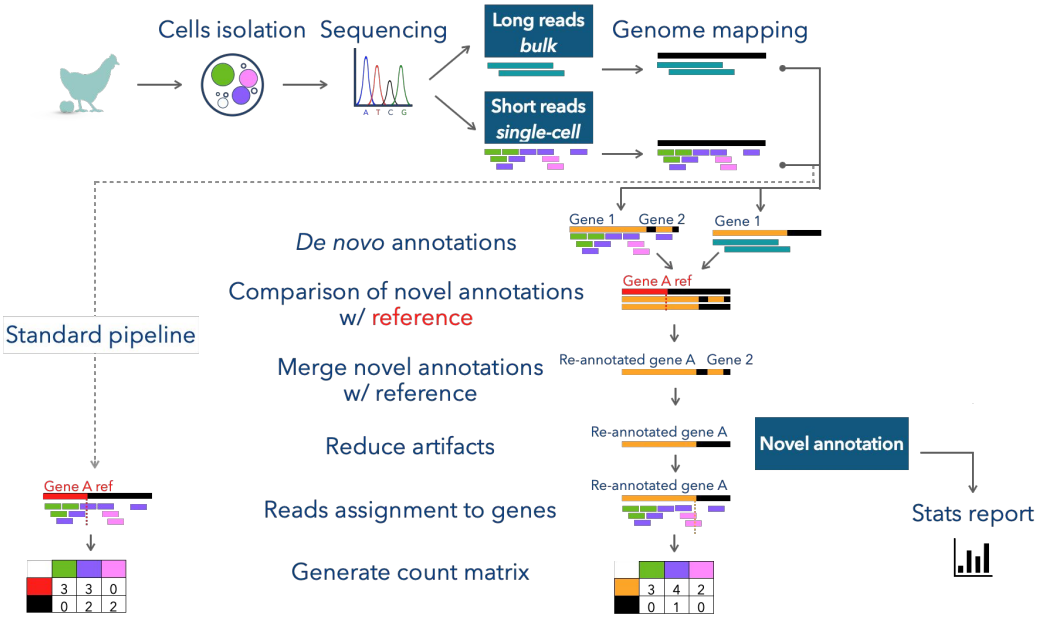


Figure 4.11 Dedicated scRNA-seq re-annotation pipeline. Our pipeline relies on both bulk long-reads and short-reads from scRNA-seq to improve the reference annotation. It also includes a step aiming at reducing the artefacts, such as small artificial genes (based on the read coverage and gene length). However, this step is still under testing. Thanks to the novel annotation, we can recover scRNA-seq signals that were lost due to lack of feature in the original annotation.

Starting from mapped reads, the first step of the pipeline aims at generating *de novo* annotations, guided by the reference in order to improve the assembly process. The output thus includes both expressed reference transcripts, as well as any novel transcripts that could be reconstructed. With Stringtie2, LR and SR can be assembled separately or simultaneously, although our results show that the quality of the annotation is higher when processing the reads separately (see below), due to inherent differences between 10x Genomics SR and ONT LR (when performed separately, the pa-

rameters can be adapted to each of their specificities). The next step is to merge the resulting annotation with the reference. For this step, we tested two tools, Stringtie merge [275] and Cuffmerge [283]. However, Stringtie merge performed poorly with Scallop re-annotation (45% of gene assigned - data not shown), while Cuffmerge showed robust performances, independently of the transcriptome reconstruction tool used. We thus decided to remain only with Cuffmerge in the pipeline. Then, we aim at including a step to reduce artefacts (such as artefactual gene fusions and small artifical genes). This step is still under testing. For this purpose, we previously attempted to automatically optimized the re-annotation tool parameters, with the Scallop and Stringtie2 parameter advisor from *Deblasio et al., 2020* [284]. However, this tool would take weeks to run, so we discarded it (tested a few times between February and May 2020, although at this time it was still a preprint).

Step	Tool	Input	Output
1. Construct novel annotation	StringTie2 Scallop	`BAM` `GTF`	`GTF`
2. Merge novel and reference annotations	cuffmerge	`GTF`	`GTF`
3. Compare novel annotations with reference	GffCompare	`GTF`	`summary.txt`
4. Extract more statistics from novel annotation	Rscript	`GTF`	`report_stats_annotation.txt`

5. Build UCSC trackhub	bash script	`BAM`
		`GTF`
			`UCSC trackhub folder`
			`bigWig`
			`bigGenePred`

6. Assign reads to genes	featureCounts	`BAM` `GTF`	`BAM`
7. Sort and index BAM file	Samtools	`BAM`	`BAM` `BAI`
8. Count unique reads per genes per cell	UMI-tools	`BAM`	`count_matrix.txt`
9. Run MultiQC	MultiQC	`BAM` `GTF`	`multiqc_report.html`

Figure 4.12 Suite of tools integrated in the scAnnotatiONT pipeline. This table shows the pipeline in its present condition. All tools are integrated into the pipeline, except the steps 4 (statistics) and 5 (automatised UCSC trackhub construction).

The next two steps consist in producing statistics summary reports out of the novel annotation (steps 3 and 4 in Figure 4.12). We then build a UCSC

trackhub in order to visualize the reads (from BAM or BigWig files) and their location compared to the reference and novel annotations. These two latter steps are not yet incorporated in the workflow, but will soon. Finally, steps 6 to 9 are common to any other 10x Genomics scRNA-seq analysis (as designed in Eoulsan for example, see Chapter 2). They include i) bulk gene assignment with featureCounts [123], ii) sorting the resulting BAM files with Samtools [120], iii) single-cell gene assignment with UMI-tools count [86], and finally iv) MultiQC [234] allows to output some general statistics on the assignment.

For the needs of the analyses, I also built a fully automated scRNA-seq downstream analyses pipeline in Nextflow, that can take as entry any scRNA-seq count matrix: https://github.com/LehmannN/scAnnotatiONT/tree/paper. Furthermore, to build the UCSC trackhubs, I developed a small utility to automatized this process (which aims at being integrated into the pipeline): https://github.com/LehmannN/makeUcscTrackhub.

4.3.5 Genome re-annotation with both scRNA-seq and bulk LR substantially improves read assignment

We have then compared this hybrid approach to using only the reference annotation, to estimate and quantify the improvement on the scRNA-seq analysis. Considering that the NCBI annotation is the most complete in terms of 3'UTRs, we decided to rely on this annotation for the following analyses. We performed two levels of comparison: at the genome level and at the gene level. Since our pipeline is not totally finalized, I will just show here a sample of some promising improvements we observed, but more analysis are needed to precisely estimate the gain.

4.3.5.1 At the genome level

Overall, we tested four different approaches: re-annotation with LR only and re-annotation with both LR and SR, for each of the two tools (Stringtie2 and Scallop). Figure 4.13 shows the results of the reads assignment after re-annotation with each of these four approaches.

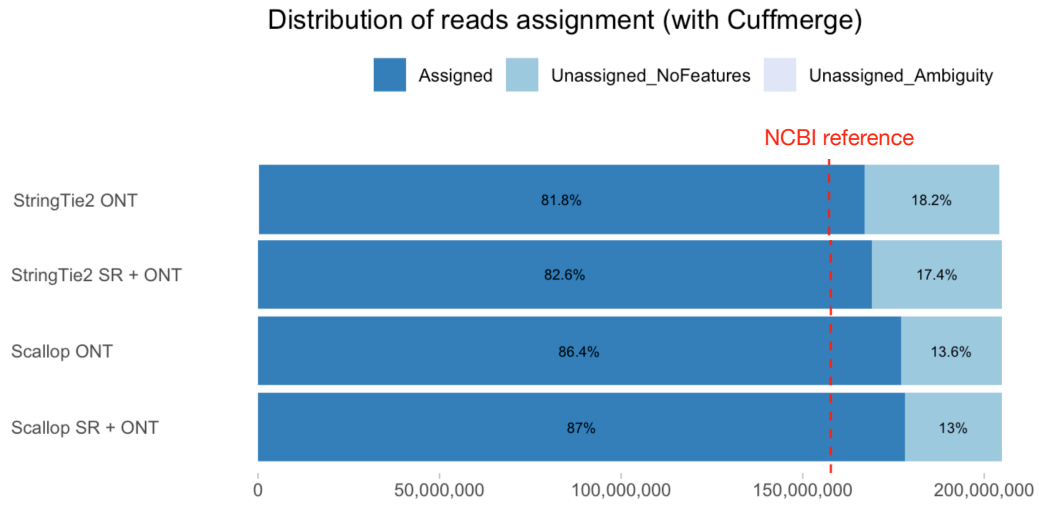


Figure 4.13 Comparison of the percentage of assigned reads with the 4 tested approaches. Based on percentage of assigned reads, Scallop with both LR and SR performs the best. The number of "unassigned ambiguity" (reads that map to a region where reference genes overlap on both strands) is too low to appear on the plots.

We can observe that the approaches enriched with SR (scRNA-seq reads) allow to consistently recover more reads than the ones with LR only. This may be due to more novel annotated genes or longer 3'extension with SR. In case of novel genes, whether they are relevant in terms of biology and gene length is still under evaluation (we may filter out the genes that are too short). Precisely, Figure 4.14 highlights the number of genes in the four novel annotations, and classify them in four categories: assembled, elongated, matching reference, and novel. These results were obtained on top of GffCompare [278] results, after comparison with the NCBI reference annotation.

Consistent with the observations on the Figure 4.13, we can observe that the two annotations that include the SR display i) a higher number of assembled genes, and ii) a higher number of elongated genes. However, all four annotations bring a significant improvement in terms of gene assignment, number of genes assembled and number of elongated genes. Further investigation are needed to assess which annotation performs best in light of the scRNA-seq analyses.

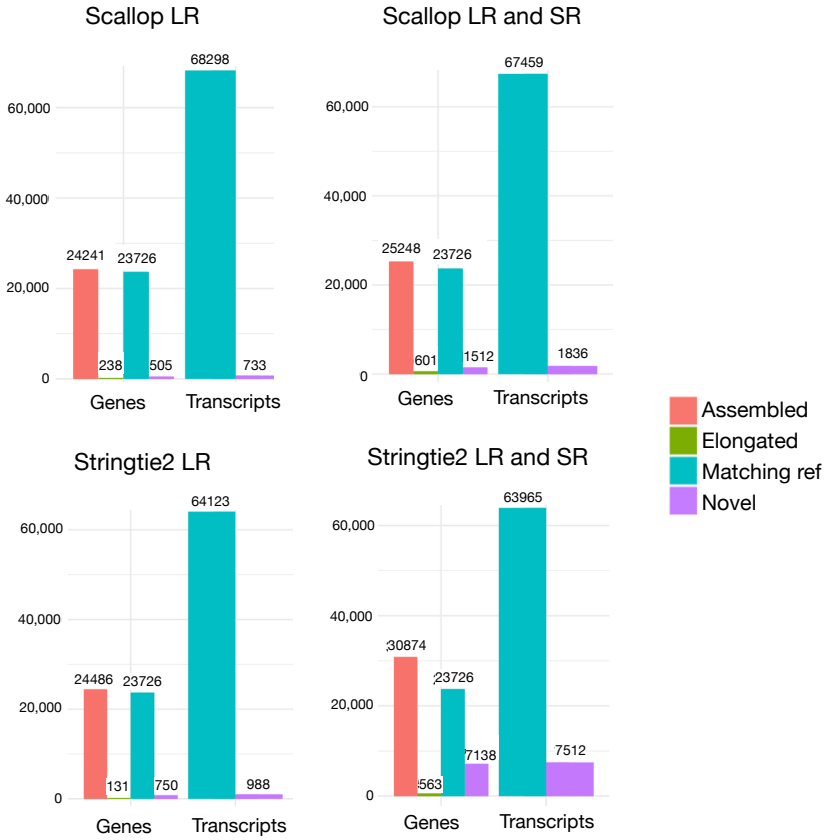


Figure 4.14 Comparison of the number of features recovered with the 4 approaches. Number of overall assembled genes (in red), elongated genes (in green), gene or transcript matching the reference (in blue) and novel genes or transcripts (in purple) within the four novel annotations.

4.3.5.2 At the gene level

Finally, we used one of the annotations (scallop LR and SR) to assess whether we could recover more scRNA-seq signal. In particular, we investigated SOX2, since it is a key marker in our study (pan-progenitor marker, see Chapter 3). It is thus expected to be expressed in the great majority of progenitors cells. Figure 4.15 highlights these differences. Before the re-annotation, SOX2 was detected in 25% of the cells (at very low levels). With the re-annotation, SOX2 is detected in 91% of the cells (3.5 times more cells). There are 30 times more reads that are assigned to SOX2 with this novel annotation. Our approach thus allows to recover a great quantity of reads in scRNA-seq, that were lost due to poor reference annotation. We

expect it will have a major impact on other genes. This analysis remains to be completed.

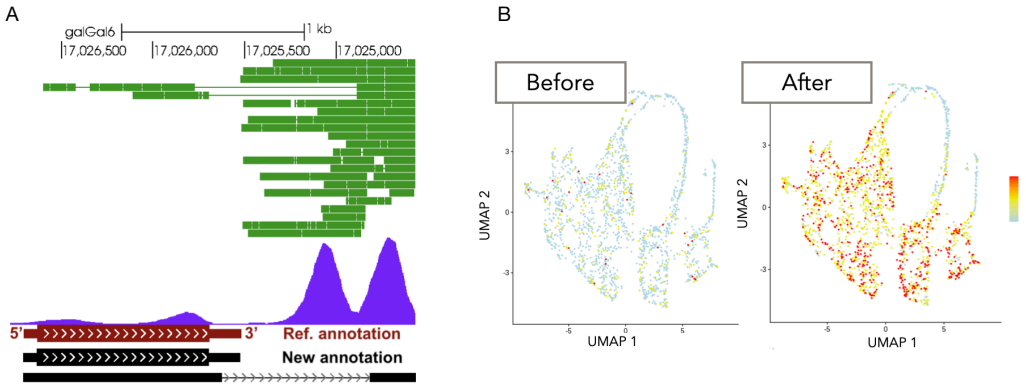


Figure 4.15 Recovery of SOX2 scRNA-seq signal with the novel annotation, extended in 3'. A) UCSC trackhub at SOX2 location. See plot description in 4.9-B. We can notice that the novel annotation (in black) gets extended in its 3'end compared to the reference annotation (in red). B) UMAPs of SOX2 expression before (left) and after (right) re-annotation.

4.4 Conclusion

We first studied the differences between three chicken reference annotations and showed that the 3'UTR regions are particularly incomplete. We then analysed our scRNA-seq dataset in parallel with the three reference annotations, and found discrepancies in gene quantification. We observed that for genes with an incomplete 3'UTR, the scRNA-seq signal is located just downstream of the gene, and thus not properly quantified. This prompted the hypothesis that improving these annotations at least on the 3'UTR could improve the scRNA-seq gene quantification. Using complementary Nanopore long-read sequencing transcriptome, we generated several novel annotations and found that it improves the recovery of scRNA-seq signal by up to 87%. We have evaluated two bioinformatics transcriptome reconstruction tools to build this novel annotation combining scRNA-seq and long-read transcripts, and propose a reusable open-source pipeline named scAnnotatiONT. Moreover, although a few transcriptome reconstruction approaches targeting

single-cell data came out since last year, they are dedicated to full-length scRNA-seq [285, 286]. We thus also demonstrate that transcriptome reconstruction tools developed for bulk datasets may be exploited to significantly improve 3' biased scRNA-seq analyses.

Chapter 5

Discussion and prospects

5.1 Contributions

The aim of my thesis project was to evaluate, develop and apply bioinformatics approaches for scRNA-seq to study the neurogenic transition in vertebrate neural progenitors. In this context, my contributions can be summarized as follows:

- I developed an automated pipeline dedicated to the pre-processing of 10x Genomics scRNA-seq data, in a scalable and reproducible manner (Chapter 2). It provides extensive quality checks and is much more flexible than the 10x Genomics proprietary pipeline. It relies on the Eoulsan workflow manager. The corresponding manuscript has been deposited in BioRxiv: https://www.biorxiv.org/content/10.1101/2021.10.13.464219v1.

- I applied scRNA-seq analysis approaches to various neural progenitors datasets (mouse and chicken), in order to isolate and uncover differences in gene expression between SYM and ASYM populations (Chapter 3, as well as Chapter 4 to a lesser extent). I have performed a custom analysis based on a public atlas dataset, and selected dedicated approaches based on both the data requirements and the biological question. I could identify a list of 15 genes that the team of Xavier Morin can

further investigate for their potential involved in the biological process, and possibly select a candidate gene for experimental validation.

- I have shown that the reference annotation is a key parameter for scRNA-seq analyses. I illustrated this point by showing that the choice between Ensembl, NCBI or UCSC reference annotations impact the downstream analyses and interpretation of the results.

- I have designed an hybrid approach to process scRNA-seq from chicken, an organism for which the annotation is not as high-quality as human or mouse. We propose to build a project-specific annotation based on bulk long-read transcripts, the scRNA-seq signal and the reference annotation (Chapter 4). Ultimately, this hybrid approach could be used to improve the annotation of any poorly annotated organism (the majority of available eukaryotic genomes) in a data-specific way and would thus bring novel insights into single-cell transcriptomics analyses.

5.2 Methodological aspects

5.2.1 Handling challenges in single-cell transcriptomics

I performed single-cell analyses on two 10x Genomics datasets, from mouse and chicken. Although the analysis of scRNA-seq data seems to reach more or less a consensus within the community in terms of steps of the analysis, and of the main tools to use, these datasets still represent some real challenges for the analyses. Stupendous technological advances are often synonymous with novel issues and challenges, and scRNA-seq is no exception. These challenges encompass all the aspects of scRNA-seq analyses: they are biological, statistical or computational in nature. Here, we will browse through some of the most notable, such as i) quantifying uncertainty, ii) dealing with highly sparse data and iii) managing different levels of resolution. I have summarized these challenges in Figure 5.1. For a thorough report of all the challenges of single cell data science, one can refer to the comprehensive study made recently by *Lähnemann et al., 2020* [287].

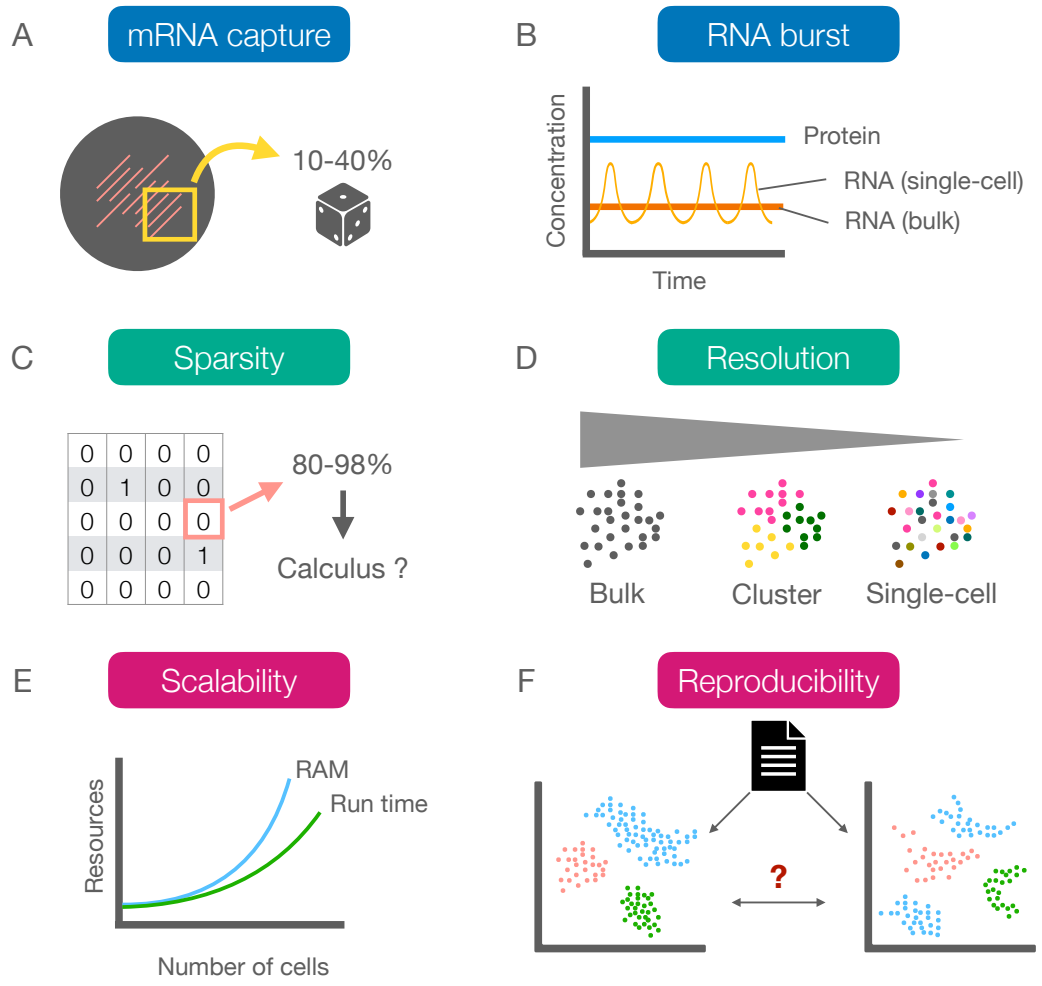


Figure 5.1 Illustration of the challenges inherent with scRNA-seq analyses. The first two challenges, in blue, are the most biology-oriented. A) Shortcomings in mRNA capture skews resulting analyses towards a random selection of all the cell's mRNA (from 10 to 30% depending on the protocol). B) Transcriptional bursting greatly affects mRNA quantification in single-cells. In bulk, this effect is imperceptible since RNA levels are averaged. Some mRNA could be missed only because of the stochasticity of transcription. The challenges in green are more specific to statistics. C) The very low amount of starting material in single-cell induces highly sparse matrices (containing 80 to 95% of zeros, depending on the protocol and sequencing depth). Statistical approaches require therefore to be adjusted accordingly. D) Identifying a suitable level of resolution is often challenging. Between bulk and authentic fine-grained single-cell resolution, there is a full range of possible cell types and subtypes definitions. The last challenges, in pink, are the most computationally-oriented. E) The increase in the number of cells within a single experiment results in a steep rise of data to handle. Tools need to be able to manage big data in order to scale rapidly. F) Reproducibility ensures a single analysis can be repeated several times and gives the same consistent results, either by the same laboratory or by external laboratories. All of these aspects have to be taken into account for accurate bioinformatics analyses.

5.2.1.1 Quantifying variability

In single-cell approaches, the amount of starting material (*e.g.* RNA molecules that are to be quantified in each cell) is extremely low by definition. With this respect, current protocols are poorly efficient in terms of RNA capture. The average sensitivity have been estimated between 10 and 40% [73, 158, 288]. Thus, transcripts that end up being actually captured and counted may lead to a significant sampling effect, which means that heterogeneity would arise among otherwise identical cells (Figure 5.1-A). In addition, mRNA capture is skewed towards the most abundant transcripts, lowly expressed genes thus being further sidelined [289]. Both of these effects are referred to as *technical noise*. Although recent development have proven to be more effective[1], this still poses a major challenge.

Moreover, biological variations such as transcriptional bursting and mRNA degradation during cell lysis contribute to add even more randomness to the single-cell RNA capture process. As shown in Figure 5.1-B, the regular biological variations of RNA transcription yield to random fluctuations that are usually wiped out by bulk approaches [290]. Thus, single-cell approaches come at the cost of a spectacular rise in uncertainty. This involves that data analysis requires customized development and thorough methodological attention [291]. Ideally, these uncertainties would need to all be quantified in order to prevent them from propagating in the downstream analyses [287].

5.2.1.2 Dealing with extreme sparsity

As a direct consequence of these biological and technical noises, scRNA-seq measurements typically suffer from a very large amount of zeros in the count matrix. For example, in the chick dataset, 92% of the counts are equal to zero. This is why scRNA-seq data is often described as highly sparse. Depending on the experimental protocol, the degree of sparsity (*e.g.* number of zeroes in the count matrix) ranges between 80 and 98%, with droplet-

[1] When I started my PhD, 10x Genomics protocol displayed a mRNA capture efficiency around 7%. With the latest chemistry, the single cell 3' reagent chemistry V3, it is now over 30%. Source: https://kb.10xgenomics.com/hc/en-us/articles/360001539051-What-fraction-of-mRNA-transcripts-are-captured-per-cell-

based protocols producing the data the most sparse [292]. This situation led to the definition of the term *dropout*, which denotes "*an event in which a transcript is not detected in the sequencing data owing to a failure to capture or amplify it*" [293]. However, the use of this ambiguous term has become controversial as it may also describe the overall zero counts, which enclose both the biologically-true absence of expression and the missed transcripts due to methodological noise [287, 294]. Additionally, the difference between a true zero or missing data is neither obvious nor clear-cut [289].

This extreme sparsity results in the data being zero-inflated, which is a well-known phenomenon in statistics. There are two common remedies to tackle this issue:

- **Use a zero-inflated model:** a specific distribution which describes a distribution that contains an excessively high proportion of zero-valued observations [295];
- **Infer novel values thanks to imputation approaches:** with the hypotheses that the dataset is incomplete, and that the missing data can be inferred either from the dataset itself or from an external source (a cell atlas for example).

In single-cell, the first approach is often preferred whenever possible, and multiple statistical models have already been successfully adapted and applied to single-cell data [150, 296, 297]. Imputation strategies have been applied to single-cell more recently and it is still debated whether they compensate for the missing data or add a supplementary noise [294].

Despite these advances, it is still challenging to appropriately handle sparsity in single-cell analyses and it is a major focus of discussion. An improper processing of the zero-valued observations can jeopardize downstream analyses quality and accuracy. Finally, a publication from *Choi et al., 2020* from last year raises questions about the very reality of zero-inflation in single-cell data [298]. In their conclusions, they recommend to get rid of zero-inflated models for single-cell data analysis, since biological signals are the primary drivers of zero-inflation, and not technical artefacts. Also, a study from *Qiu, 2020* [292] interestingly suggests to "embrace" dropouts as a useful signal

instead of attempting to correct for it, which goes in the opposite direction from all the other approaches. There is no doubt that future developments in both experimental and computational workflows will provide substantial answers towards single-cell data sparseness.

5.2.1.3 Dynamic levels of resolution

In the same way a map enables different levels of geographic areas (*e.g.* country, city, street, etc.), all scRNA-seq analyses encompass various levels of resolution (*e.g.* tissue, cell type, subtype, single cell). The key point is to define which level of resolution is the most relevant for a given biological question. Identifying a suitable level of resolution is often challenging, and depends on various factors: the biological question, the aim of the study, previous knowledge there is on the biological system and the protocol throughput. In my work, the mouse dataset was a cell atlas from a given tissue (spinal cord), and the chicken dataset was also from an heterogeneous tissue, with different cell types (mostly progenitors, but also some neurons, mesoderm and neural crest cells).

Since the beginning of the single-cell era, the very definition of cell types have been undermined and debated [55]. Which exact criteria should be kept to define a cell type ? How to define cell types when the cells evolve on a continuous spectrum ? These are the questions raised by single-cell approaches, among others, which are particularly relevant for single-cell atlases. Thus, when the study aims at providing a comprehensive cell reference, it is fundamental to investigate cellular heterogeneity at various levels of resolution. Some studies choose for instance to define different levels of definition organized just like Russian dolls: cell types 1, cell types 2 (which are subtypes of cell type 1) and cell subtypes (which are subtypes of cell type 2) [102, 299]. This provides a resolution flexibility that enables to "zoom" in and out a cellular map. It thus helps to define the best level of granularity for each specific case, and facilitates heterogeneity exploration.

Concerning the protocol throughput, it is important to emphasize that each particular experimental setup imposes its own limits. The choice of a

resolution might not be the same whether the data comes from a Smart-seq2 (few cells, high throughput) or 10x Genomics protocol (many cells, low throughput). Indeed, 10x Genomics approach is better suited to catalogue cells and define clusters, whereas Smart-seq2 approach must be considered if each cell needs to be precisely characterized with an extremely fine resolution.

5.2.2 The future of single-cell protocols

All of the single-cell protocols mentioned in this thesis rely on short-read Illumina sequencing. However a few very recent protocols target single-cell long-reads transcriptomics [300–304], a promising approach to get rid of transcriptome reconstruction steps for poorly annotated species and to uncover novel or lowly expressed isoforms.

Other recent protocols include spatial transcriptomics [305–307] and ultra-high throughput protocols (reaching several million cells in a single experiment). The former is increasingly being applied to scRNA-seq studies since the technology is now mature enough [308–310]. With spatial information added to scRNA-seq, this approach offers a unique view on tissues structure and organisation at the cellular level. Spatial transcriptomics methods are usually divided into i) FISH-based methods (*i.e.* In Fluorescence In Situ Hybridization), such as osmFISH [311] or seqFISH+ [312], and ii) scRNA-seq-based methods that rely on spatial barcodes (*e.g.* Slide-seq [313]) [310]. Most of the spatial profiling approaches however require pre-selected markers or have restricted spatial resolution (*i.e.* up to 100µm (3–30 cells)) [306]. Novel approaches, such as HDST [306], succeeded in lowering the resolution down to 2µm, opening the way to high-resolution spatial transcriptomics. Mapping the brain [314] or tumour cells [315] are just two of the remarkable applications of spatial transcriptomics.

Ultra-high throughput protocols are still under active development and fulfill its commitment in getting rid of physical cell isolation [81]. It is referred to as split-pool barcoding or single-cell combinatorial indexing, where each cell is labelled by a unique combination of several oligonucleotides (instead of just one). It has already been successfully applied to a couple of large

projects, such as cell atlases [316].

Another important point is that all the previously mentioned protocols mainly apply on whole cells, but a growing number of studies use approaches based on single-nucleus (*i.e.* snRNA-seq) [317–320]. It is key for some cell types or tissues for which cell dissociation is challenging (*e.g.* solid tissues, tumors). It also allows to remove stress-induced transcriptional response to dissociation and to capture cells which particular morphology makes it difficult to process in standard protocols (*e.g.* brain cells). Single-nucleus isolation are compatible with most technologies which explains its increasing interest in the single-cell community.

Although I focused only on transcriptomics, the SYMASYM project was intended to use single-cell multi-omics, combining scRNA-seq with scATAC-seq to study the epigenomic layer of information. These approaches enable the simultaneous measurement of distinct data modalities from single-cells. Some protocols ensure for example the joint profiling of transcriptome and targeted proteome (*e.g.* CITE-seq) [321] or the simultaneous measurement of gene expression and DNA methylation (*e.g.* scMT-seq) [322]. Several approaches even go further by combining three modalities, such as scTrio-seq2 which is able to measure transcriptome, genome and methylome simultaneously [323]. Most of these approaches are summarized and compared in the excellent review of *Lee et al., 2020* [324].

5.2.3 The pitfalls of genome mapping in scRNA-seq

Regarding mapping tools, unconventional alignment strategies recently appeared: they are often referred to as *pseudo-aligners* (a.k.a alignment-free tools), such as Kallisto [325] or Salmon [326]. Their specificity is their ability not to rely on the exact match of individual bases from a reference genome, but rather on the read assignment to known features (*e.g.* genes, transcripts) of a high quality reference transcriptome [118]. Transcriptome mapping means on one side that there was enough bulk RNA-seq data available to build an isoform-level reference, and on the other side the variety of tissues sequenced is broad enough. If not, most of the transcripts will be lost on the sole basis

they were not characterized in the reference. It also precludes the discovery of novel or unannotated transcripts [224]. However, since it is generally faster, it may be preferred in some specific cases. This approach is reserved for well annotated species though (*e.g.* human or mouse).

While all the above mentioned tools have been developed for bulk RNA-seq, they have been extensively applied to scRNA-seq data. A recent study by *Vieth et al.* [121] compared the impact of three popular bulk mapping tools (STAR, BWA and Kallisto) on five different scRNA-seq protocols. STAR was shown to perform better than Kallisto in terms of accuracy and mapping rate. Yet, another study from *Du et al.* compares both tools on scRNA-seq data and shows that Kallisto is up to four time faster and much less memory intensive than STARsolo [113], the recent version of STAR adapted to scRNA-seq [327]. The choice of the appropriate mapper then depends whether one wants to favour gene detection accuracy or computing efficiency.

Single-cell specialized pre-processing tools have emerged these recent years. They take into account scRNA-seq specificities such as high data sparsity, the increasing number of reads to handle or UMI and cell barcodes processing. The most popular ones are integrated into all-in-one pre-processing pipelines that can handle all the steps up to the expression matrix production. This include CellRanger and Alevin, the single-cell specific pipeline based on Salmon package. In transcriptome-mapped approaches, mapping and quantification can occur at the same time. Most of the tools discard multi-mapped reads, however there is an emerging movement that supports the idea of keeping multi-mapped reads. This is the strategy used in Alevin. I have tested Alevin on another 10x Genomics dataset (that we did not show here due to poor quality), but the results were very surprising, since it detected 10 times more cells than both CellRanger and Eoulsan. We did not investigated further these differences since this dataset was not suited for further analyses (high level of mRNA contamination).

5.2.4 Genome annotations are key parameters of the scRNA-seq workflow

In my work, I have shown that the choice of a reference annotation is crucial for the analysis of scRNA-seq datasets. Results obtained with different chicken reference annotations (Ensembl, NCBI/RefSeq or UCSC) were different. For 10xGenomics scRNA-seq datasets, the 3'UTR annotation is particularly important. In Ensembl, the 3'UTRs were globally shorter than in the NCBI/RefSeq annotation. We wonder whether the long-read RNA-seq datasets were effectively incorporated within the annotation pipeline. Another possible explanation of the shorter 3'UTRs could also stem from the UTR_Builder module of the Ensembl pipeline. Once a gene model is built, the module searches for the start and end boundaries of the last intron, and if these coordinates exactly match the same positions in a (longer) RNA-seq or cDNA structure, then a 3'UTR is added to the gene model from this "donor" structure (see https://www.ensembl.org/info/genome/genebuild/2018_12_chicken_gene_annotation.pdf and [272]). It would be interesting to investigate how this rule affects the UTRs in the chicken annotation.

Of note, I have been working on the current GRCg6a genome assembly. A new assembly of the chicken genome coordinated by the Vertebrate Genomes Project (VGP) seems to be soon released, with a NCBI annotation release 105 performed on these GRCg7b and GRCg7w genome assemblies: https://www.ncbi.nlm.nih.gov/bioproject/PRJNA660757/. New reference annotations are thus expected to be generated by Ensembl and NCBI.

For this project, we propose a novel strategy to handle scRNA-seq datasets from genomes that do not benefit from a high-quality annotation. We propose to generate a bulk long-read transcriptome in the same cell sample as the scRNA-seq. In my work, I showed that this matching dataset can drastically improve the annotation, and the final results. It would be interesting to evaluate whether using a long-read dataset from the same organism, but not matching the same cell types could also improve the annotation. This would be a less costly approach if using publicly-available datasets (although the cost of our ONT dataset was only about 2000 euros). Several long-read

RNA-seq datasets were previously generated in the developing chicken, with the PacBio sequencer: 8 embryonic stages [328], 1 embryo [274], 3 stages of embryonic chicken heart [329]. To our knowledge, our long-read RNA-seq is the first one sequenced on a Nanopore sequencer. It should be stressed that the Nanopore protocol used was starting from the 3'end of the transcript, thereby particularly suited for our aim of better annotating the 3'UTRs. It remains to be tested if the PacBio datasets would be also adapted for this precise aim.

Finally, it must be noted that a few very promising approaches, related to my work, came out these last 6 months:

- First, *Wang et al., 2021* propose the very first tool dedicated to improving scRNA-seq signal without the need for a reference annotation, and that can be applied to any species and protocol (full-length and droplet-based) [330]. It allows to uncover any region in the genome that is transcriptionally active, based on scRNA-seq data only. According to the authors, this approach "*recovers biologically relevant transcriptional activity beyond the scope of the best available genome annotation*". Moreover, they raise the issue that the younger the embryos are, the higher is the prevalence of unnatotated transcripts (that I have not seen mentioned elsewhere before). They specifically demonstrate that the current chicken reference annotations do not characterize the transcriptional landscape of early embryonic tissues as well as they would for later stages.
- *Shields et al., 2021* are the first ones, to my knowledge, that exploit long-reads (PacBio) in order to improve a genome annotation specifically to analyse 10x Genomics scRNA-seq data, applied on ants brain [331]. Their work has a different purpose than ours, since they also aimed at improving the ant reference annotation, and thus includes a great amount of work on manual re-annotation (they are ant experts). What is also interesting is that they claim that the extended 3'UTRs in their novel annotation resulted in the recovery of the transcriptome of 18% more cells and lead to major improvements in scRNA-seq data analyses (*e.g.* deeper single-cell resolution, identification of novel mark-

ers).

- Finally, *Botvinnik et al., 2021* propose an alignment-free, reference-independent pipeline for cross-species cell type identification for scRNA-seq data [332]. This approach would allow the analysis of any species, independently of whether a genome assembly or a reference annotation is available (which is more than 99.9% of the 10 million animal species predicted to exist).

Bibliography

1. Collins, F. S., Green, E. D., Guttmacher, A. E. & Mark, S. A vision for the future of genomics research. *Nature* **422,** 15–17 (2003).
2. Venter, J. C. *et al.* The Sequence of the Human Genome. *Science* **291,** 1304–1351 (2001).
3. Wetterstrand, K. *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)* 2014.
4. Reuter, J. A., Spacek, D. & Snyder, M. P. High-Throughput Sequencing Technologies. *Molecular Cell* **58,** 586–597 (2015).
5. Stephens, Z. D. *et al.* Big Data: Astronomical or Genomical ? *PLoS Biology* **13,** 1–11 (2015).
6. Hogeweg, P. The Roots of Bioinformatics in Theoretical Biology. **7,** 1–5 (2011).
7. Spengler, S. J. Bioinformatics in the Information Age. *Science* **287,** 1221–1223 (2000).
8. Keusch, G. T. What do - omics mean for the science and policy of the nutritional. *The American Journal of Clinical Nutrition* **83(suppl),** 520S–522S (2006).
9. Byrne, A., Cole, C., Volden, R. & Vollmers, C. Realizing the potential of full-length transcriptome sequencing. *Philosophical Transactions of the Royal Society B: Biological Sciences* **374** (2019).
10. Ozsolak, F. *et al.* Direct RNA sequencing. *Nature* **461,** 814–819 (2009).
11. Garalde, D. R. *et al.* Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods* **15,** 201–206 (2018).

12. Sanger, F., Nicklen, S. & Coulson, A. R. DNA sequencing with chain-terminating. *Proceedings of the National Academy of Sciences of the United States of America* **74,** 5463–5467 (1977).

13. Shendure, J. *et al.* DNA sequencing at 40: past, present and future. *Nature* **550,** 345–353 (2017).

14. Adams, M. D. *et al.* Complementary Sequencing : Expressed Sequence Tags and Human Genome Project. *Science* **252,** 1651–1656 (1991).

15. Adams, M. D. *et al.* Sequence identification of 2,375 human brain genes. *Nature* **355,** 632–634 (1992).

16. Nagaraj, S. H., Gasser, R. B. & Ranganathan, S. A hitchhiker's guide to expressed sequence tag (EST) analysis. *Briefings in Bioinformatics* **8,** 6–21 (2006).

17. Schena, M., Shalon, D., Davis, R. W. & Patrick, B. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science* **270,** 467–470 (1995).

18. Lowe, R., Shirley, N., Bleackley, M., Dolan, S. & Shafee, T. Transcriptomics technologies. *PLoS Computational Biology* **5,** 1–23 (2017).

19. Wang, Z., Gerstein, M. & Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics* **10,** 57–63 (2009).

20. Ozsolak, F. & Milos, P. M. RNA sequencing: advances, challenges and opportunities. *Nature Reviews Genetics* **12,** 87–98 (2011).

21. Bainbridge, M. N. *et al.* Analysis of the prostate cancer cell line LNCaP transcriptome using a sequencing-by-synthesis approach. *BMC Genomics* **7,** 1–11 (2006).

22. Nagalakshmi, U. *et al.* The Transcriptional Landscape of the Yeast Genome Defined by RNA Sequencing. *Science* **320,** 1344–1349 (2008).

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

24. Hrdlickova, R., Toloue, M. & Tian, B. RNA-Seq methods for transcriptome analysis. *Wiley Interdisciplinary Reviews: RNA* **8** (2017).

25. Mardis, E. R. DNA sequencing technologies : 2006 – 2016. *Nature Protocols* **12,** 213–218 (2017).

26. Hong, M. *et al.* RNA sequencing : new technologies and applications in cancer research. *Journal of Hematology & Oncology* **13,** 1–16 (2020).

27. Todd, E. V., Black, M. A. & Gemmell, N. J. The power and promise of RNA-seq in ecology and evolution. *Molecular Ecology* **25,** 1224–1241 (2016).

28. Rao, M. S. *et al.* Comparison of RNA-Seq and Microarray Gene Expression Platforms for the Toxicogenomic Evaluation of Liver From Short-Term Rat Toxicity Studies. **9,** 1–16 (2019).

29. Corchete, L. A. *et al.* Systematic comparison and assessment of RNA - seq procedures for gene expression quantitative analysis. *Scientific Reports* **10,** 1–15 (2020).

30. Berge, K. V. D. *et al.* RNA Sequencing Data: Hitchhiker's Guide to Expression Analysis. *Annual Review of Biomedical Data Science* **2,** 139–173 (2019).

31. Hölzer, M. & Marz, M. De novo transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers. *GigaScience* **8,** 1–16 (2019).

32. Stark, R. & Grzelak, M. RNA sequencing: the teenage years. *Nature Reviews Genetics* **20,** 631–656 (2019).

33. Pollard, M. O., Gurdasani, D., Mentzer, A. J., Porter, T. & Sandhu, M. S. Long reads: their purpose and place. *Human molecular genetics* **27,** R234–R241 (2018).

34. Dijk, E. L. V., Jaszczyszyn, Y., Naquin, D. & Thermes, C. The Third Revolution in Sequencing Technology. *Trends in Genetics* **34,** 666–681 (2018).

35. Gupta, P. K. Single-molecule DNA sequencing technologies for future genomics. *Trends in Biotechnology* **26,** 602–611 (2008).

36. Eid, J. *et al.* Real-Time DNA Sequencing from Single Polymerase Molecules. *Science* **233,** 133–139 (2009).

37. Jain, M. *et al.* Improved data analysis for the MinION nanopore sequencer. *Nature Methods* **12,** 351–359 (2015).

38. Payne, A., Holmes, N., Rakyan, V. & Loose, M. Sequence analysis for Oxford nanopore bulk FAST5 files. *Bioinformatics* **35,** 2193–2198 (2019).

39. Amarasinghe, S. L. *et al.* Opportunities and challenges in long-read sequencing data analysis. *Genome Biology* **21,** 1–16 (2020).

40. Wenger, A. M. *et al.* Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature Biotechnology* **37,** 1155–1162 (2019).

41. Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature Biotechnology* **36,** 338–345 (2018).

42. Dong, X. *et al.* The long and the short of it : unlocking nanopore long-read RNA sequencing data with short-read differential expression analysis tools. *NAR Genomics and Bioinformatics* **3,** 1–11 (2021).

43. Koren, S. *et al.* Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature Biotechnology* **30** (2012).

44. Au, K. F., Sebastiano, V., Afshar, P. T., Durruthy, J. & Lee, L. Characterization of the human ESC transcriptome by hybrid sequencing. *PNAS* **110,** 4821–4830 (2013).

45. Treutlein, B., Gokce, O., Quake, S. R. & Südhof, T. C. Cartography of neurexin alternative splicing mapped by single-molecule long-read mRNA sequencing. *PNAS* **111,** 1291–1299 (2014).

46. Sharon, D., Tilgner, H., Grubert, F. & Snyder, M. A single-molecule long-read survey of the human transcriptome. *Nature Biotechnology* **31,** 1009–1016 (2013).

47. Tilgner, H. *et al.* Comprehensive transcriptome analysis using synthetic long-read sequencing reveals molecular co-association of distant splicing events. *Nature Biotechnology* **33,** 736–743 (2015).

48. Byrne, A. *et al.* Nanopore long-read RNAseq reveals widespread transcriptional variation among the surface receptors of individual B cells. *Nature Communications* **8,** 1–11 (2017).

49. Soneson, C. *et al.* A comprehensive examination of Nanopore native RNA sequencing for characterization of complex transcriptomes. *Nature Communications* **10,** 1–14 (2019).

50. Boivin, V. *et al.* Simultaneous sequencing of coding and noncoding RNA reveals a human transcriptome dominated by a small number of highly expressed noncoding genes. *RNA* **24,** 950–965 (2018).

51. Workman, R. E. *et al.* Nanopore native RNA sequencing of a human poly(A) transcriptome. *Nature Methods* **16,** 1297–1305 (2019).

52. Adewale, B. A. Will long-read sequencing technologies replace short-read sequencing technologies in the next 10 years ? *African Journal of Laboratory Medicine* **9,** 1–5 (2020).

53. Amarasinghe, S. L., Ritchie, M. E. & Gouil, Q. Long-Read-Tools.Org: an Interactive Catalogue of Analysis Methods for Long-Read Sequencing Data. *GigaScience* **10,** 1–7 (2021).

54. Sender, R., Fuchs, S. & Milo, R. Revised Estimates for the Number of Human and Bacteria Cells in the Body. *PLoS Biology* **14,** 1–14 (2016).

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

56. Mardis, E. R. The impact of next-generation sequencing technology on genetics. *Cell* **24,** 133–141 (2008).

57. Cloonan, N. & Grimmond, S. M. Transcriptome content and dynamics at single-nucleotide resolution. *Genome Biology* **9,** 1–4 (2008).

58. Schuster, S. C. Next-generation sequencing transforms today's biology. *Nat Methods* **5,** 16–18 (2008).

59. Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* **6,** 377–382 (2009).

60. Islam, S. *et al.* Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Research* **21,** 1160–1167 (2011).

61. Goldman, S. L. *et al.* The Impact of Heterogeneity on Single-Cell Sequencing. *Frontiers in Genetics* **10,** 1–8 (2019).

62. Lafzi, A., Moutinho, C., Picelli, S. & Heyn, H. Tutorial: guidelines for the experimental design of single-cell RNA sequencing studies. *Nature Protocols* **13,** 2742–2757 (2018).

63. Picelli, S. *et al.* Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature Methods* **10** (2013).

64. Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161,** 1202–1214 (2015).

65. Zappia, L., Phipson, B. & Oshlack, A. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Computational Biology* **14** (2018).

66. Eberwine, J. *et al.* Analysis of gene expression in single live neurons. *PNAS* **89,** 3010–3014 (1992).

67. Lambolez, B. *et al.* AMPA Receptor Subunits Expressed by Single Ptirkinje Cells. *Neuron* **9,** 247–258 (1992).

68. Kurimoto, K. *et al.* An improved single-cell cDNA amplification method for efficient high-density oligonucleotide microarray analysis. *Nucleic Acids Research* **34** (2006).

69. Esumi, S. *et al.* Method for single-cell microarray analysis and application to gene-expression profiling of GABAergic neuron progenitors. *Neuroscience Research* **60,** 439–451 (2008).

70. Guo, G. *et al.* Resource Resolution of Cell Fate Decisions Revealed by Single-Cell Gene Expression Analysis from Zygote to Blastocyst. *Developmental Cell* **18,** 675–685 (2010).

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

72. Ziegenhain, C., Vieth, B., Parekh, S., Hellmann, I. & Enard, W. Quantitative single-cell transcriptomics. *Briefings in Functional Genomics* **17,** 220–232 (2018).

73. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **8** (2017).

74. Svensson, V. *et al.* Power analysis of single-cell RNA-sequencing experiments. *Nature Methods* **14,** 381–390 (2017).

75. Ziegenhain, C. *et al.* Comparative Analysis of Single-Cell RNA Sequencing Methods. *Molecular Cell* **65,** 631–643 (2017).

76. Ding, J. *et al.* Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology* **38,** 737–746 (2020).

77. Cao, J. *et al.* The single cell transcriptional landscape of mammalian organogenesis. *Nature* **566,** 496–502 (2019).

78. Cao, J. *et al.* A human cell atlas of fetal gene expression. *Science* **808** (2020).

79. Nguyen, A. & Phan, T. G. Single Cell RNA Sequencing of Rare immune Cell Populations. *Frontiers in Immunology* **9** (2018).

80. Hagemann-Jensen, M. *et al.* Single-cell RNA counting at allele and isoform resolution using Smart-seq3. *Nature Biotechnology* **38,** 708–714 (2020).

81. Li, Y., Ma, L., Wu, D. & Chen, G. OUP accepted manuscript. *Briefings In Bioinformatics* **00,** 1–18 (2021).

82. Rich-griffin, C. *et al.* Single-Cell Transcriptomics : A High-Resolution Avenue for Plant Functional Genomics. *Trends in Plant Science* **25,** 186–197 (2020).

83. Boon, W. C. *et al.* Increasing cDNA Yields from Single-cell Quantities of mRNA in Standard Laboratory Reverse Transcriptase Reactions using Acoustic Microstreaming. *Journal of Visualized Experiments* **53,** 1–4 (2011).

84. Aird, D. *et al.* Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biology* **12** (2011).

85. Islam, S. *et al.* Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods* **11,** 163–166 (2014).

86. Smith, T., Heger, A. & Sudbery, I. UMI-tools: Modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome Research* **27,** 491–499 (2017).

87. Kharchenko, P. V. The triumphs and limitations of computational methods for scRNA-seq. *Nature Methods* **18** (2021).

88. Baran-Gale, J., Chandra, T. & Kirschner, K. Experimental design for single-cell RNA sequencing. *Briefings in Functional Genomics* **17,** 233–239 (2018).

89. Chen, H., Ye, F. & Guo, G. Revolutionizing immunology with single-cell RNA sequencing. *Cellular and Molecular Immunology* **16,** 242–249 (2019).

90. He, P. *et al.* The changing mouse embryo transcriptome at whole tissue and single-cell resolution. *Nature* **583,** 760–767 (2020).

91. Mantri, M. *et al.* Spatiotemporal single-cell RNA sequencing of developing chicken hearts identifies interplay between cellular differentiation and morphogenesis. *Nature Communications* **12** (2021).

92. Zhang, Y. *et al.* Single-cell RNA sequencing in cancer research. *Journal of Experimental & Clinical Cancer Research* **40,** 1–17 (2021).

93. Yasen, A. *et al.* Progress and applications of single-cell sequencing techniques. *Infection, Genetics and Evolution* **80** (2020).

94. Marioni, J. C. & Arendt, D. How Single-Cell Genomics Is Changing Evolutionary and Developmental Biology. *Annual Review of Cell and Developmental Biology* **33,** 537–553 (2017).

95. Tanay, A. & Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541,** 331–338 (2017).

96. Mittnenzweig, M. *et al.* A single-embryo, single-cell time-resolved model for mouse gastrulation. *Cell* **184,** 1–18 (2021).

97. Moreno-ayala, R. & Junker, J. P. Single-cell genomics to study developmental cell fate decisions in zebrafish. *Briefings in Functional Genomics* **00,** 1–7 (2021).

98. Rayon, T., Maizels, R. J., Barrington, C. & Briscoe, J. Single cell transcriptome profiling of the human developing spinal cord reveals a conserved genetic programme with human specific features. *bioRxiv* **April** (2021).

99. Ton, M. L. N., Guibentif, C. & Göttgens, B. Single cell genomics and developmental biology: moving beyond the generation of cell type catalogues. *Current Opinion in Genetics and Development* **64,** 66–71 (2020).

100. Regev, A. *et al.* The Human Cell Atlas. *eLife* **6,** 1–30 (2017).

101. Li, A. H., Janssens, J., Waegeneer, M. D. & Kolluru, S. S. Fly Cell Atlas : a single-cell transcriptomic atlas of the adult fruit fly. *bioRxiv* **July** (2021).

102. The Tabula Muris Consortium. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* **562,** 367–372 (2018).

103. Karaiskos, N. *et al.* The Drosophila embryo at single-cell transcriptome resolution. *Science* **358,** 194–199 (2017).

104. Aldridge, S. & Teichmann, S. A. Single cell transcriptomics comes of age. *Nature Communications* **11** (2020).

105. Vegh, P. & Haniffa, M. The impact of single-cell RNA sequencing on understanding the functional organization of the immune system. *Briefings in Functional Genomics* **17,** 265–272 (2018).

106. Villani, A.-C. *et al.* Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* **356** (2017).

107. Sun, G. *et al.* Single-cell RNA sequencing in cancer : Applications, advances, and emerging challenges. *Molecular Therapy: Oncolytics* **21,** 183–206 (2021).

108. Saichi, M. *et al.* Single-cell RNA sequencing of blood antigen-presenting cells in severe COVID-19 reveals multi-process defects in antiviral immunity. *Nature Cell Biology* **23,** 538–551 (2021).

109. Stephenson, E. *et al.* Single-cell multi-omics analysis of the immune response in COVID-19. *Nature medicine* **27,** 904–916 (2021).

110. Huo, L. *et al.* Single-cell multi-omics sequencing : application trends, COVID-19, data analysis issues and prospects. *Briefings in Bioinformatics* **22** (2021).

111. Delorey, T. M., Ziegler, C. G. K. & Regev, A. COVID-19 tissue atlases reveal SARS-CoV-2 pathology and cellular targets. *Nature* **595** (2021).

112. Zappia, L. & Theis, F. J. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *biorxiv* **August** (2021).

113. Kaminow, B., Yunusov, D., Dobin, A. & Spring, C. STARsolo : accurate, fast and versatile mapping / quantification of single-cell and single-nucleus RNA-seq data. *bioRxiv* **May,** 1–35 (2021).

114. Nayak, R. & Hasija, Y. A hitchhiker's guide to single-cell transcriptomics and data analysis pipelines. *Genomics* **113,** 606–619 (2021).

115. Andrews, S. *et al. FastQC* Babraham Institute. Babraham, UK, 2012.

116. Robert, C. & Watson, M. Errors in RNA-Seq quantification affect genes of relevance to human disease. *Genome Biology* **16,** 1–16 (2015).

117. Zhao, S. Assessment of the Impact of Using a Reference Transcriptome in Mapping Short RNA-Seq Reads. *PloS one* **9,** e101374 (2014).

118. Schaarschmidt, S., Fischer, A., Zuther, E. & Hincha, D. K. Evaluation of Seven Different RNA-Seq Alignment Tools Based on Experimental Data from the Model Plant Arabidopsis thaliana. *International Journal of Molecular Sciences* **21,** 1–17 (2020).

119. Schulze Brüning, R., Tombor, L., Schulz, M. H., Dimmeler, S. & John, D. Comparative Analysis of common alignment tools for single cell RNA sequencing. *bioRxiv,* 2021.02.15.430948 (2021).

120. Li, H. *et al.* The Sequence Alignment / Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).

121. Vieth, B., Parekh, S., Ziegenhain, C., Enard, W. & Hellmann, I. A systematic evaluation of single cell RNA-seq analysis pipelines. *Nature Communications* **10,** 1–11 (2019).

122. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29,** 15–21 (2013).

123. Liao, Y., Smyth, G. K. & Shi, W. FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* **30,** 923–930. eprint: 1305.3347 (2014).

124. Anders, S., Pyl, P. T. & Huber, W. HTSeq — a Python framework to work with high-throughput sequencing data. *Bioinformatics* **31,** 166–169 (2015).

125. Parekh, U. *et al.* Mapping Cellular Reprogramming via Pooled Overexpression Screens with Paired Fitness and Single-Cell RNA-Sequencing Readout. *Cell Systems* **7,** 548–555.e8 (2018).

126. Petukhov, V. *et al.* dropEst : pipeline for accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. *Genome Biology* **19,** 1–16 (2018).

127. Tian, L. *et al.* scPipe : A flexible R / Bioconductor preprocessing pipeline for single-cell RNA- sequencing data. *PLoS computational biology* **14,** 1–15 (2018).

128. Srivastava, A., Malik, L., Smith, T., Sudbery, I. & Patro, R. Alevin efficiently estimates accurate gene abundances from dscRNA-seq data. *Genome Biology* **20,** 1–16 (2019).

129. Gao, M. *et al.* Comparison of high-throughput single-cell RNA sequencing data processing pipelines. *Briefings in Bioinformatics* **22** (2021).

130. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology* **15** (2019).

131. Ilicic, T. *et al.* Classification of low quality cells from single-cell RNA-seq data. *Genome Biology* **17,** 1–15 (2016).

132. Xi, N. M. & Li, J. J. Benchmarking Computational Doublet-Detection Methods for Single-Cell RNA Sequencing Data Article Benchmarking Computational Doublet-Detection Methods for Single-Cell RNA Sequencing Data. *Cell Systems* **12,** 1–19 (2021).

133. Lun, A. T. L. *et al.* EmptyDrops : distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biology* **20,** 1–9 (2019).

134. Muskovic, W. & Powell, J. E. DropletQC : improved identification of empty droplets and damaged cells in single-cell RNA-seq data. *bioRxiv* **August** (2021).

135. Osorio, D. & Cai, J. J. Systematic determination of the mitochondrial proportion in human and mice tissues for single-cell RNA-sequencing data quality control. *Bioinformatics* **37,** 963–967 (2020).

136. Hippen, A. A. *et al.* miQC : An adaptive probabilistic framework for quality control of single-cell RNA-sequencing data. *PLoS Computational Biology* **17,** e1009290 (2021).

137. Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177,** 1888–1902.e21 (2019).

138. McCarthy, D. J., Campbell, K. R., Lun, A. T. & Wills, Q. F. Scater: Preprocessing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33,** 1179–1186 (2017).

139. Bernstein, N. J. *et al.* Solo: Doublet Identification in Single-Cell RNA-Seq via Semi-Supervised Deep Learning. *Cell Systems* **11,** 95–101.e5. arXiv: arXiv:1412.6980 (2020).

140. DePasquale, E. A. *et al.* DoubletDecon: Deconvoluting Doublets from Single-Cell RNA-Sequencing Data. *Cell Reports* **29,** 1718–1727.e8 (2019).

141. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors. *Cell Systems* **8,** 329–337 (2019).

142. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data. *Cell Systems* **8,** 281–291 (2019).

143. Chen, B., Herring, C. A. & Lau, K. S. pyNVR : investigating factors affecting feature selection from scRNA-seq data for lineage reconstruction. *Bioinformatics* **35,** 2335–2337 (2019).

144. Su, K., Yu, T. & Wu, H. Accurate feature selection improves single-cell RNA-seq cell clustering. *Briefings in Bioinformatics* **00,** 1–10 (2021).

145. Brennecke, P. *et al.* Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods* **10,** 1093–1098 (2013).

146. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY : large-scale single-cell gene expression data analysis. *Genome Biology* **19,** 1–5 (2018).

147. Yip, S. H., Sham, P. C. & Wang, J. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Briefings in Bioinformatics* **20,** 1583–1589 (2019).

148. Andrews, T. S. & Hemberg, M. M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics (Oxford, England)* **35,** 2865–2867 (2019).

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

150. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology* **16,** 1–10 (2015).

151. Sheng, J. & Li, W. V. Selecting gene features for unsupervised analysis of single-cell gene expression data. *Briefings In Bioinformatics* **00,** 1–12 (2021).

152. Dillies, M. A. *et al.* A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics* **14,** 671–683 (2013).

153. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: Challenges and opportunities. *Nature Methods* **14,** 565–571 (2017).

154. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* **11** (2010).

155. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biology* **11** (2010).

156. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15,** 1–21 (2014).

157. Lun, A. T., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology* **17,** 1–14 (2016).

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

159. Vallejos, C. A., Marioni, J. C. & Richardson, S. BASiCS : Bayesian Analysis of Single-Cell Sequencing Data. *PLoS Computational Biology* **11,** e1004333 (2015).

160. Lun, A. T. L., Calero-nieto, F. J., Haim-vilmovsky, L., Göttgens, B. & Marioni, J. C. Assessing the reliability of spike-in normalization for analyses of single-cell RNA sequencing data. *Genome Research* **27,** 1795–1806 (2017).

161. Lin, L. *et al.* Normalizing single-cell RNA sequencing data with internal spike-in-like genes, 1–33 (2020).

162. Phipson, B., Zappia, L. & Oshlack, A. Gene length and detection bias in single cell RNA sequencing protocols. *F1000Research* **6** (2017).

163. Cole, M. B. *et al.* Performance Assessment and Selection of Normalization Procedures for Single-Cell RNA-Seq. *Cell Systems* **8,** 315–328 (2019).

164. Lytal, N., Ran, D. & An, L. Normalization Methods on Single-Cell RNA-seq Data: An Empirical Survey. *Frontiers in Genetics* **11,** 1–14 (2020).

165. Zhang, Z., Cui, F., Lin, C., Zhao, L. & Wang, C. Critical downstream analysis steps for single-cell RNA sequencing data. *Briefings In Bioinformatics* **00,** 1–11 (2021).

166. Heimberg, G., Bhatnagar, R., El-Samad, H. & Thomson, M. Low Dimensionality in Gene Expression Data Enables the Accurate Extraction of Transcriptional Programs from Shallow Sequencing. *Cell Systems* **2,** 239–250 (2016).

167. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology* **37,** 38–47 (2019).

168. Xiang, R. *et al.* A Comparison for Dimensionality Reduction Methods of Single-Cell RNA-seq Data. *Frontiers in Genetics* **12,** 320 (2021).

169. Pearson, K. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2,** 559–572 (1901).

170. Hyvärinen, A., Karhunen, J. & Oja, E. *Independent Component Analysis* (2001).

171. Maaten, L. V. D. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

172. Mcinnes, L., Healy, J. & Saul, N. UMAP : Uniform Manifold Approximation and Projection. *The Journal of Open Source Software* **3** (2018).

173. Coifman, R. R. *et al.* Geometric diffusions as a tool for harmonic analysis and structure definition of data : Diffusion maps. *PNAS* **102,** 7426–7431 (2005).

174. Wang, B. *et al.* SIMLR: A Tool for Large-Scale Genomic Analyses by Multi-Kernel Learning. *Proteomics* **18.** arXiv: 1703.07844 (2018).

175. Weinreb, C., Wolock, S. & Klein, A. M. SPRING : a kinetic interface for visualizing high dimensional single-cell expression data. *Bioinformatics* **34,** 1246–1248 (2018).

176. Bano, S. & Khan, M. N. A. A Survey of Data Clustering Methods. *International Journal of Advanced Science and Technology* **113,** 133–142 (2018).

177. Kim, T. *et al.* Impact of similarity metrics on single-cell RNA-seq data clustering. *Briefings in Bioinformatics* **20,** 2316–2326 (2019).

178. Freytag, S., Tian, L., Lönnstedt, I., Ng, M. & Bahlo, M. Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. *F1000Research* **7,** 1–26 (2018).

179. Inuwa-dutse, I., Liptrott, M. & Korkontzelos, I. A multilevel clustering technique for community detection. *Neurocomputing* **441,** 64–78 (2021).

180. Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics* **20,** 273–282 (2019).

181. Blondel, V. D., Guillaume, J.-l. & Lefebvre, E. Fast unfolding of communities in large networks. *arXiv* **July,** 1–12. arXiv: arXiv:0803.0476v2 (2008).

182. Freyag, S., Tang, L., Lönnstedt, I., Ng, M. & Bahlo, M. Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. *F1000Research* **7** (2018).

183. Duò, A., Robinson, M. D. & Soneson, C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research* **7** (2020).

184. Lieberman, Y., Rokach, L. & Shay, T. Correction: CaSTLe - Classification of single cells by transfer learning: Harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments(PLoS ONE (2018)13:10 (e0205499) DOI: 10.1371/journal.pone.0205499). *PLoS ONE* **13,** 1–16 (2018).

185. Kiselev, V. Y., Yiu, A. & Hemberg, M. Scmap: Projection of single-cell RNA-seq data across data sets. *Nature Methods* **15,** 359–362 (2018).

186. Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q. & Powell, J. E. ScPred: Accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biology* **20,** 1–17 (2019).

187. Abdelaal, T. *et al.* A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biology* **20,** 1–19 (2019).

188. Zhao, X., Wu, S., Fang, N., Sun, X. & Fan, J. Evaluation of single-cell classifiers for single-cell RNA sequencing data sets. *Briefings In Bioinformatics* **21,** 1581–1595 (2020).

189. Lin, X., Liu, H. & Wei, Z. An active learning approach for clustering single-cell RNA-seq data. *Laboratory Investigation,* 1–9 (2021).

190. Ranjan, B. *et al.* scConsensus: combining supervised and unsupervised clustering for cell type identification in single-cell RNA sequencing data. *BMC Bioinformatics* **22,** 1–15 (2021).

191. Moignard, V. & Göttgens, B. Dissecting stem cell differentiation using single cell expression profiling. *Current Opinion in Cell Biology* **43,** 78–86 (2016).

192. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32,** 381–386 (2014).

193. Cannoodt, R., Saelens, W. & Saeys, Y. Computational methods for trajectory inference from single-cell transcriptomics. *European Journal of Immunology* **46,** 2496–2506 (2016).

194. Griffiths, J., Scialdone, A. & Marioni, J. Using single-cell genomics to understand developmental processes and cell fate decisions. *Molecular Systems Biology* **14** (2018).

195. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nature Biotechnology* **37,** 547–554 (2019).

196. Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature Biotechnology* **34,** 637–645 (2016).

197. Street, K. *et al.* Slingshot: Cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* **19,** 1–16 (2018).

198. Wolf, F. A. *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biology* **20,** 1–9 (2019).

199. Setty, M. *et al.* Characterization of cell fate probabilities in single-cell data with Palantir. *Nature Biotechnology* **37,** 451–460 (2019).

200. Ji, Z. & Ji, H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Research* **44,** e117 (2016).

201. Chen, J., Rénia, L. & Ginhoux, F. Constructing cell lineages from single-cell transcriptomes. *Molecular Aspects of Medicine* **59,** 95–113 (2018).

202. La Manno, G. *et al.* RNA velocity of single cells. *Nature* **560,** 494–498 (2018).

203. Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology* **38,** 1408–1414 (2020).

204. Costa-silva, J., Domingues, D. & Lopes, F. M. RNA-Seq differential expression analysis : An extended review and a software tool. *PLoS ONE* **12,** e0190152 (2017).

205. Soneson, C. & Robinson, M. D. Bias, robustness and scalability in single-cell differential expression analysis. *Nature Methods* **15,** 255–261 (2018).

206. Van den Berge, K. *et al.* Observation weights to unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biology* **19,** 1–27 (2018).

207. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.-P. ZINB-WaVE: A general and flexible method for signal extraction from single-cell RNA-seq data. *bioRxiv,* 125112 (2017).

208. Finak, G. *et al.* MAST : a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology* **16,** 1–13 (2015).

209. Korthauer, K. D. *et al.* A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biology* **17,** 1–15 (2016).

210. Wang, T., Li, B., Nelson, C. E. & Nabavi, S. Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinformatics* **20,** 1–16 (2019).

211. Van den Berge, K. *et al.* Trajectory-based differential expression analysis for single-cell sequencing data. *Nature Communications* **11,** 1–13 (2020).

212. Van den Berge, K., Soneson, C., Robinson, M. D. & Clement, L. stageR: A general stage-wise method for controlling the gene-level false discovery rate in differential expression and differential transcript usage. *Genome Biology* **18,** 1–14 (2017).

213. Jiao, Y. *et al.* Improved maize reference genome with single-molecule technologies. *Nature* **546,** 524–527 (2017).

214. Palatini, U. *et al.* Improved reference genome of the arboviral vector Aedes albopictus. *Genome Biology* **21,** 1–29 (2020).

215. Nurk, S. *et al.* The complete sequence of a human genome. *bioRxiv* **May,** 1–32 (2021).

216. Guo, Y. *et al.* Improvements and impacts of GRCh38 human reference on high throughput sequencing data analysis. *Genomics* **109,** 83–90 (2017).

217. Fleischmann, R. D. *et al.* Whole-Genome Random Sequencing and Assembly of Haemophilus influenzae Rd. *Science* **269,** 496–512 (1995).

218. Stein, L. Genome Annotation: From Sequence to Biology. *Nature Reviews Genetics* **2,** 493–503 (2001).

219. Yates, A. D. *et al.* Ensembl 2020. *Nucleic Acids Research* **48,** 682–688 (2020).

220. Howe, K. L. *et al.* Ensembl 2021. *Nucleic Acids Research* **49,** D884–D891 (2021).

221. Pruitt, K. D., Tatusova, T., Brown, G. R. & Maglott, D. R. NCBI Reference Sequences (RefSeq): Current status, new features and genome annotation policy. *Nucleic Acids Research* **40,** 130–135 (2012).

222. Harrow, J. *et al.* GENCODE : The reference human genome annotation for The ENCODE Project. *Genome Research* **22,** 1760–1774 (2012).

223. Consortium, T. E. P., Michael P. Snyder & Gingeras, T. R. Perspectives on ENCODE. *Nature* **583,** 693–698 (2020).

224. Conesa, A. *et al.* A survey of best practices for RNA-seq data analysis. *Genome Biology* **17,** 1–19 (2016).

225. Wu, P. Y., Phan, J. H. & Wang, M. D. Assessing the impact of human genome annotation choice on RNA-seq expression estimates. *BMC bioinformatics* **14 Suppl 1,** 1–13 (2013).

226. Zhao, S. & Zhang, B. A comprehensive evaluation of ensembl, RefSeq, and UCSC annotations in the context of RNA-seq read mapping and gene quantification. *BMC Genomics* **16,** 1–14 (2015).

227. Chisanga, D., Liao, Y. & Shi, W. Impact of gene annotation choice on the quantification of RNA-seq data. *bioRxiv,* 2021.01.07.425794 (2021).

228. Hamaguchi, Y., Zeng, C. & Hamada, M. Impact of human gene annotations on RNA-seq differential expression analysis. *BMC Genomics preprint,* 1–20 (2021).

229. Simoneau, J., Gosselin, R. & Scott, M. S. Factorial study of the RNA-seq computational workflow identifies biases as technical gene signatures. *NAR Genomics and Bioinformatics* **2,** 1–18 (2020).

230. Deschamps-Francoeur, G., Simoneau, J. & Scott, M. S. Handling multi-mapped reads in RNA-seq. *Computational and Structural Biotechnology Journal* **18,** 1569–1576 (2020).

231. Lawson, N. D. *et al.* An improved zebrafish transcriptome annotation for sensitive and comprehensive detection of cell type-specific genes. *eLife* **9,** 1–76 (2020).

232. Jourdren, L., Bernard, M., Dillies, M. A. & Le Crom, S. Eoulsan: A cloud computing-based framework facilitating high throughput sequencing analyses. *Bioinformatics* **28,** 1542–1543 (2012).

233. Philpott, M. *et al.* Highly accurate barcode and UMI error correction using dual nucleotide dimer blocks allows direct single-cell nanopore transcriptome sequencing. *bioRxiv* **January** (2021).

234. Ewels, P., Lundin, S. & Max, K. MultiQC : summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32,** 3047–3048 (2016).

235. Delile, J. *et al.* Single cell transcriptomics reveals spatial and temporal dynamics of gene expression in the developing mouse spinal cord. *Development (Cambridge)* **146** (2019).

236. Von Bartheld, C. S., Bahney, J. & Herculano-houzel, S. The Search for True Numbers of Neurons and Glial Cells in the Human. *J Comp Neurol.* **524,** 3865–3895 (2017).

237. Willardsen, M. I. & Link, B. A. Cell biological regulation of division fate in vertebrate neuroepithelial cells. *Developmental Dynamics* **240,** 1865–1879 (2011).

238. Fischer, E. & Morin, X. Fate restrictions in embryonic neural progenitors. *Current Opinion in Neurobiology* **66,** 178–185 (2021).

239. Inaba, M. & Yamashita, Y. M. Asymmetric stem cell division: Precision for robustness. *Cell Stem Cell* **11,** 461–469 (2012).

240. Knoblich, J. A. Mechanisms of Asymmetric Stem Cell Division. *Cell* **132,** 583–597 (2008).

241. Sunchu, B. & Cabernard, C. Principles and mechanisms of asymmetric cell division. *Development (Cambridge)* **147** (2020).

242. Fernández, V., Llinares-benadero, C. & Borrell, V. Cerebral cortex expansion and folding : what have we learned ? *The EMBO Journal* **35,** 1021–1044 (2016).

243. Saade, M. *et al.* Sonic hedgehog signaling switches the mode of division in the developing nervous system. *Cell Reports* **4,** 492–503 (2013).

244. Venkei, Z. G. & Yamashita, Y. M. Emerging mechanisms of asymmetric stem cell division. *Journal of Cell Biology* **217,** 3785–3795 (2018).

245. Dréau, G. L., Saade, M., Gutiérrez-vallejo, I. & Martí, E. The strength of SMAD1/5 activity determines the mode of stem cell division in the developing spinal cord. *Journal of Cell Biology* **204,** 591–605 (2014).

246. Iacopetti, P., Michelini, M., Tuckmann, I. N. G. O. S., Jo, B. & Uttner, W. I. B. H. Expression of the antiproliferative gene TIS21 at the onset of neurogenesis identifies single neuroepithelial cells that switch from proliferative to neuron-generating division. *Proceedings of the National Academy of Sciences of the United States of America* **96,** 4639–4644 (1999).

247. Haubensak, W., Attardo, A., Denk, W. & Huttner, W. B. Neurons arise in the basal neuroepithelium of the early mammalian telencephalon : A major site of neurogenesis. *PNAS* **101** (2004).

248. Arai, Y. *et al.* Neural stem and progenitor cells shorten S-phase on commitment to neuron production. *Nature Communications* **2** (2011).

249. Danesin, C., Ferreira, M. A., Degond, P. & Theveneau, E. Anteroposterior elongation of the chicken anterior trunk neural tube is hindered by interaction with its surrounding tissues. *Cells & Development* (2021).

250. Wilcock, A. C., Swedlow, J. R. & Storey, K. G. Mitotic spindle orientation distinguishes stem cell and terminal modes of neuron production in the early spinal cord. *Development* **134,** 1943–1954 (2007).

251. MOLINA, A. *et al.* G1 phase lengthening during neural tissue development involves CDC25B induced G1 heterogeneity. *bioRxiv* **November** (2020).

252. Molina, A. & Pituello, F. Playing with the cell cycle to build the spinal cord. *Developmental Biology* **432,** 14–23 (2017).

253. Frédéric, B. *et al.* Neurogenic decisions require a cell cycle independent function of the CDC25B phosphatase. *eLife* **7** (2018).

254. Kessaris, N., Pringle, N. & Richardson, W. D. Specification of CNS glia from neural stem cells in the embryonic neuroepithelium. *Philosophical Transactions of the Royal Society B: Biological Sciences* **363,** 71–85 (2008).

255. Ribes, V. & Briscoe, J. Establishing and Interpreting Graded Sonic Hedgehog Signaling during Vertebrate Neural Tube Patterning : The Role of Negative Feedback. *Cold Spring Harbor perspectives in biology* **1** (2009).

256. Scott, K., O'Rourke, R., Winkler, C. C., Kearns, C. A. & Appel, B. Temporal single-cell transcriptomes of zebrafish spinal cord pMN progenitors reveal distinct neuronal and glial progenitor populations. *bioRxiv* **April** (2021).

257. Moreau, M. X., Saillour, Y., Cwetsch, A. W. & Pierani, A. Single-cell transcriptomics of the early developing mouse cerebral cortex disentangle the spatial and temporal components of neuronal fate acquisition. *Development* **148** (2021).

258. Chazarra-Gil, R., Dongen, S. V. & Kiselev, V. Y. Flexible comparison of batch correction methods for single-cell RNA-seq using BatchBench. *Nucleic acids research* **49,** 1–12 (2021).

259. Tran, H. T. N. *et al.* A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome biology* **21,** 12 (2020).

260. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology* **36,** 411–420 (2018).

261. Buettner, F. *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology* **33,** 155–160 (2015).

262. Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C. & Stegle, O. f-scLVM: Scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biology* **18,** 1–13 (2017).

263. Barron, M. & Li, J. Identifying and removing the cell-cycle effect from single-cell RNA-Sequencing data. *Scientific Reports* **6,** 1–10 (2016).

264. Vento-Tormo, R. *et al.* Single-cell reconstruction of the early maternal–fetal interface in humans. *Nature* **563,** 347–353 (2018).

265. Hsiao, C. J. *et al.* Characterizing and inferring quantitative cell cycle phase in single-cell RNA-seq data analysis. *Genome Research* **30,** 611–621 (2020).

266. Schwabe, D., Formichetti, S., Junker, J. P., Falcke, M. & Rajewsky, N. The transcriptome dynamics of single cells during the cell cycle. *Molecular Systems Biology* **16,** 1–20 (2020).

267. Zheng, S. C. *et al.* Universal prediction of cell cycle position using transfer learning. *bioRxiv* **April,** 1–56 (2021).

268. Burt, D. & Pourquie, O. Chicken Genome - Science Nuggets to Come Soon. *Science* **300** (2003).

269. Hubbard, S. J. *et al.* Transcriptome analysis for the chicken based on 19 , 626 finished cDNA sequences and 485 , 337 expressed sequence tags. *Genome Research* **15,** 174–183 (2005).

270. Cheng, Y. & Burt, D. W. Chicken genomics. *International Journal of Developmental Biology* **62,** 265–271 (2018).

271. Warren, W. C. *et al.* A new chicken genome assembly provides insight into avian genome structure. *G3: Genes, Genomes, Genetics* **7,** 109–117 (2017).

272. Aken, B. L. *et al.* The Ensembl gene annotation system. *The Journal of Biological Databases and Curation,* 1–19 (2016).

273. Lopez, F. *et al.* Explore, edit and leverage genomic annotations using Python GTF toolkit. *Bioinformatics* **35,** 3487–3488 (2019).

274. Kuo, R. I. *et al.* Normalized long read RNA sequencing in chicken reveals transcriptome complexity similar to human. *BMC Genomics* **18,** 1–19 (2017).

275. Kovaka, S. *et al.* Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *bioRxiv,* 694554 (2019).

276. Shao, M. & Kingsford, C. Accurate Assembly of Transcripts Through Phase-Preserving Graph Decomposition. *Nature Biotechnology* **35,** 1167–1169 (2017).

277. Venturini, L., Caim, S., Kaithakottil, G. G., Mapleson, D. L. & Swarbreck, D. Leveraging multiple transcriptome assembly methods for improved gene structure annotation. *GigaScience* **7,** 1–15 (2018).

278. Pertea, G. & Pertea, M. GFF Utilities: GffRead and GffCompare. *F1000Research* **9,** 304 (2020).

279. Rangaraju, S. *et al.* Quantitative proteomics of acutely-isolated mouse microglia identifies novel immune Alzheimer's disease-related proteins. *Molecular Neurodegeneration* **13,** 1–19 (2018).

280. Adams, K. L. *et al.* Endothelin-1 signaling maintains glial progenitor proliferation in the postnatal subventricular zone. *Nature Communications* **11,** 1–17 (2020).

281. Tung, L. H., Shao, M. & Kingsford, C. Quantifying the benefit offered by transcript assembly with Scallop-LR on single-molecule long reads. *Genome Biology* **20,** 1–18 (2019).

282. Di Tommaso, P. *et al.* Nextflow enables reproducible computational workflows. *Nature Biotechnology* **35,** 316–319 (2017).

283. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* **28,** 511–515 (2010).

284. Deblasio, D. A. N., Kim, K. & Kingsford, C. More Accurate Transcript Assembly via Parameter Advising. *Journal of Computational Biology* **27,** 1181–1189 (2020).

285. Nip, K. M. *et al.* RNA-Bloom enables reference-free and reference- guided sequence assembly for single-cell transcriptomes. *Genome Research* **30,** 1191–1200 (2020).

286. Qiao, Y. *et al.* High-resolution annotation of the mouse preimplantation embryo transcriptome using long-read sequencing. *Nature Communications* **11,** 1–13 (2020).

287. Lähnemann, D. *et al. Eleven grand challenges in single-cell data science* **1,** 1–35 (Genome Biology, 2020).

288. Wu, A. R. *et al.* Quantitative assessment of single-cell RNA-sequencing methods. *Nature Methods* **11,** 41–46 (2014).

289. Hicks, S. C., Townes, F. W., Teng, M. & Irizarry, R. A. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics* **19,** 562–578 (2018).

290. Liu, Y., Beyer, A. & Aebersold, R. On the Dependency of Cellular Protein Levels on mRNA Abundance. *Cell* **165,** 535–550 (2016).

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

292. Qiu, P. Embracing the dropouts in single-cell RNA-seq analysis. *Nature Communications* **11,** 1–9 (2020).

293. Haque, A., Engel, J., Teichmann, S. A. & Lönnberg, T. A practical guide to single-cell RNA- sequencing for biomedical research and clinical applications. *Genome Medicine* **9,** 1–12 (2017).

294. Hou, W., Ji, Z., Ji, H. & Hicks, S. C. A systematic evaluation of single-cell RNA-sequencing imputation methods. *Genome Biology* **21,** 1–30 (2020).

295. Lukusa, T. M., Lee, S.-m. & Li, C.-s. Review of Zero-Inflated Models with Missing Data. *Current Research in Biostatistics* **7,** 1–12 (2017).

296. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J. P. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications* **9** (2018).

297. He, J. *et al.* Identifying transposable element expression dynamics and heterogeneity during development at the single-cell level with a processing pipeline scTE. *Nature communications* **12,** 1456 (2021).

298. Choi, K., Chen, Y., Skelly, D. A. & Churchill, G. A. Bayesian model selection reveals biological origins of zero inflation in single-cell transcriptomics. *Genome Biology* **21,** 1–16 (2020).

299. Montoro, D. T. *et al.* A revised airway epithelial hierarchy includes CFTR-expressing ionocytes. *Nature* **560,** 319–324 (2018).

300. Karlsson, K. & Linnarsson, S. Single-cell mRNA isoform diversity in the mouse brain. *BMC Genomics* **18,** 1–11 (2017).

301. Gupta, I. *et al.* Single-cell isoform RNA sequencing characterizes isoforms in thousands of cerebellar cells. *Nature Biotechnology* **36,** 1197–1202 (2018).

302. Lebrigand, K., Magnone, V., Barbry, P. & Waldmann, R. High throughput error corrected Nanopore single cell transcriptome sequencing. *Nature Communications* **11** (2020).

303. Singh, M. *et al.* High-throughput targeted long-read single cell sequencing reveals the clonal and transcriptional landscape of lymphocytes. *Nature Communications* **10** (2019).

304. Volden, R. & Vollmers, C. Highly Multiplexed Single-Cell Full-Length cDNA Sequencing of human immune cells with 10X Genomics and R2C2. *bioRxiv* **June** (2021).

305. Stahl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353,** 78–82 (2016).

306. Vickovic, S. *et al.* High-definition spatial transcriptomics for in situ tissue profiling. *Nature Methods* **16,** 987–990 (2019).

307. Srivatsan, S. R. *et al.* Embryo-scale, single-cell spatial transcriptomics. *Science* **117,** 111–117 (2021).

308. Moor, A. E. & Itzkovitz, S. Spatial transcriptomics : paving the way for tissue-level systems biology. *Current Opinion in Biotechnology* **46,** 126–133 (2017).

309. Lein, E., Borm, L. E. & Linnarsson, S. The promise of spatial transcriptomics for neuroscience in the era of molecular cell typing. *Science* **358,** 64–69 (2017).

310. Burgess, D. J. Spatial transcriptomics coming of age. *Nature Genetics* **20,** 317 (2019).

311. Codeluppi, S. *et al.* Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature Methods* **15,** 932–935 (2018).

312. Eng, C.-H. L. *et al.* Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature* **568,** 235–239 (2019).

313. SG, R. *et al.* Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363** (2019).

314. Joglekar, A., y Prjibelski, A. & Mahfouz, A. A spatially resolved brain region- and cell typespecific isoform atlas of the postnatal mouse brain. *Nature Communications* **12** (2021).

315. Nagasawa, S., Kashima, Y., Suzuki, A. & Suzuki, Y. Single-cell and spatial analyses of cancer cells: toward elucidating the molecular mechanisms of clonal evolution and drug resistance acquisition. *Inflammation and Regeneration* **41** (2021).

316. Rosenberg, A. B. *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360,** 176–182 (2018).

317. Hu, P. *et al.* Dissecting Cell-Type Composition and Activity- Dependent Transcriptional State in Mammalian Brains by Massively Parallel Single-Nucleus Technology Dissecting Cell-Type Composition and Activity- Dependent Transcriptional State in Mammalian Brains by Massively Parallel Single-Nucleus RNA-Seq. *Molecular Cell* **68,** 1006–1015.e7 (2017).

318. Gao, R. *et al.* Nanogrid single-nucleus RNA sequencing reveals phenotypic diversity in breast cancer. *Nature Communications* **8** (2017).

319. Habib, N. *et al.* Massively parallel single- nucleus RNA-seq with. *Nature* **14** (2017).

320. Wu, P.-Y. L. Advancing precision medicine through integrative bioinformatics approaches for robust biological knowledge discovery (2018).

321. Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods* **14,** 865–868 (2017).

322. Hu, Y. *et al.* Simultaneous profiling of transcriptome and DNA methylome from a single cell. *Genome Biology* **17,** 1–11 (2016).

323. Bian, S. *et al.* Single-cell multiomics sequencing and analyses of human colorectal cancer. *Science* **362,** 1060–1063 (2018).

324. Lee, C. M. *et al.* Single-cell RNA-seq analysis revealed long-lasting adverse effects of tamoxifen on neurogenesis in prenatal and adult brains. *Proceedings of the National Academy of Sciences of the United States of America* **117,** 19578–19589 (2020).

325. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic rna-seq quantification. *Nature Biotechnology* **34,** 525–528 (2016).

326. Patro, R. *et al.* Salmon: fast and bias-aware quantification of transcript expression using dual-phase inference. *Nature Methods* **14,** 417–419 (2017).

327. Du, Y., Huang, Q., Arisdakessian, C. & Garmire, L. X. Evaluation of STAR and Kallisto on Single Cell RNA-Seq Data Alignment. *G3: Genes, Genomes, Genetics* **10,** 1775–1783 (2020).

328. Ren, J., Sun, C., Clinton, M. & Yang, N. Dynamic Transcriptional Landscape of the Early Chick Embryo. *Frontiers in Cell and Developmental Biology* **7,** 1–15 (2019).

329. Thomas, S., Underwood, J. G., Tseng, E. & Holloway, A. K. Long-Read Sequencing of Chicken Transcripts and Identification of New Transcript Isoforms. *PLoS ONE* **9,** 1–6 (2014).

330. Wang, M. F. Z. *et al.* Uncovering transcriptional dark matter via gene annotation independent single-cell RNA sequencing analysis. *Nature Communications* **12,** 1–10 (2021).

331. Shields, E. J. *et al.* Genome annotation with long RNA reads reveals new patterns of gene expression in an ant brain. *bioRxiv Bioinformatics* **April,** 1–23 (2021).

332. Botvinnik, O. B. *et al.* Single-cell transcriptomics for the 99 . 9 % of species without reference genomes Authors. *bioRxiv* **July** (2021).

333. Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L. & Rice, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa / Illumina FASTQ variants. *Nucleic Acids Research* **38,** 1767–1771 (2010).

334. Williams, C. R., Baccarella, A., Parrish, J. Z. & Kim, C. C. Trimming of sequence reads alters RNA- Seq gene expression estimates. *BMC Bioinformatics* **17,** 1–13 (2016).

335. Trapnell, C., Pachter, L. & Salzberg, S. L. TopHat : discovering splice junctions with RNA-Seq. *Bioinformatics* **25,** 1105–1111 (2009).

336. Kim, D. *et al.* TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology* **14** (2013).

337. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25,** 1754–1760 (2009).

Appendices

Appendix A

Eoulsan

A.1 Eoulsan design file

This is the design file produced by the *eoulsan createdesign* command, when processing SYMASYM data.

```
[Header]
DesignFormatVersion=2
GenomeFile=data/genome/GCF_000002315.5_GRCg6a_genomic.fna
GtfFile=data/annotation/ref_GRCg6a_top_level.corrected.gtf

[Experiments]
Exp.exp1.name=SymasymB

[Columns]
SampleId  SampleName  Reads Date  FastqFormat RepTechGroup
   Exp.exp1.Condition  Exp.exp1.Reference  UUID
S1R1001 S1_R1_001 [S1_R1_001.fastq.gz,S1_R2_001.fastq.gz]
   2019-05-24  fastq-sanger  S1_R1_001 S1_R1_001 false 4540
   b2b5-35d2-493d-9d8f-0cd59cabf439
S2R1001 S2_R1_001 [S2_R1_001.fastq.gz,S2_R2_001.fastq.gz]
   2019-05-24  fastq-sanger  S2_R1_001 S2_R1_001 false 81
   f7466f-4bb8-4c86-88f9-bab59c9f1bae
S3R1001 S3_R1_001 [S3_R1_001.fastq.gz,S3_R2_001.fastq.gz]
   2019-05-24  fastq-sanger  S3_R1_001 S3_R1_001 false 534773
   f5-8577-434f-a8cd-4b34f32c6244
```

```
14 S4R1001 S4_R1_001 [S4_R1_001.fastq.gz,S4_R2_001.fastq.gz]
      2019-05-24  fastq-sanger  S4_R1_001 S4_R1_001 false
      dc06f6d4-7509-4ddd-aa37-8f661ae5d940
```

Listing A.1 SYMASYM Eoulsan design file

A.2 Typical scRNA-seq workflow

The following pages contain the full 10x Genomics workflow that I used to pre-process SYMASYM data. The commented parts are typical steps that may be used to pre-process 10x Genomics data, which I did not needed in this case.

```
<analysis>
	<formatversion>1.0</formatversion>
	<name>SYMASYM project - dataset B - 2019</name>
	<description>10xGenomics mouse data - 5000 cells - 1TP</description>
	<author>Lehmann</author>

	<!-- The steps of the workflow -->
	<steps>
		<!-- Create STAR index -->
		<!-- This step may be skipped if needed (long time running). In this case, the step
"mapreads" will automatically generate an index. -->
		<!--step id="step5createstarindex" skip="false" requiredprocs="`nproc`">
			<module>starindexgenerator</module>
			<parameters-->
				<!-- The overhang value must be greater than the reads length -->
				<!--parameter>
					<name>overhang</name>
					<value>100</value>
				</parameter>
				<parameter>
					<name>use.gtf.file</name>
					<value>true</value>
				</parameter>
				<parameter>
					<name>gtf.feature.exon</name>
					<value>exon</value>
				</parameter>
				<parameter>
					<name>gtf.tag.exon.parent.transcript</name>
					<value>Parent</value>
				</parameter>
			</parameters>
		</step-->

		<!-- Merge technical replicates -->
		<!--step id="step0mergereplicates" skip="true">
			<module>technicalreplicatemerger</module>
			<inputs>
				<input>
					<port>input</port>
					<fromstep>step0Fastqimport</fromstep>
					<fromport>output</fromport>
				</input>
			</inputs>
			<parameters>
				<parameter>
					<name>format</name>
					<value>fastq</value>
				</parameter>
			</parameters>
		</step-->

		<!-- FastQC of non filtered reads -->
		<step id="step1fastqc" skip="false">
			<module>fastqc</module>
			<parameters/>
		</step>

		<!-- Filter reads: remove low quality reads -->
		<step id="step2filterreads" skip="false">
			<module>filterreads</module>
			<parameters>
				<parameter>
					<name>illuminaid</name>
					<value></value>
				</parameter>
				<parameter>
					<name>quality.threshold</name>
					<value>30</value>
				</parameter>
			</parameters>
		</step>

		<!-- FastQC of filtered reads -->
		<step id="step3fastqc" skip="false">
			<module>fastqc</module>
			<parameters/>
		</step>

		<!-- Extract cell barcodes and identify the most likely true barcodes using the
'knee' method. -->
```

```
		<step id="step4whitelist" skip="false">
			<module>umiwhitelist</module>
			<parameters/>
		</step>

		<!-- Extract UMI barcode from a read and add it to the read name, leaving any sample
barcode in place.  -->
		<step id="step5extract" dataproduct="match" skip="false">
			<module>umiextract</module>
			<parameters/>
		</step>

		<!-- Map reads -->
		<step id="step6mapreads" skip="false" requiredprocs="`nproc`">
			<module>mapreads</module>
			<parameters>
				<parameter>
					<name>mapper</name>
					<value>star</value>
				</parameter>
				<parameter>
					<name>mapper.arguments</name>
					<value>--outSAMunmapped Within</value>
				</parameter>
			</parameters>
		</step>

		<!-- Quality filter of SAM files -->
		<step id="step8filtersam"  skip="false">
			<module>filtersam</module>
			<parameters>
				<parameter>
					<name>removeunmapped</name>
					<value>true</value>
				</parameter>
				<parameter>
					<name>removemultimatches</name>
					<value>true</value>
				</parameter>
			</parameters>
		</step>

		<!-- Assign reads to genes -->
		<step id="step9featurecounts" requiredprocs="`nproc`" skip="false">
			<module>featurecounts</module>
			<inputs>
				<input>
					<port>input</port>
					<fromstep>step8filtersam</fromstep>
					<fromport>output</fromport>
				</input>
			</inputs>
			<parameters>
				<parameter>
					<name>strand_specificity</name>
					<value>1</value>
				</parameter>
			</parameters>
		</step>

		<!-- MutliQC of filtered and mapped reads -->
		<step id="step10multiqc" skip="false">
			<module>multiqc</module>
			<parameters>
				<parameter>
					<name>reports</name>
					<value>fastqc,mapreads,featurecounts</value>
				</parameter>
				<parameter>
					<name>use.docker</name>
					<value>false</value>
				</parameter>
			</parameters>
		</step>

		<!-- Convert SAM to BAM -->
		<step id="step11samtobam" skip="false">
			<module>sam2bam</module>
			<inputs>
				<input>
					<port>input</port>
```

```
                                <fromstep>step9featurecounts</fromstep>
                                <fromport>outputsam</fromport>
                        </input>
                </inputs>
                <parameters/>
        </step>

        <!-- Count UMIs per gene per cell -->
        <step id="step12umicounts" skip="false">
                <module>umicount</module>
                <parameters/>
        </step>

        <!-- Create a SingleCellExperiment Bioconductor Object in a RDS file -->
        <step id="step13singlecellexperiment" skip="false">
                <module>rsinglecellexperimentcreator</module>
                <parameters>
                        <parameter>
                                <name>input.matrices</name>
                                <value>true</value>
                        </parameter>
                        <parameter>
                                <name>design.prefix</name>
                                <value>Cell.</value>
                        </parameter>
                        <parameter>
                                <name>r.execution.mode</name>
                                <value>process</value>
                        </parameter>
                 </parameters>
        </step-->

        <!-- Create a Cell Ranger-like matrix file -->
        <!--step id="step14cellrangermatrix" skip="false">
                <module>matrix2cellrangermatrix</module>
                <parameters>
                        <parameter>
                                <name>input.matrices</name>
                                <value>true</value>
                        </parameter>
                        <parameter>
                                <name>use.gene.annotation</name>
                                <value>true</value>
                        </parameter>
                        <parameter>
                                <name>gene.annotation.field.name</name>
                                <value>Gene name</value>
                        </parameter>
                </parameters>
        </step-->
    </steps>

    <!-- Global configuration -->
    <globals>
        <!-- Define the location of the Docker connection -->
        <!--parameter>
                <name>main.docker.uri</name>
                <value>unix:///var/run/docker.sock</value>
        </parameter-->
    </globals>
</analysis>
```

A.3 Reads demultiplexing and filtering

Initially developed in the 2000's by the Wellcome Trust Sanger Institute [333], the FASTQ format became *de facto* the standard to store sequencing data. It is a text-based file format that stores both the raw sequence and its corresponding quality scores. The quality score of each base is encoded by a single ASCII character (one byte per quality value). This rule facilitates reading, processing and filtering of FASTQ files. An example of a FASTQ file is shown in Figure A.1.

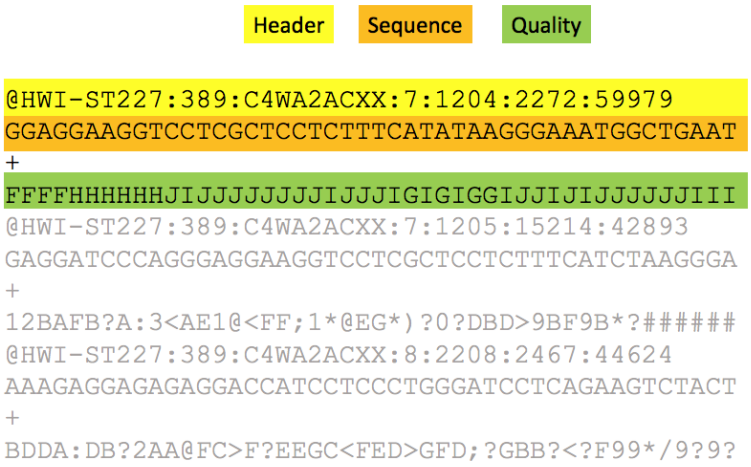


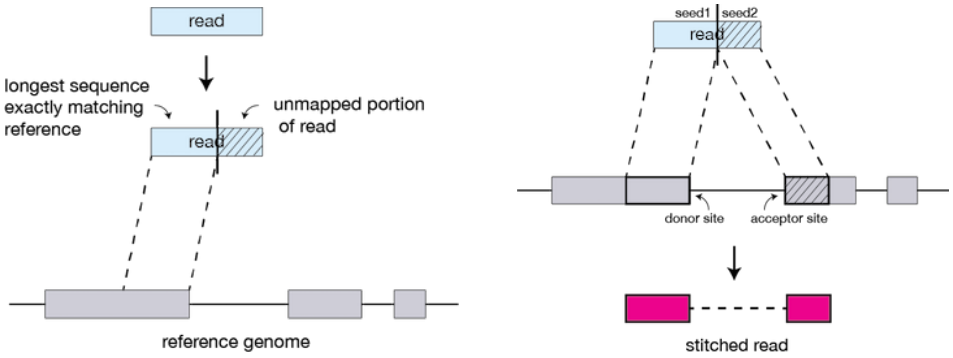
Figure A.1 Example of a FASTQ file. In the FASTQ format, each group of four lines describe a single sequencing read. Hence, a total of three reads are represented here. The header (first line) always starts with a at (@) sign and contains the sequence identifier. The second line is the read sequence itself. The third line is a separator, which is mostly a plus (+) sign. The fourth line encodes the per base quality score which is assumed to match the sequence written in the second line. Image from https://biocorecrg.github.io/PhD_course/fastq.html

The quality of each base is calculated with *Phred* scores (or Q scores). It reflects the probability that a given base is called incorrectly by the sequencer. It is defined by the equation: $Q = -10 * \log_{10} P$, where P is the probability that the base is erroneous. For example, a Phred score of Q30 means that the base call accuracy is 99.9% (*e.g.* there is 1 in 1000 possibilities that the base is wrong). This metrics thus helps to filter poor quality sequences out of the FASTQ files. The FASTQ filtering step also implies to remove poor quality

reads extremities (also called *trimming*) and non biological sequences (*e.g.* sample indexes or sequences primers) [334]. Once the filtering is completed, quality control of the remaining reads can be performed by using usual bulk RNA-seq tools such as FastQC [115].

A.4 Genome alignment or mapping

Although this might not be a major concern for most of scRNA-seq studies at the moment, the key difference between genome and transcriptome mapping is that the former one enables the identification of novel genes or transcripts. The main challenge with genome alignment is to properly identify splice junctions in order not to lose reads that may overlap two exons as shown in Figure A.2. It thus imposes the use of specialised short-reads aligners that can handle splicing (a.k.a. *splice-aware* aligners), some of the most popular ones being STAR [122] and TopHat [335, 336]. Some aligners are specialized in transcriptome alignment, such as BWA [337], but we do not recommend them for scRNA-seq (see [121]).



(a) Read matching a single exon. **(b)** Read spanning a splice junction.

Figure A.2 Illustration of a typical genome alignment strategy. Splice-aware aligners follow different strategies either they need to align unspliced or spliced reads. (a) A read (in blue) is defined as unspliced if it can be mapped with high confidence to a single exon (exons are here represented in gray little boxes). (b) Remaining unmapped reads are then splitted into two halves (here called read1 and read2) and aligned separately in order to identify splice junctions. Images from https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html

Appendix B

Re-annotation pipeline: LR data

B.1 SYMASYM bulk long-reads raw data quality report

We assessed data quality with ToulligQC, a tool dedicated to the QC analyses of raw Oxford Nanopore data: https://github.com/GenomicParisCentre/toulligQC. It is also developed at IBENS. We show here only the first page (the overall analysis is 8 pages).

ToulligQC report for LONGCHICK_A2019

Run id: LONGCHICK_A2019
Report name: 20191127_LONGCHICK_A2019
Run date: 2019-11-27T14:29:53Z
Report date : 12/04/19 08:33:43 UTC

Summary

Basic Statistics

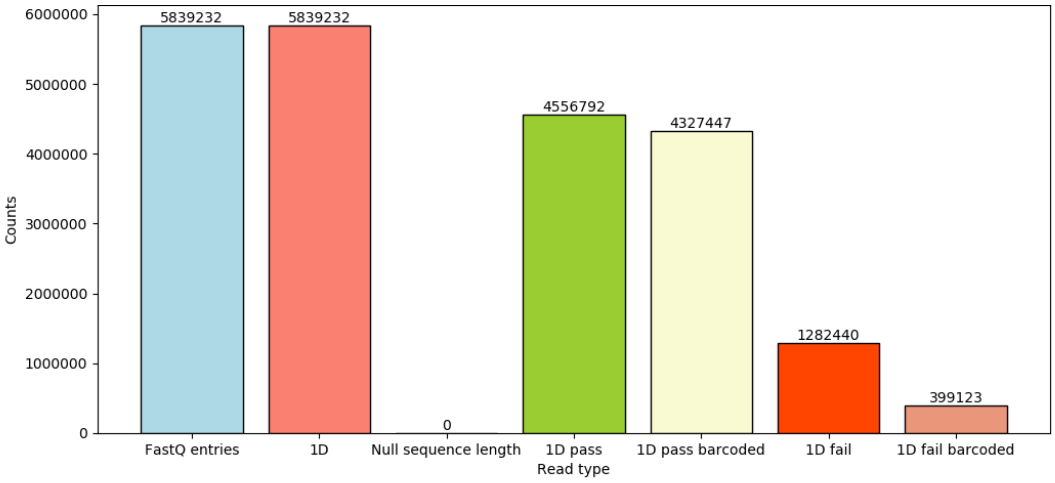
Run info

Measure	Value
Run id	LONGCHICK_A2019
Sample	LONGCHICK_A2019
Report name	20191127_LONGCHICK_A2019
Run date	2019-11-27T14:29:53Z
Run duration	71:59:13
Flowcell id	FAK97187
Flowcell version	
Kit	
Yield (Gbp)	13.85
Read count	5839232

Software info

Measure	Value
MinKNOW version	3.5.4
Basecaller name	guppy-basecalling
Basecaller version	3.3.3+fa743a6
Basecaller analysis	1d_basecalling
ToulligQC version	1.2
Hostname	MT-110298
Device	minion
Device ID	MN17734
Model file	template_r9.4.1_450bps_hac.jsn

Read count histogram



	FastQ_entries	1D	Null sequence length	1D pass	1D pass barcoded	1D fail	1D fail barcoded
count	5839232	5839232	0	4556792	4327447	1282440	399123
frequency	100.00	100.00	0.00	78.04	74.11	21.96	6.84

B.2 Eoulsan workflow file for long-reads ONT

The following pages contain the full long-reads workflow (Oxford Nanopore Technologies, ONT) that I used to pre-process SYMASYM bulk long-reads data.

```
<analysis>
    <formatversion>1.0</formatversion>

    <!-- The steps of the workflow -->
    <steps>


        <!-- Create a custom Minimap2 index feature annotation -->
        <step skip="false">
            <module>minimap2indexgenerator</module>
            <parameters>

                <!-- Create an index for splicing mode of Minimap2 -->
                <parameter>
                    <name>indexer.arguments</name>
                    <value>-x splice</value>               <!-- cDNA       -->
                    <!--value>-x splice -uf -k14</value--> <!-- Direct RNA -->
                </parameter>
            </parameters>
        </step>


        <!-- FastQC of raw reads -->
        <step id="rawfastqc" skip="false">
            <module>fastqc</module>
        </step>


        <!-- Filter reads -->
        <step skip="false" discardoutput="asap">
            <module>filterreads</module>
            <parameters>

                <!-- Remove polyN tails of the reads -->
                <parameter>
                    <name>trimpolynend</name>
                    <value></value>
                </parameter>

            </parameters>
        </step>


        <!-- Mapping of the reads -->
        <step skip="false" discardoutput="false">
            <module>mapreads</module>
            <parameters>

                <!-- Use use Minimap2 as mapper -->
                <parameter>
                    <name>mapper</name>
                    <value>minimap2</value>
                </parameter>

                <!-- The version of Minimap2 to use is 2.12 -->
                <parameter>
                    <name>mapper.version</name>
                    <value>2.12</value>
                </parameter>

                <!-- We use Minimap2 in splice mode -->
                <parameter>
                    <name>mapper.arguments</name>
                    <value>-x splice</value>               <!-- cDNA       -->
                    <!--value>-x splice -uf -k14</value--> <!-- Direct RNA -->
                </parameter>
            </parameters>
        </step>


        <!-- Filtering of the SAM alignments -->
        <step skip="false" discardoutput="false" requiredmemory="100Gb">
```

```
            <module>filtersam</module>
            <parameters>

                <!-- Remove the unmap entries in the files -->
                <parameter>
                    <name>removeunmapped</name>
                    <value>true</value>
                </parameter>

                <!-- Remove alignmments with poor quality -->
                <parameter>
                    <name>quality.threshold</name>
                    <value>1</value>
                </parameter>

                <!-- Remove supplementary alignments -->
                <parameter>
                    <name>removesupplementary</name>
                    <value>true</value>
                </parameter>

                <!-- Remove the multimaches alignments -->
                <parameter>
                    <name>removemultimatches</name>
                    <value>true</value>
                </parameter>

            </parameters>
        </step>


        <!-- Convert SAM to BAM file and sort the BAM file by coordinate -->
        <step skip="false" requiredmemory="100Gb">
            <module>sam2bam</module>
            <parameters>

                <!-- Compression level of the BAM file -->
                <parameter>
                    <name>compression.level</name>
                    <value>5</value>
                </parameter>

            </parameters>
        </step>


        <!-- MultiQC -->
        <step skip="false">
            <module>multiqc</module>
            <parameters>

                <!-- MultiQC will contain reports from FastQC, STAR and HTSeq-count -->
                <parameter>
                    <name>reports</name>
                    <value>fastqc,mapreads</value>
                </parameter>

                <!-- Docker is required to launch this step -->
                <parameter>
                    <name>use.docker</name>
                    <value>false</value>
                </parameter>

            </parameters>
        </step>
    </steps>

</analysis>
```

B.3 MultiQC summary statistics of SYMASYM bulk long-reads

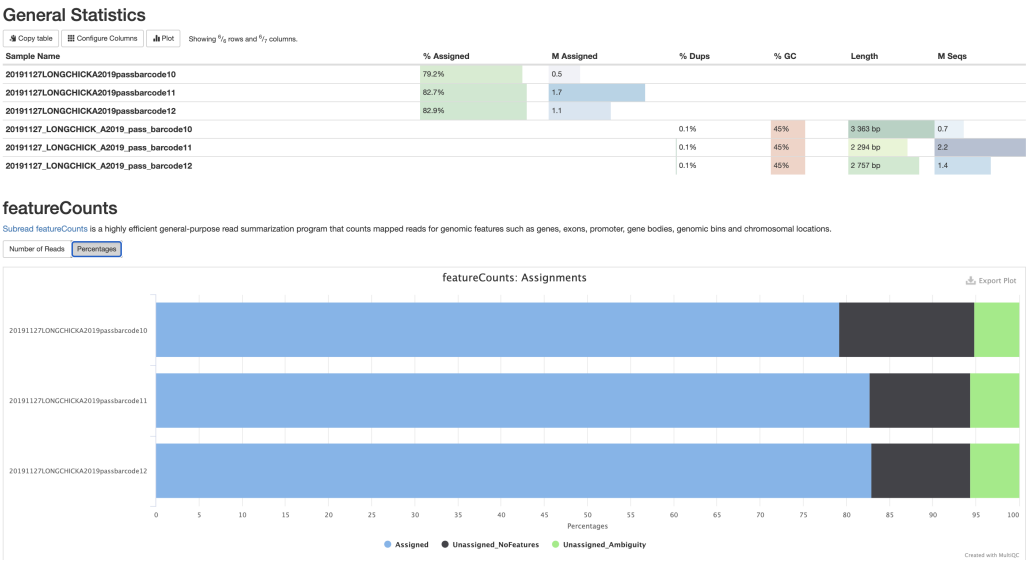


Figure B.1 MultiQC summary plot of SYMASYM bulk long-reads data.

RÉSUMÉ

Ces dernières années, l'émergence des approches en cellules uniques (scRNA-seq) a favorisé la caractérisation de l'hétérogénéité cellulaire avec une précision inégalée. Malgré leur démocratisation, l'analyse de ces données reste complexe, en particulier pour les organismes dont les annotations sont incomplètes.
Au cours ma thèse, j'ai observé que les annotations génomiques du poulet sont lacunaires, ce qui engendre la perte d'un grand nombre de lectures de séquençage. J'ai évalué à quel point une annotation améliorée affecte les résultats biologiques et les conclusions issues de ces analyses. Nous proposons une nouvelle approche basée sur la ré-annotation du génome à partir de données scRNA-seq et de RNA-seq bulk en lectures longues.
Ce projet de biologie computationnelle s'appuie sur une étroite collaboration avec l'équipe expérimentale de Xavier Morin (IBENS). Le principal objectif biologique est la recherche de signatures de mode de division symétrique et asymétrique au sein de progéniteurs neuronaux. Afin d'identifier les principaux changements transcriptionnels, j'ai mis en place des approches dédiées à la recherche de signatures géniques à partir de données scRNA-seq.

MOTS CLÉS

Bioinformatique; Pipeline d'analyse; Transcriptomique en cellule unique; Séquençage en lectures longues; Ré-annotation de génome; Progéniteurs neuronaux

ABSTRACT

In recent years, single-cell RNA-seq (scRNA-seq) has fostered the characterization of cell heterogeneity at a remarkable high resolution. Despite their democratization, the analysis of scRNA-seq remains a challenge, particularly for organisms whose genomic annotations are partial.
During my PhD, I observed that the chick genomic annotations are often incomplete, thus resulting in a loss of a large number of sequencing reads. I investigated how an enriched annotation affects the biological results and conclusions from these analyses. We developed a novel approach based on the re-annotation of the genome with scRNA-seq data and long reads bulk RNA-seq.
This computational biology project capitalises on a tight collaboration with the experimental team of Xavier Morin (IBENS). The main biological focus is the search for signatures of symmetric versus asymmetric division mode in neural progenitors. In order to identify the key transcriptional switches that occur during the neurogenic transition, I have implemented bioanalysis approaches dedicated to the search for gene signatures from scRNA-seq data.

KEYWORDS

Bioinformatics; Analysis pipeline; Single-cell RNA-seq; Long-read sequencing; Genome re-annotation; Neural progenitors