Choosing Genomics Tools

July, 2023

Table of Contents

# About this Course

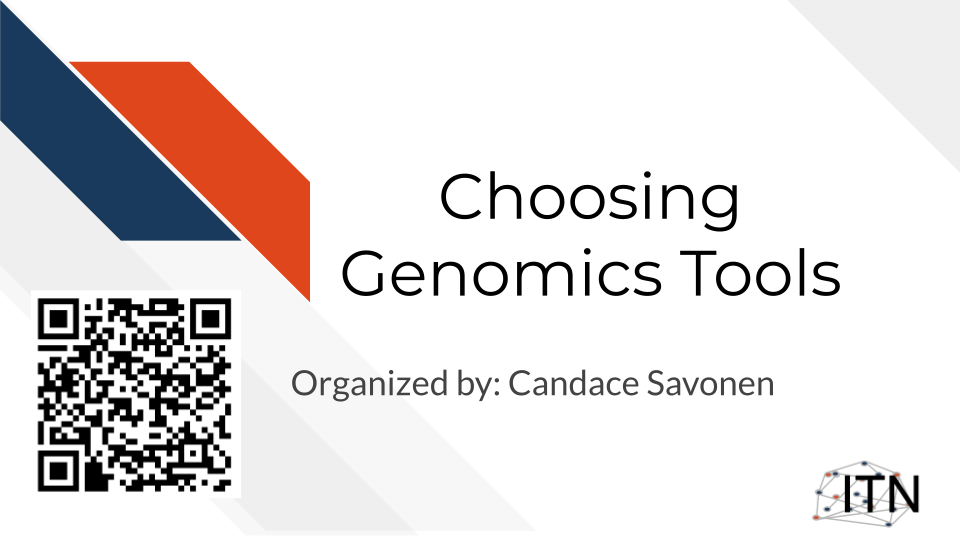
This course is part of a series of courses for the [Informatics Technology for Cancer Research (ITCR)](https://itcr.cancer.gov/) called the Informatics Technology for Cancer Research Education Resource. This material was created by the ITCR Training Network (ITN) which is a collaborative effort of researchers around the United States to support cancer informatics and data science training through resources, technology, and events. This initiative is funded by the following grant: [National Cancer Institute (NCI)](https://www.cancer.gov/) UE5 CA254170. Our courses feature tools developed by ITCR Investigators and make it easier for principal investigators, scientists, and analysts to integrate cancer informatics into their workflows. Please see our website at <www.itcrtraining.org> for more information.

## 0.1 Available course formats

This course is available in multiple formats which allows you to take it in the way that best suites your needs. You can take it for certificate which can be for free or fee.

* The material for this course can be viewed without login requirement on this [Bookdown website](https://hutchdatascience.org/Choosing_Genomics_Tools/). This format might be most appropriate for you if you rely on screen-reader technology.
* This course can be taken for free certification through Leanpub.
* This course can be taken on [Coursera for certification here](https://www.coursera.org/learn/) (but it is not available for free on Coursera).
* Our courses are open source, you can find the [source material for this course on GitHub](https://github.com/jhudsl/Choosing_Genomics_Tools).

# 1 Introduction

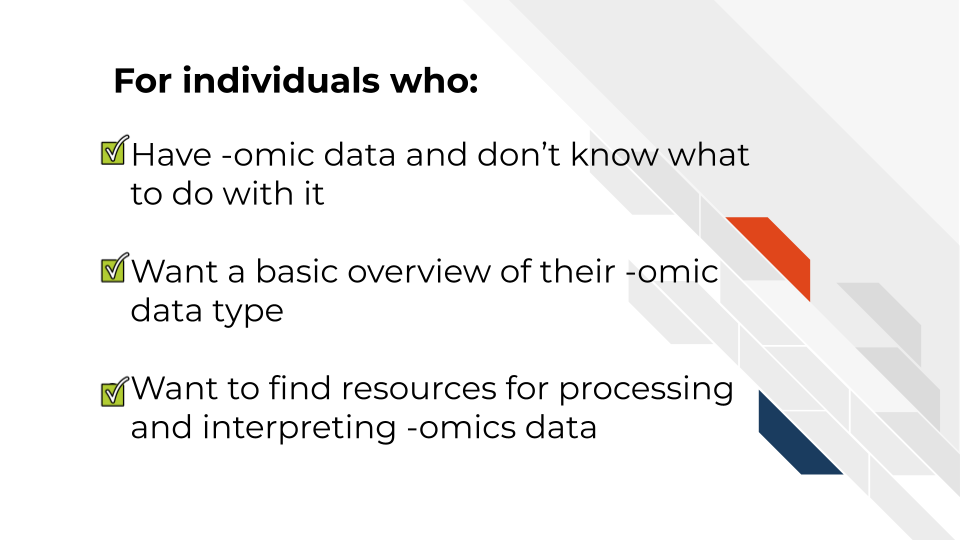


## 1.1 Target Audience

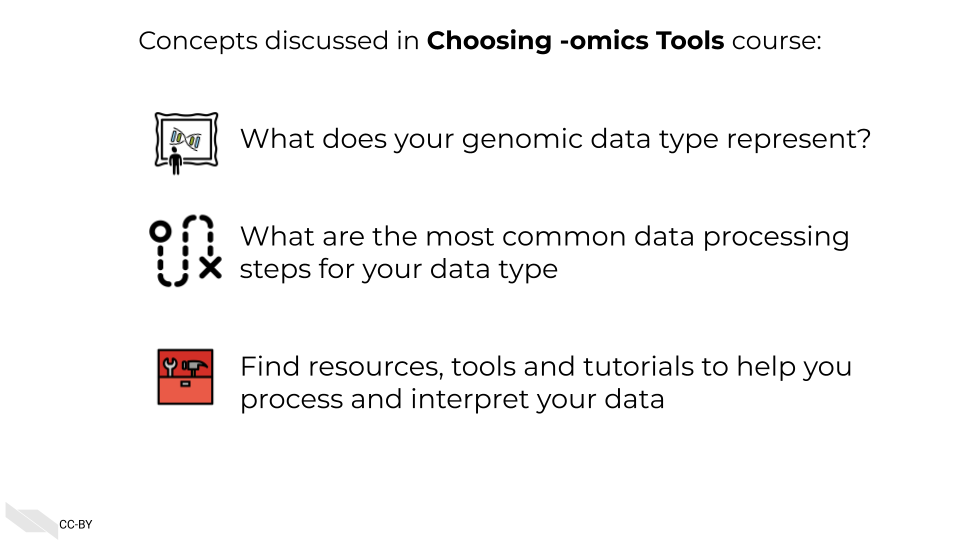
The course is intended for students in the biomedical sciences and researchers who have been given data and don’t know what to do with it or would like an overview of the different genomic data types that are out there.

*This course is written for individuals who:*

* Have genomic data and don’t know what to do with it.
* Want a basic overview of genomic data types.
* Want to find resources for processing and interpreting genomics data.



## 1.2 Topics covered:

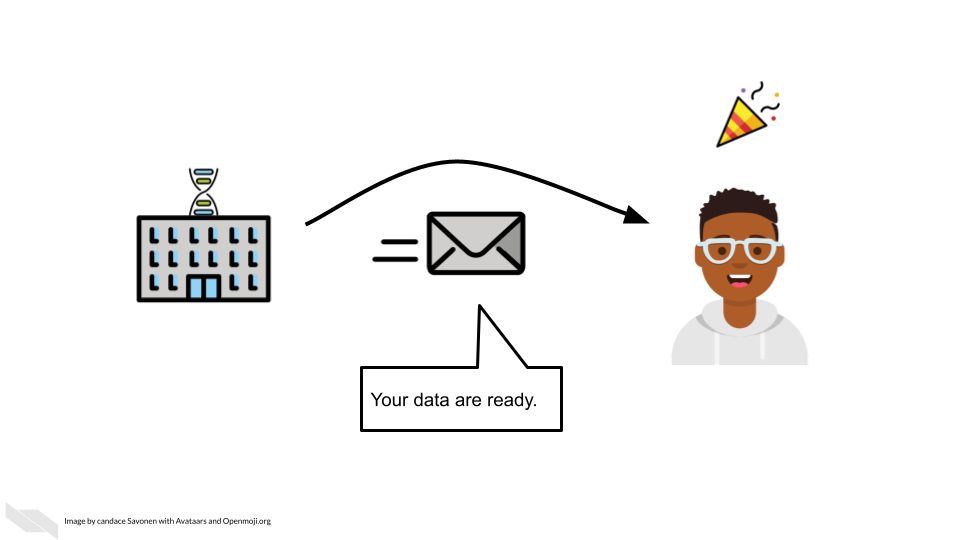


## 1.3 Motivation

Cancer datasets are plentiful, complicated, and hold untold amounts of information regarding cancer biology. Cancer researchers are working to apply their expertise to the analysis of these vast amounts of data but training opportunities to properly equip them in these efforts can be sparse. This includes training in reproducible data analysis methods.

Often students and researchers need to utilize genomic data to reach the next steps of their research but may not have formal training in computational methods or the basics of the genomic data they are attempting to utilize.

Often researchers receive their genomic data processed from another lab or institution, and although they are excited to gain insights from it to inform the next steps of their research, they may not have a practical understanding of how the data they have received came to be or what needs to be done with it.

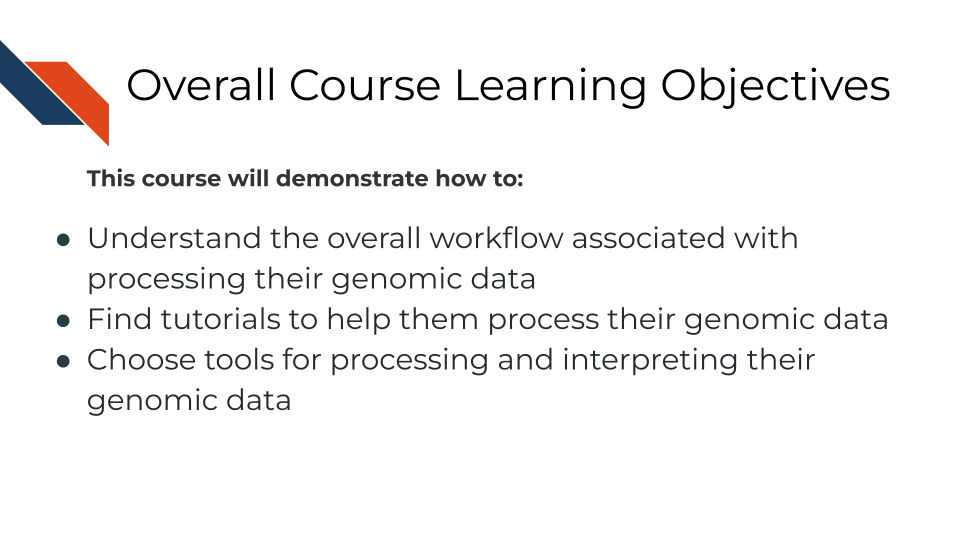


As an example, data file formats may not have been covered in their training, and the data they received seems unintelligible and not as straightforward as they hoped.



This course attempts to give this researcher the basic bearings and resources regarding their data, in hopes that they will be equipped and informed about how to obtain the insights for their researcher they originally aimed to find.

## 1.4 Curriculum



**Goal of this course:**  
Equip learners with tutorials and resources so they can understand and interpret their genomic data in a way that helps them meet their goals and handle the data properly. This includes helping learners formulate questions they will need to ask others about their data

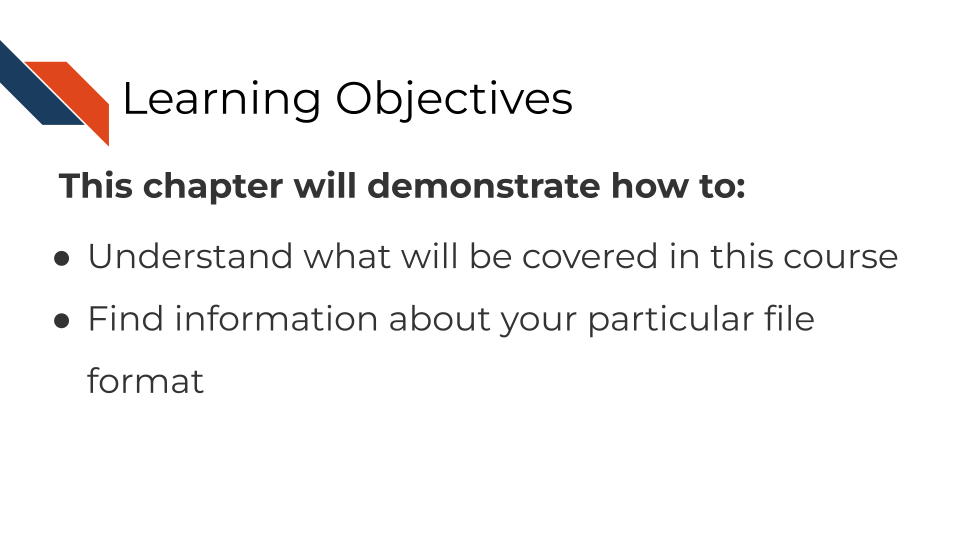
**What is not the goal**  
Teach learners about choosing parameters or about the ins and outs of every genomic tool they might be interested in. This course is meant to connect people to other resources that will help them with the specifics of their genomic data and help learners have more efficient and fruitful discussions about their data with bioinformatic experts.

## 1.5 How to use the course

This course is designed to be a jumping off point to more specific resources based on a genomic data type the learner has in mind (or currently on their computer). We encourage learners to follow links to resources we provide and feel free to jump around to chapters that are most useful for them.

# 2 A Very General Genomics Overview

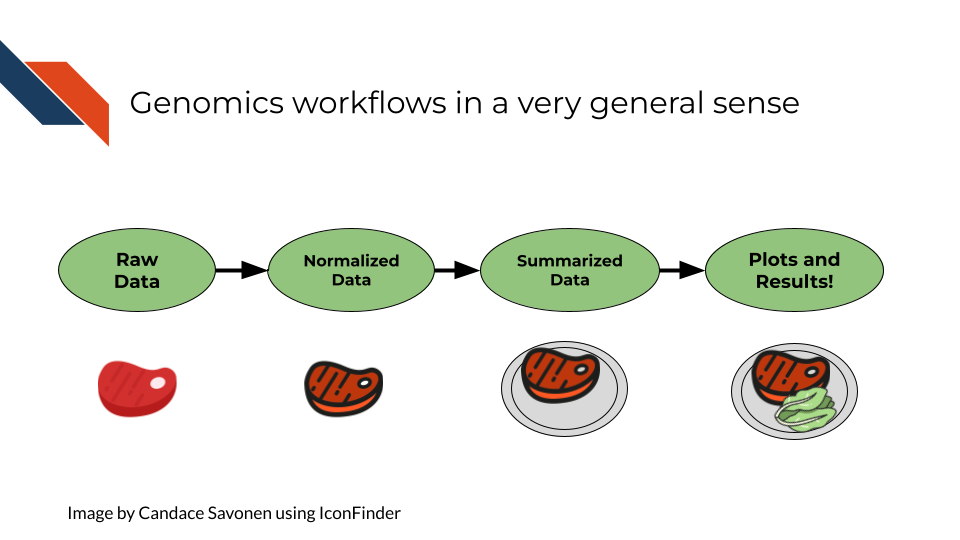
## 2.1 Learning Objectives



In this chapter we are going to cover sequencing and microarray workflows at a very general high level overview to give you a first orientation. As we dive into specific data types and experiments, we will get into more specifics. Here we will cover the most common file formats. If you have a file format you are dealing with that you don’t see listed here, it may be specific to your data type and we will discuss that more in that data type’s respective chapter. We still suggest you go through this chapter to give you a basic understanding of commonalities of all genomic data types and workflows

### 2.1.1 What do genomics workflows look like?

In the most general sense, all genomics data when originally collected is raw, it needs to undergo processing to be normalized and ready to use. Then normalized data is generally summarized in a way that is ready for it to be further consumed. Lastly, this summarized data is what can be used to make inferences and create plots and results tables.



### 2.1.2 Basic file formats

Before we get into bioinformatic file types, we should establish some general file types that you likely have already worked with on your computer. These file types are used in all kinds of applications and not specific to bioinformatics.

#### 2.1.2.1 TXT - Text

A text file is a very basic file format that contains text!

#### 2.1.2.2 TSV - Tab Separated Values

Tab separated values file is a text file is good for storing a data table. It has rows and columns where each value is separated by (you guessed it), *tabs*. Most commonly, if your genomics data has been provided to you in a TSV or CSV file, it has been processed and summarized! It will be your job to know *how* it was processed and summarized

Here the literal ⇥ represents tabs which often may show up invisible in your text editor’s preference settings.

gene\_id⇥sample\_1⇥sample\_2  
gene\_a⇥12⇥15,  
gene\_b⇥13⇥14

#### 2.1.2.3 CSV - Comma Separated Values

A comma separated values file is list just like a TSV file but instead of values being separated by tabs it is separated by… (you guessed it), *commas*!

In its raw form, a CSV file might look like our example below (but if you open it with a program for spreadsheets, like Excel or Googlesheets, it will look like a table)

gene\_id, sample\_1, sample\_2,  
gene\_a, 12, 15,  
gene\_b, 13, 14

### 2.1.3 Sequencing file formats

#### 2.1.3.1 SAM - Sequence Alignment Map

SAM Files are text based files that have sequence information. It generally has not been quantified or mapped. It is the reads in their raw form. [For more about SAM files](https://samtools.github.io/hts-specs/SAMv1.pdf).

#### 2.1.3.2 BAM - Binary Alignment Map

BAM files are like SAM files but are compressed (made to take up less space on your computer). This means if you double click on a BAM file to look at it, it will look jumbled and unintelligible. You will need to convert it to a SAM file if you want to see it yourself (but this isn’t necessary necessarily).

#### 2.1.3.3 FASTA - “fast A”

Fasta files are sequence files that can be either nucleotide or amino acid sequences. They look something like this (the example below illustrating an amino acid sequence):

>SEQ\_ID  
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT

For [more about fasta files](https://en.wikipedia.org/wiki/FASTA_format).

#### 2.1.3.4 FASTQ - “Fast q”

A Fastq file is like a Fasta file except that it also contains information about the **Q**uality of the read. By quality, we mean, how sure was the sequencing machine that the nucleotide or amino acid called was indeed called correctly?

@SEQ\_ID  
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT  
+  
!''\*((((\*\*\*+))%%%++)(%%%%).1\*\*\*-+\*''))\*\*55CCF>>>>>>CCCCCCC65

For [more about fastq files](https://en.wikipedia.org/wiki/FASTQ_format).

Later in this course we will discuss the importance of examining the quality of your sequencing data and how to do that. If you received your data from a bioinformatics core it is possible that they’ve already done this quality analysis for you.

*Sequencing data that is not of high enough quality should not be trusted!* It may need to be re-run entirely or may need extra processing (trimming) in order to make it more trustworthy. We will discuss this more in later chapters.

#### 2.1.3.5 BCL - binary base call (BCL) sequence file format

This type of sequence file is specific to Illumina data. In most cases, you will simply want to convert it to Fastq files for use with non-Illumina programs.

[More about BCL to Fastq conversion](https://medium.com/@marija190396/bcl-to-fastq-conversion-e289852823d0).

#### 2.1.3.6 VCF - Variant Call Format

VCF files are further processed form of data than the sequence files we discussed above. VCF files are specially for storing only where a particular sample’s sequences differ or are *variant* from the reference genome or each other.

This will only be pertinent to you if you care about DNA variants. We will discuss this in the DNA seq chapter.

For [more on VCF files](https://en.wikipedia.org/wiki/Variant_Call_Format).

#### 2.1.3.7 MAF - Mutation Annotation Format

MAF files are aggregated versions of VCF files. So for a group of samples for which each has a VCF file, your entire group of samples’ variants will be summarized in the form of a MAF file.

For [more on MAF files](https://docs.gdc.cancer.gov/Data/File_Formats/MAF_Format/#:~:text=Mutation%20Annotation%20Format%20(MAF)%20is,(or%20open%2Daccess).).

### 2.1.4 Microarray file formats

#### 2.1.4.1 IDAT - intensity data file

This is an Illumina microarray specific file that contains the chip image intensity information for each location on the microarray. It is a binary file, which means it will not be readable by double clicking and attempting to open the file directly.

Currently, Illumina appears to suggest directly converting IDAT files into a GTC format. We advise looking into [this package to help you do that](https://github.com/freeseek/gtc2vcf).

[For more on IDAT files](https://www.illumina.com/content/dam/illumina-marketing/documents/products/technotes/technote_array_analysis_workflows.pdf).

#### 2.1.4.2 DAT - data file

This is an Affymetrix’ microarray specific file parallel to the IDAT file in that it contains the image intensity information for each location on the microarray. It’s stored as pixels.

[For more on DAT files](https://www.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/dat.html).

#### 2.1.4.3 CEL

This is an Affymetrix microarray specific file that is made from a DAT file but translated into numeric values. It is not normalized yet but can be normalized into a CHP file.

[For more on CEL files](https://www.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/cel.html)

#### 2.1.4.4 CHP

CHP files contain the gene-level and normalized data from an Affymetrix array chip. CHP files are obtained by normalizing and processing CEL files.

[For more about CHP files](https://www.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/chp-xda.html).

## 2.2 General informatics files

At various points in your genomics workflows, you may need to use other types of files to help you annotate your data. We’ll also discuss some of these common files that you may encounter:

#### 2.2.0.1 BED - Browser Extensible Data

A BED file is a text file that has coordinates to genomic regions. THe other columns that accompany the genomic coordinates are variable depending on the context. But every BED file contains the chrom, chromStart and chromEnd columns to start.

A BED file might look like this:

chrom chromStart chromEnd other\_optional\_columns  
chr1 0 1000 good  
chr2 100 3000 bad

For [more on BED files](https://en.wikipedia.org/wiki/BED_(file_format)).

#### 2.2.0.2 GFF/GTF General Feature Format/Gene Transfer Format

A GFF file is a tab delimited file that contains information about genomic features. These types of files are available from databases and what you can use to annotate your data.

You may see there are GFF2, GFF3, and GTF files. These only refer to different versions and variations. They generally have the same information. In general, GFF2 is being phased out so using GFF3 is generally a better bet unless the program or package you are using specifies it needs an older GFF2 version.

A GFF file may look like this (borrowed example from Ensembl):

1 transcribed\_unprocessed\_pseudogene gene 11869 14409 . + . gene\_id "ENSG00000223972"; gene\_name "DDX11L1"; gene\_source "havana"; gene\_biotype "transcribed\_unprocessed\_pseudogene";

Note that it will be useful for annotating genes and what we know about them.

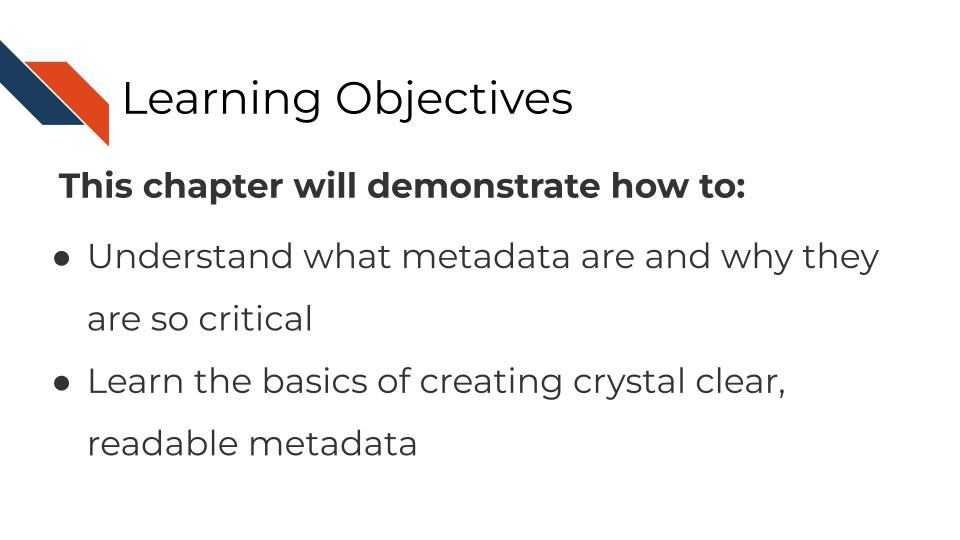
For [more about GTF and GFF files](https://useast.ensembl.org/info/website/upload/gff.html).

### 2.2.1 Other files

\* If you didn’t see a file type listed you are looking for, take a look at this [list by the BROAD](https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats). Or, it may be covered in the data type specific chapters.

# 3 What are Metadata?

## 3.1 Learning Objectives

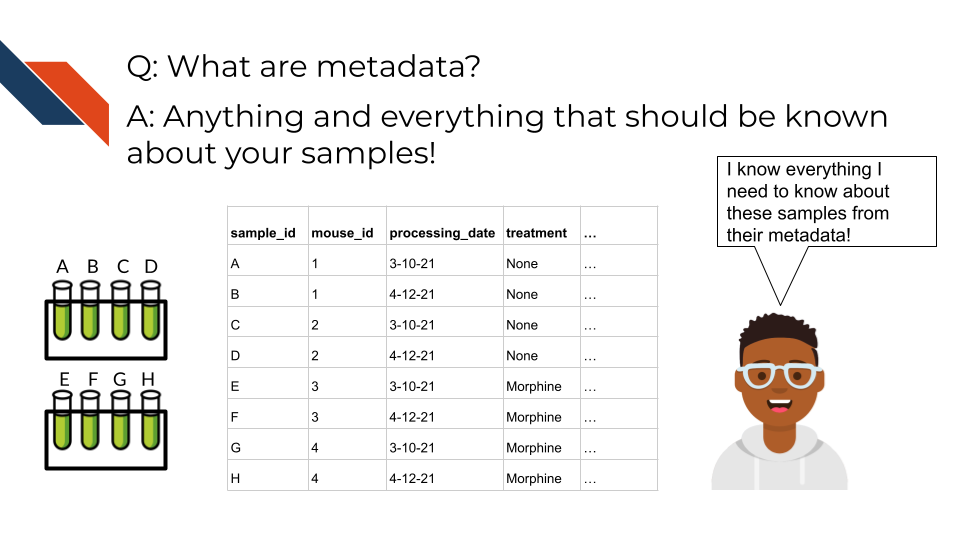


## 3.2 What are metadata?

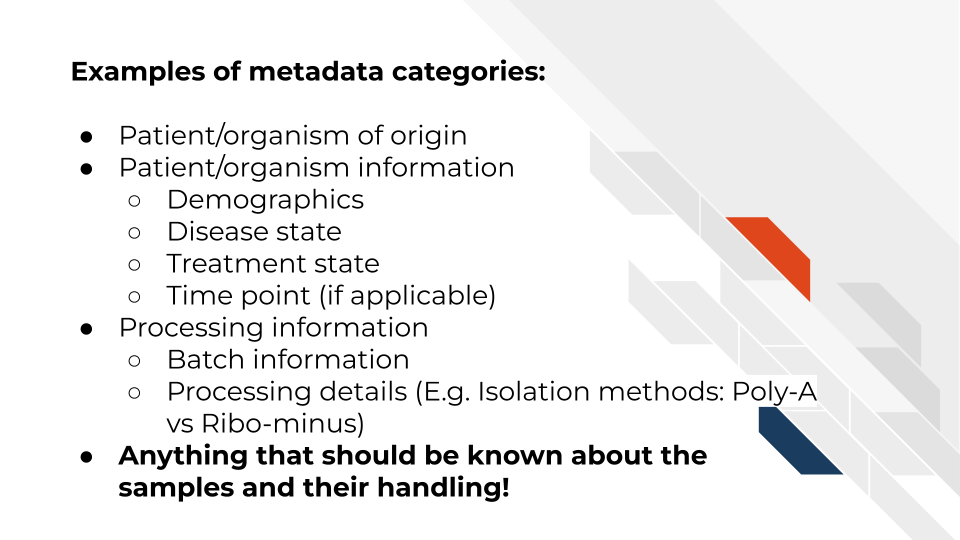
Metadata are critically important descriptive information about your data.

**Without metadata, the data themselves are useless or at best vastly limited.**

Metadata describe how your data came to be, what organism or patient the data are from and include any and every relevant piece of information about the samples in your data set.



Metadata includes but isn’t limited to, the following example categories:



At this time it’s important to note that if you work with human data or samples, your metadata will likely contain personal identifiable information (PII) and protected health information (PHI). It’s critical that you protect this information! For more details on this, we encourage you to see our [course about data management](https://jhudatascience.org/Ethical_Data_Handling_for_Cancer_Research/data-privacy.html).

## 3.3 How to create metadata?

Where do these metadata come from? The notes and experimental design from anyone who played a part in collecting or processing the data and its original samples. If this includes you (meaning you have collected data and need to create metadata) let’s discuss how metadata can be made in the most useful and reproducible manner.

### 3.3.1 The goals in creating your metadata:

#### 3.3.1.1 Goal A: Make it *crystal clear* and *easily readable* by both humans and computers!

Some examples of how to make your data crystal clear: - Look out for typos and spelling errors! - Don’t use acronyms unless you need to and then if you do need to make sure to explain what the acronym means. - Don’t add extraneous information – perhaps items that are relevant to your lab internally but not meaningful to people outside of your lab. Either explain the significance of such information or leave it out.

* Make your [data tidy](https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html#:~:text=Tidy%20data%20is%20a%20standard,Every%20row%20is%20an%20observation.). > Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types. In tidy data: > - Every column is a variable. > - Every row is an observation. > - Every cell is a single value.

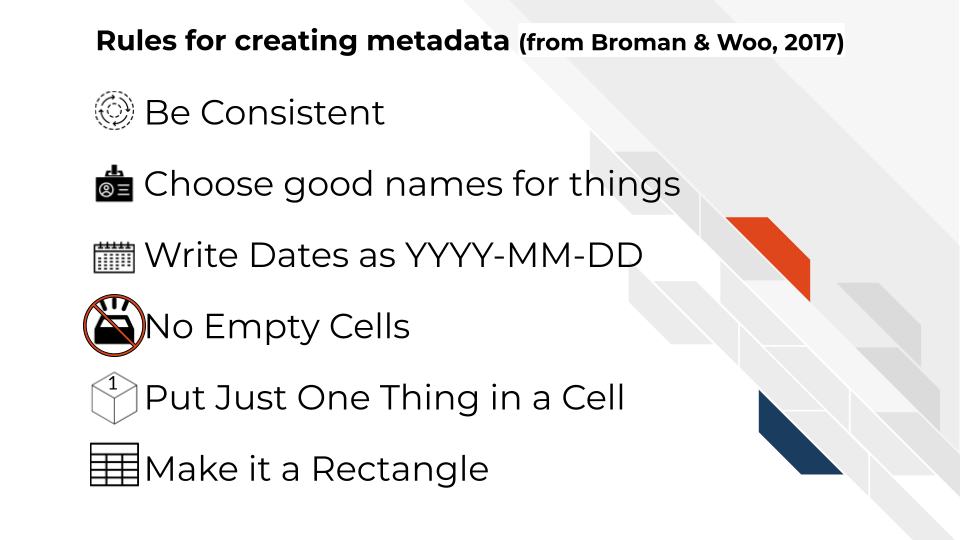
#### 3.3.1.2 Goal B: Avoid introducing errors into your metadata in the future!

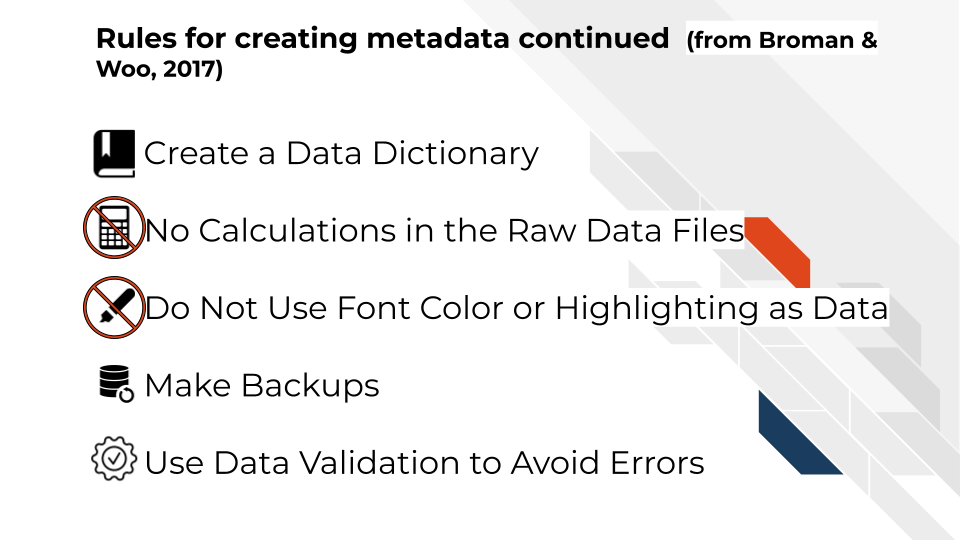
Toward these two goals, [this excellent article](https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1375989) by Broman & Woo discusses metadata design rules. We will very briefly cover the major points here but highly suggest you read the original article.

1. *Be Consistent* - Whatever labels and systems you choose, use it universally. This not only means in your metadata spreadsheet but also anywhere you are discussing your metadata variables.
2. *Choose good names for things* - avoid spaces, special characters, or within the lab jargon.
3. *Write Dates as YYYY-MM-DD* - this is a global standard and less likely to be messed up by Microsoft Excel.
4. *No Empty Cells* - If a particular field is not applicable to a sample, you can put NA but empty cells can lead to formatting errors or just general confusion.
5. *Put Just One Thing in a Cell* - resist the urge to combine variables into one, you have no limit on the number of metadata variables you can make!
6. *Make it a Rectangle* - This is the easiest way to read data, for a computer and a human. Have your samples be the rows and variables be columns.
7. *Create a Data Dictionary* - Have somewhere that you describe what your metadata mean in detailed paragraphs.
8. *No Calculations in the Raw Data Files* - To avoid mishaps, you should always keep a clean, original, raw version of your metadata that you do not add extra calculations or notes to.
9. *Do Not Use Font Color or Highlighting as Data* - This only adds to confusion to others if they don’t understand your color coding scheme. Instead create a new variable for anything you might be tempted to color code.
10. *Make Backups* - Metadata are critical, you never want to lose them because of spilled coffee on a computer. Keep the original backed up in a multiple places. We recommend keeping writing your metadata in something like GoogleSheets because it is both free and also saved online so that it is safe from computer crashes.
11. *Use Data Validation to Avoid Errors* - set data types to have googlesheets or excel check that the data in the columns is the type of data it expects for a given variable.

Note that it is very dangerous to open gene data with Excel. According to Ziemann, Eren, and El-Osta ([2016](#ref-Ziemann2016)), approximately one-fifth of papers with Excel gene lists have errors. This happens because Excel wants to interpret everything as a date. We strongly caution against opening (and saving afterward) gene data in Excel. 

### 3.3.2 To recap:

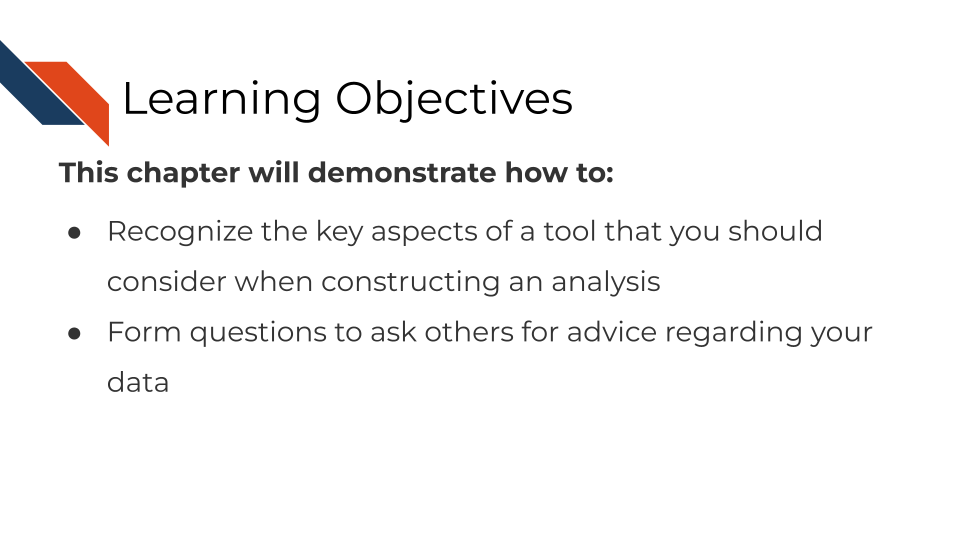




If you are not the person who has the information needed to create metadata, or you believe that another individual already has this information, make sure you get ahold of the metadata that correspond to your data. It will be critical for you to have to do any sort of meaningful analysis!

# 4 Considerations for choosing tools

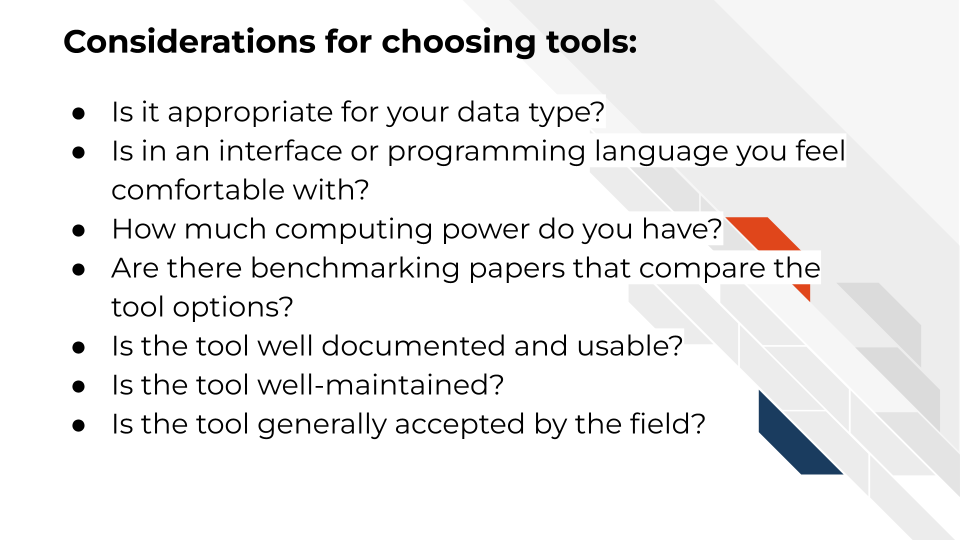
## 4.1 Learning Objectives



## 4.2 Overview

In this course, we will introduce you to the fundamentals of various data types and give you advice about choosing tutorials and tools whenever possible. However, it is critical to note that there is no “one size fits all” when it comes to genomic data decisions. Instead, our goals are to equip you with the knowledge you need as well as the questions you need to ask yourself (or others) when making decisions about your genomics data.

We will discuss the following considerations you should gather information and otherwise ponder when comparing one or more tools for your analysis:



### 4.2.1 Is this tool appropriate for your data type?

Certain tools are built for certain kinds of data. In each data-type-specific chapter we will attempt to point you tools that are appropriate for the given data type. However, note that some tools also might require tweaks in parameters for non-standard data collection methods. If you were not sure of the data collection methods used for your data type, be sure to follow the data type specific advice in the chapter to find out the information about your data that you need to know to make an informed decision.

### 4.2.2 Is this tool appropriate for your scientific question?

Some tools may be appropriate for the general data type, but might mask information you will need to answer your particular scientific question or hypothesis. For example, for RNA-seq if you are interested in splice variants, you may not be able to use certain alignment tools that do not differentiate between splice variants.

Be sure to make your goals and scientific questions clear when asking for advice or guidance. Some tools may be applicable to certain scientific questions, but other accommodations or preprocessing may need to be done

### 4.2.3 Is this tool in an interface or programming language you feel comfortable with?

Genomics and informatics tools can be classified into two groups based on how you interact with them. These groups are 1) command line or 2) graphics user interface (GUI). GUIs are tools that you can use by clicking and pointing with your mouse whereas command line tools require input through writing out commands.

Command line tools often lend to greater reproducibility of an analysis since a script can have all the steps needed to re-run analysis. This makes it so you could re-run and reproduce your results with one command instead of lots of clicking various buttons in particular order as you would need to do with a GUI based tool.

Your level of comfort or willingness/time available to learn a programming language like R or Python will influence what tool options you have. If you are unfamiliar and uncomfortable writing in R, Python, or Bash scripting, this will influence what tools you have available to you or whether you will need to enlist more outside help.

If you are interested in learning to use command line, we have many resources and recommendations for you to use for learning in this next chapter. However, if you do not have the bandwidth or motivation to learn how to code, you will want to gravitate toward tools that have GUIs.

### 4.2.4 How much computing power do you have?

Some tools require a lot more computing resources (or runtime) than others. Many institutions have cloud computing resources or high powered computing clusters for your use. [We’ll recommend you to our Computing Course for more information about this](https://jhudatascience.org/Computing_for_Cancer_Informatics/computing-resources.html).

But your computing budget access, and time allotment, may influence what tools you would like to use for a project. For example, for RNA seq data alignment, traditional aligners that use the genome take an order of magnitude greater amount of time to run than quantifying transcripts with pseudo alignment based tools. For many applications pseudoaligners are perfectly appropriate and efficient choices that can be run on a laptop. But if you prefer a traditional aligner because you are interested in something that is not detected by pseudosligners such as splice variants, then you may want to look into using some computing resources for this task. All these decisions need to be weighed in balance with each other.

### 4.2.5 Are there benchmarking papers that compare this tool to other options?

Some tools and their algorithms have been more thoroughly examined and tested than others. And this doesn’t always align to a tool’s popularity. Seek out the literature and what studies have been done comparing this tool to others like it. Keep in mind the tool developer’s own bias if the paper is coming directly from the group or individual who is the creator of the tool. Developers will be more likely to understand and know how to tweak parameters of their own tool properly, while not necessarily spending as much time testing and adjusting tools made by others. This concept has sometimes been called the [“Continental Breakfast Included”](https://www.biorxiv.org/content/10.1101/385534v4) concept.

### 4.2.6 Is the tool well documented and usable?

Well documented and usable tools can be very powerful. Poorly documented tools which may lead to unknown parameters or other mishandling of the data if it has not been made clear by the tool developers and maintainers. Good understanding of what a tool is doing with the data you give it is perhaps more important than using fancy algorithms that are unclear. Not only does documentation and usability increase your ability to use a tool, but your analysis will be more reproducible if others can also understand the tools that you used.

The existence of forums and user groups for particular tools, not only makes it a useful resource for you for analysis, troubleshooting and interpretation of your results, but it also indicates a particular drive for the tool to continue to be maintained and developed overtime.

### 4.2.7 Is the tool well maintained?

If a tool is actively being maintained this will aid in the reproducibility of your results. Tools on GitHub (an open-source platform for software) or other repositories often indicate when latest updates to a tool were made. Ideally updates are being made regularly to the tool, but a lack of updates does not speak well for the future existence of the tool. A tool that is not well maintained or supported may deprecate and make it increasingly difficult if not possible to reproduce, re-run or further develop your analysis.

### 4.2.8 Is the tool generally accepted by the field?

While tool popularity should not be the only consideration when choosing a tool, it is an aspect that can influence communication or acceptance of your results. All things being equal, it can be better to choose a tool that is more accepted by the community as tried and true, and well benchmarked as opposed to the bleeding edge technology that may have not been truly scrutinized yet. In an analysis it is perhaps more valuable to know and weigh the known limitations of an older tool than to use a newer tool whose limitations may not have been identified yet (but it certainly will have its own limitations identified in time).

## 4.3 Coming to a decision

It’s important to note that the questions we will discuss here need to be considered in balance of one another. Rarely should you make a decision about a tool without considering all of these items congruently. For example, some tools may have better benchmarking but if it is more computationally costly and you do not have access to the necessary computing resources to run the tool, then you may need to consider other options.

## 4.4 More resources

* [A longer list of tools and resources can be found here](https://hutchdatascience.org/code_review/more_resources.html)
* [DataTrail curriculum](https://datatrail-jhu.github.io/DataTrail/index.html)
* [Introduction to Reproducibility](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/introduction.html)
* [Advanced Reproducibility in Cancer Informatics](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/introduction.html)
* [Computing in Cancer Informatics](https://jhudatascience.org/Computing_for_Cancer_Informatics/)

# 5 General Data Analysis Tools

## 5.1 Learning Objectives



## 5.2 Command Line vs GUI

When using computers there are two different ways you can tell a computer program what you want it to do. You can use a a Graphics User Interface (abbreviated as GUI) where you point and click buttons or you can use a Command Line Interface where you type in commands and write scripts that tell the program what you want it to do.

Command Line Interfaces require a bit more time to learn and get used to, but they are generally easier to make more reproducible, because every step that you are using an analysis can be written in a script. Graphics User Interfaces can be more intuitive to use more quickly, but they can be difficult to repeat the analysis in the exact same way. If you know you will be doing the same analysis many times (either with different or the same samples), it is a good use of your time to make sure that you learn how to use Command Line tools. We will discuss some of the most commonly used Command line tools here.

### 5.2.1 Bash

Bash is a command language used by a lot of computers and programs. Many of the same items that you might do every day on your computer by clicking on various items on your desktop and menus, you can also perform using bash.

On a Mac computer, you can use bash commands by finding your Terminal window. Go to your search bar and search for the Terminal. You may want to keep this application handy.

In Windows, you can use bash commands by search for Command Prompt application. Go to your search bar and search for Command Prompt. You may want to keep this application handy.

### 5.2.2 R

R is a program commonly used for statistics and data analysis. It’s free and has lots of R packages built for genomics analysis purposes. Many of these packages have been highlighted in this course or otherwise listed in our [tool glossary](http://hutchdatascience.org/Choosing_Genomics_Tools/genomic-tool-glossary.html).

#### 5.2.2.1 Resources for learning R

##### 5.2.2.1.1 R and Tidyverse

* [Swirl, an interactive tutorial](https://swirlstats.com/)
* [R for Data Science](https://r4ds.had.co.nz/)
* [Tidyverse skills for Data Science](http://jhudatascience.org/tidyversecourse/) by Carrie Wright.
* [Handy R cheatsheets](https://www.rstudio.com/resources/cheatsheets/)
* [R Cookbook Second Edition](https://rc2e.com/)
* [Advanced R](https://adv-r.hadley.nz/)
* [R for Epidemiology](https://www.r4epi.com/) - has generally good R advice
* [O’Reilly books](https://www.spl.org/books-and-media/books-and-ebooks/safari-books-online) available through Seattle Public Library

##### 5.2.2.1.2 R notebooks

* [R Markdown](http://rmarkdown.rstudio.com)
* [Tutorial on R, RStudio and R Markdown](https://ismayc.github.io/rbasics-book/)
* [Handy R cheatsheets](https://www.rstudio.com/resources/cheatsheets/)
* [R Notebooks tutorial](https://bookdown.org/yihui/rmarkdown/)

##### 5.2.2.1.3 R and Genomics

* [Intro to R and Tidyverse course and exercises](https://github.com/AlexsLemonade/training-modules/tree/master/intro-to-R-tidyverse) from the Childhood Cancer Data Lab.
* [Refine.bio examples](https://alexslemonade.github.io/refinebio-examples/index.html) from the Childhood Cancer Data Lab.
* [Biostar Handbook: A Beginner’s Guide to Bioinformatics](https://www.biostarhandbook.com)

### 5.2.3 Python

Python is a program that also is used for data analysis among many other items. It can be a very powerful development tool. Some of the packages that have been highlighted in this course or otherwise are listed in our [tool glossary](http://hutchdatascience.org/Choosing_Genomics_Tools/genomic-tool-glossary.html).

#### 5.2.3.1 Resources for learning python

* [Python Data Science Handbook](https://jakevdp.github.io/PythonDataScienceHandbook/)
* [Python for Biologists](https://www.pythonforbiologists.org/)

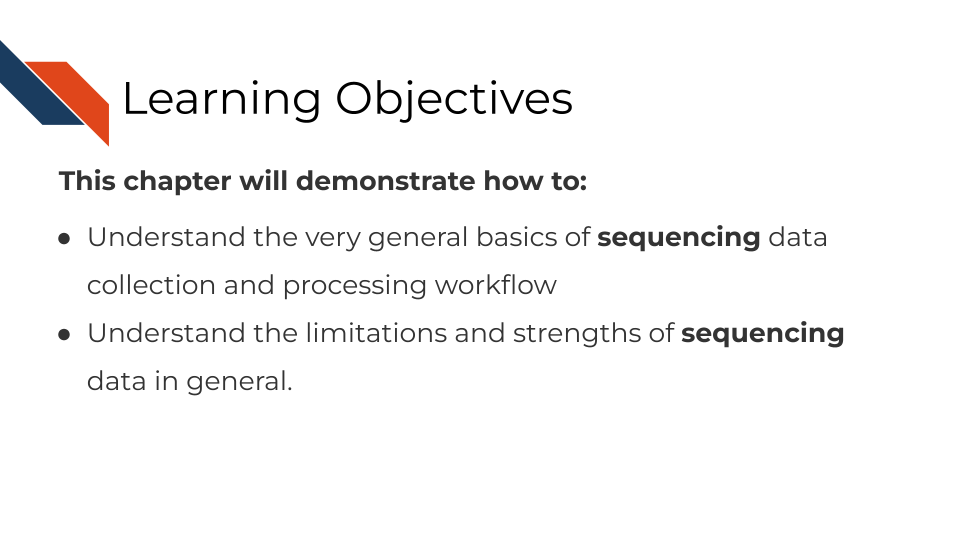
## 5.3 More resources

* [A longer list of tools and resources can be found here](https://hutchdatascience.org/code_review/more_resources.html)
* [DataTrail curriculum](https://datatrail-jhu.github.io/DataTrail/index.html)
* [Introduction to Reproducibility](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/introduction.html)
* [Advanced Reproducibility in Cancer Informatics](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/introduction.html)
* [Computing in Cancer Informatics](https://jhudatascience.org/Computing_for_Cancer_Informatics/)

# 6 Sequencing Data

This chapter is in a beta stage. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 6.1 Learning Objectives



In this section, we are going to discuss generalities that apply to all sequencing data. This is meant to be a “primer” for you which data-type specific chapters will build off of to give you more specific and practical steps and advice in regards to your data type.

## 6.2 How does sequencing work?



Sequencing methods, whether they are targeting DNA, transcriptomes, or some other target of the genome, have some commonalities in the steps as well as what types of biases and data generation artifacts to look out for.

All sequencing experiments start out with the extraction of the biological material of interest. This biological material will be processed in some way to isolate to the genomic target of interest (we will cover the various techniques for this in more detail in each respective data chapter since it is highly specific to the data type).

This set of processing steps will lead up to library generation – adding a way to catalog what molecules came from where. Sometimes for this library prep the sequences need to be fragmented before hand and an adapter bound to them.

The resulting sample material is often a very small quantity, which means Polymerase Chain Reaction (PCR) needs to be used to amplify the material to a quantity large enough to be reliably sequenced. We will talk about how this very common method not only amplifies the sequences we want to read but amplifies sequence method biases that we would like to avoid.

At the end of this process, base sequences are called for the samples (with varying degrees of confidence), creating huge amounts of data and what hopefully contains valuable research insights.

## 6.3 Sequencing concepts

### 6.3.1 Inherent biases



Sequences are not all sequenced or amplified at the same rate. In a perfect world, we could take a simple snapshot of the genome we are interested in and know exactly what and how many sequences were in a sample. But in reality, sequencing methods and the resulting data always have some biases we have to be aware of and hopefully use methods that attempt to mitigate the biases.

#### 6.3.1.1 GC bias

You may recall that with nucleotides: adenine binds with thymine and guanine binds with cytosine. But, the guanine-cytosine bond (GC) has 3 hydrogen bonds whereas the adenine-thymine bond (AT) has only 2 bonds. This means that the GC bond is stickier (to put it scientifically) and needs higher temperatures to unbind. The sequencing and PCR amplification process involves cycling through temperatures and binding and unbinding of sequences which means that if a sequence has a lot of G’s and C’s (high GC content) it will unbind at a different temperatures than a sequence of low GC content.

#### 6.3.1.2 Sequence complexity

Nonrepeating sequences are harder to sequence and amplify than repeating sequences. This means that the complexity of a target sequence influences the PCR amplification and detection.

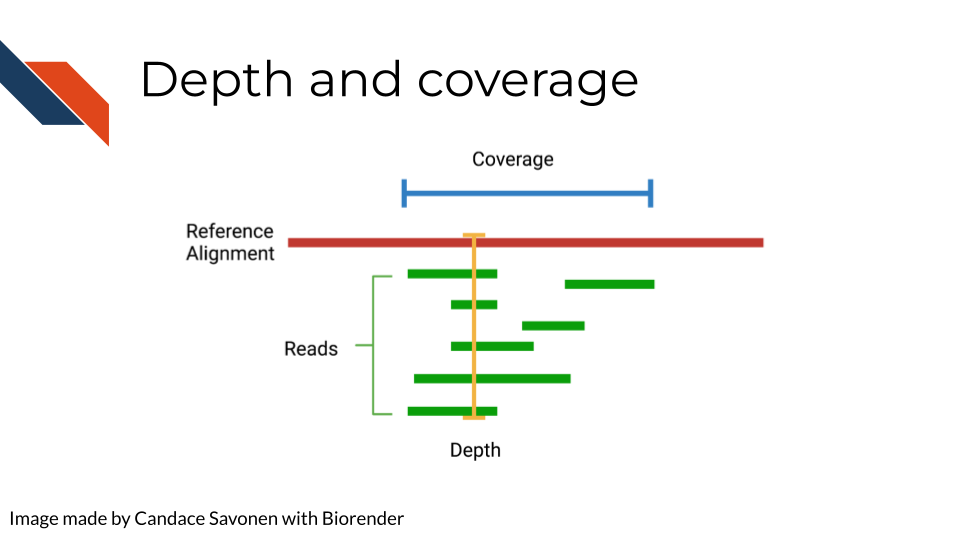
#### 6.3.1.3 Length bias

Longer sequences – whether they represent long sequence variants, long transcripts, or etc, are more likely to be identified than shorter ones! So if you are attempting to quantify the presence of a sequence, a longer sequence is much more likely to be counted more often.

### 6.3.2 PCR Amplification

All of the above biases are amplified when the sequences are being amplified! You can picture that if each of these biases have a certain effect for one copy, then as PCR steps copy the sequence exponentially, the error is also being multiplied! PCR amplification is generally a necessary part of the process. But there are tools that allow you to try to combat the biases of PCR amplification in your data analysis. These tools will be dependent on the type of sequencing methods you are using and will be something that is discussed in each data type chapter.

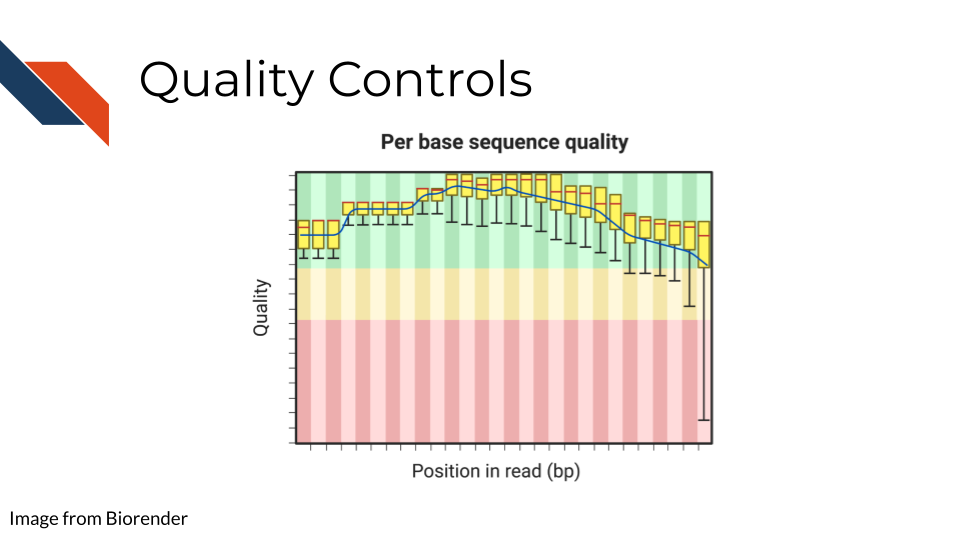
### 6.3.3 Depth of coverage



The depth of sequencing refers to how many times on average a particular base is sequenced. Obviously the more times something is sequenced, the more you can be confident that the base call is accurate. However, sequencing at greater depths also takes more time and money. Depending on your sequencing goals and methods there is an appropriate level of depth that is needed.

Coverage on the other hand has to do with how much of the target is covered. If you are doing Whole Genome Sequencing, what percentage of the whole genome were you able to sequence? You may realize how depth is related to coverage, in that the greater depth of sequencing you use the more likely you are to also cover more of the genome. As discussed in relation to the biases, some part of the genome are harder to reach than others, so by reading at greater depths some of those “hard to read” parts of the genome will be able to be covered.

### 6.3.4 Quality controls

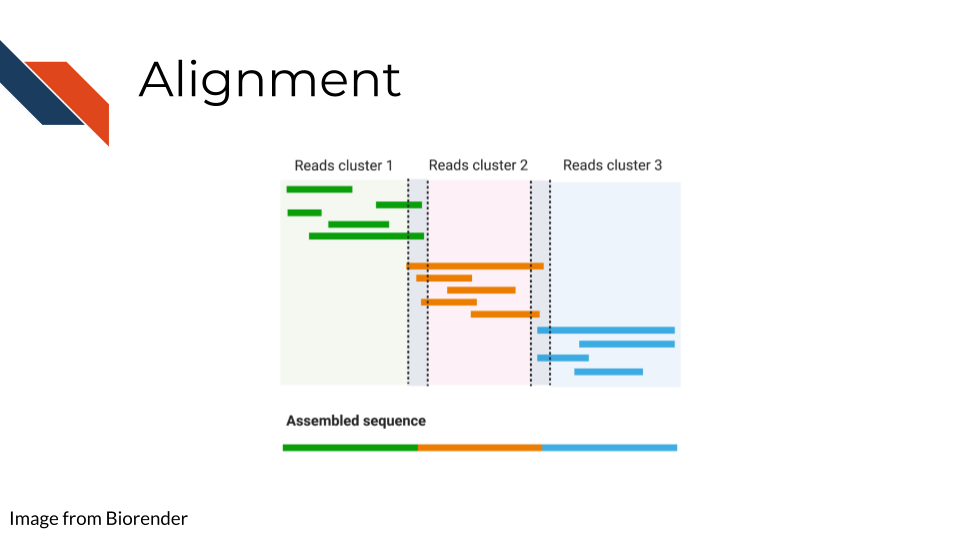


Sequencing bases involves some error/confidence rate. As mentioned, some parts of the genome are harder to read than others. Or, sometimes your sequencing can be influenced by poor quality sample that has degraded. Before you jump in to further analyzing your data, you will want to investigate the quality of the sequencing data you’ve collected.

The most common and well-known method for assessing sequencing quality controls is [FASTQC](https://www.bioinformatics.babraham.ac.uk/projects/fastqc/). FASTQC creates an abundance of sequencing quality control reports from fastq files. These reports need to be interpreted within the context of your sequencing methods, samples, and experimental goals. Often bioinformatics cores are good to contact about these reports (they may have already run FASTQC on your data if that is where you obtained your data initially). They can help you wade through the flood of quality control reports printed out by FASTQC.

FASTQC also has great documentation that can attempt to guide you through report interpretation. This also includes examples of good and bad FASTQC reports. But note that all FASTQC report interpretations must be done relative to the experiment that you have done. In other words, there is not a one size fits all quality control cutoffs for your FASTQC reports. The failure/success icons FASTQC reports back are based on defaults that may not be accurate or applicable to your data, so further investigation and consultation is warranted before you decided to trust or pitch your sequencing data.

### 6.3.5 Alignment

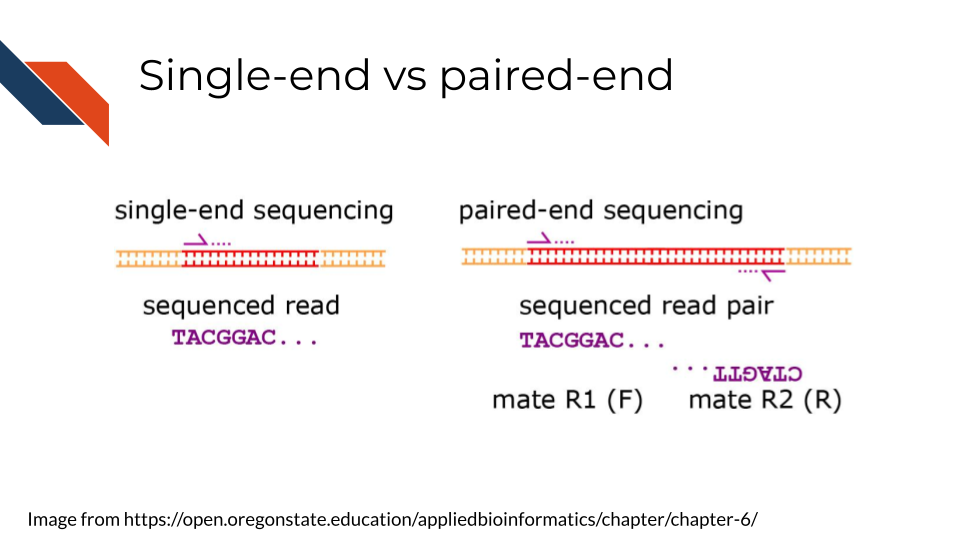


Once you have your reads and you find them reasonably trustworthy through quality control checks, you will want to align them to your reference. The reference you align your sequences to will depend on the data type you have: a reference genome, a reference transcriptome, something else?

* Traditional aligners - Align your data to a reference using standard alignment algorithms. Can be very computationally intensive.
* Pseudo aligners - much faster and the trade off for accuracy is often negligible (but again is dependent on the data you are using).

TODO: considerations for alignment.

### 6.3.6 Single End vs Paired End



Sequencing can be done single-end or paired-end. Paired end means the primers are going to bind to both sides of a sequence. This can help you avoid some 3’ bias and give you more complete coverage of the area you are sequencing. But, as you may guess, pair-end read sequencing is more expensive than single end.

You will want to determine whether your sequencing is paired end or single end. If it is paired end you will likely see file names that indicate this. You should have pairs of files that may or may not be labeled with \_1 and \_2 or \_F and \_R. We will discuss file nomenclature more specifically as it pertains to different data types in the upcoming chapters.

## 6.4 Very General Sequencing Workflow

In the data type specific chapters, we will cover the sequencing data workflows and file formats in more detail. But in the most general sense, sequencing workflows look like this:

### 6.4.1 Sequencing file formats

#### 6.4.1.1 SAM - Sequence Alignment Map

SAM Files are text based files that have sequence information. It generally has not been quantified or mapped. It is the reads in their raw form. [For more about SAM files](https://samtools.github.io/hts-specs/SAMv1.pdf).

#### 6.4.1.2 BAM - Binary Alignment Map

BAM files are like SAM files but are compressed (made to take up less space on your computer). This means if you double click on a BAM file to look at it, it will look jumbled and unintelligible. You will need to convert it to a SAM file if you want to see it yourself (but this isn’t necessary necessarily).

#### 6.4.1.3 FASTA - “fast A”

Fasta files are sequence files that can be either nucleotide or amino acid sequences. They look something like this (the example below illustrating an amino acid sequence):

>SEQ\_ID  
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT

For [more about fasta files](https://en.wikipedia.org/wiki/FASTA_format).

#### 6.4.1.4 FASTQ - “Fast q”

A Fastq file is like a Fasta file except that it also contains information about the **Q**uality of the read. By quality, we mean, how sure was the sequencing machine that the nucleotide or amino acid called was indeed called correctly?

@SEQ\_ID  
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT  
+  
!''\*((((\*\*\*+))%%%++)(%%%%).1\*\*\*-+\*''))\*\*55CCF>>>>>>CCCCCCC65

For [more about fastq files](https://en.wikipedia.org/wiki/FASTQ_format).

Later in this course we will discuss the importance of examining the quality of your sequencing data and how to do that. If you received your data from a bioinformatics core it is possible that they’ve already done this quality analysis for you.

*Sequencing data that is not of high enough quality should not be trusted!* It may need to be re-run entirely or may need extra processing (trimming) in order to make it more trustworthy. We will discuss this more in later chapters.

#### 6.4.1.5 BCL - binary base call (BCL) sequence file format

This type of sequence file is specific to Illumina data. In most cases, you will simply want to convert it to Fastq files for use with non-Illumina programs.

[More about BCL to Fastq conversion](https://medium.com/@marija190396/bcl-to-fastq-conversion-e289852823d0).

#### 6.4.1.6 VCF - Variant Call Format

VCF files are further processed form of data than the sequence files we discussed above. VCF files are specially for storing only where a particular sample’s sequences differ or are *variant* from the reference genome or each other.

This will only be pertinent to you if you care about DNA variants. We will discuss this in the DNA seq chapter.

For [more on VCF files](https://en.wikipedia.org/wiki/Variant_Call_Format).

#### 6.4.1.7 MAF - Mutation Annotation Format

MAF files are aggregated versions of VCF files. So for a group of samples for which each has a VCF file, your entire group of samples’ variants will be summarized in the form of a MAF file.

For [more on MAF files](https://docs.gdc.cancer.gov/Data/File_Formats/MAF_Format/#:~:text=Mutation%20Annotation%20Format%20(MAF)%20is,(or%20open%2Daccess).).

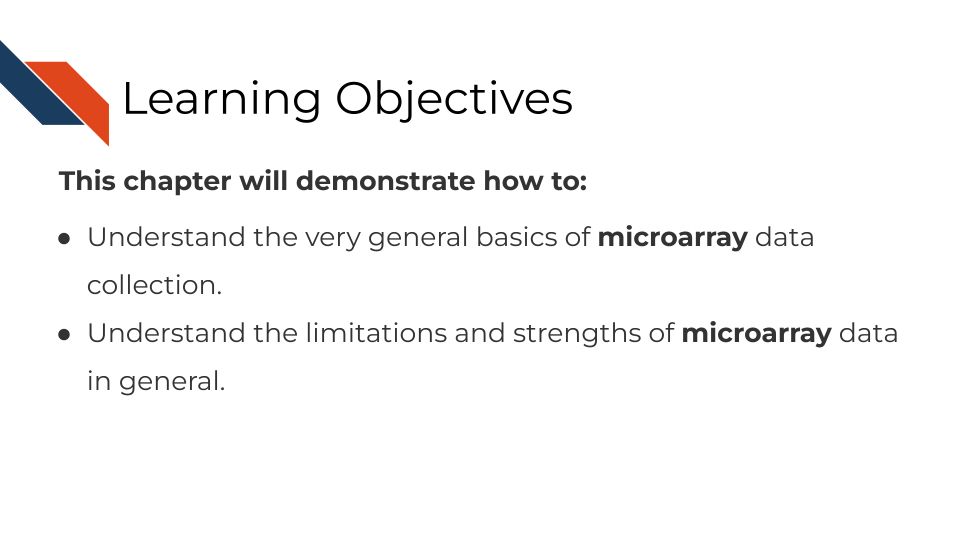
### 6.4.2 Other files

\* If you didn’t see a file type listed you are looking for, take a look at this [list by the BROAD](https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats). Or, it may be covered in the data type specific chapters.

# 7 Microarray Data

This chapter is in a beta stage. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 7.1 Learning Objectives

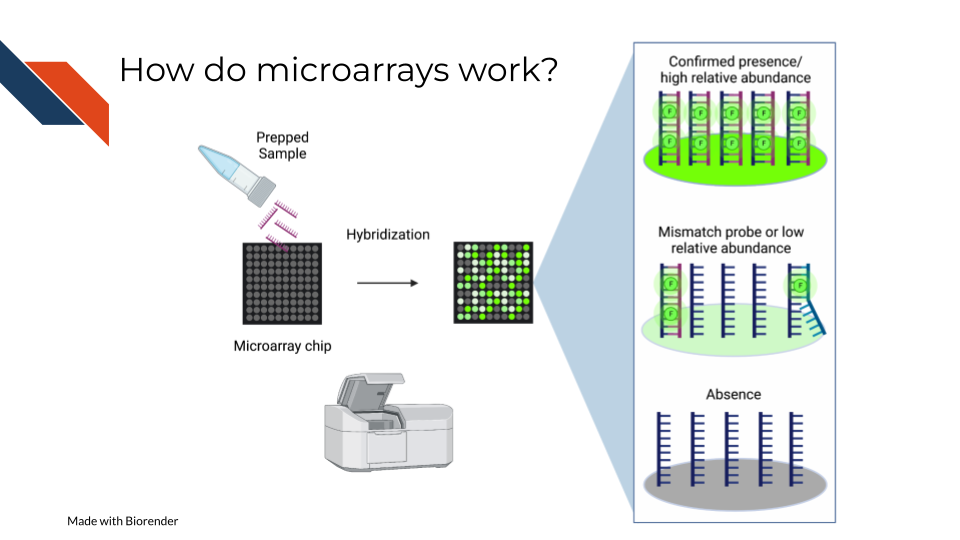


## 7.2 Summary of microarrays

Microarrays have been in use since before high throughput sequencing methods became more affordable and widespread, but they still can be a effective and affordable tool for genomic assays. Depending on your goals, microarray may be a suitable choice for your genomic study.

## 7.3 How do microarrays work?

All microarrays work on hybridization to sets of oligonucleotides on a chip. However, the preparation of the samples, and the oligonucleotides’ hybridization targets vary depending on the assay and goals.



On a basic principle, oligonucleotide probes are designed for different targets sets designed for the same targets are put together. On the whole chip, these probes are arranged in a grid like design so that after a sample is hybridized to them, you can detect how much of the target is detected by taking an image and knowing what target each location is designed to.

### 7.3.1 Pros:

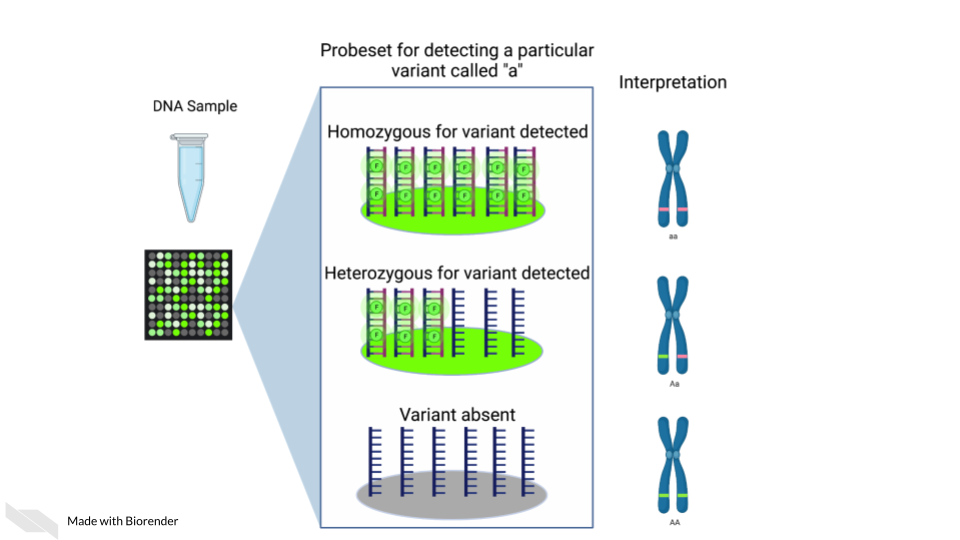
* Microarrays are much more affordable than high throughput sequencing which can allow you to run more samples and have more statistical power ([Tarca, Romero, and Draghici 2006](#ref-Tarca2006); [ALSF 2019](#ref-refinebioexamples2019)).
* Microarrays take less time to process than most high throughput sequencing methods([Tarca, Romero, and Draghici 2006](#ref-Tarca2006); [ALSF 2019](#ref-refinebioexamples2019)).
* Microarrays are generally less computationally intensive to process and you can get your results more quickly([Tarca, Romero, and Draghici 2006](#ref-Tarca2006); [ALSF 2019](#ref-refinebioexamples2019)).
* Microarrays are generally as good as sequencing methods for detecting clinical endpoints ([W. Zhang et al. 2015](#ref-Zhang2015)).

### 7.3.2 Cons:

* Microarray chips can only measure the targets they are designed for, and cannot be used for exploratory purposes ([W. Zhang et al. 2015](#ref-Zhang2015)).
* Microarrays’ probe designs can only be as up to date as the genome they were designed against at the time ([Mantione et al. 2014](#ref-Mantione2014); [**refinebioexamples?**](#ref-refinebioexamples)).
* Microarray does not escape oligonucleotide biases like GC content and sequence composition biases([ALSF 2019](#ref-refinebioexamples2019)).

## 7.4 What types of arrays are there?

### 7.4.1 SNP arrays

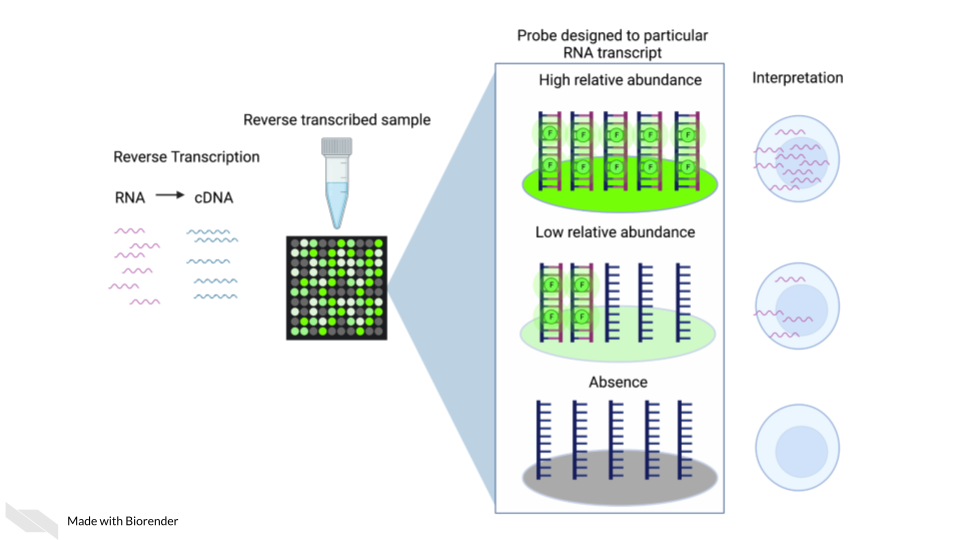


Single nucleotide polymorphism arrays are designed to measure DNA variants. They are designed to target DNA variants. When the sample is hybridized, the amount of fluorescence detected can be interpreted to indicate the presence of the variant and whether the variant is homogeneous or heterogenous. The samples prepped for SNP arrays then need to be DNA samples.

#### 7.4.1.1 Examples:

* [The 1000 genomes project](https://www.internationalgenome.org/) is a large collection of SNP data array from many populations around the world and is available for download.

### 7.4.2 Gene expression arrays

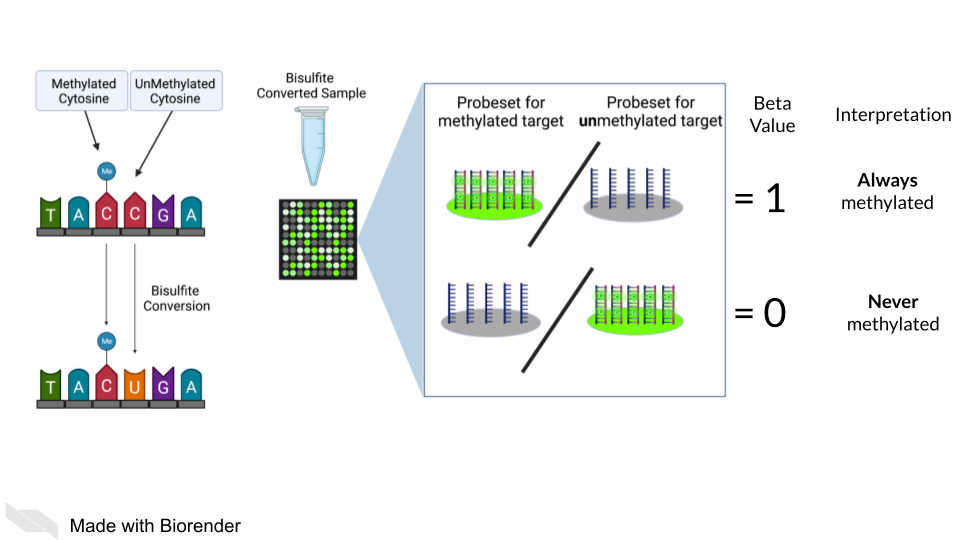


Gene expression arrays are designed to measure gene expression. They are designed to target and measure relative transcript abundance level.

#### 7.4.2.1 Examples:

* [refine.bio](https://www.refine.bio/) is the largest collection of publicly available, already normalized gene expression data (including gene expression microarrays).
* [Getting started in gene expression microarray analysis](https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000543) ([**Slonim2009?**](#ref-Slonim2009)).
* [Microarray and its applications](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3467903/) ([**Govindarajan2012?**](#ref-Govindarajan2012)).
* [Analysis of microarray experiments of gene expression profiling](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2435252/) ([Tarca, Romero, and Draghici 2006](#ref-Tarca2006)).

### 7.4.3 DNA methylation arrays

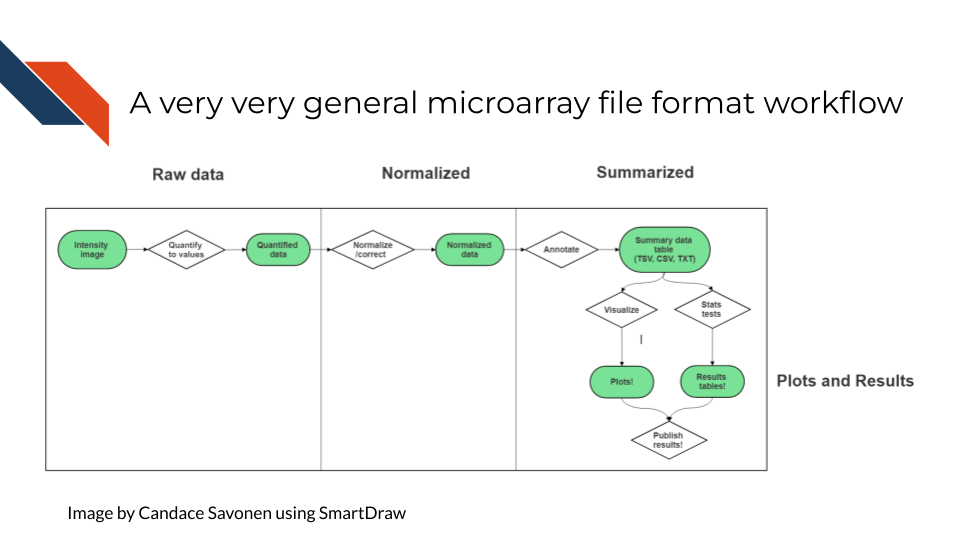


DNA methylation can also be measured by microarray. To detect methylated cytosines (5mC), DNA samples are prepped using bisulfite conversion. This converts unmethylated cytosines into uracils and leaves methylated cytosines untouched. Probes are then designed to bind to either the uracil or the cytosine, representing the unmethylated and methylated cytosines respectively.

A ratio of the fluorescence signal can be used to identify the relative abundance of the methylated and unmethylated versions of the sequence.

Additionally, 5-hydroxymethylated cytosines (5hmC) can also be detected by oxidative bisulfite bisulfite sequencing ([Booth et al. 2013](#ref-Booth2013)). Note that bisulfite conversion alone will not distinguish between 5mC and 5hmC though these often may indicate different biological mechanics.

## 7.5 General processing of microarray data



After scanning, microarray data starts as an image that needs to be quantified, normalized and further corrected and edited based on the most current genome and probe annotation.

As noted above, microarrays do not escape the base sequence biases that accompany most all genomic assays. The normalization methods you use ideally will mitigate these sequence biases and also make sure to remove probes that may be outdated or bind to multiple places on the genome.

The tools and methods by which you normalize and correct the microarray data will be dependent not only on the type of microarray assay you are performing (gene expression, SNP, methylation), but most of all what kind of microarray chip design/platform you are using.

### 7.5.1 Examples

* [Refine.bio describes their processing methods](https://docs.refine.bio/en/latest/main_text.html#processing-information).
* [Brainarray keeps up to date microarray annotation for all kinds of platforms](http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/genomic_curated_CDF.asp)

### 7.5.2 Microarray Platforms

There are so many microarray chip designs out there designed to target different things. Three of the largest commercial manufacturers have ready to use microarrays you can purchase. You can also design microarrays to hit your own targets of interest.

Here are full lists of platforms that have been published on Gene Expression Omnibus.

* [Affymetrix platforms](https://www.ncbi.nlm.nih.gov/geo/browse/?view=platforms&submitter=5&zsort=series&display=20)
* [Agilent platforms](https://www.ncbi.nlm.nih.gov/geo/browse/?view=platforms&submitter=63&display=20&zsort=series).
* [Illumina platforms](https://www.ncbi.nlm.nih.gov/geo/browse/?view=platforms&submitter=1242&zsort=series&display=20).

## 7.6 Very General Microarray Workflow

In the data type specific chapters, we will cover the microarray workflow and file formats in more detail. But in the most general sense, microarray workflows look like this, note that the exact file formats are specific to the chip brand and type you use (e.g. Illumina, Affymetrix, Agilent, etc.):

### 7.6.1 Microarray file formats

#### 7.6.1.1 IDAT - intensity data file

This is an Illumina microarray specific file that contains the chip image intensity information for each location on the microarray. It is a binary file, which means it will not be readable by double clicking and attempting to open the file directly.

Currently, Illumina appears to suggest directly converting IDAT files into a GTC format. We advise looking into [this package to help you do that](https://github.com/freeseek/gtc2vcf).

[For more on IDAT files](https://www.illumina.com/content/dam/illumina-marketing/documents/products/technotes/technote_array_analysis_workflows.pdf).

#### 7.6.1.2 DAT - data file

This is an Affymetrix’ microarray specific file parallel to the IDAT file in that it contains the image intensity information for each location on the microarray. It’s stored as pixels.

[For more on DAT files](https://www.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/dat.html).

#### 7.6.1.3 CEL

This is an Affymetrix microarray specific file that is made from a DAT file but translated into numeric values. It is not normalized yet but can be normalized into a CHP file.

[For more on CEL files](https://www.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/cel.html)

#### 7.6.1.4 CHP

CHP files contain the gene-level and normalized data from an Affymetrix array chip. CHP files are obtained by normalizing and processing CEL files.

[For more about CHP files](https://www.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/chp-xda.html).

## 7.7 General informatics files

At various points in your genomics workflows, you may need to use other types of files to help you annotate your data. We’ll also discuss some of these common files that you may encounter:

#### 7.7.0.1 BED - Browser Extensible Data

A BED file is a text file that has coordinates to genomic regions. THe other columns that accompany the genomic coordinates are variable depending on the context. But every BED file contains the chrom, chromStart and chromEnd columns to start.

A BED file might look like this:

chrom chromStart chromEnd other\_optional\_columns  
chr1 0 1000 good  
chr2 100 3000 bad

For [more on BED files](https://en.wikipedia.org/wiki/BED_(file_format)).

#### 7.7.0.2 GFF/GTF General Feature Format/Gene Transfer Format

A GFF file is a tab delimited file that contains information about genomic features. These types of files are available from databases and what you can use to annotate your data.

You may see there are GFF2, GFF3, and GTF files. These only refer to different versions and variations. They generally have the same information. In general, GFF2 is being phased out so using GFF3 is generally a better bet unless the program or package you are using specifies it needs an older GFF2 version.

A GFF file may look like this (borrowed example from Ensembl):

1 transcribed\_unprocessed\_pseudogene gene 11869 14409 . + . gene\_id "ENSG00000223972"; gene\_name "DDX11L1"; gene\_source "havana"; gene\_biotype "transcribed\_unprocessed\_pseudogene";

Note that it will be useful for annotating genes and what we know about them.

For [more about GTF and GFF files](https://useast.ensembl.org/info/website/upload/gff.html).

### 7.7.1 Other files

\* If you didn’t see a file type listed you are looking for, take a look at this [list by the BROAD](https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats). Or, it may be covered in the data type specific chapters.

### 7.7.2 Microarray processing tutorials:

For the most common microarray platforms, you can see these examples for how to process the data:

#### 7.7.2.1 General arrays

* [Using Bioconductor for Microarray Analysis](https://www.bioconductor.org/packages/devel/workflows/vignettes/arrays/inst/doc/arrays.html).

#### 7.7.2.2 Gene Expression Arrays

* [An end to end workflow for differential gene expression using Affymetrix microarrays](https://f1000research.com/articles/5-1384).

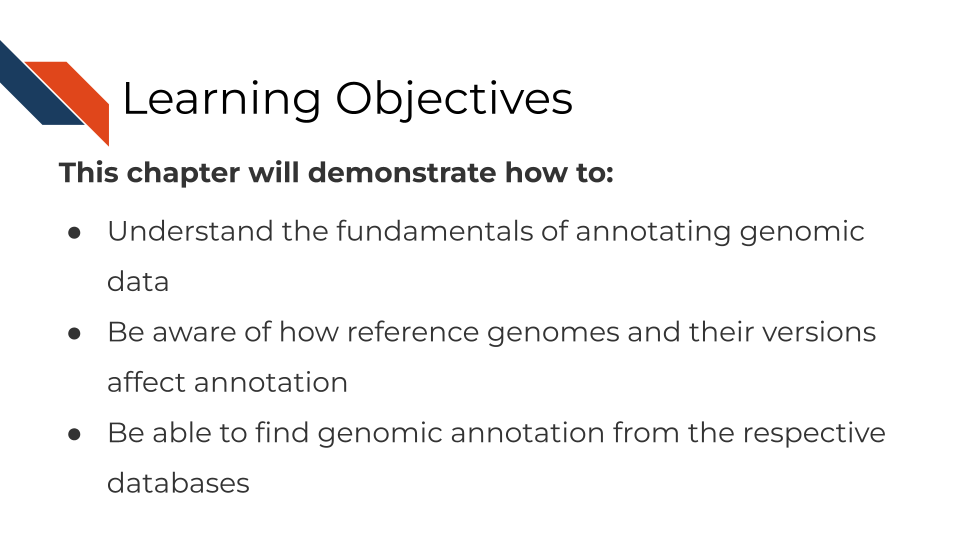
#### 7.7.2.3 DNA Methylation Arrays

* [DNA Methylation array workflow](https://nbis-workshop-epigenomics.readthedocs.io/en/latest/content/tutorials/methylationArray/Array_Tutorial.html).

# 8 Annotating Genomes

This chapter is in a beta stage. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 8.1 Learning Objectives



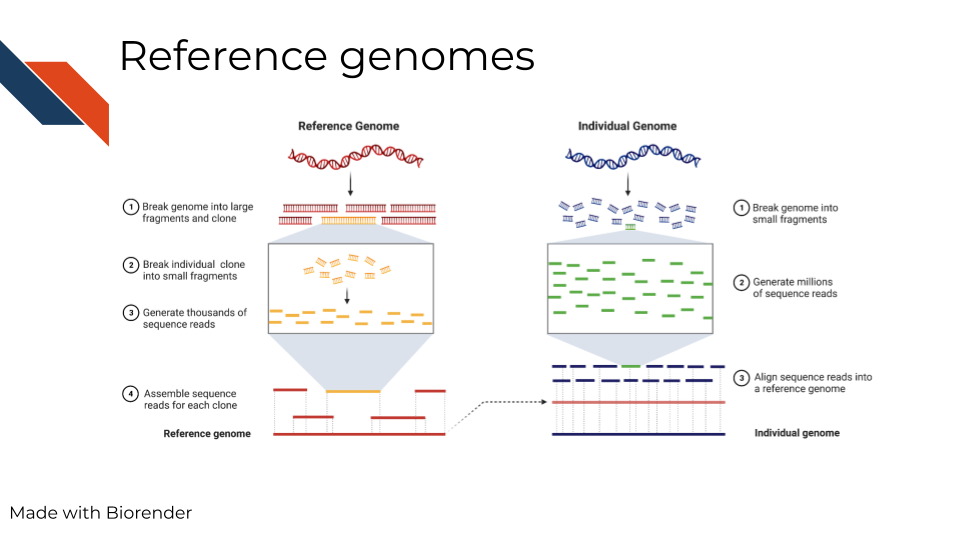
In this chapter, we are going to discuss methods that affect every genomic method and may take up the majority of your time as a genomic data analyst: Annotation.

We know that the sequencing or array data is not useful on its own – for our human minds to comprehend it and apply it to something we need a tangible piece of information to be attached to it. This is where annotation comes in. At best annotation helps you and others interpret genomic data. At its worst, its a time consuming activity that, done incorrectly, can lead to erroneous conclusions and labeling.

Proper annotation requires an understanding of how the annotation data you are using was derived as well as the realization that all annotation data is constantly changing and the confidence for these data are never 100%. Some organism’s genomes are better annotated than others but nearly all are at least somewhat incomplete.

## 8.2 What are reference genomes?

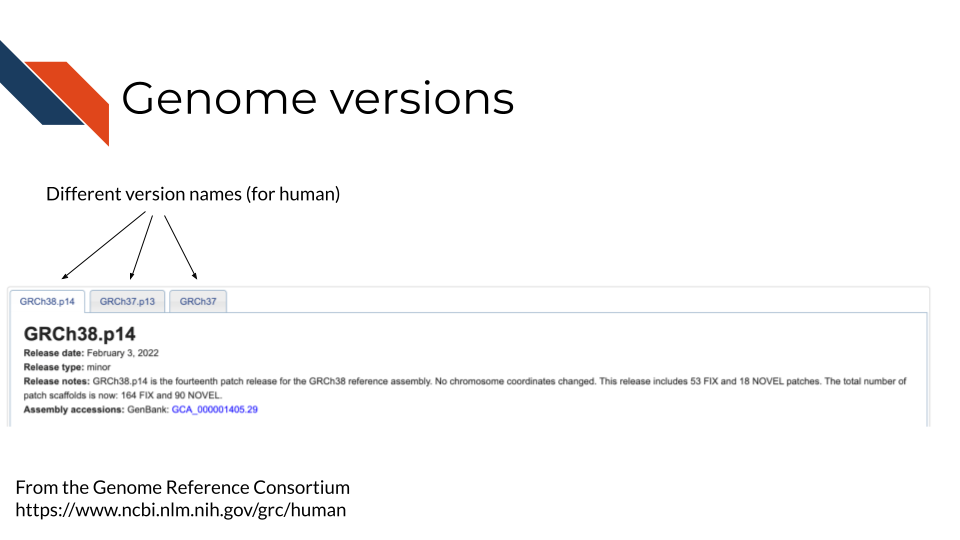
Every individual organism has its own DNA sequence that is unique to it. So how can we compare organisms to each other? In some studies, sequencing data is obtained and the genome is built de novo (aka from scratch) but this takes a lot of time and computing power. So instead, most genomic studies use the imperfect method of comparing to a reference genome. Reference genomes are built from prior data and available online. They inherently have biases in them. For example, human genomes are generally not made from diverse populations but instead from mostly males of european descent. It is inherently bad for both ethical and scientific reasons to to have [genome references that are too white](https://www.sciencenews.org/article/genetics-race-dna-databases-reference-genome-too-white). For more on the problems with reference genomes, [read this](https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1774-4).



In summary, reference genomes are used for comparison and as a ‘source of truth’ of sorts, but its important to note that this method is biased and better alternatives need to be realized.

## 8.3 What are genome versions?

If you are familiar with software development, or have used any app before, you’re familiar with software updates and releases. Similarly, the genome has updates and releases as continued cloning and assemblies of organisms teaches us more. In the image below we are showing an example of what a genome version may be noted as (note that different databases may have different terminology – here we are showing the Genome Reference Consortium). You may also notice on their website it shows the date the genome version was released and what was fixed.

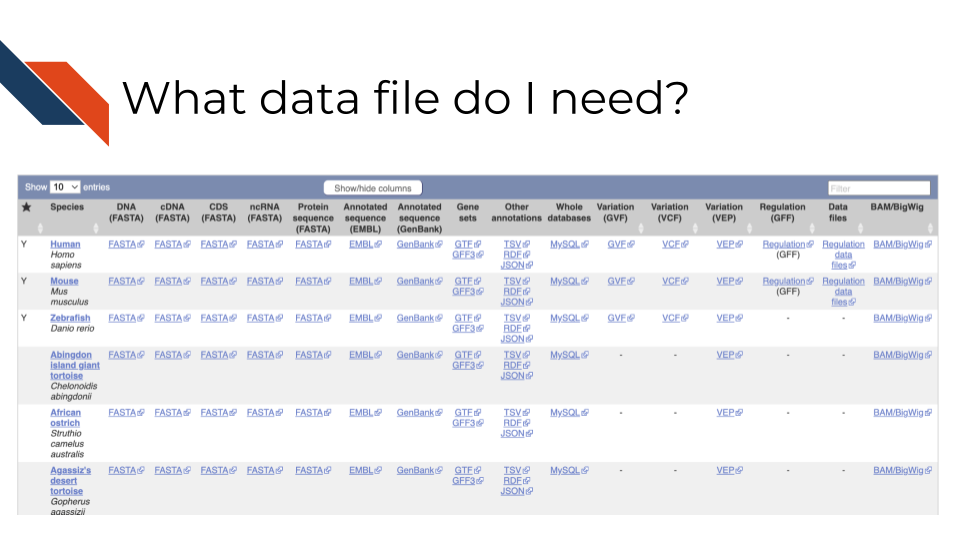


The details of how genome versions are fixed and released are not really of concern for your data analysis. This is merely to explain that genomes change and what is most important in your analysis is that:

1. You choose one genome version and consistently use it in all your analyses.
2. Choose a genome version that the rest of your field has generally had a consensus on and is also using. Generally this means sticking with major releases of a genome instead of always going with the latest version. Most databases will try to point you to their major release, so just stick with that. We will point you where you can find genome annotation for a lot of the major organisms.

## 8.4 What are the different files?

Although we can’t walk you through every organism and database set up, we will walkthrough the files and structure of one example here.



In the above screenshot, [from Ensembl](https://useast.ensembl.org/info/data/ftp/index.html), it shows different organisms in the rows, but also a variety of different files across the columns. In this example, DNA reference to the DNA sequence of the organism’s genome, but cDNA refers to complementary DNA – aka DNA that has been reversed transcribed from RNA. If you are working with RNA data you may want to use the cDNA file. Whereas CDS files are referring to only coding sequences and ncRNA files are showing only non coding sequences. Gene sets are also annotated and are in their own files. Most of these files are FASTA files. For a reminder on what these different file types are [see the previous chapter](http://hutchdatascience.org/Choosing_Genomics_Tools/a-very-general-genomics-overview.html#basic-file-formats).

Depending on the tool you are using, the data file and type you need will vary. Some tools have these data built in or are compatible with other packages that have annotation. If a tool automatically includes annotation within it, you will need to ensure that any additional tools you are using are also pulling from the same genome and version. Look into a tool’s documentation to find out what genome versions it is based on. If it doesn’t tell you at all, you don’t want to be using that tool. You cannot assume that cross genome analyses will translate.

### 8.4.1 How to download annotation files

For another database example we’ll look at the human data on ENA’s servers. Note that if you see FTP that just means “Fast Transfer Protocol” and it just means its where you can get the files themselves. For more on computing lingo, you can take our [Computing in Cancer Informatics course](https://www.itcrtraining.org/courses#h.civy2cnri95t).

[There’s many ways you can download these files and they are described here](https://ena-docs.readthedocs.io/en/latest/retrieval/file-download.html). In summary: - If you don’t feel comfortable using command line, [you can use the browser downloader for ENA here](https://www.ebi.ac.uk/ena/browser/home) - If you are using command line to write a script, then you can write use the [wget or curl instructions described here](https://ena-docs.readthedocs.io/en/latest/retrieval/file-download.html#using-wget). Be sure to read the README files to understand what it is you are downloading.

Also note that if you are working from a high power computing cluster or other online server, these annotation files may already be available to you. You don’t want to take up more computing resources by downloading extra files, so check with an administrator or informatics expert who also uses the cluster or cloud to check if the annotation files already exist in your workspace.

## 8.5 Considerations for annotating genomic data

### 8.5.1 Make sure you have the right file to start!

1. Is the annotation from the right organism? You may think this is a dumb question, but its very critical that you make sure you have the genome annotation for the organism that matches your data. Indeed the author of this has made this mistake in the past, so double check that you are using the correct organism.
2. Are all analyses utilizing coordinates from the same genome/transcriptome version? Genome versions are constantly being updated. Files from older genome versions cannot be used with newer ones (without some sort of liftover conversion). This also goes for transcriptome and genome data. All analysis need to be done using the same genomic versions so that is ensured that any chromosomal coordinates can translate between files. For example, it could be in one genome version a particular gene was said to be at chromosome base pairs 300 - 400, but in the next version its now been changed to 305 - 405. This can throw off an analysis if you are not careful. This type of annotation mapping becomes even more complicated when considering different splice variants or non-coding genes or regulatory regions that have even less confidence and annotation about them.

### 8.5.2 Be consistent in your annotations

1. If at all possible avoid making cross species analyses - unless you are an evolutionary genomics expert and understand what you are doing. But for most applications cross species analyses are hopeful wishing at best, so stick to one organism.
2. Avoid mixing genome/transcriptome versions. Yes there is liftover annotation data to help you identify what loci are parallel between releases, but its really much simpler to stick with the same version throughout your analyses’ annotations.

### 8.5.3 Be clear in your write ups!

Above all else, not matter what you end up doing, make sure that your steps, what files you use, and what tool versions you use are clear and reproducible! Be sure to clearly link to and state the database files you used and include your code and steps so others can track what you did and reproduce it. For more information on how to create reproducible analyses, you can take our reproducibility in cancer informatics courses: [Introduction to Reproducibility](https://www.itcrtraining.org/courses#h.n5yoq68qj0rz) and [Advanced Reproducibility in Cancer Informatics](https://www.itcrtraining.org/courses#h.i5zyiyjyttr4).

## 8.6 Resources you will need for annotation!

### 8.6.1 Annotation databases

* [Ensembl](https://useast.ensembl.org/downloads.html)
* [EMBL-EBI](https://www.ebi.ac.uk/services)
* [UCSCGenomeBrowser](https://genome.ucsc.edu/cgi-bin/hgGateway?hgsid=1571980135_Ym6A5aa3nDyOfZKtGishprdrhLDm)
* [NCBI Genomes download page](https://ftp.ncbi.nlm.nih.gov/genomes/)

### 8.6.2 GUI based annotation tools

* [UCSCGenomeBrowser](https://genome.ucsc.edu/cgi-bin/hgGateway?hgsid=1571980135_Ym6A5aa3nDyOfZKtGishprdrhLDm)
* [BROAD’s IGV](https://software.broadinstitute.org/software/igv/)
* [Ensembl’s biomart](https://useast.ensembl.org/info/data/biomart/index.html)

### 8.6.3 Command line based tools

#### 8.6.3.1 R-based packages:

* [annotatr](https://bioconductor.org/packages/release/bioc/html/annotatr.html)
* [ensembldb](https://bioconductor.org/packages/release/bioc/html/ensembldb.html)
* [GenomicRanges](https://www.bioconductor.org/packages/release/bioc/html/GenomicRanges.html) - useful for manipulating and identifying sequences.
* [GO.db](https://bioconductor.org/packages/3.16/data/annotation/html/GO.db.html) - Gene ontology annotation
* [org.Hs.eg.db](https://bioconductor.org/packages/release/data/annotation/html/org.Hs.eg.db.html)
* [RSamtools](https://www.bioconductor.org/packages/release/bioc/html/Rsamtools.html)
* [A full list of Bioconductors annotation packages](https://bioconductor.org/packages/3.16/data/annotation/) - contains annotation for all kinds of species and versions of genomes and transcriptomes.

#### 8.6.3.2 Python-based packages:

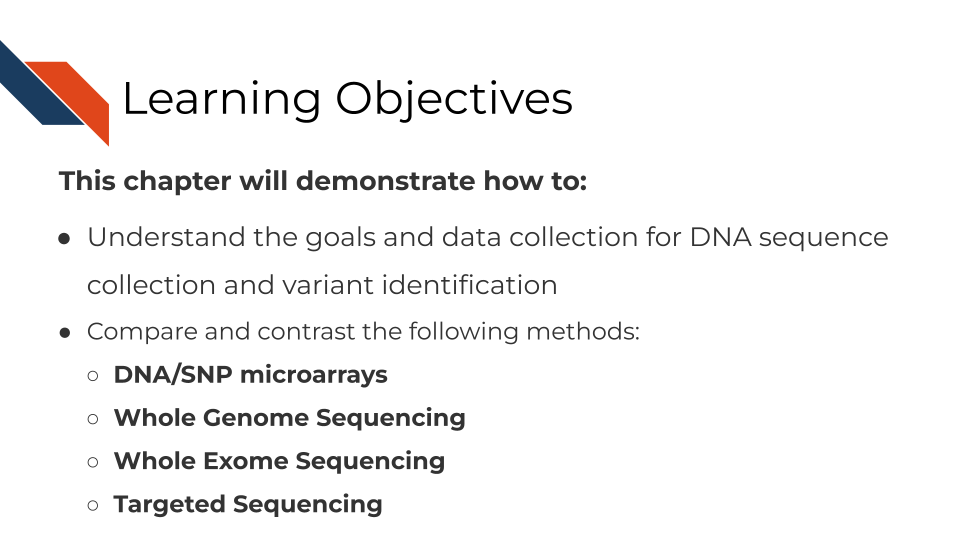
* [BioPython](https://biopython.org/)
* [genetrack](https://github.com/ialbert/chipexo)

### 8.6.4 More resources about genome annotation

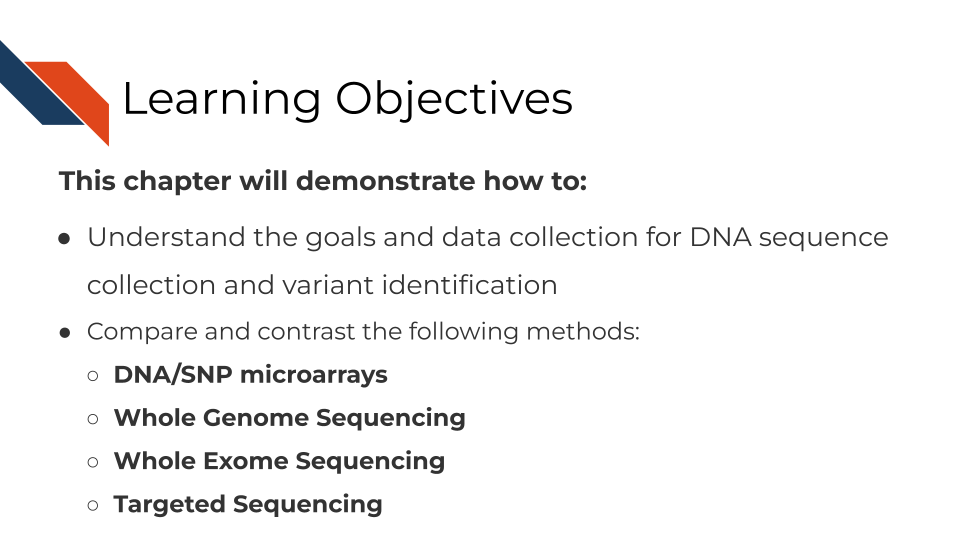
# 9 DNA Methods Overview

This chapter is in a beta stage. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

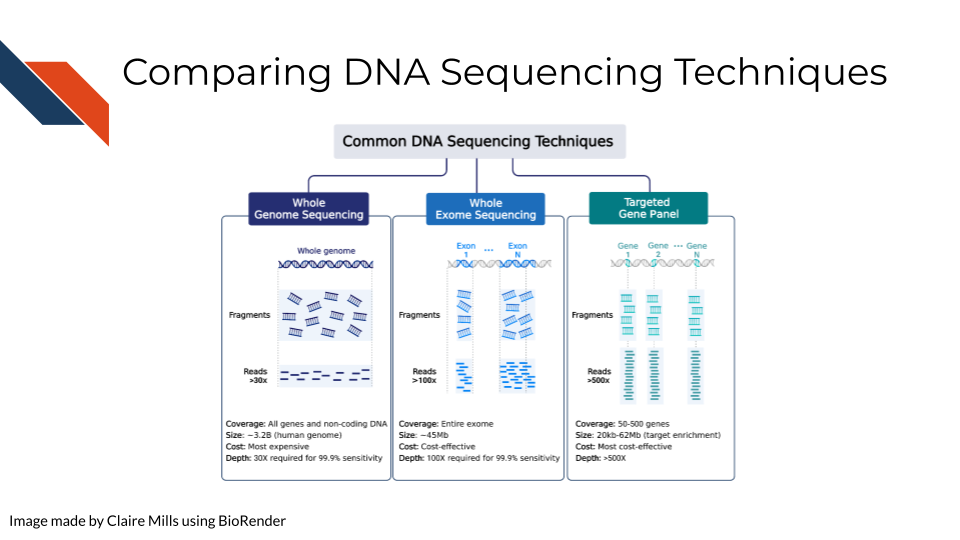
## 9.1 Learning Objectives



## 9.2 What are the goals of analyzing DNA sequences?



## 9.3 Comparison of DNA methods

 Compared to WXS and Targeted Gene Sequencing, WGS is the most expensive but requires the lowest depth of coverage to achieve 95% sensitivity. In other words, WGS requires sequencing each region of the genome (3.2 billion bases) 30 times in order to confidently be able to pick up all possible meaningful variants. ([Sims et al. 2014](#ref-Sims2014)) goes into more depth on how these depths are calculated.

Alternatively, WXS is a more cost effective way to study the genome, focusing places in the genome that have open reading frames – aka generally genes that are able to be expressed. This focuses on enriching for exons and not introns so splicing variants may be missed. In this case, each gene must be sequenced 80-100x for sufficient sensitivity to pick up meaningful variants.

In targeted gene sequencing, a panel of 50-500 regions of interest are selected. This technique is very applicable for studying a set of specific genes of interest at great depth to identify all varieties of mutations within those specific genes. These genes must be sequenced at much greater depth (>500x) to confidently identify all meaningful variants. [This page from Illumina](https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/coverage.html) also provides information regarding sequencing depth considerations for different modalities.

Additional references: WGS: ([Bentley et al. 2008](#ref-Bentley2008)) WES: ([Clark et al. 2011](#ref-Clark2011)) Targeted: ([Bewicke-Copley et al. 2019](#ref-BewickeCopley2019))

## 9.4 How to choose a DNA sequencing method

Before starting any sequencing method, you likely have a research question or hypothesis in mind. In order to choose a DNA sequencing method, you will need to consider a few items in balance of each other:

### 9.4.1 1. What region(s) of the genome pertain to your research question?

Is this unknown? Can it be narrowed down to non-coding or coding regions? Is there an even more specific subset of interest?

### 9.4.2 2. What does your project budget allow for?

Some methods are much more costly than others. Cost is not only a factor for the reagents needed to sequence, but also the computing power needed to process and store the data and people’s compensation for their work on the data. All of these costs increase as the amounts of data that are collected increase. For more information on computing decisions see our [Computing in Cancer Informatics course](https://www.itcrtraining.org/courses#h.civy2cnri95t).

### 9.4.3 3. What is your detection power for these variants?

Detecting DNA variants is not simply a matter of yes or no, but a confidence level due to sequencing errors in data collection. Are the variants you are looking for very rare and/or small (single nucleotide or very few copy number differences)? If so you will need more samples and potentially more sequencing depth to detect these variants with confidence.

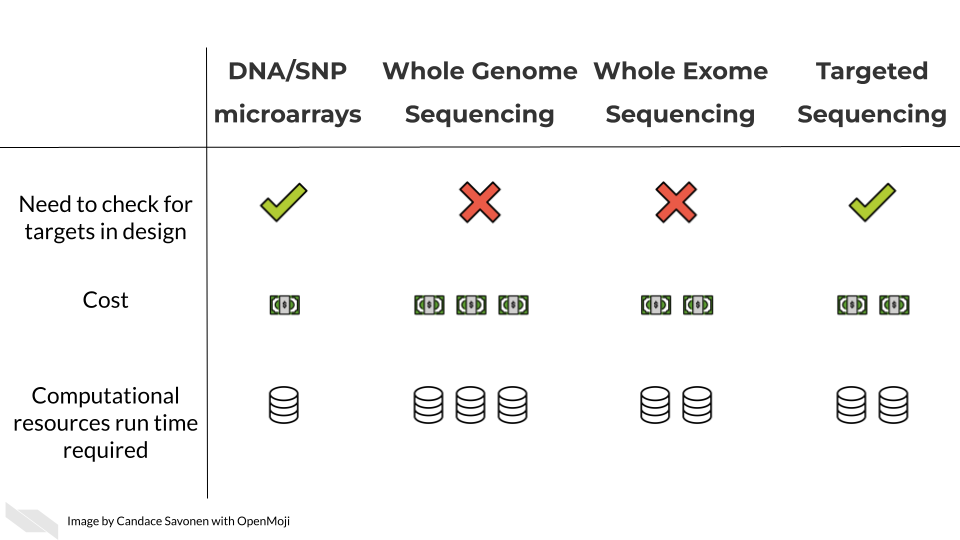
## 9.5 Strengths and Weaknesses of different methods

Is not much known about DNA variants in your organism or disease in question? In this instance you may want to cast a large net to explore more variants by using WGS.

If previous research has identified sections of the genome that are of interest to your research question, then it’s highly advisable to not sequence the entire genome with WGS methods. Not only will whole genome sequencing be more costly, but it will decrease your statistical power to discover true positive variants of interest and increase your chances of discovering false positive variants. This is because multiple testing correction needs to be applied in instances where many tests are being done currently. In this instance, the tests being performed are across the whole genome.

If your research question does not pertain to non-coding regions of the genome or splicing, then its advisable to use WXS. Recall that only about 1-2% of the genome is coding sequences meaning that if you are uninterested in noncoding regions but still use WGS then 98-99% of your data will be uninteresting to you and will only serve to increase your chances of finding false positives or cost you a lot of funding. Not only does sequencing more of the genome take more money and time but it will be more costly in time and resources in terms of the computing power needed to analyze it.

Furthermore, if you are able to narrow down even further what regions are of interest this would be better in terms of cost and detection abilities. A targeted sequencing panel or DNA microarray are ideal for assaying known groups of targets. DNA microarrays are the least costly of all the methods to identify DNA variants, but with both targeted sequencing and DNA microarray you will need to find or create a custom probe or primer set. Ideally a probe or primer set that hits your regions of interest already exists commercially but if not, then you will have to design your own – which also costs time and money.

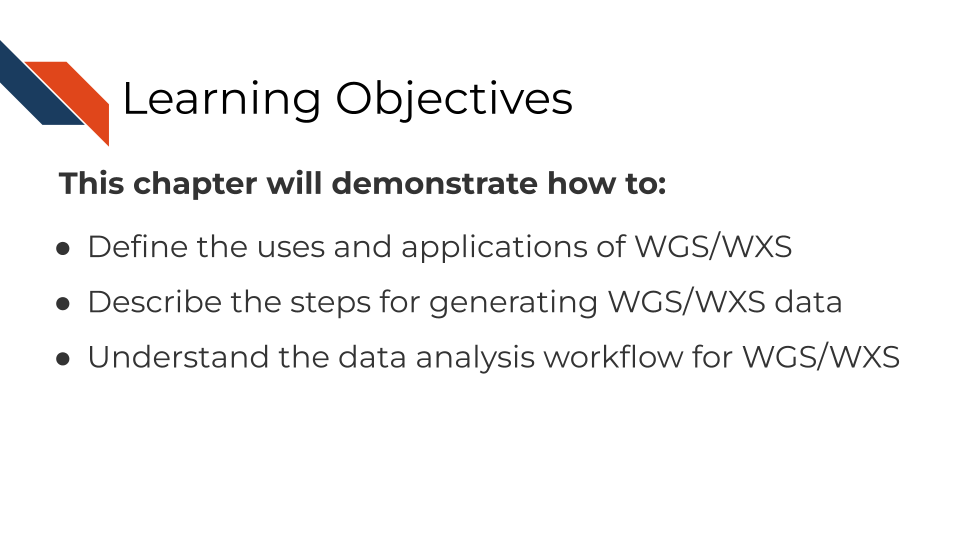


In these upcoming chapters we will discuss in more detail each of these methods, what the data represent, what you need to consider, and what resources you can consult for analyzing your data.

# 10 Whole Genome/Exome Sequencing

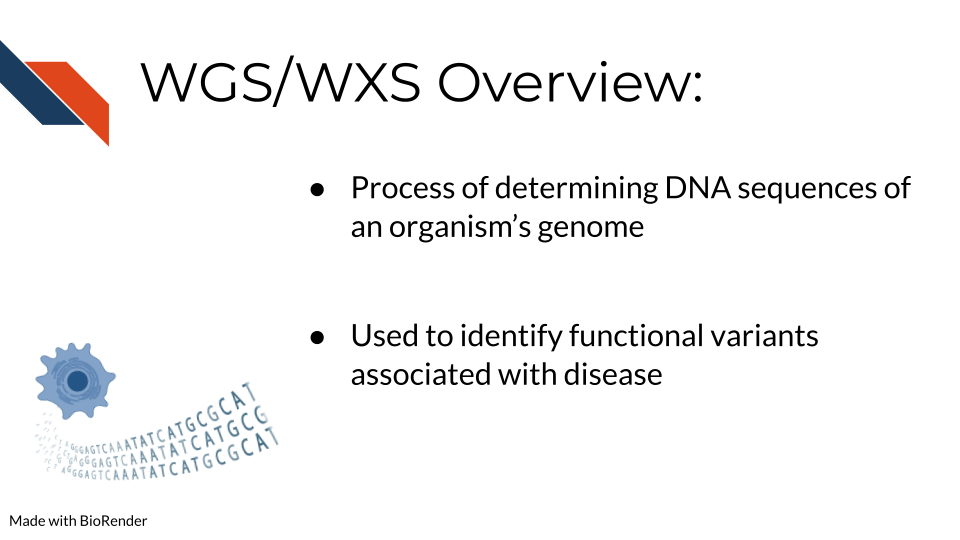
This chapter is in a beta stage. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 10.1 Learning Objectives

 The learning objectives for this course are to explain the use and application of Whole Genome Sequencing (WGS) and Whole Exome Sequencing (WES/WXS) for genomics studies, outline the technical steps in generating WGS/WXS data, and detail the processing steps for analyzing and interpreting WGS/WXS data.

**To familiarize yourself with sequencing methods as a whole, we recommend you read our** [**chapter on sequencing first**](http://hutchdatascience.org/Choosing_Genomics_Tools/sequencing-data.html)**.**

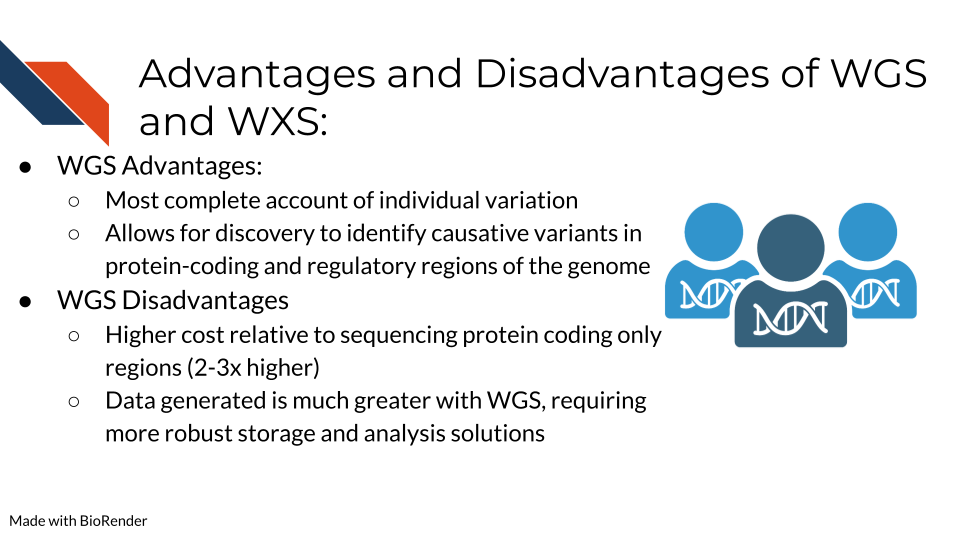
## 10.2 WGS and WGS Overview

 The difference between WGS and WXS sequencing is whether or not the open reading frames and thus coding regions are targeted in sequencing. WGS attempts to sequence the whole genome, while for WXS only exons with open reading frames are targeted for sequencing. Both of these methods can be massively beneficial for studying rare and complex diseases.

Thus, whole genome sequencing is a technique to thoroughly analyze the entire DNA sequence of an organism’s genome. This includes sequencing all genes both coding and non-coding and all mitochondrial DNA. WGS is beneficial for identifying new and previously established variants related to disease and the regulatory elements of the genome including promoters, enhancers, and silencers. Increasingly non-coding RNAs have also been identified to play a functional role in biological mechanisms and diseases. In order to learn more about the non-coding regions of the genome, WGS is necessary.

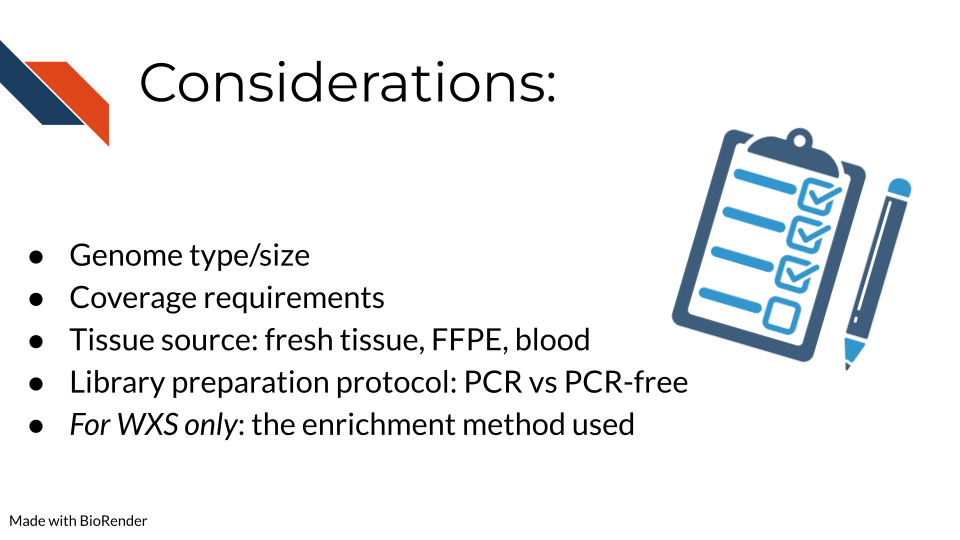
Alternatively whole exome sequencing is used to sequence the coding regions of an organism’s genome. Although non-coding regions can sometimes reveal valuable insights, coding regions can be a useful area of the genome to focus sequencing methods on, since changes in a protein coding sequence of the genome generally have more information known about them. Often protein coding sequences can have more clearly functional changes - like if a stop codon is introduced or a codon is changed to a predictable amino acid. This can more easily lead to downstream investigations on the functional implications of the protein affected.

## 10.3 Advantages and Disadvantages of WGS vs WXS



We more thoroughly discuss how to choose DNA sequencing methods [here in the previous chapter](http://hutchdatascience.org/Choosing_Genomics_Tools/dna-methods.html), but we will briefly cover this here. Alternatives to WGS include Whole Exome Sequencing (WES/WXS), which sequences the open reading frame areas of the genome or Targeted Gene Sequencing where probes have been designed to sequence only regions of interest. The main advantages of WGS include the ability to comprehensively analyze all regions of a genome, the ability to study structural rearrangements, gene copy number alterations, insertions and deletions, single nucleotide polymorphisms (SNPs), and sequencing repeats. Some disadvantages include higher sequencing costs and the necessity for more robust storage and analysis solutions to manage the much larger data output generated from WGS.

## 10.4 WGS/WXS Considerations

 Some important considerations for WGS/WXS include:

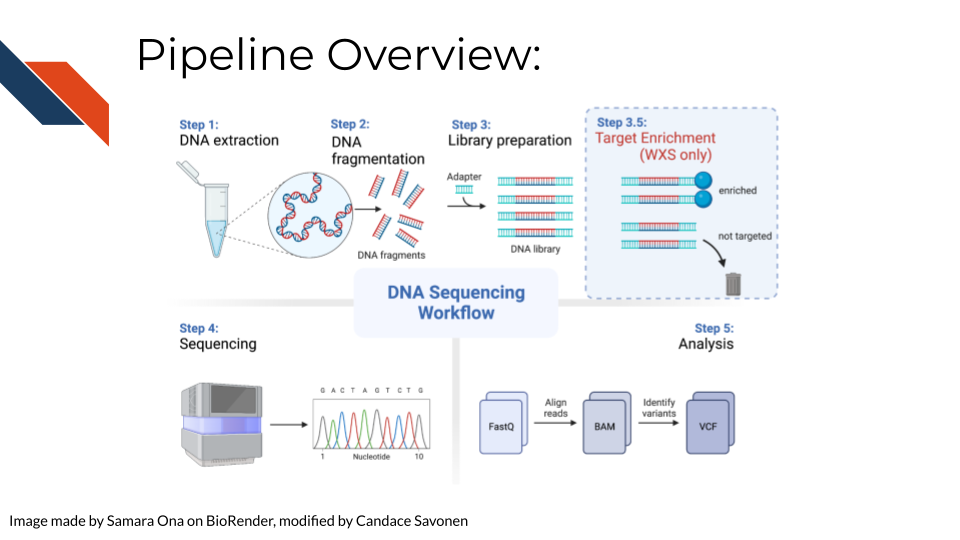
* What genome you are studying and the size of this genome. Included in this considerations is whether this genome has been sequenced before and you will have a “reference” genome to compare your data against or whether you will have to make a reference genome yourself. [This bioinformatics resource](https://eriqande.github.io/eca-bioinf-handbook/alignment-of-sequence-data-to-a-reference-genome-and-associated-steps.html) provides a great overview of genome alignment.
* The depth of coverage for sequencing is an important consideration. The typical recommendation for WGS coverage is 30x, but this is on the lower side and many researchers find it does not provide sufficient coverage compared to 50x. Illumina has an [infographic](https://www.illumina.com/content/dam/illumina-marketing/documents/techniques/wgs-wes-infographic-web.pdf) that explains this information
* The tissue source and whether genetic alterations were introduced during processing are important. Fixation for formalin-fixed paraffin embedded (FFPE) can introduce mutations/genetic changes that will need to be accounted for during data analysis. [This page from Beckman](https://www.beckman.com/resources/technologies/next-generation-sequencing/challenges-with-ffpe-tissue-samples) addresses many of the questions researchers often have about utilizing FFPE samples for their sequencing studies.
* The library preparation method of DNA amplification via PCR is very important as PCR can often introduce duplicates that interfere with interpreting whether a mutant gene is truly frequent or just over amplified during sequencing preparation. [Illumina](https://www.illumina.com/products/by-type/sequencing-kits/library-prep-kits/dna-pcr-free-prep.html) provides a comparison of using PCR and PCR-free library preparation methods on their website.

### 10.4.1 Target enrichment techniques

For WXS or other targeted sequencing specifically (so not relevant to WGS data), what methods were used to enrich for the targeted sequences? (Which is the entire exome in the case of general WXS) These methods are generally summarized into two major categories: Hybridization based and amplicon based enrichment.

- [Hybridization based enrichment](https://www.paragongenomics.com/target-enrichment/). This includes a variety of widely used methods that we will broadly categorize in two groups: Array-based and In-solution:  
 - [Array-based capture](https://en.wikipedia.org/wiki/Exome\_sequencing#:~:text=Target%2Denrichment%20strategies-,Array%2Dbased%20capture,-In%2Dsolution%20capture) uses microarrays that have probes designed to bind to known coding sequences. Fragments that do not bind to these probes are washed away, leaving the sample with known coding sequences bound and ready for PCR amplification [@Hodges2007; @Turner2009].  
 - [In-solution capture](https://en.wikipedia.org/wiki/Exome\_sequencing#In-solution\_capture) has become more popular in recent years because it [requires less sample DNA than array-base capture](https://sequencing.roche.com/global/en/article-listing/what-is-ngs-target-enrichment-and-why-is-it-important.html). To enrich for coding sequences, in-solution capture has a pool of custom probes that are designed to bind to the coding regions in the sample. Attached to these probes are beads which can be physically separated from DNA that is not bound to the probes (this should be the non-coding sequences) [@Mamanova2010].   
- [PCR/Amplicon based enrichment](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9318977/) requires even less sample than the other two strategies and so is ideal for when the amount of sample is limited or the DNA has been otherwise processed harshly (e.g. with paraffin embedding). Because the other two enrichment methods are done after PCR amplification has been done to the whole genomic DNA sample, its thought that this method of selective PCR amplification for enrichment can result in more uniformly amplified DNA in the resulting sample. However this is less suitable the more gene targets you have (like if you truly need to sequence all of the exome) since amplicons need to be designed for each target. Overall it is much more affordable of a method. There are several variations of this method that are [discussed thoroughly by @Singh2022](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9318977/).

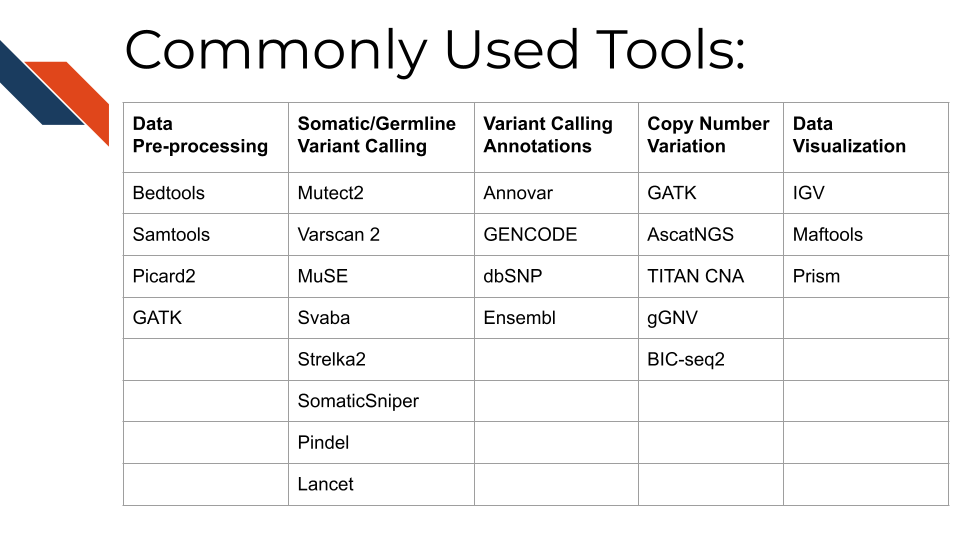
## 10.5 DNA Sequencing Pipeline Overview

 In order to create WGS/WXS data, DNA is first extracted from a specific sample type (tissue, blood samples, cells, FFPE blocks, etc.). Either traditional (involving phenol and chloroform) or commercial kits can be used for this first step. Next, the DNA sequencing libraries are prepared. This involves fragmenting the DNA, adding sequencing adapters, and DNA amplification if the input DNA is not of sufficient quantity. Recall that for WXS After sequencing, data is analyzed by converting and aligning reads to generate a BAM file. Many analysis tools will use the BAM file to identify variants, which then generates a VCF file. More information about sequencing and BAM and VCF file generation can be found [here](http://hutchdatascience.org/Choosing_Genomics_Tools/sequencing-data.html) in the sequencing data chapter.

## 10.6 Data Pre-processing

 Raw sequencing reads are first transformed into a fastq file (more information about fastq files can be found [here](http://hutchdatascience.org/Choosing_Genomics_Tools/sequencing-data.html) in the sequencing data chapter in the Quality Controls section. Then the sequencing reads are aligned to a reference genome to create a BAM file. This data is sorted and merged, and PCR duplicates are identified. The confidence that each read was sequenced correctly is reflected in the base quality score. This score must be recalibrated at this step before variants are called. A final BAM file is thus created. This can be used for future analysis steps include variant or mutation identification, which is outlined on the following slide.

## 10.7 Commonly Used Tools

 The following link provides the data analysis pipeline written by researchers in the NCI division of the NIH and provides a helpful overview of the typical steps necessary for [WGS analysis](https://docs.gdc.cancer.gov/Data/Bioinformatics_Pipelines/DNA_Seq_Variant_Calling_Pipeline/).

Here are many of the tools and resources used by researchers for analyzing WGS data.

## 10.8 Data pre-processing tools

In most cases, all of these tools will be used sequentially to prepare the data for downstream mutational and copy number variation (CNV) analysis.

* [Bedtools](https://bedtools.readthedocs.io/en/latest/) including the bamtofastq function, which is the first step in converting data off the sequencer to a usable format for downstream analysis
* [Samtools](http://www.htslib.org/) including tools for converting fastq to BAM files while mapping genes to the genome, duplicate read marking, and sorting reads
* [Picard2](https://broadinstitute.github.io/picard/) including tools to covert fastq to SAM files, filter files, create indices, mark read duplicates, sort files, and merge files
* [GATK](https://gatk.broadinstitute.org/hc/en-us) is a comprehensive set of tools from the Broad Institute for analyzing many types of sequencing data. For pre-processing, the print read function is very beneficial for writing the reads from a BAM or SAM file that pass specific criteria to a new file

## 10.9 Tools for somatic and germline variant identification

These tools are used to identify either somatic or germline mutations from a sequenced sample. Many researchers will often use a combination of these tools to narrow down only variants that are identified using a combination of these analysis algorithms. All of these mutation calling tools except SvABA can be used on both WGS and WXS data.

* [Mutect2](https://gatk.broadinstitute.org/hc/en-us/articles/9570422171291-Mutect2) This is a beneficial variant calling tool with functions including using a “panel of normals” (samples provided by the user of many normal controls) to better compare disease samples to normal and filtering functions for samples with orientation bias artifacts (FFPE samples) called F1R2, which is explained in the link above.
* [Varscan 2](https://varscan.sourceforge.net/) This is a helpful tool that utilizes a heuristic/statistic approach to variant calling. This means that it detects somatic CNAs (SCNAs) as deviations from the log-ratio of sequence coverage depth within a tumor–normal pair, and then quantify the deviations statistically. This approach is unique because it accounts for differences in read depth between the tumor and normal sample. Varscan 2 can also be used for identifying copy number alterations in tumor-normal pairs.
* [MuSE](https://bioinformatics.mdanderson.org/public-software/muse/) This is a beneficial mutation calling tool when you have both tumor and normal datasets. [The Markov Substitution Model for Evolution](https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1029-6) utilized in this tool models the evolution of the reference allele to the allelic composition of the tumor and normal tissue at each genomic locus.
* [SvABA](https://github.com/walaj/svaba) This tool is especially useful for calling insertions and deletions (indels) because it assembles aberrantly aligned sequence reads that reflect indels or structural variants using a custom [String Graph Assembler](https://github.com/jts/sga). Indels can be difficult to detect with standard alignment-based variant callers.
* [Strelka2](https://github.com/Illumina/strelka) This is a small variant caller designed by Illumina. It is used for identifying germline variants in cohorts of samples and somatic variants in tumor/normal sample pairs.
* [SomaticSniper](https://gmt.genome.wustl.edu/packages/somatic-sniper/) SomaticSniper can be used to identify SNPs in tumor/normal pairs. It calculates the probability that the tumor and normal genotypes are different and reports this probability as a somatic score.
* [Pindel](https://www.sanger.ac.uk/tool/pindel/) Pindel is a tool that uses a pattern growth approach to detect breakpoints of large deletions, medium size insertion/inversion, tandem duplications.
* [Lancet](https://github.com/nygenome/lancet) This is a newer variant calling tool that uses colored de Bruijn graphs to jointly analyze tumor and normal pairs, offering strong indel detection. More information about the processes used in this variant calling tool can be found [here](https://www.nature.com/articles/s42003-018-0023-9)

Researchers may want to create a consensus file based on the mutation calls using multiple tools above. [OpenPBTA-analysis](https://github.com/AlexsLemonade/OpenPBTA-analysis/tree/master/analyses/snv-callers) shows an open source code example of how you might compare and contrast different SNV caller’s results.

For researchers who prefer GUI based platforms: [Gene Pattern has a great set of variant based tutorials](https://www.genepattern.org/variant-and-copy-number-analysis#gsc.tab=0). GenePattern is an open software environment providing access to hundreds of tools for the analysis and visualization of genomic data.

## 10.10 Tools for variant calling annotation

These are beneficial for providing functional meaning to the mutational hits identified above.

* [Annovar](https://annovar.openbioinformatics.org/en/latest/) This is a helpful tool for annotating, filtering, and combining the output data from the above tools. It can be used for gene-based, region-based, or filter-based annotations.
* [GENCODE](https://www.gencodegenes.org/#) This tool can be used to identify and classify gene features in human and mouse genomes.
* [dbSNP](https://www.ncbi.nlm.nih.gov/snp/) This is a resource to look up specific human single nucleotide variations, microsatellites, and small-scale insertions and deletions.
* [Ensembl](https://uswest.ensembl.org/index.html) This resource is a genome browser for annotating genes from a wide variety of species.
* [pVACtools](http://www.pvactools.org) supports identification of altered peptides from different mechanisms, including point mutations, in-frame and frameshift insertions and deletions, and gene fusions.

## 10.11 Tools for copy number variation analysis

Similar to the mutation calling tools, many researchers will use several of these tools and investigate the overlapping hits seen with different copy number variant calling algorithms:

* [GATK](https://gatk.broadinstitute.org/hc/en-us/articles/360035531092--How-to-part-I-Sensitively-detect-copy-ratio-alterations-and-allelic-segments) GATK has a variety of tools that can be used to study changes in copy numbers of genes. This link provides a tutorial for how to use the tools.
* [AscatNGS](https://www.crick.ac.uk/research/labs/peter-van-loo/software) These tools (allele-specific copy number analysis of tumors) are specific for WGS copy number variation analysis. They can be used to dissect allele-specific copy numbers of tumors by estimating and adjusting for tumor ploidy and nonaberrant cell admixture.
* [TitanCNA](https://github.com/gavinha/TitanCNA) This tool is used to analyze copy number variation and loss of heterozygosity at the subclonal level for both WGS and WXS data in tumors compared to matched normals. It accounts for mixtures of cell populations and estimates the proportion of cells harboring each event. The Ha lab has developed a [snakemake pipeline](https://github.com/gavinha/TitanCNA/tree/master/scripts/snakemake) to more easily use this tool. [Ha et al. published a paper describing this tool in detail here](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4216928/)
* [gGNV](https://gatk.broadinstitute.org/hc/en-us/articles/4404604788891-GermlineCNVCaller) This is a germline CNV calling tool that can be used on both WGS and WXS data. This tool has booth COHORT and CASE modes. COHORT mode is used when providing a cohort of germline samples where CASE mode is used for individual samples. More details about these modes are described in the link above.
* [BIC-seq2](http://compbio.med.harvard.edu/BIC-seq/) This tool is used to detect CNVs with or without control samples. The steps involved in this data processing tool include normalization and CNV detection.

## 10.12 Tools for data visualization

These tools are often used in parallel to look at regions of the genome, develop plots, and create other relevant figures:

* [OpenCRAVAT](https://run.opencravat.org) uses variation data in many popular variant file formats and its outputs are variant annotations and visualizations.
* [IGV](https://software.broadinstitute.org/software/igv/) IGV is an interactive tool used to easily visualize genomic data. It is available as a desktop application, web application, and JavaScript to embed in web pages. This application is very beneficial for visualizing both mutational and CNV data for WGS and WXS. IGV has many [tutorials on YouTube](https://www.youtube.com/channel/UCb5W5WqauDOwubZHb-IA_rA/featured) that are helpful for using the tool to its full potential.
* [Maftools](https://www.bioconductor.org/packages/devel/bioc/vignettes/maftools/inst/doc/maftools.html) Maftools is an R package that can be used to create informative plots from your WGS data output. It has tools to import both VCF files and ANNOVAR output for data analysis.
* [Prism](https://www.graphpad.com/scientific-software/prism/) Prism is a widely used tool in scientific research for organizing large datasets, generating plots, and creating readable figures. WGS or WXS data regarding mutations and CNV can be used as input for creating plots with this tool.

## 10.13 Resources for WGS

Online tutorials:

* [Galaxy tutorials](https://training.galaxyproject.org/training-material/topics/sequence-analysis/)
* [NCI resources](https://gdc.cancer.gov/about-data/gdc-data-processing/genomic-data-processing)
* [Bioinformaticsdotca tutorial](https://bioinformaticsdotca.github.io/CAN_2021_module4_lab)

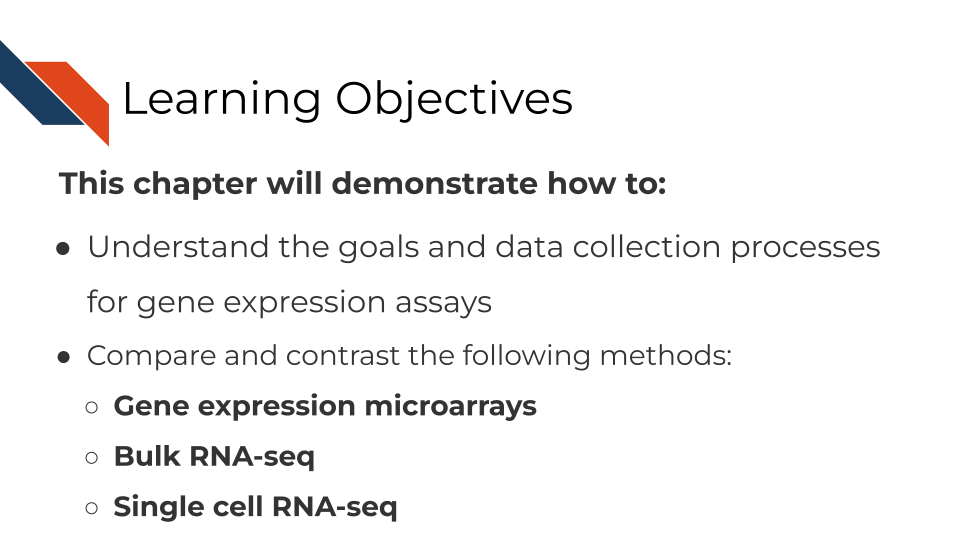
Papers comparing analysis tools:

* ([Hwang et al. 2019](#ref-Hwang2019))
* ([Naj et al. 2019](#ref-Naj2019))
* ([He et al. 2020](#ref-He2020))

# 11 RNA Methods Overview

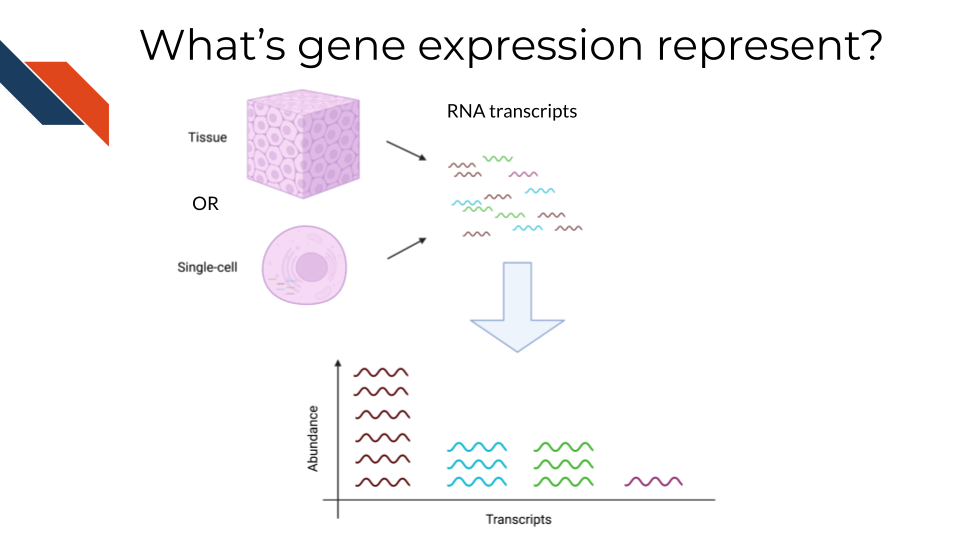
This chapter is in a beta stage. Some of it has been written with AI tools. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 11.1 Learning Objectives



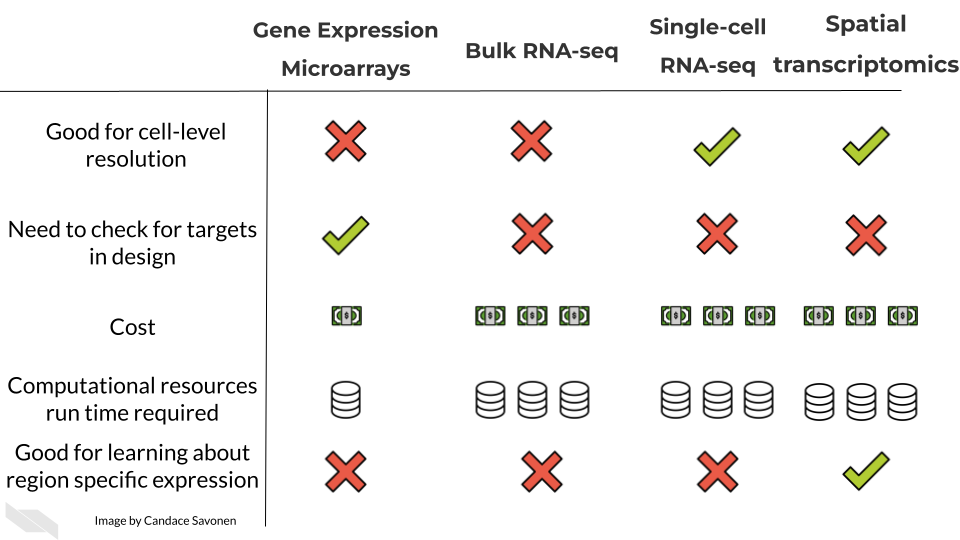
## 11.2 What are the goals of gene expression analysis?

The goal of gene expression analysis is to quantify RNAs across the genome. This can signify the extent to which various RNAs are being transcribed in a particular cell. This can be informative for what kinds of activity a cell is undergoing and responding to.



## 11.3 Comparison of RNA methods

There are three general methods we will discuss for evaluating gene expression. RNA sequencing (whether bulk or single-cell) allows you to catch more targets than gene expression microarrays but is much more costly and computationally intensive. Gene expression microarrays have a lower dynamic range than RNA-seq generally but are much more cost effective. Spatial transcriptomics is the newest method on the block and has the ability to relate gene expression to tissue regions and subpopulations.



### 11.3.1 Single-cell RNA-seq (scRNA-seq):

* Cost: scRNA-seq methods can be relatively expensive due to the need for specialized protocols and reagents. Droplet-based methods (e.g., 10x Genomics) are generally more cost-effective than full-length methods (e.g., SMART-seq) because they require fewer sequencing reads per cell.
* Experimental Goals: scRNA-seq is suitable when studying cellular heterogeneity and characterizing gene expression profiles at the single-cell level. It provides insights into cell types, cell states, and cell-cell interactions.
* Specific Requirements: scRNA-seq requires single-cell isolation techniques, and the choice of method depends on the desired cell throughput, desired coverage, and the need for full-length transcript information.

### 11.3.2 Bulk RNA-seq:

* Cost: Bulk RNA-seq is generally more cost-effective compared to scRNA-seq because it requires fewer sequencing reads per sample. The cost primarily depends on the sequencing depth required.
* Experimental Goals: Bulk RNA-seq is appropriate for analyzing average gene expression profiles across a population of cells. It provides information on gene expression levels and can be used for differential gene expression analysis.
* Specific Requirements: Bulk RNA-seq requires a sufficient quantity of RNA from the sample, typically obtained through RNA extraction and purification.

### 11.3.3 Gene Expression Microarray:

* Cost: Gene expression microarrays are usually less expensive compared to RNA-seq methods. The cost includes array production and hybridization.
* Experimental Goals: Microarrays are useful for profiling gene expression levels across a large number of genes in a cost-effective manner. They can be employed for differential gene expression analysis and identification of gene expression patterns.
* Specific Requirements: Microarrays require labeled cDNA or cRNA targets, and they are limited to the detection of known transcripts represented on the array platform.

### 11.3.4 Spatial Transcriptomics:

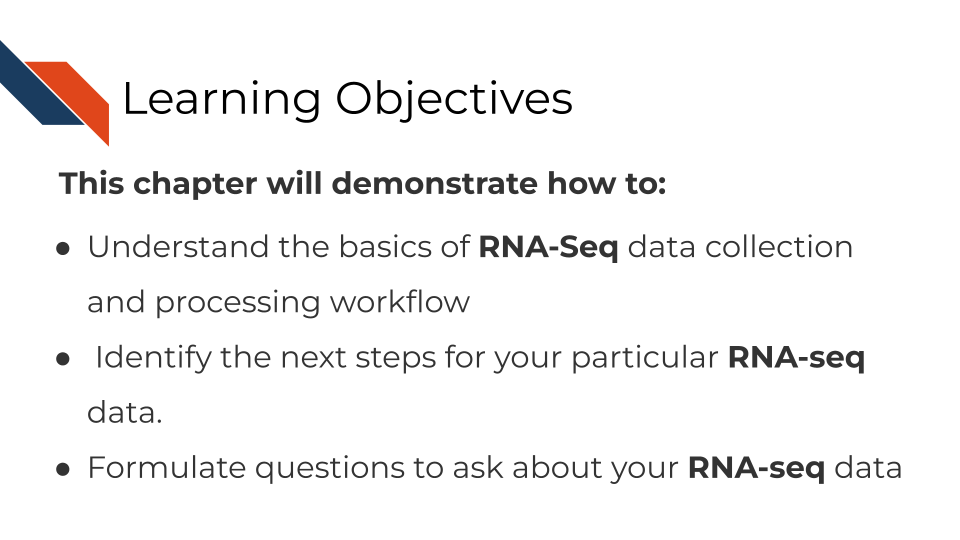
* Cost: Spatial transcriptomics methods can vary in cost depending on the technique used. Some methods involve additional steps and specialized equipment, making them relatively more expensive.
* Experimental Goals: Spatial transcriptomics allows the investigation of gene expression patterns within the context of tissue or cellular spatial organization. It provides spatial information on gene expression, enabling the identification of cell types and their interactions.
* Specific Requirements: Spatial transcriptomics requires intact tissue sections or samples, and the choice of method depends on factors such as desired spatial resolution, throughput, and compatibility with downstream analyses.

In these upcoming chapters we will discuss in more detail each of these methods, what the data represent, what you need to consider, and what resources you can consult for analyzing your data.

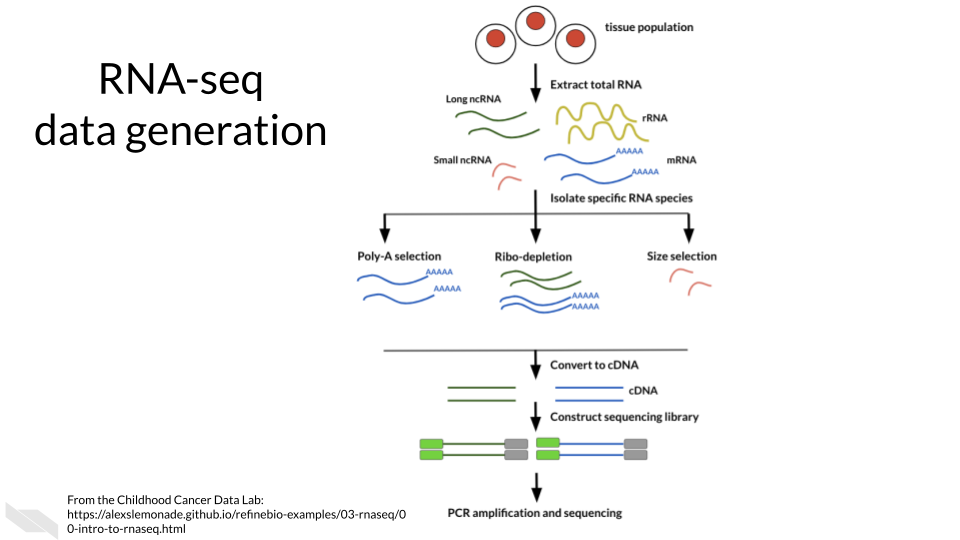
# 12 Bulk RNA-seq

This chapter is in a beta stage. If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 12.1 Learning Objectives

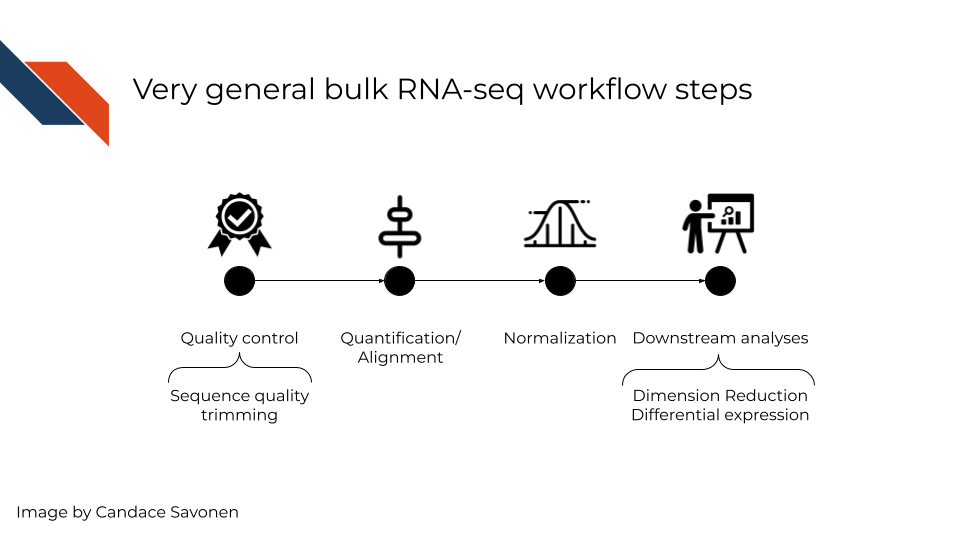


## 12.2 Where RNA-seq data comes from



## 12.3 RNA-seq workflow

In a very general sense, RNA-seq workflows involves first quantification/alignment. You will also need to conduct quality control steps that check the quality of the sequencing done. You may also want to trim and filter out data that is not trustworthy. After you have a set of reliable data, you need to normalize your data. After data has been normalized you are ready to conduct your downstream analyses. This will be highly dependent on the original goals and questions of your experiment. It may include dimension reduction, differential expression, or any number of other analyses.



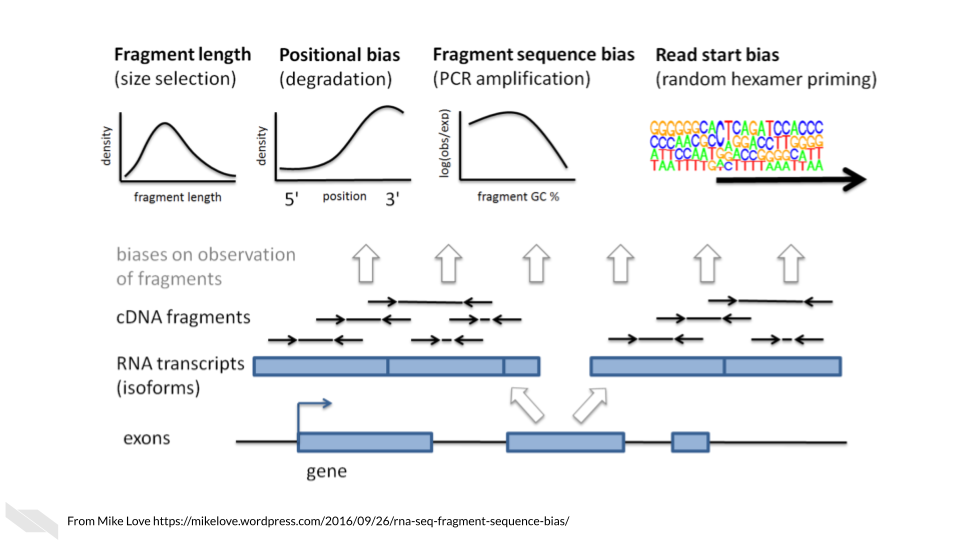
In this chapter we will highlight some of the more popular RNA-seq tools, that are generally suitable for most experiment data but there is no “one size fits all” for computational analysis of RNA-seq data ([Conesa et al. 2016](#ref-Conesa2016)). You may find tools out there that better suit your needs than the ones we discuss here.

## 12.4 RNA-seq data **strengths**

* RNA-seq can give you an idea of the transcriptional activity of a sample.
* RNA-seq has a more dynamic range of quantification than gene expression microarrays are able to measure.
* RNA-seq is able to be used for transcript discovery unlike gene expression microarrays.

## 12.5 RNA-seq data **limitations**

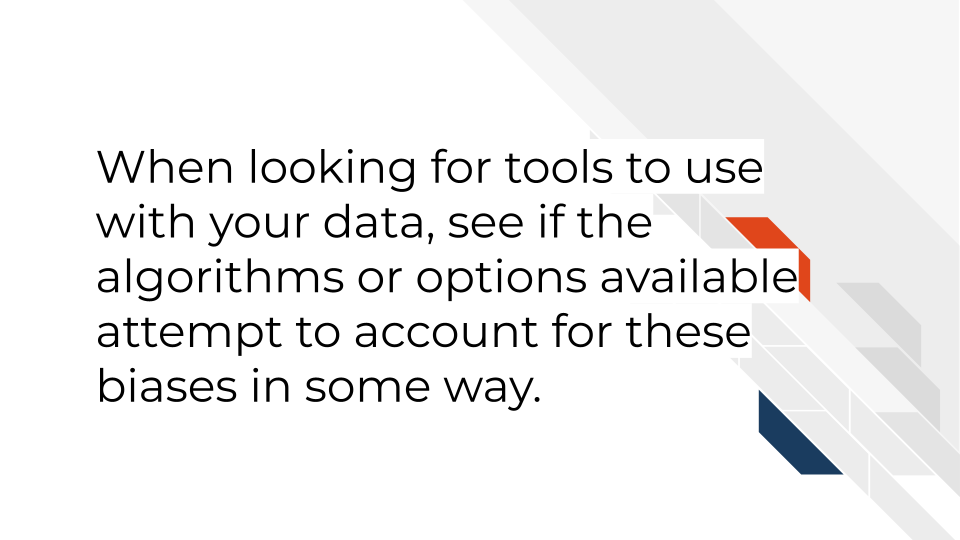
RNA-seq suffers from a lot of the common sequence biases which are further worsened by PCR amplification steps. We discussed some of the sequence biases in the previous sequencing chapter.



These biases are nicely covered in [this blog by Mike Love](https://mikelove.wordpress.com/2016/09/26/rna-seq-fragment-sequence-bias/) and we’ll summarize them here:

* **Fragment length**: Longer transcripts are more likely to be identified than shorter transcripts because there’s more material to pull from.
* **Positional bias**: 3’ ends of transcripts are more likely to be sequenced due to faster degradation of the 5’ end.
* **Fragment sequence bias**: The complexity and GC content of a sequence influences how often primers will bind to it (which influences PCR amplification steps as well as the sequencing itself).
* **Read start bias**: Certain reads are more likely to be bound by random hexamer primers than others.

*Main Takeaway*: When looking for tools, you will want to see if the algorithms or options available attempt to account for these biases in some way.



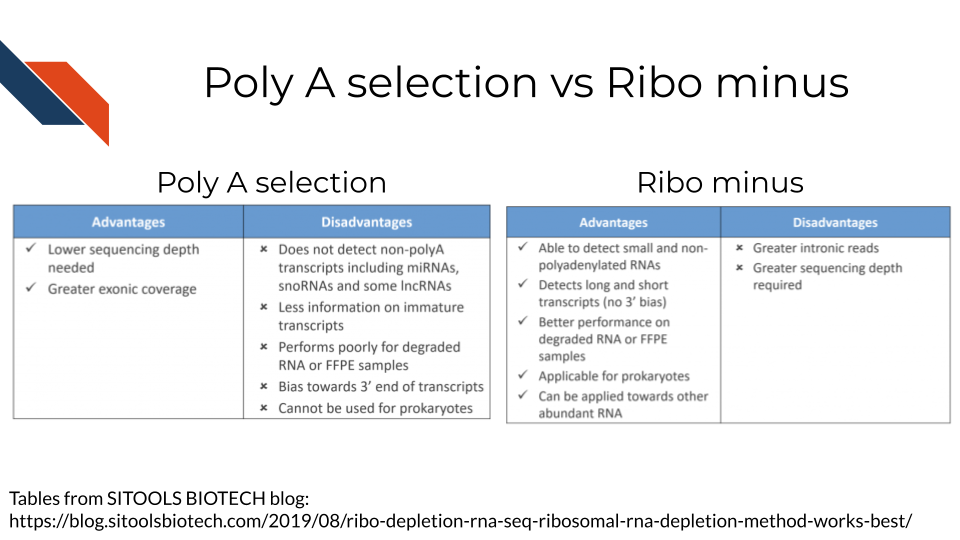
## 12.6 RNA-seq data considerations

### 12.6.1 Ribo minus vs poly A selection

Most of the RNA in the cell is not mRNA or noncoding RNAs of interest, but instead loads of ribosomal RNA a. So before you can prepare and sequence your data you need to isolate the RNAs to those you are interested in. There are two major methods to do this:

* **Poly A selection** - Keep only RNAs that have poly A tails – remember that mRNAs and some kinds of noncoding RNAs have poly A tails added to them after they are transcribed. A drawback of this method is that transcripts that are not generally polyadenylated: microRNAs, snoRNAs, certain long noncoding RNAs, or immature transcripts will be discarded. There is also generally a worse 3’ bias with this method since you are selecting based on poly A tails on the 3’ end.
* **Ribo-minus** - Subtract all the ribosomal RNA and be left with an RNA pool of interest. A drawback of this method is that you will need to use greater sequencing depths than you would with poly A selection (because there is more material in your resulting transcript pool).

[This blog by Sitools Biotech does a good summary](https://blog.sitoolsbiotech.com/2019/08/ribo-depletion-rna-seq-ribosomal-rna-depletion-method-works-best/) of the pros and cons of either selection method.

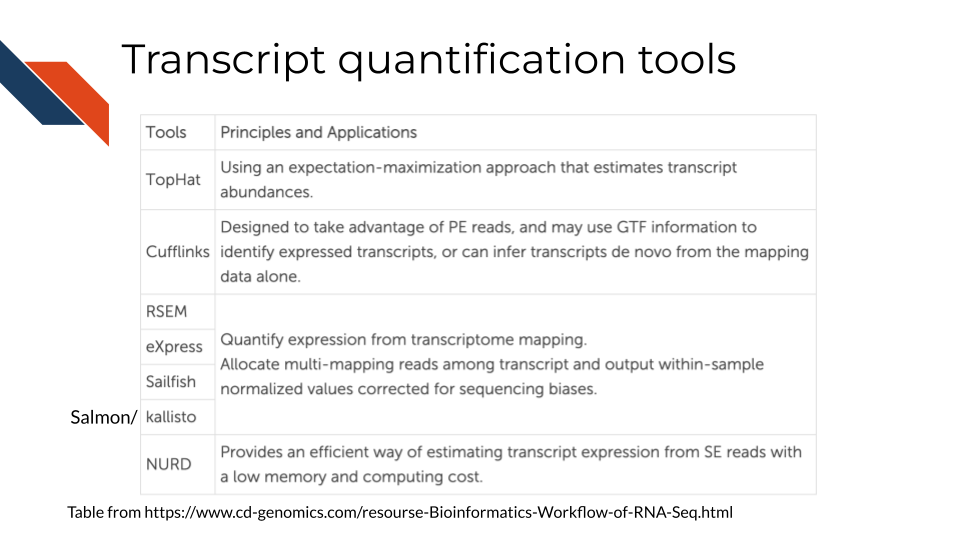


### 12.6.2 Transcriptome mapping

How do you know which read belongs to which transcript? This is where alignment comes into play for RNA-seq There are two major approaches we will discuss with examples of tools that employ them.

* **Traditional aligners** - Align your data to a reference using standard alignment algorithms. Can be very computationally intensive. Traditional alignment is the original approach to alignment which takes each read and finds where and how in the genome/transcriptome it aligns. If you are interested in identifying the intracacies of different splices and their boundaries, you may need to use one of these traditional alignment methods. But for common quantification purposes, you may want to look into pseudo alignment to save you time. *Examples of traditional aligners*:
  + [STAR](https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf)
  + [HISAT2](http://daehwankimlab.github.io/hisat2/manual/)
* This blog compares some of the traditional alignment tools
* [**Pseudo aligners**](https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html) - much faster and the trade off for accuracy is often negligible (but as always, this is likely dependent on the data you are using). The biggest drawback to pseudoaligners is that if you care about local alignment (e.g. perhaps where splice boundaries occur) instead of just transcript identification then a traditional alignment may be better for your purposes. These pseudo aligners often include a verification step where they compare a subset of the data to its performance to a traditional aligner (and for most purposes they usually perform well). Pseudo aligners can potentially save you hours/days/weeks of processing time as compared to traditional aligners so are worth looking into. *Examples of pseudo aligners*:
  + [Salmon](https://salmon.readthedocs.io/en/latest/salmon.html#using-salmon)
  + [Kallisto](https://pachterlab.github.io/kallisto/)
* **Reference free assembly** - The first two methods we’ve discussed employ aligning to a reference genome or transcriptome. But alternatively, if you are much more interested in transcript identification or you are working with a model organism that doesn’t have a well characterized reference genome/transcriptome, then de novo assembly is another approach to take. As you may suspect, this is the most computationally demanding approach and also requires deeper sequencing depth than alignment to a reference. But depending on your goals, this may be your preferred option.

These strategies are discussed at greater length [in this excellent manuscript by Conesa et al, 2016](https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8).



### 12.6.3 Abundance measures

If your RNA-seq data has already been processed, it may have abundance measure reported with it already. But there are various types of abundance measures used – what do they represent?

* **raw counts** - this is a raw number of how many times a transcript was counted in a sample.

Two considerations to think of:  
1. **Library sizes**: Raw counts does not account for differences between samples’ library sizes. In other words, how many reads were obtained from each sample? Because library sizes are not perfectly equal amongst samples and not necessarily biologically relevant, its important to account for this if you wish to compare different samples in your set.  
2. **Gene length**: Raw counts also do not account for differences in gene length (remember how we discussed longer transcripts are more likely to be counted).

Because of these items, some sort of transformation needs to be done on the raw counts before you can interpret your data.

These other abundance measures attempt to account for library sizes and gene length. [This blog and video by StatQuest](https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/) does an excellent job summarizing the differences between these quantifications and we will quote from them:

* **Reads per kilobase million (RPKM)**

1. Count up the total reads in a sample and divide that number by 1,000,000 – this is our “per million” scaling factor.
2. Divide the read counts by the “per million” scaling factor. This normalizes for sequencing depth, giving you reads per million (RPM)
3. Divide the RPM values by the length of the gene, in kilobases. This gives you RPKM.

* **Fragments per kilobase million (FPKM)**

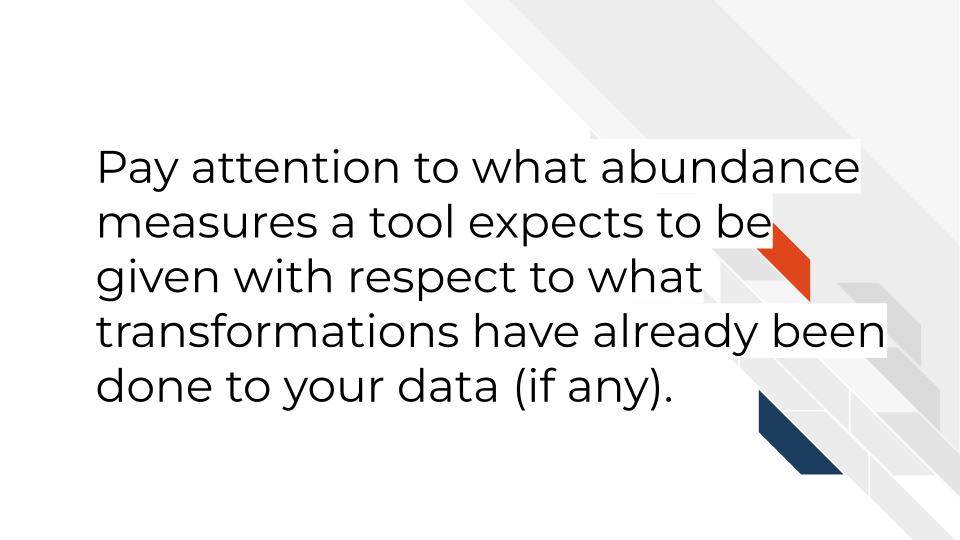
FPKM is very similar to RPKM. RPKM was made for single-end RNA-seq, where every read corresponded to a single fragment that was sequenced. FPKM was made for paired-end RNA-seq. With paired-end RNA-seq, two reads can correspond to a single fragment, or, if one read in the pair did not map, one read can correspond to a single fragment. The only difference between RPKM and FPKM is that FPKM takes into account that two reads can map to one fragment (and so it doesn’t count this fragment twice).

* **Transcripts per million (TPM)**

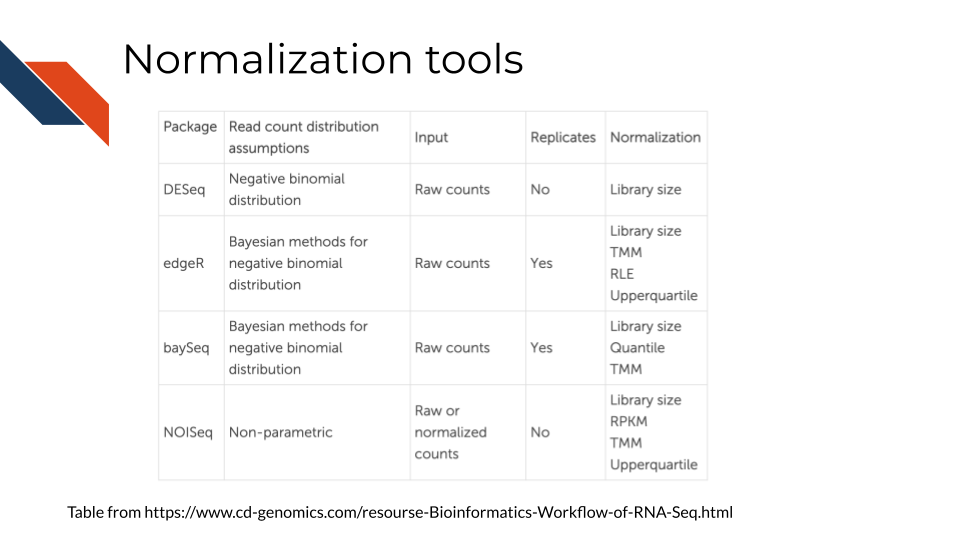
1. Divide the read counts by the length of each gene in kilobases. This gives you reads per kilobase (RPK).
2. Count up all the RPK values in a sample and divide this number by 1,000,000. This is your “per million” scaling factor.
3. Divide the RPK values by the “per million” scaling factor. This gives you TPM.

TPM has gained a popularity in recent years because it is more intuitive to understand:

When you use TPM, the sum of all TPMs in each sample are the same. This makes it easier to compare the proportion of reads that mapped to a gene in each sample. In contrast, with RPKM and FPKM, the sum of the normalized reads in each sample may be different, and this makes it harder to compare samples directly.



### 12.6.4 RNA-seq downstream analysis tools



* [ComplexHeatmap](https://bioconductor.org/packages/release/bioc/html/ComplexHeatmap.html#:~:text=Complex%20heatmaps%20are%20efficient%20to,and%20supports%20various%20annotation%20graphics.) is great for visualizations
* [DESEq2](https://www.bioconductor.org/packages/release/bioc/html/DESeq2.html) and [edgeR](https://www.bioconductor.org/packages/release/bioc/html/edgeR.html) are great for differential expression analyses.
* [CTAT](https://github.com/NCIP/Trinity_CTAT/wiki) - Using RNA-seq as input, CTAT modules enable detection of mutations, fusion transcripts, copy number aberrations, cancer-specific splicing aberrations, and oncogenic viruses including insertions into the human genome.
* [Gene Set Enrichment Analysis (GSEA)](https://www.gsea-msigdb.org/gsea/index.jsp) is a method to identify the coordinate activation or repression of groups of genes that share common biological functions, pathways, chromosomal locations, or regulation, thereby distinguishing even subtle differences between phenotypes or cellular states.
* [Gene Pattern’s RNA-seq tutorials](https://www.genepattern.org/rna-seq-analysis#gsc.tab=0) - an open software environment providing access to hundreds of tools for the analysis and visualization of genomic data.

## 12.7 Visualization GUI tools

* [WebMeV](https://webmev.tm4.org) uniquely provides a user-friendly, intuitive, interactive interface to processed analytical data uses cloud-computing elasticity for computationally intensive analyses and is compatible with single cell or bulk RNA-seq input data.
* [UCSC Xena](http://xena.ucsc.edu/) is a web-based visualization tool for multi-omic data and associated clinical and phenotypic annotations. It can be used with single cell RNA-seq data.
* [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a track-based browser for interactively exploring genomic data mapped to a reference genome.
* [Network Data Exchange (NDEx)](https://www.ndexbio.org/#/) is a project that provides an open-source framework where scientists and organizations can store, share and publish biological network knowledge.

## 12.8 RNA-seq data resources

* [ARCHS4](https://maayanlab.cloud/archs4/) (All RNA-seq and ChIP-seq sample and signature search) is a resource that provides access to gene and transcript counts uniformly processed from all human and mouse RNA-seq experiments from GEO and SRA.
* [Refine.bio](https://www.refine.bio/) - a repository of uniformly processed and normalized, ready-to-use transcriptome data from publicly available sources.

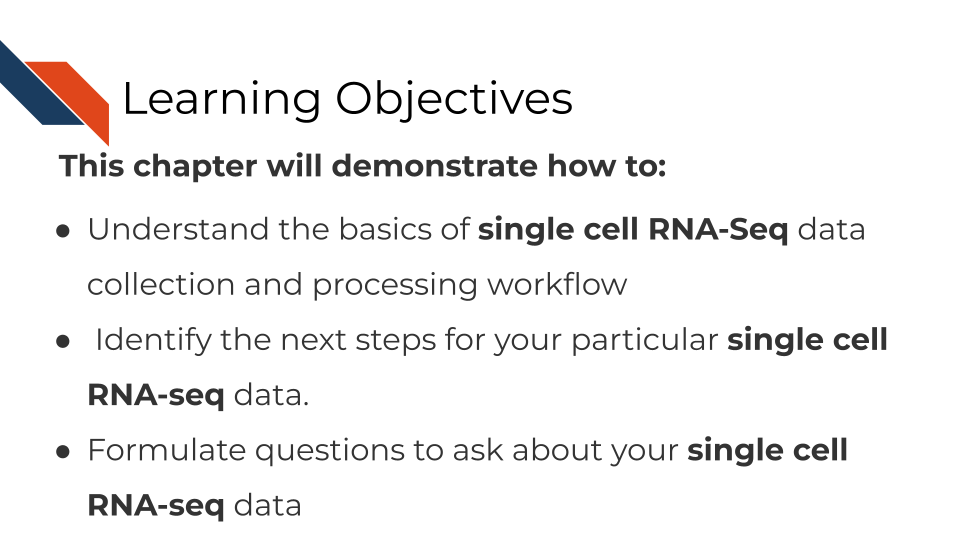
## 12.9 More reading about RNA-seq data

* [Refine.bio’s introduction to RNA-seq](https://alexslemonade.github.io/refinebio-examples/03-rnaseq/00-intro-to-rnaseq.html)
* [StatQuest: A gentle introduction to RNA-seq](https://www.youtube.com/watch?v=tlf6wYJrwKY) ([**Starmer2017-rnaseq?**](#ref-Starmer2017-rnaseq)).
* [A general background on the wet lab methods of RNA-seq](https://bitesizebio.com/13542/what-everyone-should-know-about-rna-seq/) ([**Hadfield2016?**](#ref-Hadfield2016)).
* [Modeling of RNA-seq fragment sequence bias reduces systematic errors in transcript abundance estimation](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5143225/) ([**Love2016?**](#ref-Love2016)).
* [Mike Love blog post about sequencing biases](https://mikelove.wordpress.com/2016/09/26/rna-seq-fragment-sequence-bias/) ([**bias-blog?**](#ref-bias-blog))
* [Biases in Illumina transcriptome sequencing caused by random hexamer priming](https://pdfs.semanticscholar.org/9d16/997f5de72d6c606fef3d673db70e5d1d8e1e.pdf?_ga=2.131436679.965169313.1600175795-124991789.1600175795) ([**Hansen2010?**](#ref-Hansen2010)).
* [Computation for RNA-seq and ChIP-seq studies](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4121056/) ([**Pepke2009?**](#ref-Pepke2009)).

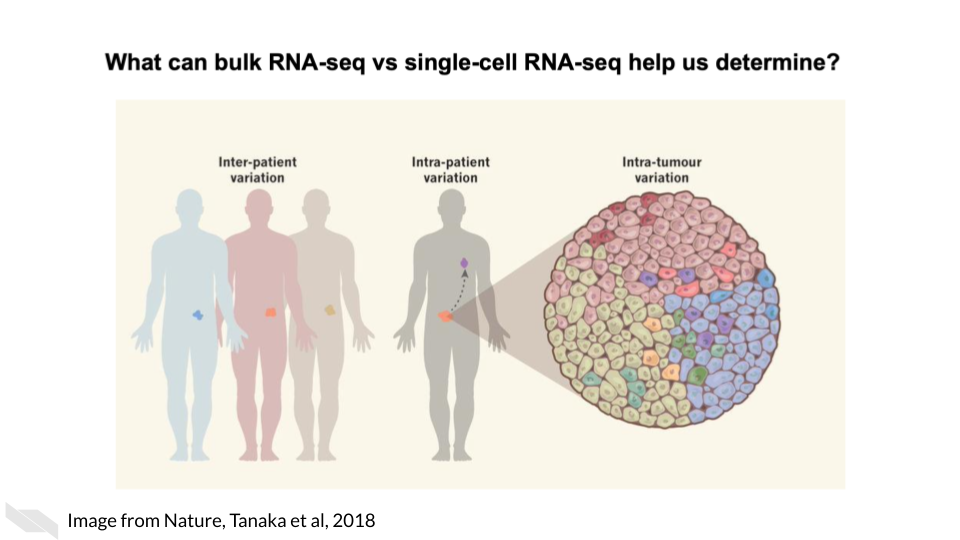
# 13 Single-cell RNA-seq

This chapter is incomplete! If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 13.1 Learning Objectives

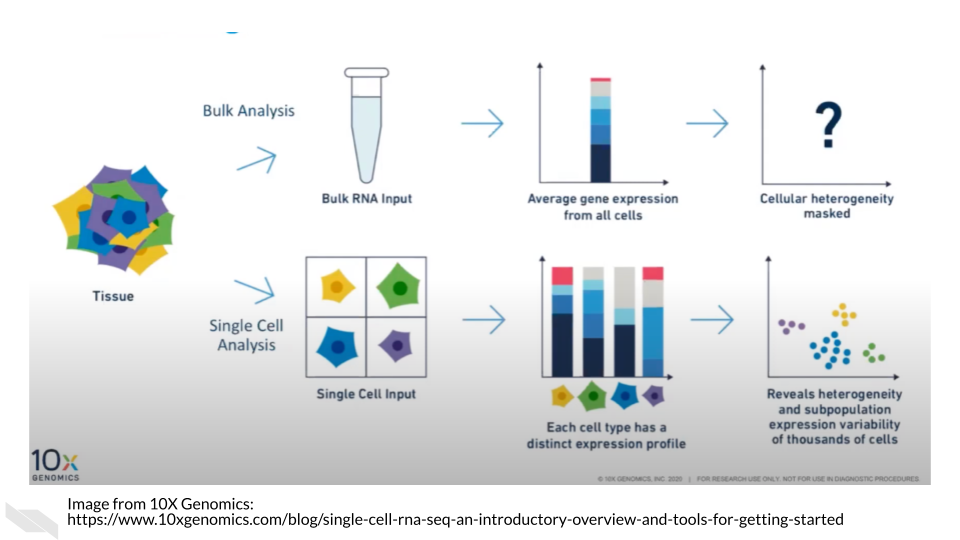


## 13.2 Where single-cell RNA-seq data comes from



As opposed to bulk RNA-seq which can only tell us about tissue level and within patient variation, single-cell RNA-seq is able to tell us cell to cell variation in transcriptomics including intra-tumor heterogeneity.

Single cell RNA-seq can give us cell level transcriptional profiles. Whereas bulk RNA-seq masks cell to cell heterogeneity. If your research questions require cell-level transcriptional information, single-cell RNA-seq will on interest to you.



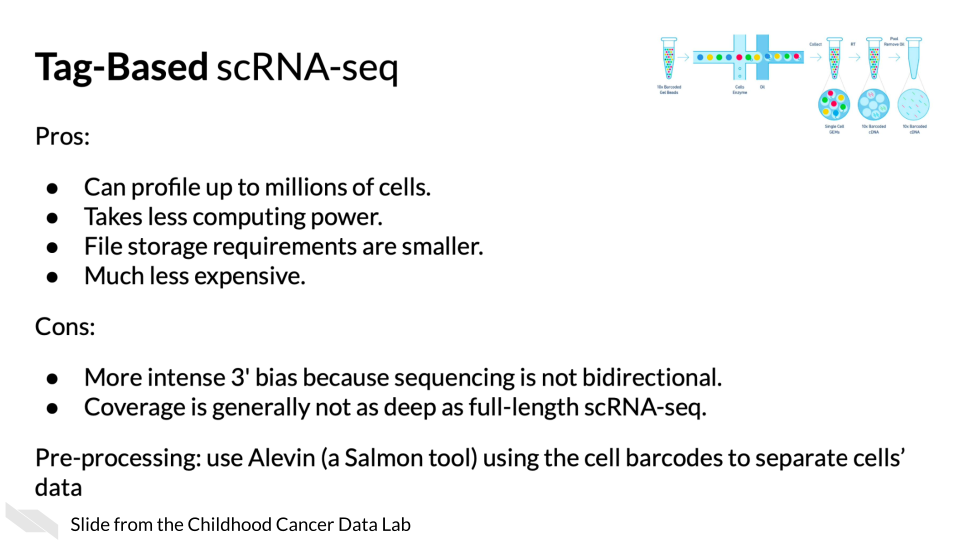
## 13.3 Single-cell RNA-seq data types

There are broadly two categories of single-cell RNA-seq data methods we will discuss.

* **Full length RNA-seq**: Individual cells are physically separated and then sequenced.
* **Tag Based RNA-seq**: Individual cells are tagged with a barcode and their data is separated computationally.

Depending on your goals for your single cell RNA-seq analysis, you may want to choose one method over the other.

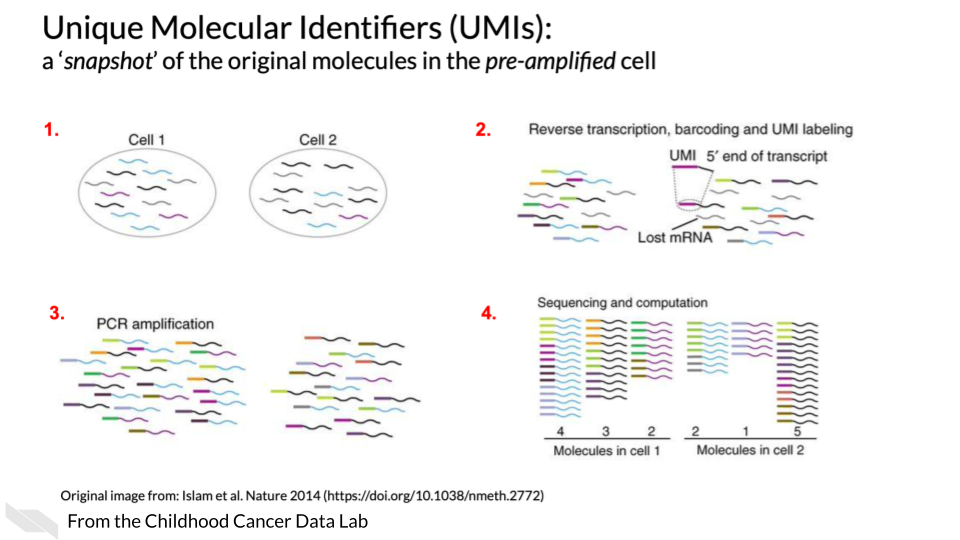




(Material borrowed from ([“Alex’s Lemonade Training Modules” 2022](#ref-AlexsLemonade2022))).

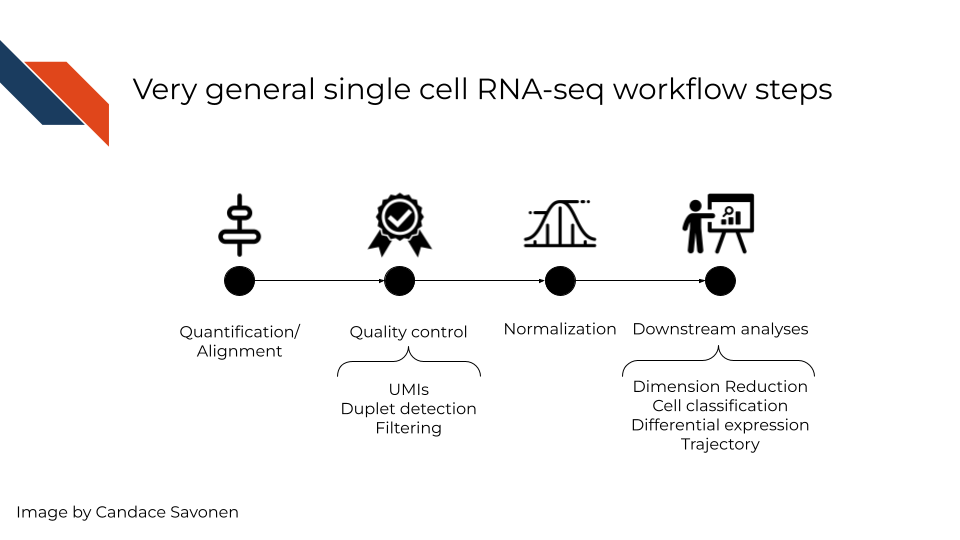
### 13.3.1 Unique Molecular identifiers

Often Tag based single cell RNA-seq methods will include not only a cell barcode for cell identification but will also have a unique molecular identifier (UMI) for original molecule identification. The idea behind the UMIs is it is a way to have insight into the original snapshot of the cell and potentially combat PCR amplification biases.



## 13.4 Single cell RNA-seq tools

There are a lot of scRNA-seq tools for various steps along the way.



In a very general sense, single cell RNA-seq workflows involves first quantification/alignment. You will also need to conduct quality control steps that may involve using UMIs to check for what’s detected, detecting duplets, and using this information to filter out data that is not trustworthy. After you have a set of reliable data, you need to normalize your data. Single cell data is highly skewed - a lot of genes barely or not detected and a few genes that are detected a lot. After data has been normalized you are ready to conduct your downstream analyses. This will be highly dependent on the original goals and questions of your experiment. It may include dimension reduction, cell classification, differential expression, detecting cell trajectories or any number of other analyses.

Each step of this very general representation of a workflow can be conducted by a variety of tools. We will highlight some of the more popular tools here. But, to look through a full list, you can consult the [scRNA-tools website](https://www.scrna-tools.org/table).

## 13.5 Quantification and alignment tools

This following pros and cons sections have been written by AI and may need verification by experts. This is meant to give you a basic idea of the pros and cons of these tools but should ultimately be used with your own judgment.

* [STAR](https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html):
  + **Pros**: Accurate alignment of RNA-seq reads to the genome. Can handle a wide range of RNA-seq protocols, including scRNA-seq. Provides read counts and gene-level expression values.
  + **Cons**: Requires a significant amount of memory and computational resources. May be difficult to set up and run for beginners.
* [HISAT2](http://daehwankimlab.github.io/hisat2/):
  + **Pros**: Accurate alignment of RNA-seq reads to the genome. Provides transcript-level expression values. Supports splice-aware alignment.
  + **Cons**: May require significant computational resources for large datasets. May not be as accurate as some other alignment tools.

This following pros and cons sections have been written by AI and may need verification by experts. This is meant to give you a basic idea of the pros and cons of these tools but should ultimately be used with your own judgment.

* [STAR](https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html) ([Dobin et al. 2013](#ref-dobin2013star)):
  + **Pros**: Accurate alignment of RNA-seq reads to the genome. Can handle a wide range of RNA-seq protocols, including scRNA-seq. Provides read counts and gene-level expression values.
  + **Cons**: Requires a significant amount of memory and computational resources. May be difficult to set up and run for beginners.
* [HISAT2](http://daehwankimlab.github.io/hisat2/) ([Kim, Langmead, and Salzberg 2015](#ref-kim2015hisat)):
  + **Pros**: Accurate alignment of RNA-seq reads to the genome. Provides transcript-level expression values. Supports splice-aware alignment.
  + **Cons**: May require significant computational resources for large datasets. May not be as accurate as some other alignment tools.
* [Kallisto bustools](https://www.kallistobus.tools/) ([Bray et al. 2016](#ref-bray2016near)):
  + **Pros**: Fast and accurate quantification of RNA-seq reads without the need for alignment. Provides transcript-level expression values. Requires less memory and computational resources than alignment-based methods.
  + **Cons**: May not be as accurate as alignment-based methods for lowly expressed genes. Cannot provide allele-specific expression estimates.

[Alevin/Salmon](https://salmon.readthedocs.io/en/latest/alevin.html) ([Patro et al. 2017](#ref-patro2017salmon)): - **Pros**: Fast and accurate quantification of RNA-seq reads without the need for alignment. Provides transcript-level expression values. Supports both single-end and paired-end sequencing. - **Cons**: May not be as accurate as alignment-based methods for lowly expressed genes. Cannot provide allele-specific expression estimates.

* [Cell Ranger](https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger) ([Zheng et al. 2017](#ref-zheng2017massively)):
  + **Pros**: Specifically designed for 10x Genomics scRNA-seq data, with optimized workflows for alignment and quantification. Provides read counts and gene-level expression values. Offers a streamlined pipeline with minimal input from the user.
  + **Cons**: Limited options for customizing parameters or analysis methods. May not be suitable for datasets from other scRNA-seq platforms.

## 13.6 Downstream tools Pros and Cons

* [Seurat](https://satijalab.org/seurat/):
  + **Pros**: Has a wide range of functionalities for preprocessing, clustering, differential expression, and visualization. Can handle multiple modalities, including CITE-seq and ATAC-seq. Has a large and active user community, with extensive documentation and tutorials available.
  + **Cons**: Can be computationally intensive, especially for large datasets. Requires some knowledge of R programming language.
* [Scanpy](https://scanpy.readthedocs.io/en/stable/):
  + **Pros**: Written in Python, a widely used programming language in bioinformatics. Has a user-friendly interface and extensive documentation. Offers a variety of preprocessing, clustering, and differential expression methods, as well as interactive visualizations.
  + **Cons**: May not be as feature-rich as some other tools, such as Seurat. Does not yet support multiple modalities.
* [Monocle](https://cole-trapnell-lab.github.io/monocle3/):
  + **Pros**:Focuses on trajectory analysis, allowing users to explore developmental trajectories and cell fate decisions. Has a user-friendly interface and extensive documentation. Can handle data from multiple platforms, including Smart-seq2 and Drop-seq.
  + **Cons**: May not be as feature-rich for clustering or differential expression analysis as some other tools. Requires some knowledge of R programming language.
* [Monocle](https://cole-trapnell-lab.github.io/monocle3/):
  + **Pros**:Focuses on trajectory analysis, allowing users to explore developmental trajectories and cell fate decisions. Has a user-friendly interface and extensive documentation. Can handle data from multiple platforms, including Smart-seq2 and Drop-seq.
  + **Cons**: May not be as feature-rich for clustering or differential expression analysis as some other tools. Requires some knowledge of R programming language.

### 13.6.1 Doublet Tool Pros and Cons

* [DoubletFinder](https://github.com/chris-mcginnis-ucsf/DoubletFinder)([McGinnis, Murrow, and Gartner 2020](#ref-mcginnis2020doubletfinder)):
  + **Pros**: Uses a machine learning approach to detect doublets based on transcriptome similarity. Can be used with a variety of scRNA-seq platforms. Offers a user-friendly interface and extensive documentation.
  + **Cons**: Can be computationally intensive for large datasets. May require some knowledge of R programming language.
* [Scrublet](https://github.com/swolock/scrublet) ([Wolock, Krishnaswamy, and Huang 2019](#ref-wolock2019scrublet)):
  + **Pros**: Uses a density-based approach to detect doublets based on barcode sharing. Fast and computationally efficient, making it suitable for large datasets. Offers a user-friendly interface and extensive documentation.
  + **Cons**:May not be as accurate as other methods, especially for low-quality data. Limited to 10x Genomics data.
* [DoubletDecon](https://github.com/EDePasquale/DoubletDecon) ([De Pasquale and Dudoit 2019](#ref-de2019doubletdecon)):
  + **Pros**: Uses a statistical approach to identify doublets based on the distribution of the number of unique molecular identifiers (UMIs) per cell. Can be used with different platforms and species. Offers a user-friendly interface and extensive documentation.
  + **Cons**: May not be as accurate as other methods, especially for data with low sequencing depth or low cell numbers. Requires some knowledge of R programming language.

It’s important to note that no doublet detection method is perfect, and it’s often a good idea to combine multiple methods to increase the accuracy of doublet identification. Additionally, manual inspection of the data is always recommended to confirm the presence or absence of doublets.

## 13.7 More scRNA-seq tools and tutorials

* [AlevinQC](https://bioconductor.org/packages/release/bioc/html/alevinQC.html)
* [Gene Pattern’s single cell RNA-seq tutorials](https://notebook.genepattern.org/single-cell/) - an open software environment providing access to hundreds of tools for the analysis and visualization of genomic data.
* [Single Cell Genome Viewer](https://github.com/KrasnitzLab/SCGV)
* For normalization [scater](https://bioconductor.org/packages/devel/bioc/vignettes/scater/inst/doc/overview.html)
* [TumorDecon](https://people.math.umass.edu/~aronow/TumorDecon) can be used to generate customized signature matrices from single-cell RNA-sequence profiles. It is available on Github (<https://github.com/ShahriyariLab/TumorDecon>) and PyPI (<https://pypi.org/project/TumorDecon/>).

## 13.8 Visualization GUI tools

* [WebMeV](https://webmev.tm4.org) uniquely provides a user-friendly, intuitive, interactive interface to processed analytical data uses cloud-computing elasticity for computationally intensive analyses and is compatible with single cell or bulk RNA-seq input data.
* [UCSC Xena](http://xena.ucsc.edu/) is a web-based visualization tool for multi-omic data and associated clinical and phenotypic annotations. It can be used with single cell RNA-seq data.
* [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a track-based browser for interactively exploring genomic data mapped to a reference genome.

## 13.9 Useful tutorials

These tutorials cover explicit steps, code, tool recommendations and other considerations for analyzing RNA-seq data.

* [Orchestrating Single Cell Analysis with Bioconductor](http://bioconductor.org/books/3.15/OSCA.intro/) - An excellent tutorial for processing single cell data using Bioconductor.
* [Advanced Single Cell Analysis with Bioconductor](http://bioconductor.org/books/3.15/OSCA.advanced/) - a companion book to the intro version that contains code examples.
* [Alex’s Lemonade scRNA-seq Training module](https://alexslemonade.github.io/training-modules/scRNA-seq/) - A cancer based workshop module based in R, with exercise notebooks.
* [Sanger Single Cell Course](https://www.singlecellcourse.org/) - a general tutorial based on using R.
* [ASAP: Automated Single-cell Analysis Pipeline](https://asap.epfl.ch/) is a web server that allows you to process scRNA-seq data.
* [Processing raw 10X Genomics single-cell RNA-seq data (with cellranger)](https://swaruplab.bio.uci.edu/tutorial/cellranger/cellranger-rna.html) - a tutorial based on using CellRanger.

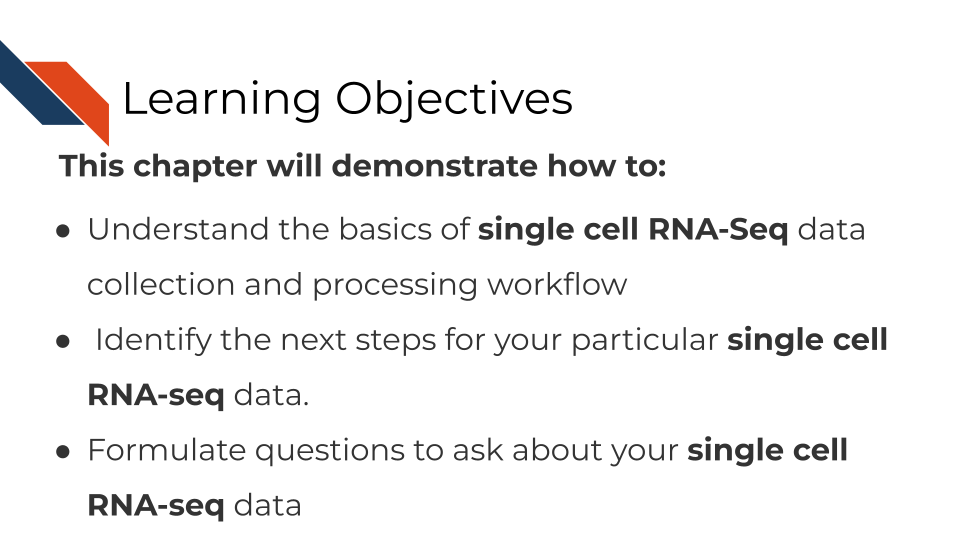
## 13.10 Useful readings

* [An Introduction to the Analysis of Single-Cell RNA-Sequencing Data](https://doi.org/10.1016/j.omtm.2018.07.003) ([**AlJanahi2018?**](#ref-AlJanahi2018)).
* [Orchestrating single-cell analysis with Bioconductor](https://www.nature.com/articles/s41592-019-0654-x) ([**Amezquita2019?**](#ref-Amezquita2019)).
* [UMIs the problem, the solution and the proof](https://cgatoxford.wordpress.com/2015/08/14/unique-molecular-identifiers-the-problem-the-solution-and-the-proof/) ([Smith 2015](#ref-Smith2015)).
* [Experimental design for single-cell RNA sequencing](https://doi.org/10.1093/bfgp/elx035) ([Baran-Gale, Chandra, and Kirschner 2018](#ref-BaranGale2018)).
* [Tutorial: guidelines for the experimental design of single-cell RNA sequencing studies](https://doi.org/10.1038/s41596-018-0073-y) ([**Lafzi2019?**](#ref-Lafzi2019)).
* [Comparative Analysis of Single-Cell RNA Sequencing Methods](http://dx.doi.org/10.1016/j.molcel.2017.01.023) ([**Ziegenhain2018?**](#ref-Ziegenhain2018)).
* [Comparative Analysis of Droplet-Based Ultra-High-Throughput Single-Cell RNA-Seq Systems](https://doi.org/10.1016/j.molcel.2018.10.020) ([**Zhang2018?**](#ref-Zhang2018)).
* [Single cells make big data: New challenges and opportunities in transcriptomics](http://dx.doi.org/10.1016/j.coisb.2017.07.004) ([Angerer et al. 2017](#ref-Angerer2017)).
* [Comparative Analysis of common alignment tools for single cell RNA sequencing](https://www.biorxiv.org/content/10.1101/2021.02.15.430948v2) ([Brüning et al. 2021](#ref-Bruning2021)).
* [Current best practices in single-cell RNA-seq analysis: a tutorial](https://doi.org/10.15252/msb.20188746) ([Luecken and Theis 2019](#ref-Luecken2019)).

# 14 Spatial transcriptomics

This chapter chapter has currently been written by ChatGPT and has not been verified by experts. We need help writing and reviewing it! If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 14.1 Learning Objectives



## 14.2 What are the goals of spatial transcriptomic analysis?

Characterize Spatial Gene Expression Patterns: One primary goal of spatial transcriptomic analysis is to characterize gene expression patterns within a tissue or cellular context. By examining gene expression profiles at different locations within the tissue, researchers aim to identify spatially regulated gene expression patterns and understand how gene expression varies across different regions.

1. **Identify Cell Types and Subpopulations:** Spatial transcriptomic analysis helps in the identification and characterization of different cell types and subpopulations within a tissue sample. By examining the gene expression profiles at a spatial resolution, researchers can differentiate and define distinct cell types based on their unique transcriptional signatures. This information contributes to the understanding of cellular heterogeneity and cellular composition within the tissue.
2. **Uncover Spatially Regulated Biological Processes:** Another goal of spatial transcriptomic analysis is to uncover spatially regulated biological processes. By examining gene expression patterns within the tissue context, researchers can identify spatially restricted expression of genes involved in specific biological processes, such as tissue gradients, morphogenesis, cell differentiation, or signaling pathways. This knowledge aids in understanding how spatial organization influences cellular behavior and tissue function.
3. **Investigate Cell-Cell Interactions and Signaling Networks:** Spatial transcriptomic analysis enables the investigation of cell-cell interactions and signaling networks within a tissue. By examining the gene expression profiles of neighboring cells, researchers can identify genes that are specifically expressed in close proximity, indicating potential cell-cell interactions and communication. This information helps in deciphering cellular communication networks and understanding the functional relationships between cells.
4. **Integrate Spatial Transcriptomics with Anatomical or Imaging Data:** Spatial transcriptomic analysis aims to integrate gene expression data with anatomical or imaging data, such as histological images or imaging modalities like fluorescence microscopy. By overlaying gene expression information onto the tissue architecture, researchers can correlate gene expression patterns with tissue morphology, cellular structures, or specific anatomical regions. This integration enhances the interpretation of spatial transcriptomic data and provides a more comprehensive understanding of gene expression within the tissue.
5. **Generate Hypotheses and Identify Biomarkers:** Spatial transcriptomic analysis can generate hypotheses and identify potential biomarkers associated with specific tissue functions or disease states. By examining gene expression patterns in the spatial context, researchers can discover spatially restricted gene signatures and potential diagnostic or prognostic markers. This knowledge has implications for disease research, precision medicine, and therapeutic targeting.

## 14.3 Spatial transcriptomic general workflow overview

Here are the basic steps involved in collecting spatial transcriptomic data:

1. **Sample Preparation:** The first step is to prepare the tissue sample for spatial transcriptomic analysis. This involves obtaining the tissue of interest, which could be a section from a biopsy, a thin slice from a tissue block, or a whole-mount tissue. The sample should be properly preserved and stabilized to maintain the integrity of RNA molecules.
2. **Spatially Resolved Capture of RNA Molecules:** In this step, the tissue sample is typically placed on a solid substrate, such as a glass slide or a microarray chip. One common approach is to use spatially barcoded arrays or grids, where each region corresponds to a specific spatial location. Alternatively, techniques like in situ sequencing or imaging-based methods can be used. These approaches allow the capture of RNA molecules while retaining spatial information.
3. **RNA Extraction and Amplification:** After spatial capture, the captured RNA molecules need to be extracted and amplified to generate sufficient material for downstream analysis. This step involves lysing the cells, extracting the RNA, and performing reverse transcription to convert RNA into complementary DNA (cDNA). Amplification methods such as PCR or in vitro transcription (IVT) are employed to generate amplified cDNA libraries.
4. **Library Preparation:** The amplified cDNA libraries are then prepared for sequencing. This involves adding specific sequencing adapters and barcodes to each library, which enable sample multiplexing and identification during sequencing.
5. **Sequencing:** The prepared libraries are subjected to high-throughput sequencing using platforms like Illumina or Ion Torrent. The choice of sequencing platform depends on the desired read length, throughput, and cost considerations.
6. **Data Analysis:** After sequencing, the generated data undergoes bioinformatics analysis. This typically includes preprocessing steps such as read alignment or mapping to a reference genome, removing duplicate reads, and quantifying gene expression levels. The spatial coordinates associated with each read are also considered for spatial analysis, which involves assigning gene expression profiles to specific locations within the tissue.
7. **Visualization and Interpretation:** The final step involves visualizing and interpreting the spatial transcriptomic data. Various tools and algorithms are available for spatial visualization, such as generating heatmaps, spatial expression plots, or clustering analysis. The data can be integrated with anatomical or imaging data to gain further insights into cellular organization and interactions within the tissue.

## 14.4 Spatial transcriptomic data **strengths**:

* **Preservation of Spatial Information:** Spatial transcriptomics allows the investigation of gene expression patterns within the context of tissue or cellular spatial organization. By preserving the spatial information of RNA molecules, it provides insights into how gene expression varies across different regions within a tissue sample.
* **Identification of Cell Types and Subpopulations:** Spatial transcriptomics enables the identification and characterization of different cell types and subpopulations within a tissue sample based on their unique gene expression profiles. This information helps in understanding cellular heterogeneity and cell-cell interactions within the tissue.
* **Visualization of Gene Expression Patterns:** By generating spatially resolved gene expression data, spatial transcriptomics allows for the visualization of gene expression patterns across the tissue sample. This visualization can be in the form of heatmaps, spatial expression plots, or spatially resolved gene expression atlases, providing a comprehensive view of gene expression within the tissue.
* **Integration with Anatomical and Imaging Data:** Spatial transcriptomics data can be integrated with anatomical or imaging data, such as histological images or imaging modalities like fluorescence microscopy. This integration enhances the interpretation of spatial transcriptomic data by correlating gene expression patterns with tissue morphology or specific cellular structures.
* **Discovery of Novel Cell-Cell Interactions and Signaling Pathways:** By examining gene expression profiles in the spatial context, spatial transcriptomics can uncover novel cell-cell interactions and signaling pathways within a tissue. It allows the identification of genes that are specifically expressed in close proximity to each other, revealing potential regulatory relationships and functional interactions between cells.
* **Exploration of Spatially Regulated Biological Processes:** Spatial transcriptomics enables the investigation of spatially regulated biological processes, such as tissue gradients or developmental processes occurring in specific regions. It provides insights into spatially restricted gene expression patterns associated with tissue patterning, morphogenesis, or cellular differentiation.
* **Hypothesis Generation and Biomarker Discovery:** Spatial transcriptomic analysis can generate hypotheses and identify potential biomarkers related to specific tissue functions or disease states. By linking gene expression patterns to tissue organization and pathology, spatial transcriptomics facilitates the discovery of spatially restricted gene signatures and potential diagnostic or prognostic markers.

## 14.5 Spatial transcriptomic data **weaknesses**:

* **Limited Spatial Resolution:** Spatial transcriptomic techniques have a finite spatial resolution, which determines the level of detail at which gene expression patterns can be analyzed. The resolution is influenced by factors such as the size of capture spots or the resolution of imaging techniques. Obtaining fine-grained spatial information may be challenging, especially in complex tissues or samples with high cellular density.
* **Technical Variability and Experimental Artifacts:** Spatial transcriptomic analysis involves multiple experimental steps, including tissue processing, capture techniques, and sequencing. Each step introduces technical variability and potential experimental artifacts, which can impact the accuracy and reproducibility of the results. Controlling and minimizing these sources of variation is crucial but can be challenging.
* **Limited Coverage of Transcripts:** Spatial transcriptomic methods often have limited coverage of the entire transcriptome. The capture techniques used may bias the representation of certain RNA molecules or favor highly expressed transcripts, potentially leading to a biased view of gene expression. Additionally, spatial transcriptomic methods may have limitations in capturing non-polyadenylated RNAs or low-abundance transcripts.
* **Complex Data Analysis:** Analyzing spatial transcriptomic data requires advanced computational methods and expertise. Data analysis involves multiple steps, including image processing, spatial registration, normalization, and interpretation. The complexity of the data and the need for specialized bioinformatics tools and pipelines can pose challenges, particularly for researchers without extensive computational skills.
* **Validation and Integration Challenges:** Spatial transcriptomic analysis generates hypotheses and provides spatially resolved gene expression information. However, validating the functional significance of identified gene expression patterns or cellular interactions may require additional experiments or techniques. Integrating spatial transcriptomic data with other omics data or imaging modalities can also be complex and may require careful data integration strategies.
* **Cost and Time Considerations:** Spatial transcriptomic analysis can be relatively expensive and time-consuming compared to traditional transcriptomic techniques. The specialized protocols, reagents, and instrumentation required can add to the cost of the analysis. Moreover, the data generation and analysis processes can be time-intensive, which may limit the scalability of studies involving large sample sizes.
* **Limited Accessibility of Techniques:** Some spatial transcriptomic techniques may have limited availability, accessibility, or may require specialized equipment or expertise. This can restrict their widespread use and accessibility to researchers who do not have access to the necessary resources or infrastructure.

### 14.5.1 Tools for spatial transcriptomics

#### 14.5.1.1 Seurat:

* **Pros:** Seurat is a widely used R package that offers a comprehensive suite of tools for spatial transcriptomic analysis. It provides a variety of functions for data preprocessing, dimensionality reduction, clustering, and visualization. Seurat has a large user community, extensive documentation, and tutorials, making it accessible to researchers.
* **Cons:** Seurat can be memory-intensive, particularly when working with large datasets. It may require familiarity with R programming and bioinformatics concepts for effective use. Additionally, Seurat’s performance can depend on the specific computational resources available.

#### 14.5.1.2 Scanpy:

* **Pros:** Scanpy is a Python-based library specifically designed for single-cell and spatial transcriptomic analysis. It offers a range of functionalities for data preprocessing, clustering, trajectory analysis, and visualization. Scanpy is known for its scalability, efficiency, and flexibility. It integrates well with other Python libraries and frameworks, making it suitable for integration with other analysis pipelines.
* **Cons:** Similar to Seurat, Scanpy may require some familiarity with Python programming and bioinformatics concepts. Users without prior programming experience may need to invest time in learning Python.

#### 14.5.1.3 spaceranger:

* **Pros:** spaceranger is a software package developed by 10x Genomics specifically for processing and analyzing spatial transcriptomic data generated by their platform. It provides a streamlined workflow for processing raw data, including image processing, spot calling, and counting transcripts. spaceranger offers integration with the Seurat R package, enabling downstream analysis using Seurat’s extensive tools.
* **Cons:** spaceranger is optimized for 10x Genomics data and may have limited compatibility with other spatial transcriptomic platforms. It may not offer the same level of flexibility as more general-purpose tools like Seurat or Scanpy.

#### 14.5.1.4 ST Pipeline:

* **Pros:** ST Pipeline is an open-source bioinformatics pipeline developed by the Spatial Transcriptomics consortium. It provides a complete workflow for spatial transcriptomic data analysis, including preprocessing, normalization, spot detection, and visualization. ST Pipeline supports various spatial transcriptomic platforms, making it versatile.
* **Cons:** ST Pipeline may require some command-line proficiency and familiarity with Linux environments. Users may need to invest time in setting up the pipeline and configuring parameters based on their specific datasets and platforms.

#### 14.5.1.5 SpatialDE:

* **Pros:** SpatialDE is a Python package designed for detecting spatially variable genes from spatial transcriptomic data. It offers statistical methods to identify genes that exhibit spatial heterogeneity in expression across the tissue. SpatialDE provides a straightforward implementation for identifying spatially variable genes and their spatial patterns.
* **Cons:** SpatialDE focuses on the identification of spatially variable genes and may not provide a comprehensive set of tools for the entire spatial transcriptomic analysis pipeline. Additional tools may be required for preprocessing, visualization, and downstream analysis.

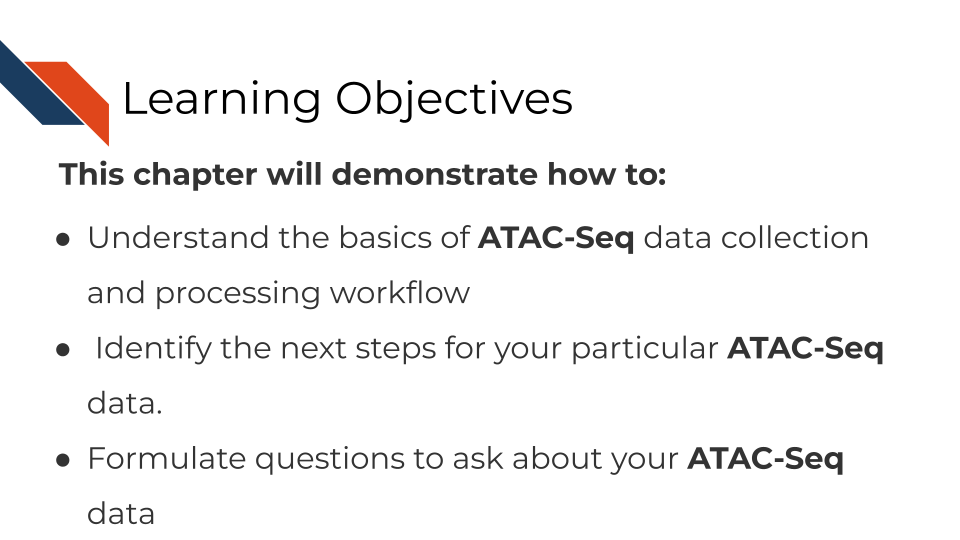
## 14.6 More tools and tutorials regarding spatial transcriptomics

* [Analysis, visualization, and integration of spatial datasets with Seurat](https://satijalab.org/seurat/articles/spatial_vignette.html)
* [Sheffield Bioinformatics tutorial for spatial transcriptomics](https://github.com/sheffield-bioinformatics-core/spatial_transcriptomics_tutorial)
* [Theis Lab SCOG workshop materials for spatial transcriptomics](https://github.com/theislab/spatial_scog_workshop_2022)

# 15 ATAC-Seq

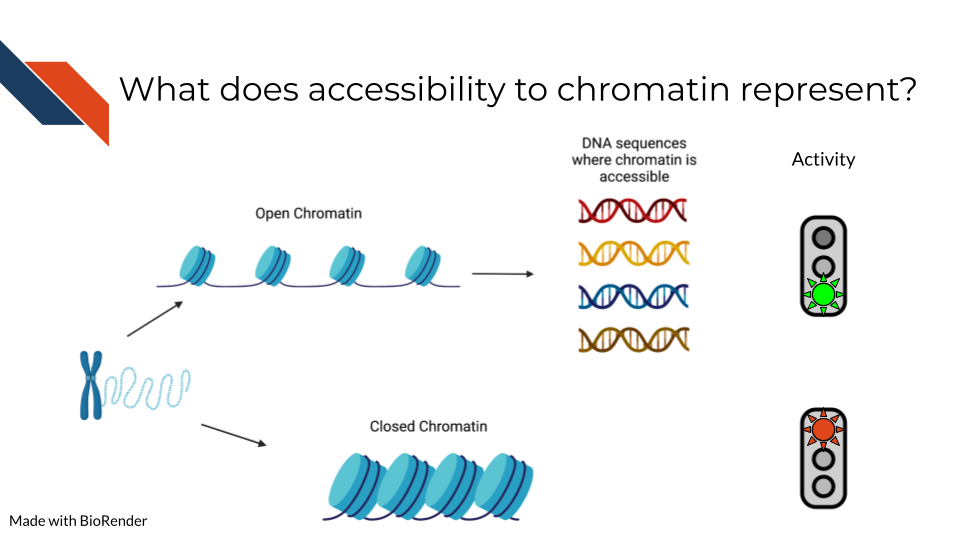
This chapter is incomplete! If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 15.1 Learning Objectives

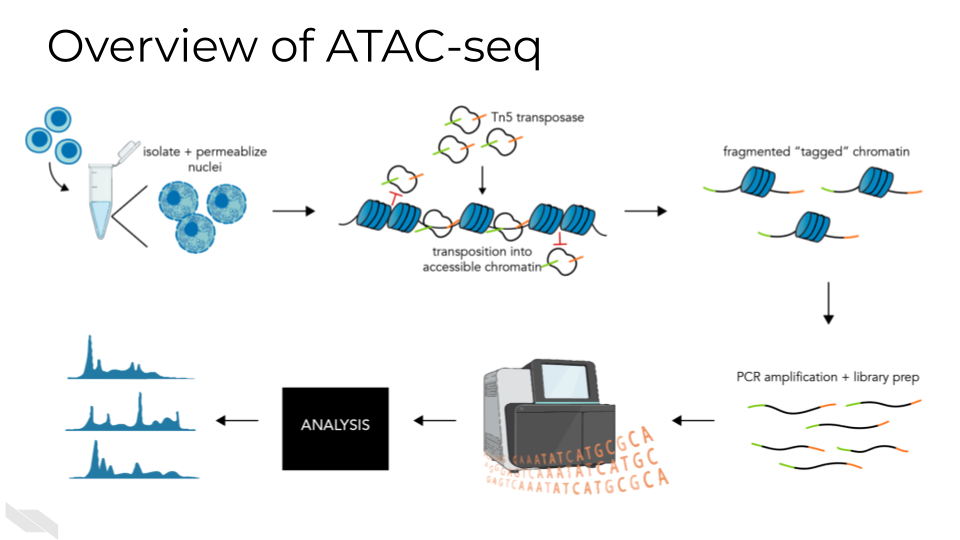


## 15.2 What are the goals of ATAC-Seq analysis?

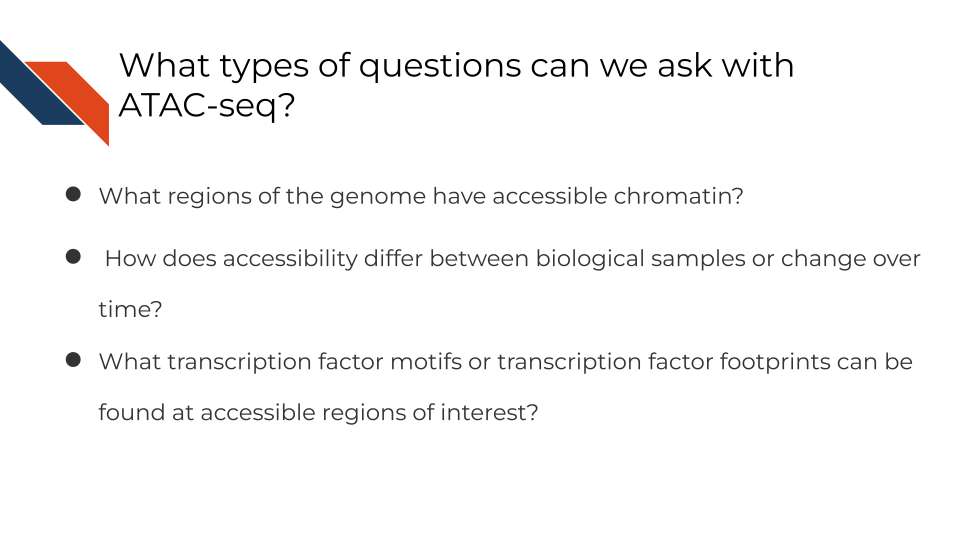
The goals of ATAC-seq are to identify the accessible regions of the genome in a particular set of samples. These data allow us to understand the relationships between the chromatin accessibility patterns and cell states, and to understand the mechanistic causes and consequences of these chromatin accessibility patterns.



ATAC-seq data is generated by fragmenting the genome with the Tn5 endonuclease and sequencing the shorter DNA fragments. While most of the genome is associated with protein complexes that preclude the digestion of DNA by Tn5, some regions of the genome have accessible chromatin that can be cleaved by Tn5 resulting in short (<500bp) fragments. These regions of the genome are of biological interest as they are likely to harbor transcription factor binding sites and to constitute cis-regulatory elements, genomic regions that are involved in the regulation of gene expression.

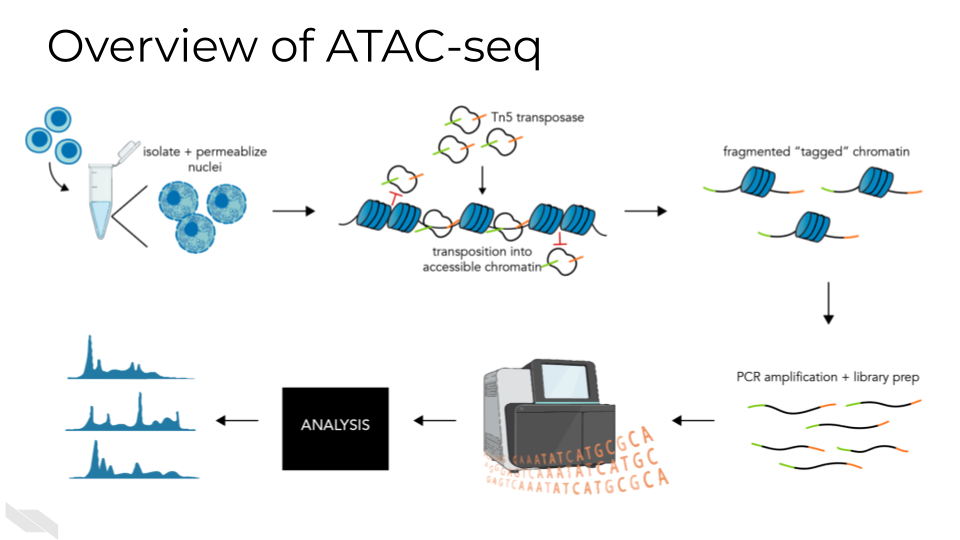


### 15.2.1 What questions can be answered with ATAC-seq?



## 15.3 ATAC-Seq general workflow overview

A basic ATAC-seq workflow involves mapping sequence reads to the genome, identifying peaks, assessing data quality, and identifying patterns of interest through clustering or identification of differentially accessible regions or other statistical means.



### 15.3.1 Data quality metrics:

#### 15.3.1.1 Pre-sequencing QC:

#### 15.3.1.2 Sequencing considerations:

{r, fig.alt = "Single end sequencing. Cheaper. OK for most standard applications. Paired-end sequencing. More expensive. Useful for looking at nucleosome positioning and transcription factor footprinting ", out.width = "100%", echo = FALSE} ottrpal::include\_slide("https://docs.google.com/presentation/d/1YwxXy2rnUgbx\_7B7ENH9wpDX-j6JpJz6lGVzOkjo0qY/edit#slide=id.g1567d947351\_0\_57")

{r, fig.alt = "Single end sequencing. Cheaper. OK for most standard applications. Paired-end sequencing. More expensive. Useful for looking at nucleosome positioning and transcription factor footprinting ", out.width = "100%", echo = FALSE} ottrpal::include\_slide("https://docs.google.com/presentation/d/1YwxXy2rnUgbx\_7B7ENH9wpDX-j6JpJz6lGVzOkjo0qY/edit#slide=id.g1567d947351\_0\_63")

#### 15.3.1.3 Pre-alignment QC:

{r, fig.alt = "Post-sequencing. Signal to noise ratio (link resources at end) Comparison with DNase hypersensitivity datasets (or other computational QC method- check current resources available)", out.width = "100%", echo = FALSE} ottrpal::include\_slide("https://docs.google.com/presentation/d/1YwxXy2rnUgbx\_7B7ENH9wpDX-j6JpJz6lGVzOkjo0qY/edit#slide=id.g1567d947351\_0\_50")

A tool like FastQC or similar should be used to check for GC content, read quality and length, and primer or adapter reads prior to alignment. Trimmomatic is a useful tool for removing primer and adapter sequences if they are present. ATAC-seq experiments should be sequenced with paired-end sequencing, and existing pipelines will expect paired-end. (2 files \*\_R1.fastq and \*\_R2.fastq)

* Use fasterq-dump to download files from NCBI Sequence Read Archive - this tool will automatically split the reads in multiple files

#### 15.3.1.4 Number of mapped reads

As for all DNA-sequencing based genomics technologies, a sufficient number of mapped reads is required to obtain meaningful results from a sample. You can read more about [general sequencing technologies in our previous chapter here](http://hutchdatascience.org/Choosing_Genomics_Tools/sequencing-data.html). For experiments on human samples this number should be greater than 20 million mapped unique reads.

[Bowtie2](https://bowtie-bio.sourceforge.net/index.shtml) is commonly used for mapping fragments to the genome.

As for all DNA-sequencing based genomics technologies, a sufficient number of mapped reads is required to obtain meaningful results from a sample. You can read more about [general sequencing technologies in our previous chapter here](http://hutchdatascience.org/Choosing_Genomics_Tools/sequencing-data.html). For experiments on human samples this number should be greater than 20 million mapped unique reads.

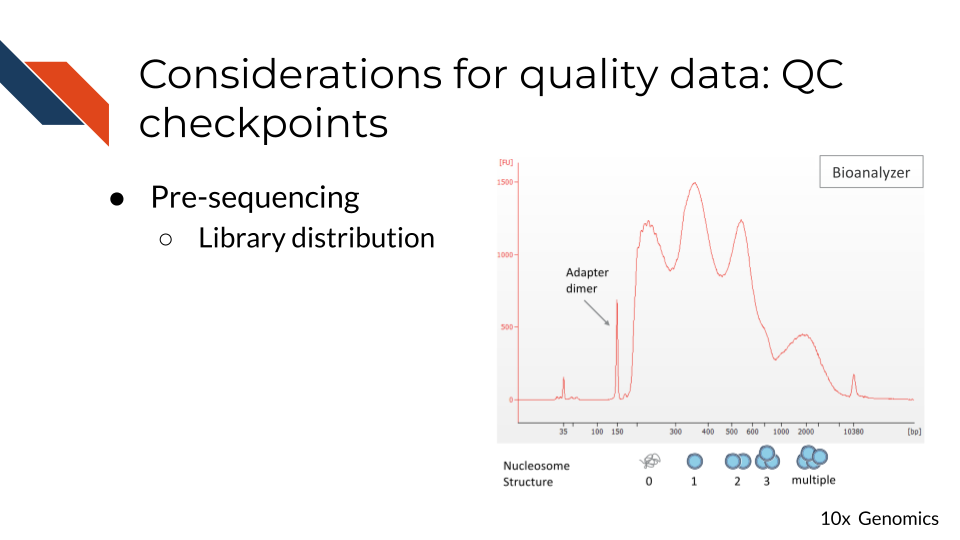
#### 15.3.1.5 Post-alignment QC:

{r, fig.alt = "Post-sequencing. Signal to noise ratio (link resources at end) Comparison with DNase hypersensitivity datasets (or other computational QC method- check current resources available)", out.width = "100%", echo = FALSE} ottrpal::include\_slide("https://docs.google.com/presentation/d/1YwxXy2rnUgbx\_7B7ENH9wpDX-j6JpJz6lGVzOkjo0qY/edit#slide=id.g1567d947351\_0\_50")

Post alignment: check percent of matched, unmatched, unpaired and duplicated reads. Reads which are duplicated or unmatched should be filtered out. [Picard](https://broadinstitute.github.io/picard/) is a useful tool for this step. Reads on the + strand should be shifted +4bp, reads on the - strand should be shifted -5 bp.

#### 15.3.1.6 Fragment size distribution:

ATAC-seq data is often generated using paired end sequencing technologies, which allow for characterization of ATAC-seq fragments. Histograms of these distributions using single base pair resolution bins reveal patterns of enrichment relative to the nucleosome scale of 147bp and the DNA-helix scale ~10.5bp.



When comparing ATAC-seq samples, it is important to consider the fragment size distributions of the samples being compared. Differences in the distributions could lead to results that are unrelated to biology.

#### 15.3.1.7 Peak calling:

ATAC-seq peak calling typically makes use of analysis tools developed for ChIP-seq. MACS2 is one of the most common choices for a peak calling tool, but HOMER or other common ChIP-seq peak callers are also acceptable. An input sample is not typically generated for ATAC-seq as it would be for a ChIP-seq experiment, so the major requirement for the peak caller is that it does not require the input control to call peaks.

 #### Number of peaks:

Although the number of accessible chromatin regions can vary from one cell type to another, there are several regions that appear to be constitutively accessible across most cell types. At least 20,000 peaks can be identified in a high quality experiment. The deeper the sequencing the more peaks will be detected in an ATAC-seq experiments. At a very high sequencing depth some of the statistically significant peaks might not be of biological interest. In an analysis of such data sets the fold enrichment relative to background, or absolute peak signal, in addition to statistical significance, ought to be taken into account.

#### 15.3.1.8 FRiP score (fraction of reads in peaks)

In high quality ATAC-seq data a large fraction of reads overlap with peaks, while in low quality data there is a high level of fragments that map to background regions. Ideally, the FRiP score is greater than 0.3 (30 percent or more of reads overlap with peaks), with a score below 0.2 indicating low-quality data

#### 15.3.1.9 Overlap with other chromatin accessibility data

Thousands of ATAC-seq samples have been produced in human and mouse. High quality ATAC-seq data will share a substantial proportion of peaks with many of these datasets. Publicly available ATAC-seq data can be found and comparisons made at the Cistrome Data Browser [<http://cistrome.org/db/>].

#### 15.3.1.10 Overlap with promoters

The promoter regions of many genes are constitutively accessible. Examining peak overlap with regions close to known protein coding gene transcription start sites can be used as a check for data quality.

### 15.3.2 Information from ATAC-seq analysis:

#### 15.3.2.1 Major approaches:

* Compare changes in transcription factor motif enrichment in accessible regions between samples
* Compare changes in accessibility of regions (differential accessibility) between samples
* Footprinting - identify regions where insertion is below expected level

#### 15.3.2.2 Differential accessibility analysis:

Differential accessibility analysis typically uses packages for RNA-seq differential expression analysis such as DEseq2, edgeR, or limma.

All three are available as R packages and can be installed using Bioconductor, a bioinformatics package manager for R. Unfortunately, there are no well-established packages for this analysis in other languages such as Python. Differential accessibility analysis is an approach with high potential, but care must be taken in processing and normalizing the data for accurate results.

#### 15.3.2.3 Motif analysis:

Motif analysis in ATAC-seq is more complex than for ChIP-seq because a larger set of TFs are responsible for the emergence of chromatin accessible regions than for the binding sites of a particular TF. Nevertheless, in the analysis of differential ATAC-seq peaks motif analysis can be used to reveal the TFs related to differences between conditions. This type of analysis is most likely to be successful when the ATAC-seq between closely related conditions or cell types is being compared.

[The MEME suite](https://meme-suite.org/meme/) has a variety of tools for motif analysis available in both web and command-line versions.

#### 15.3.2.4 Motif Scanning

Motif scanning is an analysis technique which identifies putative transcription factor binding sites (TFBS) which sufficiently match a given TF motif’s position-weight matrix. PWMscan is a straightforward online tool, but not the best option for high throughput. FIMO is an alternative which can be used either on the web or the command line. This approach will identify all sites within the genome which are likely to bind a single transcription factor.

#### 15.3.2.5 Motif discovery:

Homer or MEME. These tools identify overrepresented sequences within the accessible peaks, regardless of whether they match a previously defined motif. Once the ATAC-seq peaks are determined, the next step is to search for enriched DNA sequence motifs within these regions. This is accomplished by using motif discovery algorithms such as MEME Suite, HOMER, or DREME. These tools scan the ATAC-seq peaks for overrepresented sequence patterns, which may correspond to binding sites for specific transcription factors or other regulatory elements. The motifs discovered can be compared against existing motif databases, such as JASPAR or TRANSFAC, to annotate the potential transcription factor binding sites.

#### 15.3.2.6 Motif Enrichment:

These motif enrichment tools will scan through and identify matches to known motif sequences within accessible sites, and additionally will quantify whether the motif is significantly enriched compared to a control sample (input, uncommon with ATAC-seq) or a shuffled sequence to mimic background.

After identifying the enriched motifs, researchers can perform motif enrichment analysis to determine the significance of these motifs in the ATAC-seq peaks. This is often done using statistical tools like Fisher’s exact test or hypergeometric test, which assess the enrichment of specific motifs compared to their background occurrence in the genome. Additionally, tools like GREAT or HOMER can be employed to perform gene ontology analysis and assess the functional relevance of the identified motifs in biological processes and pathways.

Overall, ATAC-seq motif enrichment analysis provides researchers with valuable insights into the regulatory landscape of the genome. By identifying enriched motifs within accessible chromatin regions, researchers can gain a deeper understanding of the transcriptional regulatory networks and potentially uncover novel transcription factors involved in specific biological processes or diseases. This analysis serves as a powerful tool for unraveling the intricacies of gene regulation and can pave the way for further investigations in functional genomics and therapeutic development. Homer or MEME suite tools.

## 15.4 ATAC-Seq data **strengths**:

* The ATAC-seq is easy to adopt and has been used by many laboratories to generate high quality data for characterizing accessible chromatin in cell lines or sorted cells derived from tissues.
* In principle, ATAC-seq can identify a large proportion of cis-regulatory elements.
* In contrast to ChIP-seq, ATAC-seq does not require specific antibodies- ATAC-seq is a time-efficient protocol which requires low cell input.
* In comparison with histone modification ChIP-seq, ATAC-seq provides a higher resolution assessment of the cis-regulatory genomic regions. Histone modification ChIP-seq, in contrast, tends to be localized on nucleosomes flanking the site of interest and can spread to nucleosomes beyond the immediate flanking ones.

## 15.5 ATAC-Seq data **limitations**:

* ATAC-seq does not precisely identify the transcription factors or other chromatin associated factors that bind in or around chromatin accessible regions. This type of information needs to be inferred through analysis of transcription factor binding motif analysis or ChIP-seq data.
* Whereas ATAC-seq indicates the presence of a putative cis-regulatory element, H3K27ac ChIP-seq is able to separate accessible regions from those that are accessible and active.
* Accessible regions are not necessarily cis-regulatory regions, although many of them are.
* The genes that are regulated by cis-regulatory elements cannot be identified conclusively by ATAC-seq alone.
* ATAC-seq data can be biased, and affected by batch effects like any other genomics data type. When comparing ATAC-seq data good experimental design principles like the inclusion of biological replicates and consideration of controls, are needed for a meaningful outcome. .

## 15.6 ATAC-Seq data considerations

The nucleosome is the fundamental unit of chromatin packaging in the genome and nucleosomal DNA is far less likely to be cleaved by the Tn5 nuclease than linker DNA. When DNA is fragmented by Tn5 the positions of the endpoints relative to the nucleosomes is an important consideration. When the ends are less than 147bp apart it is likely that both ends originate from the same linker region. Longer fragments can result from cuts on opposite sides of the same nucleosome, or even opposite sides of a genomic interval that encompasses multiple nucleosomes. The short fragments are therefore most likely to be nucleosome free and provide stronger evidence for transcription factor binding sites.

As will other genomics protocols, ATAC-seq data is subject to biases introduced in the ATAC-seq protocol and in the sequencing itself. Comparison of ATAC-seq data generated in different batches, by different laboratories or using different protocols might not be directly comparable. In addition, the Tn5 endonuclease does have biases in the precise DNA sequences it can cut. This should be taken into consideration when carrying out base pair resolution analyses including footprinting analysis and analysis of the effects of sequence variants on chromatin accessibility.

Read depth will impact ATAC-seq signal, but enzyme strength and conditions can also alter the distribution of cuts.

When using ATAC-seq data to answer biological questions it is important to understand what types of bias could impact the results. To ensure valid results the analysis needs to use appropriate statistical methods, ensure enough high quality ATAC-seq data is available, including controls, and possibly reframing the questions.

## 15.7 ATAC-seq analysis tools

This section has been written by AI and needs verification by experts. This is meant to give you a basic idea of the pros and cons of these tools but should ultimately be used with your own judgment.

* [MACS2](https://github.com/macs3-project/MACS)([Y. Zhang et al. 2008](#ref-zhang2008model)):
  + **Pros**: widely used, handles both paired-end and single-end sequencing data, allows for differential peak calling between different samples.
  + **Cons**: assumes that all peaks have the same shape, may not be as accurate as other peak-calling tools in some cases.
* [HOMER](http://homer.ucsd.edu/homer/introduction/programs.html)([Heinz et al. 2010](#ref-heinz2010simple)):
  + **Pros**: includes tools for peak-calling, motif analysis, and annotation of nearby genes, user-friendly interface, handles both paired-end and single-end sequencing data.
  + **Cons**: may not be as accurate as other peak-calling tools in some cases.
* [ATACseqQC](https://bioconductor.org/packages/release/bioc/html/ATACseqQC.html)([Schep et al. 2017](#ref-schep2017chromvar)):
  + **Pros:** provides several metrics and plots for evaluating data quality, identifies potential issues with data such as batch effects, sequencing depth, and library complexity.
  + **Cons**: does not perform peak-calling or downstream analysis.
* [deeptools](https://deeptools.readthedocs.io/en/develop/)([Ramı́rez et al. 2016](#ref-ramirez2016deeptools2)):
  + **Pros**: includes tools for normalization, visualization, and comparison of ATAC-seq data, generates heatmaps, profiles, and other plots for visualizing chromatin accessibility.
  + **Cons**: may require some programming skills to use effectively.
* [DFilter](https://reggenlab.github.io/DFilter/tutorial.html) ([Ghavi-Helm et al. 2019](#ref-ghavi2019highly)):
  + **Pros**: uses a deep learning approach to predict the likelihood of a genomic region being an ATAC-seq peak, can handle both paired-end and single-end sequencing data, has been shown to outperform other peak-calling tools in some cases.
  + **Cons**: may require more computational resources than other tools.

## 15.8 Additional tutorials and tools

This section has been written by AI and needs verification by experts. This is meant to give you a basic idea of the pros and cons of these tools but should ultimately be used with your own judgment.

* [MACS2](https://github.com/macs3-project/MACS)([Y. Zhang et al. 2008](#ref-zhang2008model)):
  + **Pros**: widely used, handles both paired-end and single-end sequencing data, allows for differential peak calling between different samples.
  + **Cons**: assumes that all peaks have the same shape, may not be as accurate as other peak-calling tools in some cases.
* [HOMER](http://homer.ucsd.edu/homer/introduction/programs.html)([Heinz et al. 2010](#ref-heinz2010simple)):
  + **Pros**: includes tools for peak-calling, motif analysis, and annotation of nearby genes, user-friendly interface, handles both paired-end and single-end sequencing data.
  + **Cons**: may not be as accurate as other peak-calling tools in some cases.
* [ATACseqQC](https://bioconductor.org/packages/release/bioc/html/ATACseqQC.html)([Schep et al. 2017](#ref-schep2017chromvar)):
  + **Pros:** provides several metrics and plots for evaluating data quality, identifies potential issues with data such as batch effects, sequencing depth, and library complexity.
  + **Cons**: does not perform peak-calling or downstream analysis.
* [deeptools](https://deeptools.readthedocs.io/en/develop/)([Ramı́rez et al. 2016](#ref-ramirez2016deeptools2)):
  + **Pros**: includes tools for normalization, visualization, and comparison of ATAC-seq data, generates heatmaps, profiles, and other plots for visualizing chromatin accessibility.
  + **Cons**: may require some programming skills to use effectively.
* [DFilter](https://reggenlab.github.io/DFilter/tutorial.html) ([Ghavi-Helm et al. 2019](#ref-ghavi2019highly)):
  + **Pros**: uses a deep learning approach to predict the likelihood of a genomic region being an ATAC-seq peak, can handle both paired-end and single-end sequencing data, has been shown to outperform other peak-calling tools in some cases.
  + **Cons**: may require more computational resources than other tools.

## 15.9 Additional tutorials and tools

* [A Galaxy based tutorial for ATAC-seq](https://training.galaxyproject.org/training-material/topics/epigenetics/tutorials/atac-seq/tutorial.html) - Galaxy is a good recommendation for those new to informatics who would like a cloud-based GUI option to use for the analysis of their data.
* [MACS - Model-based analysis for ChIP-Seq](https://macs3-project.github.io/MACS/) - A command line tool for the identification of transcription factor binding sites. Can be used with ChIP-seq or ATAC-seq.
* [CHIPS](https://github.com/liulab-dfci/CHIPS) - A Snakemake pipeline for quality control and reproducible processing of chromatin profiling data. This tool will require some snakemake and coding knowledge. For more recommendations about coding see our later chapter about general data analysis tools.
* [Cistrome DB](http://cistrome.org/db/#/) - a visual tool to allow you to browse your ATAC-seq data.
* [SELMA - Simplex Encoded Linear Model for Accessible Chromatin](https://github.com/zang-lab/SELMA) - SELMA is a python based tool for the assessment of biases in Chromatin based data.

## 15.10 Online Visualization tools

* [Cistrome DB](http://cistrome.org/db/#/) - a visual tool to allow you to browse your ATAC-seq data.
* [UCSC Xena](http://xena.ucsc.edu/) is a web-based visualization tool for multi-omic data and associated clinical and phenotypic annotations. It can be used with ATAC-seq data.
* [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a track-based browser for interactively exploring genomic data mapped to a reference genome.

## 15.11 More resources about ATAC-seq data

* [ATAC-seq overview from Galaxy](https://training.galaxyproject.org/training-material/topics/epigenetics/tutorials/atac-seq/slides.html#1) - these slides explain the overarching concepts of ATAC-seq.
* [ATAC seq guidelines from Harvard](https://informatics.fas.harvard.edu/atac-seq-guidelines.html) - this workflow runs through step by step how to analysis ATAC-seq data and what different parameters mean.
* [ATAC-seq review](https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-1929-3) - this paper gives a great overview of ATAC-seq data and step by step what needs to be considered.
* [Identifying and mitigating bias in chromatin](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4473780/)
* [CHIP Snakemake pipeline for analyzing ChIP-seq and chromatin accessibility data](https://f1000research.com/articles/10-517)
* [Paper on bias in DNase-seq footprinting analysis and fragment size effects, similar comments apply to ATAC-seq](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4018771/)
* [SELMA Method for evaluating footprint bias in ATAC-seq](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9492688/)

# 16 Single cell ATAC-Seq

This chapter is incomplete! If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 16.1 Learning Objectives

{r, fig.alt = "Learning objectives This chapter will demonstrate how to: Understand the basics of single cell ATAC-Seq data collection and processing workflow Identify the next steps for your particular single cell ATAC-Seq data. Formulate questions to ask about your single cell ATAC-Seq data", out.width = "100%", echo = FALSE} ottrpal::include\_slide("https://docs.google.com/presentation/d/1YwxXy2rnUgbx\_7B7ENH9wpDX-j6JpJz6lGVzOkjo0qY/edit#slide=id.g227d7dd1e08\_0\_41")

## 16.2 What are the goals of scATAC-seq analysis?

The primary goal of single-cell ATAC-seq is to obtain a high-resolution map of chromatin accessibility at the single-cell level. It is often used for the identification of cell type-specific cis-regulatory elements (CREs) or transcription factor (TF) binding sites because single-cell resolution enables researchers to parse heterogeneous subgroups within a sample. Single-cell ATAC-seq is often applied to questions in developmental biology and cell differentiation.

## 16.3 scATAC-seq general workflow overview

Align reads to genome and assign to cells based on barcodes This step can be performed using Cell Ranger if the data were generated using a 10X Genomics kit (commercially available). For other methods, this step largely resembles the alignment step of bulk ATAC-seq analysis, using aligners such as Bowtie2 or BWA, filtering tools such as Picard, and adapter-trimming tools such Trimmomatic. Prior to adapter trimming barcodes should be matched to the list of known barcodes generated in the experiment and either assigned to a cell or assigned as ambiguous. At this stage unique molecular identifiers (UMIs) added to fragments during library preparation are also extracted and associated with each read to allow for PCR deduplication. Quality control

The most important considerations for single-cell ATAC-seq are the number of unique fragments per cell, the transcription start site (TSS) enrichment score and detection of doublets.

The number of unique fragments in a cell is a critical quality control metric for single-cell ATAC-seq. Cells with a low fragment count do not provide enough information to draw conclusions about their characteristics, and cells with extremely high fragment counts are likely to be doublets containing reads from multiple cells. To determine the number of unique reads per cell, short random barcodes termed unique molecular identifiers (UMIs) are added to the fragments during library preparation. After the reads have been aligned to the genome and grouped by their cell barcodes, the UMIs can be used to remove PCR duplicates by retaining only one copy of reads with the same UMI and genomic location. The resulting UMI counts can be used as a more accurate measure of chromatin accessibility at specific genomic regions in individual cells. An additional step is typically taken to filter out reads mapping to the mitochondrial genome, so that the final unique fragment counts consist of only unique reads corresponding to nuclear DNA.

The TSS enrichment score in ATAC-seq measures the preferential accessibility of chromatin regions near gene promoters. This approach was established in pipelines for bulk ATAC-seq, such as the ENCODE pipeline (cite), and is also applicable to single-cell ATAC-seq. In brief, the TSS enrichment score quantifies the enrichment of open chromatin regions at TSSs versus a non-TSS background (e.g. +/-2000 bp beyond TSSs). A high TSS enrichment score therefore indicates that the number of accessible regions at TSSs, where high accessibility is expected, is significantly higher than background (cite), while a low TSS enrichment score indicates that the data quality is not high enough to distinguish accessible regions from background insertion patterns.

Doublet detection is any approach that attempts to computationally identify cell barcodes which contain reads from a mixture of single cells. Although an extremely high number of fragment counts may indicate that a cell is in fact a doublet, doublet detection provides a more targeted approach by assigning a score or a probability that each cell is a doublet. These approaches may compare cells to simulated doublets generated randomly from the data, or may rely on the fact that the number of ATAC-seq reads in a single cell is limited to only two reads per cell for diploid organisms. This step is not as common in scATAC-seq analysis as it is in single cell RNA-seq analysis owing to the difficulty of estimating doublets from the highly sparse data, but can be done for additional rigor or if there is particular concern that the dataset contains a high number of doublets.

Additionally, the fragment size distribution of the library should exhibit nucleosomal periodicity, where fragments are enriched at ~147 bp intervals corresponding to the length of nucleosome-bound DNA that are refractory to Tn5 insertion.

## 16.4 Peak calling

Peak calling in ATAC-seq is performed in a similar manner to bulk ATAC-seq [ref bulk chapter]. Importantly, it should be performed by treating data from all cells within a cluster as a pseudo-bulk replicate. This is because scATAC-seq data is highly sparse and any individual cell only has enough information to convey whether a region is accessible or inaccessible, due to the maximum of 2 reads per locus per cell. Peak calling is commonly performed using MACS2, but other peak callers suitable for ATAC-seq could be used as well, as described in our chapter on bulk ATAC-seq (reference).

## 16.5 Dimensionality reduction

As ATAC-seq data is extremely high dimensional, with counts for hundreds of thousands of peaks in thousands of cells, dimensionality reduction must be performed to represent the data in a way which reflects the major sources of variation while allowing for efficient computation. Many of the most popular dimensionality reduction approaches for ATAC-seq are borrowed from natural language processing, including latent semantic indexing (LSI) as well as probabilistic approaches such as latent Dirichlet allocation (LDA) and probabilistic LSI (pLSI). LSI and its variations are commonly used and are a simple, efficient approach based on PCA. Probabilistic approaches calculate the probability of information in a dataset being related to specific ‘topics’ identified by the statistical model. They are more mathematically complex than LSI but attempt to more accurately reconstruct the latent (not observable) structure in the data.

## 16.6 Embedding (visualization)

Embedding is the process of representing the high-dimensional scATAC-seq dataset in two (or occasionally three) dimensions for visualization. First, dimensionality reduction must have been performed using one of the methods described in the section above. Then, the result of dimensionality reduction can be provided as input to the chosen embedding approach. The most common method for generating ATAC-seq embeddings is UMAP (Uniform Manifold approximation) but other methods, such as force-directed graph layouts or t-SNE (t-distributed Stochastic Neighbor Embedding) can also be used.

## 16.7 Clustering

Clustering is the process of computationally detecting populations of cells with similar characteristics - in this case, cells with similar accessibility profiles. Leiden clustering, which uses the similarity of cells to their neighbors to group cells into clusters, is a common choice for identifying clusters in scATAC-seq data.

## 16.8 Cell type annotation

Cell type annotation on scATAC-seq data alone can be performed based on the enrichment of cell-type-specific CREs, or alternatively can be performed based on gene expression patterns observed in integrated scRNA-seq data. Gene scores are a measure of the accessibility of a gene locus and putative CREs within a defined window of the gene. Gene scores significantly above the expected background suggest a gene is active in a given cell type, and these scores can be used to identify markers for cell type annotation. Integration with scRNA-seq data can allow for identification of cell types which may be difficult to distinguish based on ATAC-seq profiles alone(ref), but requires an scRNA-seq dataset of a comparable population of cells.

Trajectory analysis, which is used to infer and visualize the developmental or differentiation paths of individual cells within a population, can be performed on processed single-cell ATAC-seq data using tools developed for single-cell RNA-seq data. These approaches aim to reconstruct the temporal progression and identify the key intermediate states or cell fate decisions during biological processes such as embryonic development, tissue regeneration, or disease progression.

Trajectory inference algorithms, such as:

* [Monocle](https://cole-trapnell-lab.github.io/monocle3/docs/trajectories/) Qiu et al. ([2017](#ref-Qiu2017))
* [Slingshot](https://bioconductor.org/packages/devel/bioc/vignettes/slingshot/inst/doc/vignette.html) Street et al. ([2018](#ref-Street2018))
* [Palantir](https://github.com/dpeerlab/Palantir) Setty et al. ([2019](#ref-Setty2019))
* [PAGA](https://github.com/theislab/paga) Wolf et al. ([2019](#ref-Wolf2019))

These are commonly used to reconstruct the developmental trajectories and order the cells along these trajectories. The resulting trajectory models provide valuable insights into the underlying regulatory dynamics, lineage relationships, and critical regulatory genes or pathways governing cellular differentiation and development.

Much like peak calling, it is not possible to obtain enough information from individual cells to perform differential accessibility analysis at the single cell level. Because of this limitation, differential accessibility analysis is performed in a similar manner to bulk ATAC-seq analysis using pseudo-bulk data at the cluster or cell type level, where counts from many single cells are aggregated together and treated as though they are a single sample generated from a bulk experiment. Common tools for differential accessibility analysis include deSeq2 and EdgeR, which were both developed for differential gene expression analysis.

## 16.9 scATAC-seq data **strengths**:

* scATAC-seq is the gold-standard for showing heterogeneity in chromatin accessibility between populations of cells and within tissues because single-cell resolution enables analysis of subpopulations that are challenging to isolate experimentally.
* scATAC-seq can be paired with scRNAseq to obtain transcriptome and chromatin accessibility measurements from the same cells. This is a powerful approach for gaining understanding of how specific patterns of chromatin accessibility affect gene expression.
* scATAC-seq is also a relatively high throughput technique, particularly with droplet based techniques. A single dataset can cover thousands of cells.

## 16.10 scATAC-seq data **limitations**:

* scATAC-seq has very high sparsity compared to single-cell RNA-seq since there are only two copies of each locus in a diploid cell compared to many copies of mRNAs. Like other single-cell techniques This results in the data essentially being binary at the single cell level - a region either has reads and is considered accessible in that cell or has no reads.
* Like bulk ATAC-seq, the Tn5 transposase has a sequence bias, so regions with a preferred sequence will undergo higher levels of transposition. Highly accessible regions of DNA will also be overrepresented in the final library.
* Single-cell ATAC-seq is an expensive technique regardless of the experimental approach chosen. Plate-based methods are generally cheaper but have lower throughput, while droplet-based methods are higher throughput but extremely costly and reliant on proprietary technology. Large datasets require significant investment and often use of droplet-based techniques.
* Many scATAC-seq datasets have low cell numbers due to the cost and technical difficulty of the assay. This presents a challenge for analysis since the data is highly sparse and noisy, which in combination with a small dataset can lead to difficulty interpreting the data.

## 16.11 scATAC-seq data considerations

scATAC-seq will always be sequenced with paired-end reads. There are two major experimental approaches for generating single-cell ATAC-seq data: droplet based methods, such as the commercially available [10X Chromium platform](https://www.10xgenomics.com/products/single-cell-atac), where nuclei are separated into individual droplets, and plate-based methods, which use multiple pooling and barcoding steps to tag each cell with a unique combination of barcodes (with a level of expected barcode collisions).

The procedure for demultiplexing the reads will depend on the method used to generate the data. Data generated using 10X platforms can be de-multiplexed and aligned using the Cell Ranger software, while plate-based approaches typically use an alignment and peak-calling approach similar to that used for bulk ATAC-seq, with the additional step of matching the barcodes in each read to the known set of combinatorial barcodes. Correctly matching the reads to cells and filtering reads with non-matching barcodes is a critical step for scATAC-seq analysis.

## 16.12 scATAC-seq analysis tools

* [Cellranger](https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger) is a popular preprocessing tool specifically designed for scATAC-seq data generated using the 10x Genomics platform. It performs essential steps such as demultiplexing, barcode processing, read alignment, and filtering, providing a streamlined workflow for 10x-generated scATAC-seq data. However, it cannot be used for data generated by other methods.
* [Bowtie2](https://bowtie-bio.sourceforge.net/bowtie2/manual.shtml), [Picard tools](https://broadinstitute.github.io/picard/), and [Trimmomatic](http://www.usadellab.org/cms/?page=trimmomatic): These tools are commonly used for preprocessing scATAC-seq data generated using plate-based or combinatorial indexing approaches. Bowtie is a fast and widely used aligner for mapping sequencing reads to a reference genome, while Picard provides a suite of command-line tools for manipulating and analyzing BAM files and Trimmomatic can remove adapter sequences from reads. These tools can be utilized for aligning reads, removing duplicates, sorting, and filtering the data to obtain the necessary inputs for downstream analysis.
* [ArchR](https://www.archrproject.com/) is a comprehensive scATAC-seq preprocessing tool implemented in R. It accepts both 10x fragment files and BAM files as input, making it suitable for data generated using different protocols. ArchR performs quality control, peak calling, peak annotation, normalization, and data transformation steps. It is one of the most popular tools for analyzing standalone scATAC-seq data and provides a user-friendly interface for exploratory data analysis.
* [Scanpy](https://scanpy.readthedocs.io/en/stable/) is a Python-based tool widely used for visualizing and manipulating single-cell omics data, including scATAC-seq. After processing scATAC-seq data with tools like ArchR, the output can be exported as a matrix (data) or CSV (metadata) and formatted into a Scanpy data object. Scanpy offers various analytical functionalities, including dimensionality reduction, clustering, trajectory inference, differential accessibility analysis, and visualization. This tool is the tool of choice if you plan to perform your analysis primarily in Python.
* [Seurat](https://satijalab.org/seurat/) is an R-based tool that is extensively used for analyzing and visualizing single-cell omics data, including scATAC-seq. Similar to Scanpy, after preprocessing the data with tools like ArchR, Seurat can be employed for downstream analysis. It provides a wide range of functions for quality control, dimensionality reduction, clustering, differential accessibility analysis, cell type identification, and visualization. Seurat integrates well with other existing R-based tools for single-cell data analysis, offering flexibility and compatibility. This is a useful core tool to use if you plan to perform your analysis in R.
* [Signac](https://stuartlab.org/signac/) is an R package specifically designed for the analysis of single-cell epigenomics data, including scATAC-seq. It offers a comprehensive set of functions for preprocessing, quality control, dimensionality reduction, clustering, trajectory analysis, differential accessibility, and visualization. Signac integrates well with Seurat, providing an additional tool for exploring and analyzing scATAC-seq data.

Additional quality checking tools: Quality checking and filtering steps in scATAC-seq analysis can be performed using various tools depending on the workflow and programming language. Some commonly used tools with QC capabilities useful for examining library quality measures such as GC bias, overrepresented sequences, and quality scores include FastQC and deepTools.

#### 16.12.0.1 Doublet detection

[ArchR](https://www.archrproject.com/) has a tool for doublet detection - it generates synthetic doublets from combinations of cells in the dataset and uses the similarity of cells in the dataset to these synthetic doublets to identify doublets. This is a common approach, and variations of it are used by most doublet detection algorithms. Many are specifically designed to expect transcriptomic data (such as the commonly used Scrublet) and identify barcodes with mixed transcriptional signatures of multiple clusters/cell types, and these methods do not accept scATAC-seq input. Some transcription based tools can be given modified input to detect doublets in scATAC-seq data, as described in documentation from the Demuxafy project. There are also tools like AMULET which leverage the fact that the number of ATAC-seq reads at any locus in a single cell are limited by the number of copies of a chromosome to detect doublets. Overall, doublet detection is not as common of a step in scATAC-seq analysis as it is in scRNA-seq analysis, owing to the limited tools available and the difficulty of performing this analysis on extremely sparse data.

#### 16.12.0.2 Visualization

[Scanpy](https://scanpy.readthedocs.io/en/stable/) (Python) and [Seurat](https://satijalab.org/seurat/) (R) are the most commonly used tools for visualizing scATAC-seq data. These tools allow you to plot the accessibility of specific peaks or gene scores, as well as metadata such as cell type, clusters, etc. on the UMAP (or other) embedding at the single-cell level. Both packages include built-in functions to perform this plotting in a streamlined manner and to manipulate the data objects for additional quantification and visualization using general plotting packages such as [matplotlib](https://matplotlib.org/) or [ggplot](https://ggplot2.tidyverse.org/). The choice between these tools is primarily determined by the programming language you choose for your analysis, as they share many of the same core features. Additionally, tools such as deepTools or enrichedHeatmap may be useful for visualizing heatmaps of pseudo-bulk data, and bedGraph or BigWig representations of pseudo-bulk data can be visualized using genome browsers such as IGV or UCSC genome browser. pyGenomeBrowser is a package which allows more customizable visualization of browser tracks and may be useful for generating publication-quality figures.

## 16.13 Trajectory analysis

Several tools are available for single-cell trajectory analysis. These approaches are primarily distinguished by variations used in their mathematical approaches for calculating trajectories, but most make use of graph-based approaches which model the similarity or connections between cells in a dataset. The distinct approaches of the tools discussed here lead to varying levels of performance on different types of data, and extensive benchmarking has been performed (here) and (here) on synthetic datasets to determine the accuracy of different approaches. The most important consideration here is whether there are any cyclic trajectories expected in the dataset, where the end of the trajectory would connect back to the start, or disconnected trajectories, where not all trajectories originate from the same starting state. Not all approaches can reconstruct these trajectories accurately. Most popular methods expect a tree-like structure, with a single starting point and branches which lead toward terminal cell fates.

[Monocle](http://cole-trapnell-lab.github.io/monocle-release/) is a popular choice that offers a comprehensive workflow for trajectory inference, visualization of trajectory analysis, pseudotime ordering of cells, and identification of differentially expressed genes along trajectories. Another commonly used tool is Slingshot, which utilizes a graph-based approach to infer trajectories, compute pseudotime ordering, and generate smooth curves to visualize trajectories. Additionally, it has the ability to infer multiple disconnected trajectories within a single dataset. PAGA (Partition-based Graph Abstraction) uses a distinct strategy with the goal of maintaining connections between similar groups of cells as well as the overall structure of the data. [Palantir](https://github.com/dpeerlab/Palantir) is a tool which uses a probabilistic approach to assign cell fate probabilities to each cell in a dataset, which can be used to define cells belonging to a specific trajectory.

## 16.14 Motif detection (ex. ChromVar)

[Single-cell chromVAR analysis](https://bioconductor.org/packages/release/bioc/html/chromVAR.html) is a computational approach used to assess cell-to-cell variation in chromatin accessibility profiles across a population of single cells. It aims to identify TF activity differences between cell types or states and elucidate the underlying regulatory dynamics. Single-cell chromVAR leverages the concept of TF motif enrichment or depletion within cell-specific accessible regions to infer TF activity. It compares the chromatin accessibility profiles of individual cells to a background model derived from the aggregate accessibility profiles of all cells, enabling the detection of cell-specific TF binding patterns. By quantifying the enrichment or depletion of TF motifs within accessible regions, single-cell chromVAR provides insights into TF activity variation, potential regulatory networks, and cell-type-specific transcriptional regulation. It serves as a valuable tool for understanding the contribution of TFs to cellular heterogeneity and regulatory processes in single-cell chromatin accessibility data.

## 16.15 Regulatory network detection

[CisTopic](https://github.com/aertslab/cisTopic) is a computational tool used for the analysis of single-cell chromatin accessibility data to identify and characterize cell subpopulations with distinct regulatory patterns. It employs a topic modeling approach to capture the variability in chromatin accessibility profiles across cells and identifies the major regulatory patterns driving cell heterogeneity. CisTopic assigns cells to topics based on the similarity of their accessibility landscapes. By analyzing the differential accessibility of genomic regions within each topic, CisTopic facilitates the discovery of transcription factor binding motifs and CREs associated with specific cell subpopulations.

## 16.16 Tools for data type conversion

A comprehensive explanation of packages to convert between single-cell data object types used by Python and R packages is found here.

The most common data types for processed scATAC-seq data are:

* [SingleCellExperiment](https://bioconductor.org/packages/release/bioc/html/SingleCellExperiment.html)
* [Seurat/h5Seurat](https://mojaveazure.github.io/seurat-disk/articles/h5Seurat-load.html)
* [annData objects](https://anndata.readthedocs.io/en/latest/)

H5seurat objects can be [converted to annData objects using SeuratDisk](https://mojaveazure.github.io/seurat-disk/articles/convert-anndata.html).

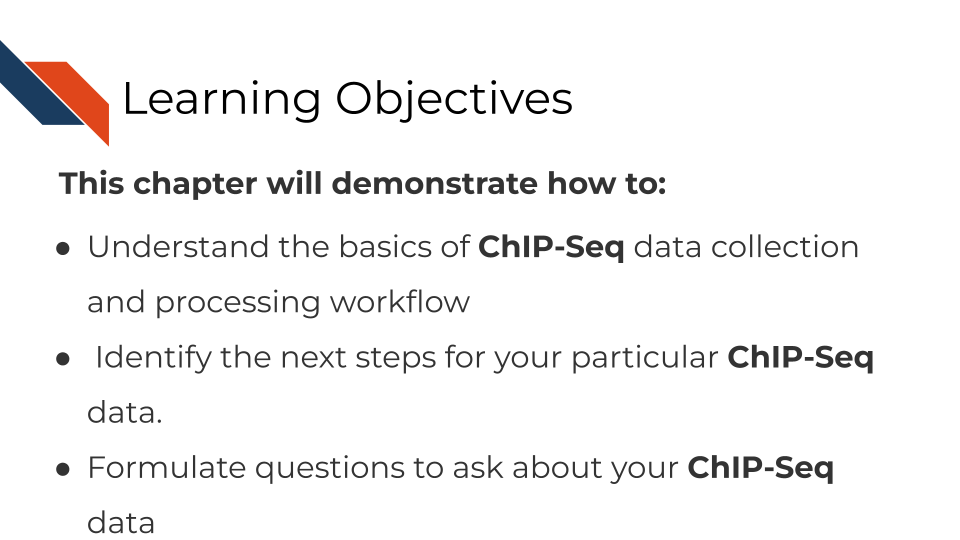
## 16.17 More resources and tutorials about scATAC-seq data

* [Galaxy tutorial for sc-ATAC-seq analysis](https://training.galaxyproject.org/training-material/topics/single-cell/tutorials/scatac-preprocessing-tenx/tutorial.html)
* [Signac scATAC-seq tutorial with pbmcs](https://stuartlab.org/signac/articles/pbmc_vignette.html)
* [sc ATAC-seq chapter - Intro to Bioinformatics and Comp Bio](https://liulab-dfci.github.io/bioinfo-combio/scatac.html)
* [Single Cell ATAC-seq youtube video](https://www.youtube.com/watch?v=ufUVMHLDa00)
* [Comprehensive analysis of single cell ATAC-seq data with SnapATAC](https://www.nature.com/articles/s41467-021-21583-9)

# 17 ChIP-Seq

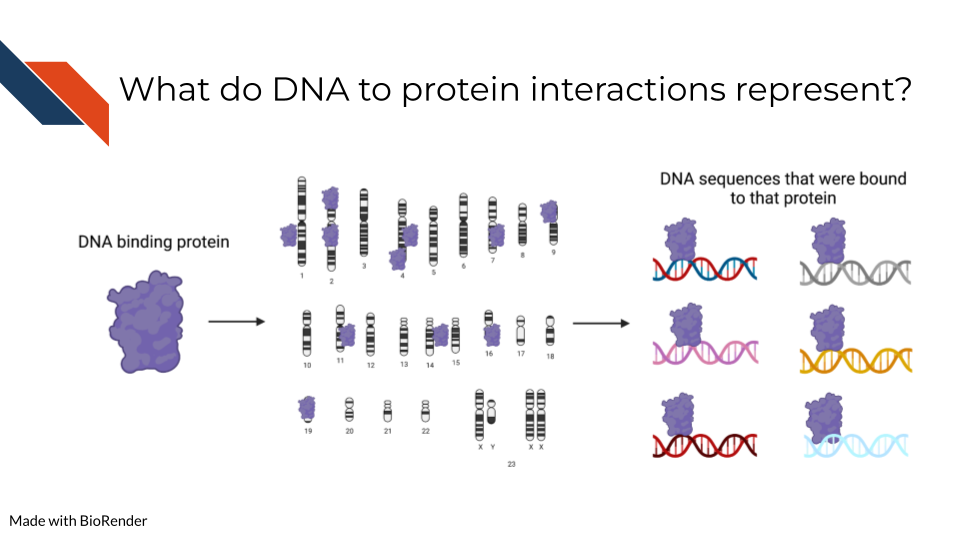
This chapter is incomplete! If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 17.1 Learning Objectives



## 17.2 What are the goals of ChIP-Seq analysis?

The goal of ChIP-Seq is to….

 ## What are the goals of ChIP-Seq analysis?

ChIP-Seq (chromatin immunoprecipitation sequencing) and related approaches are used to identify genome-wide binding sites of specific proteins or protein complexes. Given the diversity of interactions at the DNA-protein interface, sequencing-based methods for targeted chromatin capture have evolved to meet precise research needs and improve the quality of the results. Specifically, ChIP-Seq builds on protein immunoprecipitation techniques (IP) by applying next generation sequencing to a pulldown product. IP followed by sequencing can be applied to any nucleic-acid binding protein for which an antibody is available, including a known or putative transcription factor (TF), chromatin remodeler or histone modifications, or other DNA- or chromatin-specific factors. ChiP-Seq approaches have been honed to increase signal-to-noise, reduce input material, and more specifically map protein-DNA interactions, for example by treating the IP product with a exonuclease that chews-back unprotected DNA end (e.g. ChIP-exo).

The main goals of analysis for ChIP-Seq approaches are:

* Identify the genomic regions where a specific protein or protein complex binds. This can be achieved by sequencing both the IP input and product, and then calculating the enrichment in the product sample over the input.
* Annotate binding sites via comparison to other datasets and genome annotations. This may include transcription start sites (TSSs) or gene-regulatory regions. Oftentimes it is best to validate your data against previous profiling of similar epitopes.
* Comparison of binding sites: Many ChIP-Seq experiments compare changes in protein-DNA interactions across different conditions. This type of analysis can leverage statistical tools for pairwise comparison and multiple hypothesis testing.
* Identification of co-occurring motifs: Many chromatin proteins exhibit a sequence-specific binding pattern that is shaped by evolutionary forces. These sequence patterns, or motifs, are thought to capture contacts between specific base pairs and the DNA-binding domain of a protein and are often represented as a position weight matrix (PWM) for computational analysis. Statistical tools have been developed for de novo motif discovery within a given set of genomic intervals, like a ChIP-seq peaklist. The list of discovered motifs can be meaningfully interpreted by cross referencing with a motif database and recovery of known motifs represent another means of data validation.
* Integration with other -omics data: Given the expansive repositories of publicly available sequencing data, creating a comprehensive narrative from a ChIP-Seq experiment usually involves comparison with other types of sequencing data. Just like how a ChIP-Seq peak list can be interpreted through existing genome annotations, other sequencing data can be interpreted through the binding sites identified from a given ChIP-Seq experiment. For example, a sequence variant might be enriched for or against in protein binding sites versus previously identified motifs. This would suggest that a mutation would alter DNA-protein interactions. Binding of a specific gene-regulatory element might also correlate with changes in gene expression.

## 17.3 ChIP-Seq general workflow overview

—<TODO: add data formats in a graphical format>

A key contribution of large consortia, such as the ENCODE consortium, are standardized processing workflows to facilitate the integration of ChIP-seq data generated in different labs. While the exact data processing needs of any given experiment may vary, established pipelines provide a helpful starting point. In choosing a data processing workflow, it is essential to note the input data format. For example, the read length should be considered, as well as the sequencing paradigm (i.e. whether the data is single-end or paired-end).

The most generic steps for processing ChIP-Seq data are:

* Quality control: The first step in ChIP-Seq data processing is to perform quality control checks on the raw sequencing data to assess its quality and identify any potential issues, such as poor sequencing quality or adapter contamination which can be assessed via FASTQC.
* Read alignment: The next step is to align the ChIP-Seq reads to a reference genome using a suitable alignment tool such as Bowtie or BWA. Notably, many publicly available ChIP-Seq datasets are single-ended and it is important to use the correct alignment parameters for a given sequencing approach. In the case of ChIP-seq approaches that include exonuclease treatment, such as ChIP-exo and ChIP-nexus, a paired-end sequencing approach is often taken and then insert size can be useful for validating alignment. For example, profiling of a histone modification should yield nucleosome-sized fragments, ranging up from 120 bp for mononucleosomes, whereas TFs should yield smaller, sub-nucleosomal fragments and polymerase is in between at 20-50bps (PMID: 30030442).
* Peak calling: After the reads have been aligned to the genome, the next step is to identify the genomic regions where the protein or protein complex of interest is bound. This is done using peak-calling algorithms, such as MACS2, SICER, or HOMER, which can calculate enrichment as fold change over the input control with statistical testing.
* Quality control of peaks: Once the peaks have been called, it is important to perform quality control checks to ensure that the peaks are of high quality and biologically relevant. This can be done by assessing the number of peaks, fraction of reads in peaks (FRiPs), enrichment of the peaks in specific genomic regions, comparing the peaks to known gene annotations, or performing motif analysis. Often, peaks will be merged across replicates to create a consensus peak set. Peaks should be assessed visually with tools like IGV or the UCSC genome browser to ensure they overlap regions of high coverage. The Cistrome Data Browser is another useful resource for comparing with published ChIP-seq, DNase-seq and ATAC-seq data.
* Differential binding analysis: If the ChIP-Seq experiment involves comparing the binding of the protein or protein complex in different conditions or cell types, statistical testing can be performed to identify the regions of the genome where the protein or protein complex binds differentially. Tools developed for multiple comparison testing, like Limma, Deseq2, and EdgeR are useful for this type of comparative analysis.
* Integrative analysis: Finally, integrative analysis with other -omics data can be performed to gain biological insights into the ChIP-Seq data. This can involve interpreting ChiP-Seq data through existing annotations by looking at signal enrichment in different genomic regions, like transcription start sites (TSSs), gene bodies, and previously-identified cis-regulatory elements (CREs). ChIP-Seq data can even be interpreted through other ChIP-seq data to see if features overlap with statistical testing for similarity using packages like BEDTools and Bedops.

## 17.4 ChIP-Seq data **strengths**:

ChIP-Seq (chromatin immunoprecipitation sequencing) is a powerful tool for understanding the genomic locations where a specific protein or protein complex binds.

ChIP-Seq is particularly good at showing or illustrating:

* Identification of regulatory elements: ChIP-Seq can be used to identify the genomic regions where a protein or protein complex binds to regulatory elements, such as promoters, enhancers, and silencers. For example, certain histone modifications characterize active promoters and enhancers, such as H3K4 methylation and H3K27 acetylation.
* Characterization of protein-protein interactions: ChIP-Seq can be used to identify the genomic regions where multiple proteins bind. In this way, cobinding can be inferred to provide insight into the protein-protein interactions that are involved in regulating gene expression.
* Identification of binding site motifs: ChIP-Seq can be used to identify the DNA motifs that are enriched in the binding sites of a protein or protein complex. This information can be used to identify other transcription factors or cofactors that are involved in the same regulatory network. Databases of known TF binding motifs include JASPAR, Cis-BP, Hocomoco [include links].
* Differential binding analysis: ChIP-Seq can be used to compare the binding of a protein or protein complex in different conditions or cell types, which can provide insight into the mechanisms that regulate protein binding and the impact of different cellular states on the regulatory networks.

## 17.5 ChIP-Seq data **limitations**:

ChIP-Seq (chromatin immunoprecipitation sequencing) is a powerful technique, but there are several biases, caveats, and problems that can arise when analyzing ChIP-Seq data.

Some of the most common biases, caveats, and problems are:

* Accessibility bias: ChIP-Seq relies on fragmentation of chromatin prior to immunoprecipitation, which is observed to enrich for genomic regions that are highly accessible to TFs in general .
* Antibody specificity and cross-reactivity: The specificity of the antibody used in ChIP-Seq is crucial for the accuracy of the results. Finding an antibody for specific epitopes can pose a challenge because antibodies can have cross-reactivity with other epitopes, which can result in false positives or misinterpretation of the data.
* DNA fragmentation bias: The length and quality of the DNA fragments used in ChIP-Seq can impact the results. Shorter fragments are often located in regions with more highly accessible chromatin, especially nucleosome linker regions and promoters of active genes. Sequencing depth bias: The amount of sequencing depth can impact the results of ChIP-Seq analysis. Insufficient sequencing depth can result in false negatives or miss important binding sites.
* Reproducibility and sample variation: ChIP-Seq experiments can be highly variable, and reproducibility between replicates can be an issue. Additionally, the composition and quality of the sample can also impact the results. Peak-calling algorithm choice: The choice of peak-calling algorithm can impact the results of ChIP-Seq analysis, as different algorithms have different strengths and weaknesses.
* Interpretation of binding sites: Finally, the interpretation of binding sites identified by ChIP-Seq can be complex and requires additional validation to confirm their biological relevance and function. Notably, ChIP-Seq cannot distinguish direct protein-DNA interaction from indirect binding (e.g. where a protein may bind another protein that binds to DNA).

## 17.6 ChIP-Seq data considerations

<What variations of data exist within ChIP-seq that someone might need to know about? For example, are there paired end and single end reads? Or different ways of isolating data? >

As a general guideline, a minimum sequencing depth of 20 million reads is recommended for ChIP-seq experiments in Drosophila, whereas 40–50 million reads is a practical minimum for most marks in human tissue (PMID: 24598259). However, this depth may not be sufficient for some analyses, particularly for studies that require high resolution or low signal-to-noise ratio. In such cases, deeper sequencing may be necessary to achieve the desired level of sensitivity and specificity.

In general, epitopes that cover large sequence space (e.g. repressive histone modification such as H3K27me3) require greater sequencing depth than epitopes confined to more narrow genomic regions (e.g. active histone modifications such as H3K4 methylation and H3K27ac). ChIP-seq for TFs may require even less sequencing depth; however, low antibody specificity may necessitate deeper sequencing due to low signal-to-noise. In practice, the depth of sequencing required for ChIP-seq experiments can vary widely depending on the specific experimental design and research question. It is important to perform a pilot study or use appropriate statistical methods to estimate the necessary sequencing depth for a given experiment. Choosing a specific antibody is essential, otherwise even deep sequencing may not recover signal over high background. Sequencing depth should also account for genome size (e.g. larger genome requires deeper sequencing).

## 17.7 ChiP-seq analysis tools

### 17.7.1 Tools for quality checks

* [FastQC](https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) is a widely used tool that is used to assess the quality of sequencing data. It analyzes the raw sequencing data and generates a report that provides an overview of various metrics such as base quality, sequence length distribution, and GC content.
* [Picard tools](https://broadinstitute.github.io/picard/) and [SAMtools](http://www.htslib.org/): Picard tools and SAMtools are two collections of command-line tools that are used to manipulate and analyze high-throughput sequencing data. They can be used to check the quality of the data, remove duplicates, and generate summary statistics.
* [MACS2 (Model-based Analysis of ChIP-Seq)](https://pypi.org/project/MACS2/) is a software tool that is specifically designed for the analysis of ChIP-Seq data. It is used to identify regions of the genome that are enriched for DNA-protein interactions.
* [ENCODE Uniform Processing Pipelines](https://www.encodeproject.org/pipelines/): The ENCODE (Encyclopedia of DNA Elements) Uniform Processing Pipelines are a set of standardized protocols and tools that are used to process and analyze ChIP-Seq data. They ensure that the data generated by different labs are consistent and can be easily compared.

These tools are just a few examples of the many quality control tools available for ChIP-Seq analysis. The choice of tool(s) to use will depend on the specific analysis being performed and the preferences of the user.

### 17.7.2 Tools for Peak calling:

* [MACS2 (Model-based Analysis of ChIP-Seq)](https://pypi.org/project/MACS2/) is a widely used tool for peak calling in ChIP-Seq data. It uses a Poisson distribution to model the local noise and identifies peaks based on the fold enrichment over the background noise.
* [SICER](https://zanglab.github.io/SICER2/): Spatial Clustering for Identification of ChIP-Enriched Regions (SICER) is a peak caller that takes into account the spatial clustering of enriched regions in ChIP-Seq data. It uses a clustering algorithm to identify peaks based on the local density of enriched regions.
* [HOMER (Hypergeometric Optimization of Motif EnRichment)](http://homer.ucsd.edu/homer/) is a suite of tools that includes a peak caller for ChIP-Seq data. It uses a sliding window approach to identify peaks based on the local enrichment of reads.
* [PeakSeq](https://github.com/gersteinlab/PeakSeq) is a peak caller that uses a Bayesian approach to identify enriched regions in ChIP-Seq data. It models the relationship between the read counts and the signal-to-noise ratio and identifies peaks based on the posterior probability of enrichment.

### 17.7.3 Tools for Differential Analysis

* [DESeq2](https://bioconductor.org/packages/release/bioc/html/DESeq2.html): This is a widely used R package for differential analysis of sequencing count data, including ChIP-seq. It uses a negative binomial model to normalize and test for differential enrichment of ChIP-seq peaks.
* [edgeR](https://bioconductor.org/packages/release/bioc/html/edgeR.html): Another popular R package for differential expression analysis of RNA-seq data, edgeR can also be used for differential analysis of ChIP-seq data. It uses a generalized linear model to estimate differential enrichment and has been shown to be effective for ChIP-seq data with low read counts. Annotation
* [ChIPseeker](https://bioconductor.org/packages/release/bioc/html/ChIPseeker.html): This R package can be used for annotating ChIP-seq peaks with genomic features such as gene annotation, gene ontology, and pathway analysis. It can also generate plots and heatmaps for visualization.
* [HOMER](http://homer.ucsd.edu/homer/): This suite of tools includes several programs for motif discovery, peak annotation, and visualization. The annotatePeaks.pl program can be used for assigning genomic regions to specific functional categories, including promoter, exon, intron, intergenic, and enhancer regions.
* [GREAT](http://great.stanford.edu/public/html/): This web-based tool can be used for annotating genomic regions with functional annotations such as gene ontology terms and regulatory domains. It uses a statistical approach to associate genomic regions with biological functions.
* [Cistrome-GO](http://go.cistrome.org/): A web-based tool for determining the gene ontologies of genes likely to be regulated by regions discovered through TF ChIP-seq.
* [GenomicRanges](https://bioconductor.org/packages/release/bioc/html/GenomicRanges.html): This R package provides a framework for working with genomic ranges, including intersection, overlap, and annotation of genomic regions with functional categories. It can be used in conjunction with other R packages for ChIP-seq analysis, such as ChIPseeker and DiffBind.
* [ChIP-Enrich](http://chip-enrich.med.umich.edu/): This web-based tool can be used for annotating ChIP-seq peaks with functional categories such as gene ontology, pathway analysis, and transcription factor binding sites. It uses a hypergeometric test to identify overrepresented functional categories.
* [Cistrome DB](http://dbtoolkit.cistrome.org/): The website allows users to upload their enriched regions, returning TF ChIP-seq, DNase-seq or ATAC-seq samples with similar profiles.

### 17.7.4 Motif Analysis

* [MEME Suite](https://meme-suite.org/meme/): The MEME Suite is a comprehensive suite of tools for motif analysis, including motif discovery and motif-based sequence analysis. It includes tools for discovering de novo motifs from ChIP-Seq data and for searching for known motifs in the regions bound by the protein of interest.
* [HOMER](http://homer.ucsd.edu/homer/) is a suite of tools for motif discovery and analysis. It includes tools for identifying de novo motifs from ChIP-Seq data, as well as for searching for known motifs in the regions bound by the protein of interest. HOMER also provides tools for performing gene ontology analysis and pathway analysis based on the identified motifs.
* [MEME-ChIP](https://meme-suite.org/meme/tools/meme-chip) is a specialized version of the MEME Suite that is specifically designed for motif analysis in ChIP-Seq data. It includes tools for discovering de novo motifs from ChIP-Seq data, as well as for searching for known motifs in the regions bound by the protein of interest.
* [CentriMo](https://meme-suite.org/meme/doc/centrimo.html)is a tool for identifying enriched motifs in ChIP-Seq data based on the position of the motif relative to the peak summit. It can be used to identify motifs that are enriched at the center of the peak, as well as those that are enriched near the edges of the peak.

### 17.7.5 Tools for preprocessing

* [Trimmomatic](http://www.usadellab.org/cms/?page=trimmomatic) is a widely used tool for trimming and filtering Illumina sequencing data. It is often used to remove low-quality reads, adapter sequences, and other artifacts that can affect downstream analysis.
* [Cutadapt](https://cutadapt.readthedocs.io/en/stable/) is another popular tool for trimming adapter sequences from high-throughput sequencing data. It is particularly useful for removing adapters that contain degenerate nucleotides or that have been ligated with variable lengths.
* [Bowtie2](https://bowtie-bio.sourceforge.net/bowtie2/manual.shtml) is a fast and memory-efficient tool for aligning sequencing reads to a reference genome. It is often used to map ChIP-Seq reads to the genome prior to peak calling.
* [SAMtools](http://www.htslib.org/) is a suite of tools for manipulating SAM/BAM files, which are commonly used to store alignment data from high-throughput sequencing experiments. It can be used for filtering and sorting reads, as well as for generating summary statistics.
* [BEDTools](http://www.htslib.org/) is a powerful suite of tools for working with genomic intervals, such as those generated by ChIP-Seq peak calling. It can be used for operations such as intersecting, merging, and subtracting intervals.

### 17.7.6 Tools for making visualizations

* [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a popular genome browser that is widely used for the visualization of genomic data, including ChIP-Seq data. It provides a user-friendly interface for exploring genomic data at different levels of resolution, from the whole-genome level down to individual nucleotides.
* The [UCSC Genome Browser](https://genome.ucsc.edu/) is another widely used genome browser that can be used to visualize ChIP-Seq data. It provides an intuitive interface for navigating and visualizing genomic data, including the ability to zoom in and out and to overlay multiple data tracks.
* [Genome Visualization Tool (GViz)](http://www.sthda.com/english/wiki/gviz-visualize-genomic-data) is a package for the R statistical computing environment that provides functions for generating publication-quality visualizations of genomic data, including ChIP-Seq data. It offers a high degree of flexibility and customization, allowing users to create complex and informative plots that convey the relevant information in a clear and concise manner.
* [UCSC Xena](http://xena.ucsc.edu/) is a web-based visualization tool for multi-omic data and associated clinical and phenotypic annotations. It can be used with ChIP-seq data.
* [Cistrome-Explorer](http://gehlenborglab.org/research/projects/cistrome-explorer/) A web-based visualization of compendia of ATAC-seq and histone modification ChIP-seq data for diverse samples, represented as a heatmap. Users can upload their ChIP-seq peak sets to assess the tissue specificity of their regions on the genome.

### 17.7.7 Tools for making heatmaps

* [Deeptools](https://github.com/fidelram/deepTools) is a widely used package for analyzing ChIP-seq data, and it includes a tool called “plotHeatmap” that can generate heatmaps from ChIP-seq data.
* [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a popular tool for visualizing and exploring genomic data. It includes a heatmap function that can be used to generate heatmaps from ChIP-seq data.
* [EnrichedHeatmap](https://bioconductor.org/packages/release/bioc/html/EnrichedHeatmap.html)is an R package for making heatmaps that visualize the enrichment of genomic signals on specific target regions.
* [SeqMonk](https://www.bioinformatics.babraham.ac.uk/projects/seqmonk/) is a software package designed for the visualization and analysis of large-scale genomic data. It includes a heatmap function that can generate heatmaps from ChIP-seq data.
* [ngs.plot](https://github.com/shenlab-sinai/ngsplot) is a tool that can generate different types of plots, including heatmaps, from NGS data. It includes a ChIP-seq specific mode that can be used to generate heatmaps from ChIP-seq data.
* [ChAsE: ChAsE (ChIP-seq Analysis Engine)](http://chase.cs.univie.ac.at/overview) is a web-based platform for ChIP-seq analysis that includes a heatmap function that can generate heatmaps from ChIP-seq data.

These tools allow users to generate heatmaps of ChIP-seq data, which can be used to identify enriched regions of binding and to visualize patterns of binding across genomic regions.

The Cistrome Project has a large collection of human and mouse ChIP-seq, DNase-seq and ATAC-seq data, as well as tools for analyzing user generate ChIP-seq data with publicly available samples. These tools include the Cistrome Data Browser toolkit function that can find publicly available datasets that are similar to a ChIP-Seq peak set, and Cistrome-GO for gene ontology analysis of TF ChIP-seq target genes.

## 17.8 More resources about ChiP-seq data

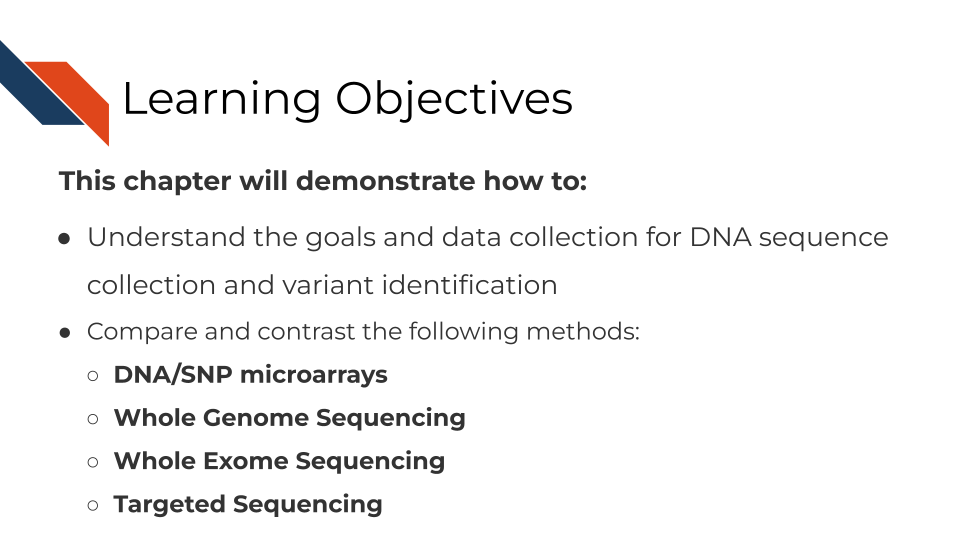
<TODO: Put links to any resources and tutorials that are useful for ChIP-Seq data>

* [Shirley Liu’s Computational biology course](https://liulab-dfci.github.io/bioinfo-combio/chip.html)
* [Galaxy ChIP-seq tutorial](https://galaxyproject.org/tutorials/chip/)
* [ENCODE ChiP-seq tutorial](https://www.encodeproject.org/tutorials/chip-pipeline-howto/)
* [Crazyhottommy’s ChIp-seq tutorial](https://github.com/crazyhottommy/ChIP-seq-analysis)
* [Harvard CUT&RUN tutorial](https://hbctraining.github.io/Intro-to-ChIPseq-flipped/CUT&RUN/Intro_to_CUT&RUN.html)
* [4DN CUT&RUN tutorial](https://data.4dnucleome.org/resources/data-analysis/cut-and-run-pipeline)
* [Henikoff Lab CUT&Tag tutorial](https://yezhengstat.github.io/CUTTag_tutorial/)
* [ARCHS4](https://maayanlab.cloud/archs4/) (All RNA-seq and ChIP-seq sample and signature search) is a resource that provides access to gene and transcript counts uniformly processed from all human and mouse RNA-seq experiments from GEO and SRA.
* [UCSC Xena](http://xena.ucsc.edu/) is a web-based visualization tool for multi-omic data and associated clinical and phenotypic annotations. It can be used with ChIP-seq data.
* [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a track-based browser for interactively exploring genomic data mapped to a reference genome.

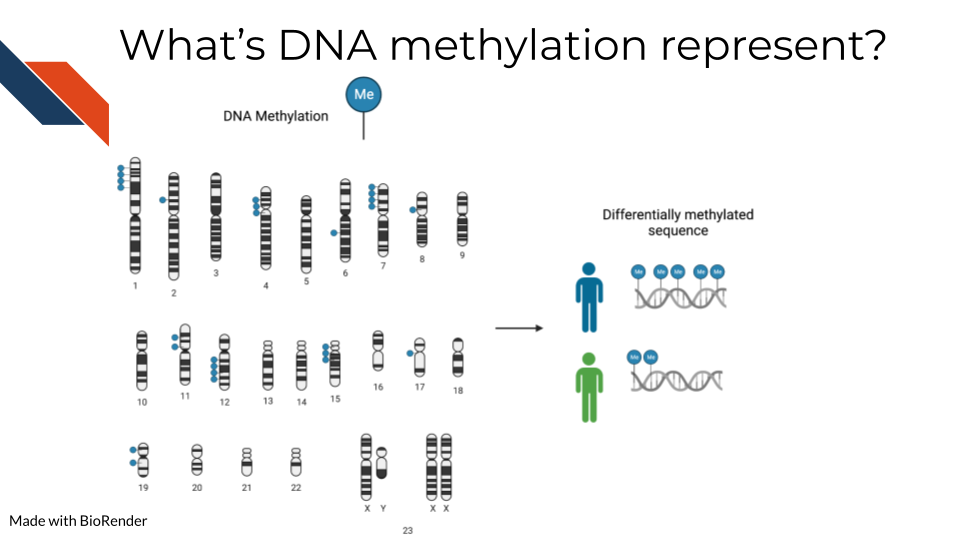
# 18 DNA Methylation Sequencing

This chapter is incomplete! If you wish to contribute, please [go to this form](https://forms.gle/dqYgmKH8XXE2ohwD9) or our [GitHub page](https://github.com/fhdsl/Choosing_Genomics_Tools).

## 18.1 Learning Objectives



## 18.2 What are the goals of analyzing DNA methylation?

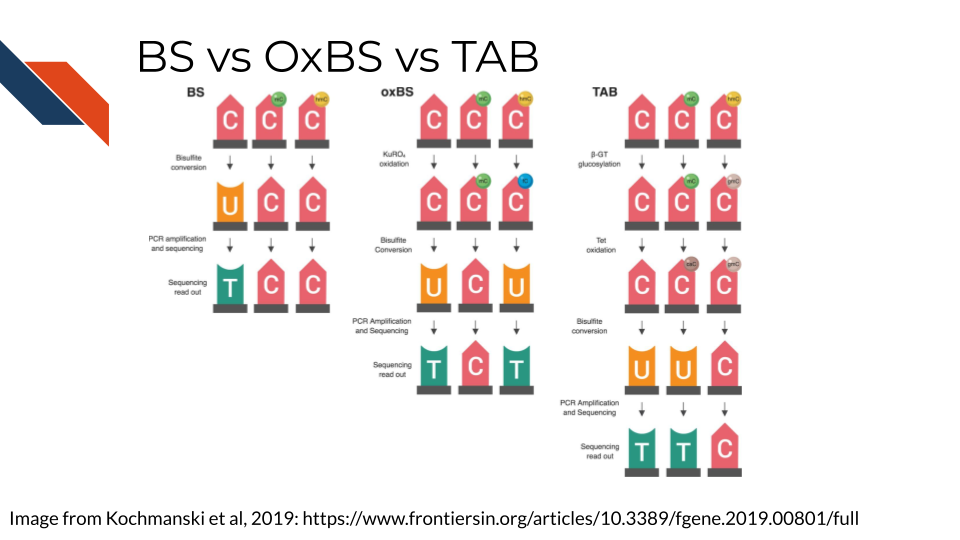


To detect methylated cytosines (5mC), DNA samples are prepped using bisulfite (BS) conversion. This converts unmethylated cytosines into uracils and leaves methylated cytosines untouched. Probes are then designed to bind to either the uracil or the cytosine, representing the unmethylated and methylated cytosines respectively.

For a given sample, you will obtain a fraction, known as the Beta value, that indicates the relative abundance of the methylated and unmethylated versions of the sequence. Beta values exist then on a scale of 0 to 1 where 0 indicates none of this particular base is methylated in the sample and 1 indicates all are methylated.

Note that bisulfite conversion alone will not distinguish between 5mC and 5hmC though these often may indicate different biological mechanics.

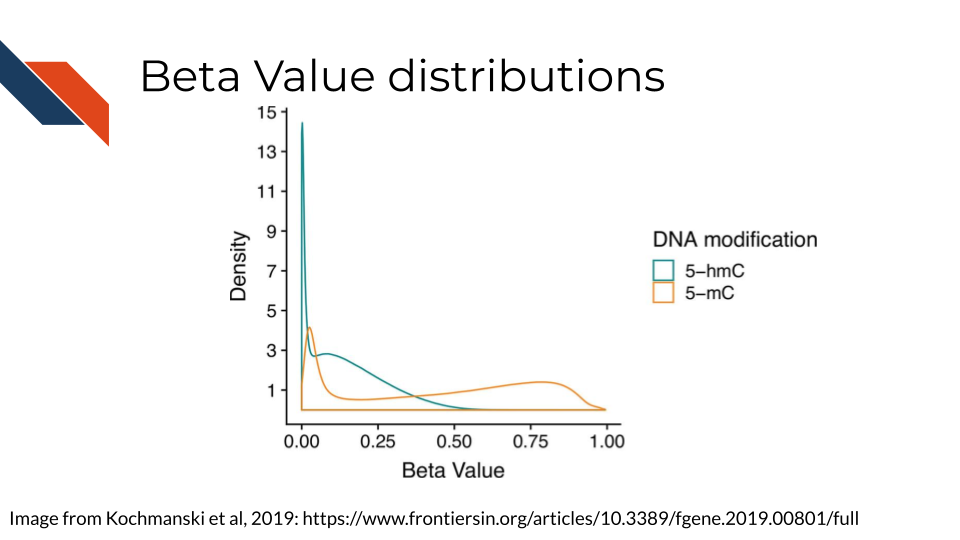
Additionally, 5-hydroxymethylated cytosines (5hmC) can also be detected by oxidative bisulfite sequencing (OxBS) [Booth et al. ([2013](#ref-Booth2013)). oxidative bisulfite conversion measures both 5mC and 5hmC. If you want to identify 5hmC bases you either have to pair oxBS data with BS data OR you have to use Tet-assisted bisulfite (TAB) sequencing which will exclusively tag 5hmC bases ([Yu et al. 2012](#ref-Yu2012)).



## 18.3 Methylation data considerations

### 18.3.1 Beta values binomially distributed

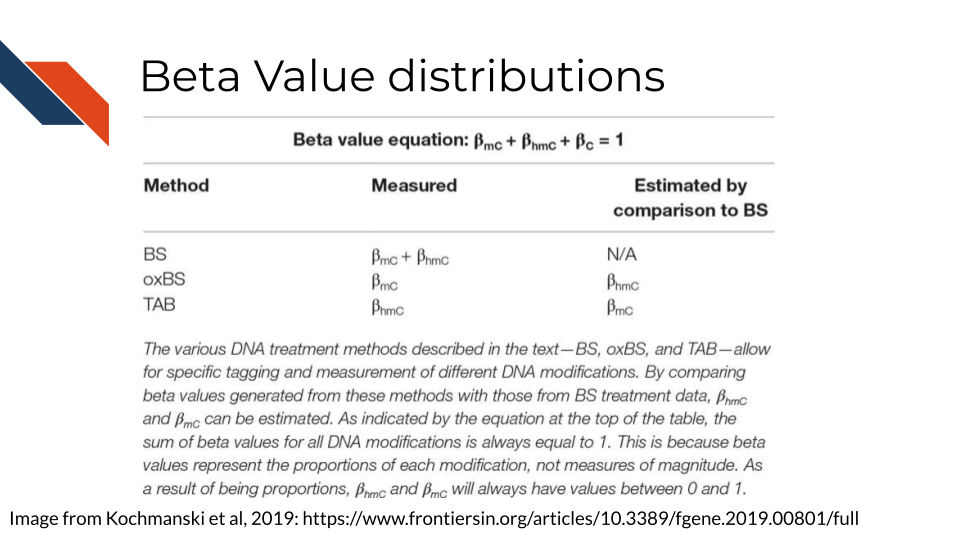
Because beta values are a ratio, by their nature, they are not normally distributed data and should be treated appropriately. This means data models (like those used by the limma package) built for RNA-seq data should not be used on methylation data. More accurately, Beta values follow a binomial distribution.



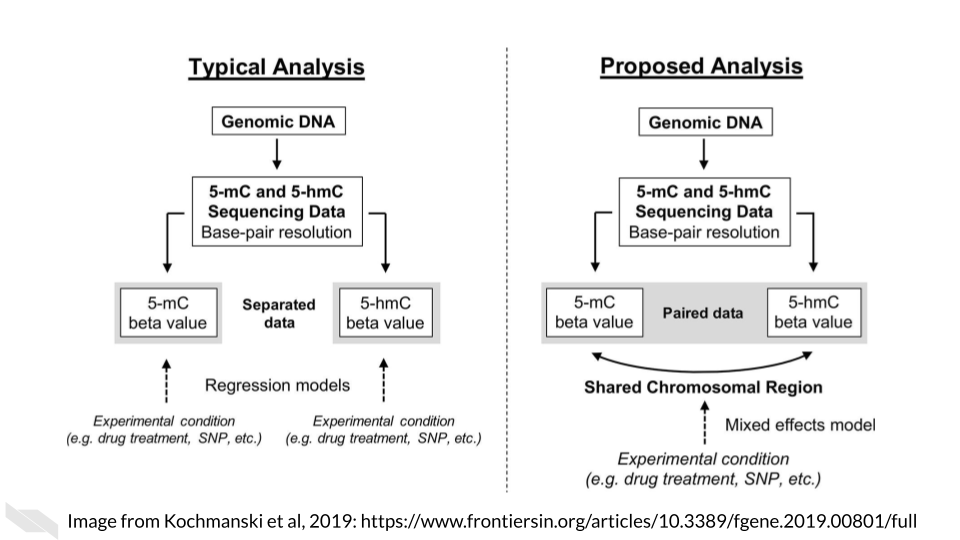
This generally involves applying a generalized linear model.

### 18.3.2 Measuring 5mC and/or 5hmC

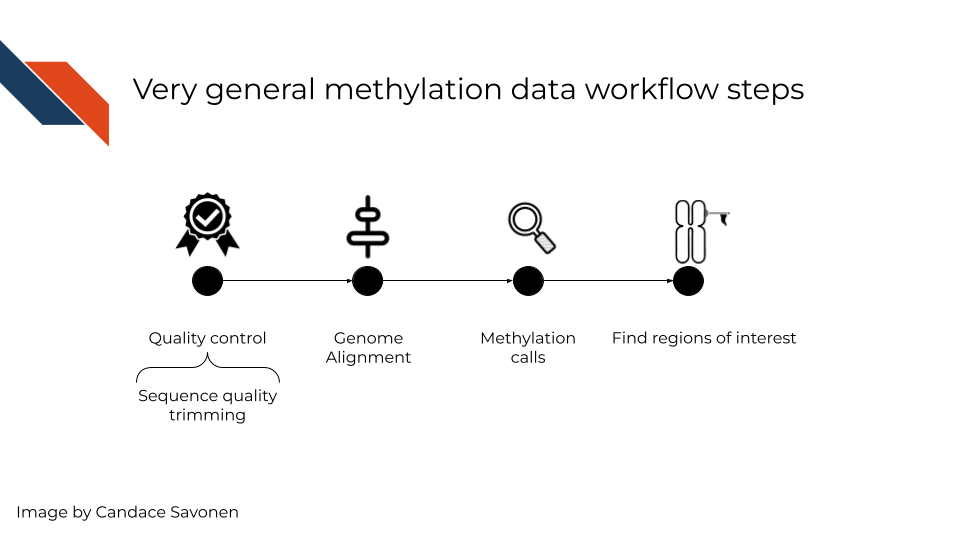
If your data and questions are interested in both 5mC and 5hmC, you will have separate sequencing datasets for each sample for both the BS and OBS processed samples. 5mC is often a step toward 5hmC conversion and therefore the 5mC and 5hmC measurements are, by nature, not independent from each other. In theory, 5mC, 5hmC and unmethylated cytosines should add up to 1.



Because of this, its been proposed that the most appropriate way to model these data is to combine them together in a model ([Kochmanski, Savonen, and Bernstein 2019](#ref-Kochmanski2019)).



## 18.4 Methylation data workflow



Like other sequencing methods, you will first need to start by quality control checks. Next, you will also need to align your sequences to the genome. Then, using the base calls, you will need to make methylation calls – which are methylated and which are not. This details of step depends on whether you are measuring 5mC and/or 5hmC methylation calls. Lastly, you will likely want to use your methylation calls as a whole to identify differentially methylated regions of interest.

## 18.5 Methylation Tools Pros and Cons

This following pros and cons sections have been written by AI and may need verification by experts. This is meant to give you a basic idea of the pros and cons of these tools but should ultimately be used with your own judgment.

### 18.5.1 Quality control:

* [FastQC](https://www.bioinformatics.babraham.ac.uk/projects/fastqc/): A popular tool for evaluating the quality of sequencing reads, generating various quality control plots and statistics. It is fast, easy to use and has a simple user interface ([Andrews, n.d.](#ref-andrews2010fastqc)).
  + **Pros**: Fast and easy to use. Very commonly used. Provides various quality control metrics and plots. Can generate reports that can be easily shared with collaborators
  + **Cons**: Does not perform any trimming or filtering of low-quality reads Not specifically designed for bisulfite sequencing data
* [Trim Galore!](https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/): A wrapper tool for Cutadapt and FastQC that provides a simple way to trim adapters and low-quality reads. It also has built-in support for bisulfite sequencing data ([Krueger and Andrews, n.d.](#ref-krueger2015trim)).
  + **Pros**: Easy to use, with a simple command line interface. Automatically trims adapters and low-quality reads. Specifically designed for bisulfite sequencing data
  + **Cons**: Limited flexibility in terms of the trimming and filtering options. Does not provide quality control metrics or plots

### 18.5.2 Analysis:

* [Bismark](https://www.bioinformatics.babraham.ac.uk/projects/bismark/): A widely used tool for aligning bisulfite sequencing reads to a reference genome. It allows for paired-end and single-end reads, provides many options for handling sequencing errors and can output methylation calls in various formats ([Liu et al. 2019](#ref-liu2019bismark)).
  + **Pros**: Performs alignment, quantification and methylation calling in a single tool. Can output methylation calls in various formats. Provides many options for handling sequencing errors and optimizing methylation calling parameters
  + **Cons**:Can be computationally intensive for large datasets. Requires a pre-built bisulfite-converted reference genome
* [Bowtie2](https://github.com/BenLangmead/bowtie2): A fast and efficient aligner that can be used for bisulfite sequencing data, and can align reads to bisulfite-converted genomes or to an unconverted genome with a pre-built bisulfite index ([Langmead and Salzberg 2012](#ref-langmead2012fast)).
  + **Pros**: Very fast and efficient, making it suitable for large datasets. Can align reads to either a bisulfite-converted genome or to an unconverted genome with a pre-built bisulfite index. Provides options for handling sequencing errors and optimizing alignment parameters
  + **Cons**: Does not perform methylation calling or quantification

### 18.5.3 Methylation calling:

* [Bismark](https://www.bioinformatics.babraham.ac.uk/projects/bismark/): As well as performing alignment, Bismark can also be used to call methylation from aligned reads. It reports the percentage of cytosines methylated at each site ([Liu et al. 2019](#ref-liu2019bismark)).
  + **Pros**: Performs both alignment and methylation calling in a single tool. Can output methylation calls in various formats. Provides many options for handling sequencing errors and optimizing methylation calling parameters
  + **Cons**:Can be computationally intensive for large datasets. Requires a pre-built bisulfite-converted reference genome
* [MethylDackel](https://github.com/dpryan79/MethylDackel): A fast and efficient tool for methylation calling from bisulfite sequencing data. It can output methylation calls in various formats, including a methylation bedGraph.
  + **Pros**: Very fast and efficient, making it suitable for large datasets. Provides options for handling sequencing errors and optimizing methylation calling parameters. Can output methylation calls in various formats, including a methylation bedGraph
  + **Cons**:Does not perform alignment or methylation quantification

### 18.5.4 Methylation quantification:

* [MethylKit](https://www.bioconductor.org/packages/release/bioc/html/methylKit.html): A popular tool for quantifying methylation levels from bisulfite sequencing data. It can handle various types of data and provides options for filtering out low-quality data and detecting differentially methylated regions ([Akalin et al. 2012](#ref-akalin2012methylome)).
  + **Pros**: Provides various options for filtering out low-quality data and detecting differentially methylated regions. Can handle various types of data, including bisulfite sequencing and reduced representation bisulfite sequencing. Provides many visualization tools for analyzing methylation data
  + **Cons**: Can be computationally intensive for large datasets. Requires some knowledge of R programming language to use effectively
* [Bismark](https://www.bioinformatics.babraham.ac.uk/projects/bismark/): As well as methylation calling, Bismark can also quantify methylation levels at each cytosine site. It reports the number of methylated and unmethylated reads, as well as the percentage of methylation ([Liu et al. 2019](#ref-liu2019bismark)).

### 18.5.5 Analysis:

* [DSS](http://www.bioconductor.org/packages/release/bioc/vignettes/DSS/inst/doc/DSS.html): A popular tool for identifying differentially methylated regions (DMRs) between groups of samples. It uses a statistical model to detect significant changes in methylation levels and reports DMRs with associated p-values ([Feng and Conneely 2016](#ref-feng2014dss)).
  + **Pros**: Uses a statistical model to identify differentially methylated regions between groups of samples. Provides various options for controlling false discovery rate and adjusting for multiple comparisons. Suitable for large datasets.
  + **Cons**: Requires some knowledge of statistical methods and programming language to use effectively. May not be suitable for smaller datasets or datasets with low coverage.
* [MethylKit](https://www.bioconductor.org/packages/release/bioc/html/methylKit.html): As well as methylation quantification, MethylKit can also be used for downstream analysis, such as clustering samples based on methylation patterns and performing functional annotation of differentially methylated regions ([Akalin et al. 2012](#ref-akalin2012methylome)).

## 18.6 More resources

* [DNA methylation analysis with Galaxy tutorial](https://training.galaxyproject.org/training-material/topics/epigenetics/tutorials/methylation-seq/tutorial.html)
* [The mint pipeline](https://github.com/sartorlab/mint/blob/master/README.md) for analyzing methylation and hydroxymethylation data.
* [Book chapter about finding methylation regions of interest](https://compgenomr.github.io/book/extracting-interesting-regions-differential-methylation-and-segmentation.html)

# 19 ITCR -omic Tool Glossary

Here’s all the tools that have been mentioned in this course or are otherwise recommended for your use. The list is in alphabetical order.

* [ARCHS4](#archs4)
* [Bioconductor](#bioconductor)
  + [Notable Bioconductor genomics tools:](#notable-bioconductor-genomics-tools-)
* [Cancer Models](#cancer-models)
* [CIViC](#civic)
* [CTAT](#ctat)
* [DeepPhe](#deepphe)
* [Genetic Cancer Risk Detector (GARDE)](#genetic-cancer-risk-detector--garde-)
* [GenePattern](#genepattern)
* [Gene Set Enrichment Analysis (GSEA)](#gene-set-enrichment-analysis--gsea-)
* [Integrative Genomics Viewer (IGV)](#integrative-genomics-viewer--igv-)
* [NDEx](#ndex)
* [MultiAssayExperiment](#multiassayexperiment)
* [OpenCRAVAT](#opencravat)
* [pVACtools](#pvactools)
* [TumorDecon](#tumordecon)
* [WebMeV](#webmev)
* [Xena](#xena)

## 19.1 ARCHS4

All RNA-seq and ChIP-seq sample and signature search (ARCHS4) (<https://maayanlab.cloud/archs4/>) is a resource that provides access to gene and transcript counts uniformly processed from all human and mouse RNA-seq experiments from GEO and SRA. The ARCHS4 website provides the uniformly processed data for download and programmatic access in H5 format, and as a 3-dimensional interactive viewer and search engine. Users can search and browse the data by metadata enhanced annotations, and can submit their own gene sets for search. Subsets of selected samples can be downloaded as a tab delimited text file that is ready for loading into the R programming environment. To generate the ARCHS4 resource, the kallisto aligner is applied in an efficient parallelized cloud infrastructure. Human and mouse samples are aligned against the most recent Ensembl annotation (Ensembl 107).

## 19.2 Bioconductor

The mission of the Bioconductor project is to develop, support, and disseminate free open source software that facilitates rigorous and reproducible analysis of data from current and emerging biological assays. We are dedicated to building a diverse, collaborative, and welcoming community of developers and data scientists.

Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, and an active user community. Bioconductor is also available as Docker images.

### 19.2.1 Notable Bioconductor genomics tools:

* [annotatr](https://bioconductor.org/packages/release/bioc/html/annotatr.html)
* [ensembldb](https://bioconductor.org/packages/release/bioc/html/ensembldb.html)
* [GenomicRanges](https://www.bioconductor.org/packages/release/bioc/html/GenomicRanges.html) - useful for manipulating and identifying sequences.
* [GO.db](https://bioconductor.org/packages/3.16/data/annotation/html/GO.db.html) - Gene ontology annotation
* [org.Hs.eg.db](https://bioconductor.org/packages/release/data/annotation/html/org.Hs.eg.db.html)
* [RSamtools](https://www.bioconductor.org/packages/release/bioc/html/Rsamtools.html)
* [A full list of Bioconductors annotation packages](https://bioconductor.org/packages/3.16/data/annotation/) - contains annotation for all kinds of species and versions of genomes and transcriptomes.
* [ComplexHeatmap](https://bioconductor.org/packages/release/bioc/html/ComplexHeatmap.html#:~:text=Complex%20heatmaps%20are%20efficient%20to,and%20supports%20various%20annotation%20graphics.)
* [MultiAssayExperiment](https://bioconductor.org/packages/MultiAssayExperiment/)
* [limma](https://www.bioconductor.org/packages/release/bioc/html/limma.html)
* [DESEq2](https://www.bioconductor.org/packages/release/bioc/html/DESeq2.html)
* [edgeR](https://www.bioconductor.org/packages/release/bioc/html/edgeR.html)
* [curatedTCGAData](https://bioconductor.org/packages/curatedTCGAData/)
* [cBioPortalData](https://bioconductor.org/packages/cBioPortalData/)
* [SingleCellMultiModal](https://bioconductor.org/packages/SingleCellMultiModal/)

## 19.3 Cancer Models

Patient Derived Cancer Models Finder (www.cancermodels.org) is a cancer research platform that aggregates clinical, genomic and functional data from patient-derived xenografts, organoids and cell lines. The PDCM Finder standardises, harmonises and integrates the complex and diverse data associated with PDCMs for cancer community.

Data types used are model meta data, related clinical metadata from the sample for which the model was derived, e.g. molecular and treatment-based. Data are preprocessed, consistently semantically annotated, harmonised and FAIR.

PDCM Finder contains >6200 models across 13 cancer types, including rare pediatric models (17%) and models from minority ethnic backgrounds (33%), making it the largest free to consumer and open access resource of this kind. Get started at www.cancermodels.org to browse and query models by cancer type

## 19.4 CIViC

[CIViC](www.civicdb.org) is a knowledgebase and curation interface for the clinical interpretation of variants in cancer. Evidence is curated from published literature describing the diagnostic, prognostic, predictive, predisposing, oncogenic, or functional role of variants in specific cancer types. Evidence submitted by community curators is revised and moderated by expert editors. Individual evidence is synthesized into gene summaries, variant summaries and variant-disease assertions of specific clinical relevance. Anyone can make use of CIViC knowledge through the open web interface or API. Information on how to use or contribute to CIViC is available in our help docs (docs.civicdb.org). The main distinguishing feature of CIViC compared to similar resources it is total commitment to open data sharing. All data are available in the Public Domain (CC0). The code is available for any use under an MIT license.

## 19.5 CTAT

The Trinity Cancer Transcriptome Analysis Toolkit (CTAT, <https://github.com/NCIP/Trinity_CTAT/wiki>) provides a diverse collection of tools to gain insights into the biology of cancer through the lens of the transcriptome. Using RNA-seq as input, CTAT modules enable detection of mutations, fusion transcripts, copy number aberrations, cancer-specific splicing aberrations, and oncogenic viruses including insertions into the human genome. CTAT uses both read mapping and de novo assembly methods to analyze RNA-seq, leveraging tumor bulk and single cell transcriptomes. CTAT modules provide interactive visualizations as outputs, are easily installed for local execution or run via cloud computing (eg. Terra), have detailed user guides and tutorials, and are well-supported through user forums.

## 19.6 DeepPhe

[DeepPhe: Natural Language Processing Tools for Cancer Research](https://deepphe.github.io/)

Under development since 2014, the DeepPhe suite of software tools aims to extract deep phenotype information from the Electronic Medical Records from patients with cancer. DeepPhe combines:

1. multiple natural language processing (NLP) techniques based on cTAKES,1
2. a structured cancer information model including concepts from the NCIT and the HemOnc ontology
3. a graph data model supporting persistence of extracted details including links between patient data enabling semantically informed interpretation, aggregation, and disaggregation of key attributes,
4. visual analytics tools supporting patient- and cohort-level displays of extracted data5 including identification of patients matching key research criteria and the examination of individual patient records such as exploration of links between summary items and supporting text mentions, and
5. multiple strategies for use, including containerized REST services and GUIs for installation and pipeline execution.

DeepPhe tools are available for download and installation from the [DeepPhe website](https://deepphe.github.io/) under an open-source license for non-commercial use.

## 19.7 Genetic Cancer Risk Detector (GARDE)

[Genetic Cancer Risk Detector (GARDE)](https://reimagineehr.utah.edu/innovations/garde/) screens and identifies patients who meet National Comprehensive Cancer Network (NCCN) criteria for genetic evaluation of familial cancer risk based on their family history in the EHR using both structured data and natural language processing of free-text data. Patients identified by GARDE are imported into an EHR’s population health management dashboard (e.g., Epic’s Healthy Planet module) where genetic counseling staff review individual cases, select, and send bulk outreach messages to patients via chatbot and/or through the patient portal.

GARDE is a population clinical decision support (CDS) platform based on Fast Healthcare Interoperability Resources (FHIR) and CDS Hooks standards to support interoperability and logic sharing beyond single vendor solutions.

## 19.8 GenePattern

GenePattern, www.genepattern.org, is an open software environment providing access to hundreds of tools for the analysis and visualization of genomic data. Analyses include general machine learning methods, the gene set enrichment analysis suite, ’omics-specific tools for bulk and single-cell gene expression, proteomics, flow cytometry, variant annotation, sequence variation and others, as well as cancer-specific analyses. Also included are data preprocessing and utility tools. A web-based interface provides easy, non-programmatic access to these tools and allows the creation of multi-step analysis pipelines that enable reproducible in silico research.

The GenePattern Notebook interface, notebook.genepattern.org, extends the Jupyter Notebook system to allow users to combine GenePattern analyses with text, graphics, and code to create complete research narratives. It includes many additional features to make notebooks accessible to non-programmers. The online GenePattern Notebook Workspace allows investigators to create, run, and collaborate on notebooks using only a web browser. A library of GenePattern Notebooks implementing common scientific workflows is available for investigators to use as templates and adapt to their own requirements.

To get started with GenePattern you can go through the GenePattern Quick Start Tutorial, view the GenePattern User Guide, or the videos on our YouTube channel. To learn more about GenePattern Notebook, view the GenePattern Notebook Quick Start, GenePattern Notebook documentation, run through the tutorial notebooks (click the Tutorial button), or view the videos on the GenePattern Notebooks YouTube channel.

## 19.9 Gene Set Enrichment Analysis (GSEA)

[Gene Set Enrichment Analysis (GSEA)](https://www.gsea-msigdb.org/gsea/index.jsp) is a method to identify the coordinate activation or repression of groups of genes that share common biological functions, pathways, chromosomal locations, or regulation, thereby distinguishing even subtle differences between phenotypes or cellular states. Gene set-based enrichment analysis is now standard practice for interpreting global transcription profiling experiments and elucidating the biological mechanisms associated with disease and other biological phenotypes of interest. The method is more powerful than typical single-gene approaches to comparing phenotypes, as it can identify sets of genes (e.g., perturbation signatures or molecular pathways) that are coordinately up- or downregulated when each gene in the set may not be significantly differentially expressed. The GSEA software provides useful visualizations and reports for the exploration and interpretation of results. GSEA bundles direct access to the Molecular Signatures Database (MSigDB) – a comprehensive curated repository of annotated gene sets representing signatures derived from publications, pathway databases, and other sources of public data; MSigDB can also be used independently.

The website for the GSEA-MSigDB resource can be found at gsea-msigdb.org. To get started with GSEA you can view the GSEA User Guide, and access the GSEA software through the downloads page or through the GSEA modules available on GenePattern. See the MSigDB section of the website for more information about MSigDB and to interactively explore the gene sets and their annotations. User support for GSEA and MSigDB is available through our help forum.

## 19.10 Integrative Genomics Viewer (IGV)

The [Integrative Genomics Viewer (IGV)](https://software.broadinstitute.org/software/igv/) is a track-based browser for interactively exploring genomic data mapped to a reference genome. IGV supports all the standard genomic data types (aligned reads, variants, signal peaks, genome annotations, copy number variation, etc.) as well as sample information, such as clinical, phenotypic, or other attributes. IGV provides great flexibility in loading data, whether investigator generated or publicly available, directly from multiple disparate sources without the need for any pre-processing. Supported data sources include local file systems; web servers on the user’s intranet or the Internet; commercial cloud providers (Google, Amazon, Azure, Dropbox); web links to data in public repositories. Authentication to access private data on the web is supported with the industry standard OAuth protocol.

IGV is available in multiple forms, including both end-user applications and versions for use by developers. The IGV website at <https://igv.org> provides access to all modalities of IGV. Download and install the IGV Desktop application from the downloads page. To learn about using the application see the tutorial videos on the IGV YouTube channel and the online User Guide. The IGV-Web app is available at <https://igv.org/app>. To learn about using the app, the Help link in the menu bar provides access to the documentation, and see also the tutorial videos on the YouTube channel. The igv.js JavaScript component is for web developers who wish to embed IGV in their web apps or portals. More information can be found in the Readme file and the Wiki in the igv.js GitHub repository. IGV user support is available through the igv-help online forum and the GitHub repositories.

## 19.11 NDEx

The [Network Data Exchange (NDEx)](https://www.ndexbio.org/#/) project provides an open-source framework where scientists and organizations can store, share and publish biological network knowledge. A distinctive feature of NDEx is that it serves as a home for models that are currently available only as figures, tables, or supplementary information, such as networks produced via systematic mining and integration of large-scale molecular data.

NDEx includes features to support data distribution and access according to FAIR principles. Its full integration with Cytoscape, the popular desktop application for network analysis and visualization, provides the cloud back-end component for data I/O; so, if a network file format can be opened in Cytoscape, it can also be stored in (and retrieved from) NDEx.

NDEx can be accessed via its web user interface or programmatically, via REST API and client libraries in Python, R, Java. Web applications can interface with NDEx via JavaScript: MSigDB, CRAVAT, cBioPortal and IQuery, are all examples of web applications integrated with NDEx.

For more information, please review the About NDEx page. To get started, visit the NDEx public server: there, you can review the NDEx FAQ, access documentation, contact us, and search or browse thousands of biological network models.

## 19.12 MultiAssayExperiment

[MultiAssayExperiment](https://bioconductor.org/packages/MultiAssayExperiment/) is an R/Bioconductor package that harmonizes data management, manipulation, and subsetting of multiple experimental assays performed on an overlapping set of specimens. It supports on-disk and remote data storage, and provides reshaping tools for adaptability to arbitrary downstream analysis.

MultiAssayExperiment is distinct from alternative approaches in its focus on multi’omic data management and manipulation and in its integration with the Bioconductor ecosystem: it is used by more than 50 other Bioconductor packages, it provides a familiar Bioconductor user experience by extending concepts from SummarizedExperiment while supporting an open-ended mix of data classes for individual assays, and it allows subsetting by genomic ranges, row names, phenotypic data, and assays.

You can get started with the MultiAssayExperiment Bioconductor package documentation, or start with prebuilt MultiAssayExperiments objects from [curatedTCGAData](https://bioconductor.org/packages/curatedTCGAData/), [cBioPortalData](https://bioconductor.org/packages/cBioPortalData/), or [SingleCellMultiModal](https://bioconductor.org/packages/SingleCellMultiModal/).

## 19.13 OpenCRAVAT

[OpenCRAVAT](https://run.opencravat.org) uses variation data in many popular variant file formats and its outputs are variant annotations and visualizations. To get started go to opencravat.org. Download and run on your local machine, multi-user servers, at <https://run.opencravat.org> or in the cloud. We offer a broader selection of annotation tools than comparable software and results can be explored with an interactive GUI that provides customized filtering options, interactive tables and widgets. Use it for a single sample or a large cohort, or pull single variant reports with a structured url (Example: <https://run.opencravat.org/webapps/variantreport/index.html?chrom=chr11&pos=48123823&ref_base=A&alt_base=C> )

## 19.14 pVACtools

Identification of neoantigens is a critical step in predicting response to checkpoint blockade therapy and design of personalized cancer vaccines. We have built a computational framework called pVACtools that, when paired with a well-established genomics pipeline, produces an end-to-end solution for neoantigen characterization. pVACtools supports identification of altered peptides from different mechanisms, including point mutations, in-frame and frameshift insertions and deletions, and gene fusions. Prediction of peptide:MHC binding is accomplished by supporting an ensemble of MHC Class I and II binding algorithms within a framework designed to facilitate the incorporation of additional algorithms. Prioritization of predicted peptides occurs by integrating diverse data, including mutant allele expression, peptide binding affinities, and determination whether a mutation is clonal or subclonal. Interactive visualization via a Web interface allows clinical users to efficiently generate, review, and interpret results, selecting candidate peptides for individual patient vaccine designs. Additional modules support design choices needed for competing vaccine delivery approaches. One such module optimizes peptide ordering to minimize junctional epitopes in DNA vector vaccines. Downstream analysis commands for synthetic long peptide vaccines are available to assess candidates for factors that influence peptide synthesis. All of the aforementioned steps are executed via a modular workflow consisting of tools for neoantigen prediction from somatic alterations (pVACseq and pVACfuse), prioritization, and selection using a graphical Web-based interface (pVACview), and design of DNA vector–based vaccines (pVACvector) and synthetic long peptide vaccines. pVACtools is available at <http://www.pvactools.org>.

## 19.15 TumorDecon

It is only software that includes these four digital cytometry methods in one platform, so that users can compare the results of these methods. It is the only software that includes a method for creating signature matrix from single cell gene expression data.

TumorDecon software includes four deconvolution methods (DeconRNAseq [Gong2013], CIBERSORT [Newman2015], ssGSEA [Şenbabaoğlu2016], Singscore [Foroutan2018]) and several signature matrices of various cell types, including LM22. The input of this software is the gene expression profile of the tumor, and the output is the relative number of each cell type and several visualization plots. Users have an option to choose any of the implemented deconvolution methods and included signature matrices or import their own signature matrix to get the results. Additionally, TumorDecon can be used to generate customized signature matrices from single-cell RNA-sequence profiles.

In addition to the 3 tutorials provided on GitHub (tutorial.py, sig\_matrix\_tutorial.py, & full\_tutorial.py) there is a User Manual available at: <https://people.math.umass.edu/~aronow/TumorDecon>

TumorDecon is available on Github (<https://github.com/ShahriyariLab/TumorDecon>) and PyPI (<https://pypi.org/project/TumorDecon/>).

For more info please see: Rachel A. Aronow, Shaya Akbarinejad, Trang Le, Sumeyye Su, Leili Shahriyari, TumorDecon: A digital cytometry software, SoftwareX, Volume 18, 2022, 101072, <https://doi.org/10.1016/j.softx.2022.101072>.

## 19.16 WebMeV

[WebMeV](https://webmev.tm4.org) is an online tool that facilitates analysis of large-scale RNA-seq and other multi-omic datasets by providing intuitive access to advanced analytical methods and high-performance computing for a wide range of basic, clinical, and translational researchers. Although WebMeV provides support for “bulk” RNA-seq data, single-cell RNA-seq, and other types of -omic data and provides easy access to public data resources such as The Cancer Genome Atlas (TCGA) and the Genotype-Tissue Expression project (GTEx)—as well as user-provided data. WebMeV uniquely provides a user-friendly, intuitive, interactive interface to processed analytical data uses cloud-computing elasticity for computationally intensive analyses that are increasingly required for genomic data analysis. WebMeV’s design places an emphasis on user-driven data analysis by providing users the ability to visualize, interact with, and dissect genomic data at each step in the analysis with a “point-and-click” interactive data environment. Although the primary input is normalized “count matrices,” WebMeV does include tools for data normalization and quality control and uses Dropbox and Google Drive as means of easily uploading data. Analytical methods include statistical tests for comparing cohorts, for identifying gene seats, for doing functional enrichment analysis on gene sets (GSEA), and for inferring gene regulatory network models and comparing these networks between phenotypes to understand the drivers of disease. WebMeV also provides a platform to support reproducible research and makes code for the entire system and its component methods available as open-source software code.

## 19.17 Xena

[UCSC Xena](http://xena.ucsc.edu/) is a web-based visualization tool for multi-omic data and associated clinical and phenotypic annotations.

Xena showcases seminal cancer genomics datasets from TCGA, the Pan-Cancer Atlas, GDC, PCAWG, ICGC, and more; a total of more than 1500 datasets across 50 cancer types. We support virtually any type of functional genomics data (sometimes known as level 3 or 4 data). This includes SNPs, INDELs, copy number variation, gene expression, ATAC-seq, DNA methylation, exon-, transcript-, miRNA-, lncRNA-expression and structural variants. We also support clinical data such as phenotype information, subtype classifications and biomarkers. All of our data is available for download via python or R APIs, or through our URL links.

### 19.17.1 Questions Xena can help you answer include:

* Is overexpression of this gene associated with better survival?
* What genes are differentially expressed between these two groups of samples?
* What is the relationship between mutation, copy number, expression, etc for this gene?

Our tool differentiates itself by its ability to visualize more uncommon data types, such as DNA methylation, its visual integration of multiple types of genomic data side-by-side, and its ability to easily privately visualize your own data.

Get started with our tutorials: <https://ucsc-xena.gitbook.io/project/tutorials>. If you use us please cite us: <https://www.nature.com/articles/s41587-020-0546-8>

# About the Authors

These credits are based on our [course contributors table guidelines](https://github.com/jhudsl/OTTR_Template/wiki/How-to-give-credits).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [Candace Savonen](https://www.cansavvy.com/) |
| Lecturer(s) | [Candace Savonen](https://www.cansavvy.com/) |
| Content Contributor(s) | [Cailin Jordan](https://www.linkedin.com/in/cailin-jordan-3297b7199/) - sc-ATAC-Seq, [Carrie Wright](https://carriewright11.github.io/), [Claire Mills](https://www.linkedin.com/in/claire-mills-dds-50883553) - Whole Genome Sequencing, [Jacob Greene](https://www.linkedin.com/in/jacob-greene-890aa318a/) - ChIP-seq |
| Content Directors | [Jeff Leek](https://jtleek.com/) |
| Content Consultants | [Carrie Wright](https://carriewright11.github.io/), [Cliff Meyer](https://ds.dfci.harvard.edu/our-people/clifford-meyer-phd/) - ATAC-seq, [Frederick Tan](https://bse.carnegiescience.edu/dr-frederick-tan) |
| Acknowledgments |  |
| **Production** |  |
| Content Publisher | [Ira Gooding](https://publichealth.jhu.edu/faculty/4130/ira-gooding) |
| Content Publishing Reviewers | [Ira Gooding](https://publichealth.jhu.edu/faculty/4130/ira-gooding) |
| **Technical** |  |
| Course Publishing Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([ottrpal](https://github.com/jhudsl/ottrpal))[Candace Savonen](https://www.cansavvy.com/), [John Muschelli](https://johnmuschelli.com/), [Carrie Wright](https://carriewright11.github.io/) |  |
| **Art and Design** |  |
| Illustrator | [Candace Savonen](https://www.cansavvy.com/) |
| Figure Artist | [Candace Savonen](https://www.cansavvy.com/) and [Claire Mills](https://www.linkedin.com/in/claire-mills-dds-50883553) |
| Videographer | [Candace Savonen](https://www.cansavvy.com/) |
| Videography Editor | [Candace Savonen](https://www.cansavvy.com/) |
| **Funding** |  |
| Funder | [National Cancer Institute (NCI)](https://www.cancer.gov/) UE5 CA254170 |
| Funding Staff | [Sandy Ormbrek](https://hutchdatascience.org/ourteam/), [Shasta Nicholson](https://hutchdatascience.org/ourteam/) |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.3 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2023-07-11   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.3)   
## bookdown 0.24 2022-02-15 [1] Github (rstudio/bookdown@88bc4ea)   
## callr 3.4.4 2020-09-07 [1] RSPM (R 4.0.2)   
## cli 2.0.2 2020-02-28 [1] RSPM (R 4.0.0)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.14 2019-05-28 [1] RSPM (R 4.0.3)   
## fansi 0.4.1 2020-01-08 [1] RSPM (R 4.0.0)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.6.1 2022-01-22 [1] CRAN (R 4.0.2)   
## htmltools 0.5.0 2020-06-16 [1] RSPM (R 4.0.1)   
## knitr 1.33 2022-02-15 [1] Github (yihui/knitr@a1052d1)   
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)   
## magrittr 2.0.2 2022-01-26 [1] CRAN (R 4.0.2)   
## memoise 1.1.0 2017-04-21 [1] RSPM (R 4.0.0)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.3.4 2020-08-11 [1] RSPM (R 4.0.2)   
## purrr 0.3.4 2020-04-17 [1] RSPM (R 4.0.3)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 0.4.10 2022-02-15 [1] Github (r-lib/rlang@f0c9be5)   
## rmarkdown 2.10 2022-02-15 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.2 2020-11-15 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2022-02-15 [1] Github (R-lib/testthat@e99155a)   
## usethis 2.1.5.9000 2022-02-15 [1] Github (r-lib/usethis@57b109a)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2022-02-15 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# References

Akalin, Altuna, Matthias Kormaksson, Sheng Li, Francine E Garrett-Bakelman, Maria E Figueroa, Ari Melnick, and Christopher E Mason. 2012. “methylKit: A Comprehensive r Package for the Analysis of Genome-Wide DNA Methylation Profiles.” *Genome Biology* 13 (10): R87. <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2012-13-10-r87>.

“Alex’s Lemonade Training Modules.” 2022. <https://github.com/AlexsLemonade/training-modules>.

ALSF, CCDL for. 2019. “Introduction to Microarray Data.” <https://alexslemonade.github.io/refinebio-examples/02-microarray/00-intro-to-microarray.html>.

Andrews, Simon. n.d. “FastQC: A Quality Control Tool for High Throughput Sequence Data.” <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.

Angerer, Philipp, Lukas Simon, Sophie Tritschler, F. Alexander Wolf, David Fischer, and Fabian J. Theis. 2017. “Single Cells Make Big Data: New Challenges and Opportunities in Transcriptomics.” *Current Opinion in Systems Biology* 4 (August): 85–91. <https://doi.org/10.1016/j.coisb.2017.07.004>.

Baran-Gale, Jeanette, Tamir Chandra, and Kristina Kirschner. 2018. “Experimental Design for Single-Cell RNA Sequencing.” *Briefings in Functional Genomics* 17 (4): 233–39. <https://doi.org/10.1093/bfgp/elx035>.

Bentley, David R., Shankar Balasubramanian, Harold P. Swerdlow, Geoffrey P. Smith, John Milton, Clive G. Brown, Kevin P. Hall, et al. 2008. “Accurate Whole Human Genome Sequencing Using Reversible Terminator Chemistry.” *Nature* 456 (7218): 53–59. <https://doi.org/10.1038/nature07517>.

Bewicke-Copley, Findlay, Emil Arjun Kumar, Giuseppe Palladino, Koorosh Korfi, and Jun Wang. 2019. “Applications and Analysis of Targeted Genomic Sequencing in Cancer Studies.” *Computational and Structural Biotechnology Journal* 17: 1348–59. <https://doi.org/10.1016/j.csbj.2019.10.004>.

Booth, Michael J, Tobias W B Ost, Dario Beraldi, Neil M Bell, Miguel R Branco, Wolf Reik, and Shankar Balasubramanian. 2013. “Oxidative Bisulfite Sequencing of 5-Methylcytosine and 5-Hydroxymethylcytosine.” *Nature Protocols* 8 (10): 1841–51. <https://doi.org/10.1038/nprot.2013.115>.

Bray, Nicolas L, Harold Pimentel, Páll Melsted, and Lior Pachter. 2016. “Near-Optimal Probabilistic RNA-Seq Quantification.” *Nature Biotechnology* 34 (5): 525–27. <https://www.nature.com/articles/nbt.3519>.

Brüning, Ralf Schulze, Lukas Tombor, Marcel H. Schulz, Stefanie Dimmeler, and David John. 2021. “Comparative Analysis of Common Alignment Tools for Single Cell RNA Sequencing.” bioRxiv. <https://doi.org/10.1101/2021.02.15.430948>.

Clark, Michael J, Rui Chen, Hugo Y K Lam, Konrad J Karczewski, Rong Chen, Ghia Euskirchen, Atul J Butte, and Michael Snyder. 2011. “Performance Comparison of Exome DNA Sequencing Technologies.” *Nature Biotechnology* 29 (10): 908–14. <https://doi.org/10.1038/nbt.1975>.

Conesa, Ana, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, et al. 2016. “A Survey of Best Practices for RNA-Seq Data Analysis.” *Genome Biology* 17 (1). <https://doi.org/10.1186/s13059-016-0881-8>.

De Pasquale, Elisa, and Sandrine Dudoit. 2019. “DoubletDecon: Deconvoluting Doublets from Single-Cell RNA-Sequencing Data.” *Cell Reports* 29 (6): 1718–1727. e8. <https://www.sciencedirect.com/science/article/pii/S2211124719312860>.

Dobin, Alexander, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. 2013. “STAR: Ultrafast Universal RNA-Seq Aligner.” *Bioinformatics* 29 (1): 15–21. <https://academic.oup.com/bioinformatics/article/29/1/15/272537>.

Feng, Hui, and Karen N Conneely. 2016. “Differential Methylation Analysis for BS-Seq Data Under General Experimental Design.” *Bioinformatics* 32 (2): 289–91. <https://pubmed.ncbi.nlm.nih.gov/26819470/>.

Ghavi-Helm, Yad, Aleksander Jankowski, Sascha Meiers, Rebecca R Viales, Jan O Korbel, and Eileen EM Furlong. 2019. “Highly Rearranged Chromosomes Reveal Uncoupling Between Genome Topology and Gene Expression.” *Nature Genetics* 51 (8): 1272–82.

He, Xiaoyu, Shanyu Chen, Ruilin Li, Xinyin Han, Zhipeng He, Danyang Yuan, Shuying Zhang, Xiaohong Duan, and Beifang Niu. 2020. “Comprehensive Fundamental Somatic Variant Calling and Quality Management Strategies for Human Cancer Genomes.” *Briefings in Bioinformatics* 22 (3). <https://doi.org/10.1093/bib/bbaa083>.

Heinz, S, Christopher Benner, Nathanael Spann, Eric Bertolino, Yen Ching Lin, Petra Laslo, Jia Cheng, et al. 2010. “Simple Combinations of Lineage-Determining Transcription Factors Prime Cis-Regulatory Elements Required for Macrophage and b Cell Identities.” *Molecular Cell* 38 (4): 576–89. <https://pubmed.ncbi.nlm.nih.gov/20513432/>.

Hwang, Kyu-Baek, In-Hee Lee, Honglan Li, Dhong-Geon Won, Carles Hernandez-Ferrer, Jose Alberto Negron, and Sek Won Kong. 2019. “Comparative Analysis of Whole-Genome Sequencing Pipelines to Minimize False Negative Findings.” *Scientific Reports* 9 (1). <https://doi.org/10.1038/s41598-019-39108-2>.

Kim, Daehwan, Ben Langmead, and Steven L Salzberg. 2015. “HISAT: A Fast Spliced Aligner with Low Memory Requirements.” *Nature Methods* 12 (4): 357–60. <https://www.nature.com/articles/nmeth.3317>.

Kochmanski, Joseph, Candace Savonen, and Alison I. Bernstein. 2019 10. <https://doi.org/10.3389/fgene.2019.00801>.

Krueger, Felix, and Simon R Andrews. n.d. “Trim Galore!: A Wrapper Tool Around Cutadapt and FastQC to Consistently Apply Quality and Adapter Trimming to FastQ Files.” <https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/>.

Langmead, Ben, and Steven L Salzberg. 2012. “Fast Gapped-Read Alignment with Bowtie 2.” *Nature Methods* 9 (4): 357–59. <https://www.nature.com/articles/nmeth.1923>.

Liu, Yi, Kimberly D Siegmund, Peter W Laird, and Benjamin P Berman. 2019. “Bismark: A Flexible Aligner and Methylation Caller for Bisulfite-Seq Applications.” *Bioinformatics* 36 (22-23): 5280–82. <https://academic.oup.com/bioinformatics/article/27/11/1571/216956>.

Luecken, Malte D, and Fabian J Theis. 2019. “Current Best Practices in Single‐cell RNA‐seq Analysis: A Tutorial.” *Molecular Systems Biology* 15 (6). <https://doi.org/10.15252/msb.20188746>.

Mantione, K. J., R. M. Kream, H. Kuzelova, R. Ptacek, J. Raboch, J. M. Samuel, and G. B. Stefano. 2014. “Comparing Bioinformatic Gene Expression Profiling Methods: Microarray and RNA-Seq.” *Medical Science Monitor Basic Research* 20 (August): 138–42. <https://doi.org/10.12659/MSMBR.892101>.

McGinnis, Christopher S, Lillian M Murrow, and Zev J Gartner. 2020. “DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors.” *Cell Systems* 8 (4): 329–337. e4. <https://pubmed.ncbi.nlm.nih.gov/30954475/>.

Naj, Adam C., Honghuang Lin, Badri N. Vardarajan, Simon White, Daniel Lancour, Yiyi Ma, Michael Schmidt, et al. 2019. “Quality Control and Integration of Genotypes from Two Calling Pipelines for Whole Genome Sequence Data in the Alzheimers Disease Sequencing Project.” *Genomics* 111 (4): 808–18. <https://doi.org/10.1016/j.ygeno.2018.05.004>.

Patro, Rob, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. 2017. “Salmon Provides Fast and Bias-Aware Quantification of Transcript Expression.” *Nature Methods* 14 (4): 417–19. <https://pubmed.ncbi.nlm.nih.gov/28263959/>.

Qiu, Xiaojie, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A. Pliner, and Cole Trapnell. 2017. “Reversed Graph Embedding Resolves Complex Single-Cell Trajectories.” *Nature Methods* 14 (10): 979–82. <https://doi.org/10.1038/nmeth.4402>.

Ramı́rez, Fidel, Devon P Ryan, Bjørn Grüning, Vivek Bhardwaj, Fabian Kilpert, Andreas S Richter, Steffen Heyne, et al. 2016. “deepTools2: A Next Generation Web Server for Deep-Sequencing Data Analysis.” *Nucleic Acids Research* 44 (W1): W160–65.

Schep, Alicia N, Bin Wu, Jason D Buenrostro, and William J Greenleaf. 2017. “chromVAR: Inferring Transcription-Factor-Associated Accessibility from Single-Cell Epigenomic Data.” *Nature Methods* 14 (10): 975–78. <https://www.nature.com/articles/nmeth.4401>.

Setty, Manu, Vaidotas Kiseliovas, Jacob Levine, Adam Gayoso, Linas Mazutis, and Dana Pe’er. 2019. “Characterization of Cell Fate Probabilities in Single-Cell Data with Palantir.” *Nature Biotechnology* 37 (4): 451–60. <https://doi.org/10.1038/s41587-019-0068-4>.

Sims, David, Ian Sudbery, Nicholas E. Ilott, Andreas Heger, and Chris P. Ponting. 2014. “Sequencing Depth and Coverage: Key Considerations in Genomic Analyses.” *Nature Reviews Genetics* 15 (2): 121–32. <https://doi.org/10.1038/nrg3642>.

Smith, Tom. 2015. “Unique Molecular Identifiers – the Problem, the Solution and the Proof.” *CGAT*. <https://cgatoxford.wordpress.com/2015/08/14/unique-molecular-identifiers-the-problem-the-solution-and-the-proof/>.

Street, Kelly, Davide Risso, Russell B. Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. 2018. “Slingshot: Cell Lineage and Pseudotime Inference for Single-Cell Transcriptomics.” *BMC Genomics* 19 (1): 477. <https://doi.org/10.1186/s12864-018-4772-0>.

Tarca, A. L., R. Romero, and S. Draghici. 2006. “Analysis of Microarray Experiments of Gene Expression Profiling.” *American Journal of Obstetrics and Gynecology* 195 (2): 373–88. <https://doi.org/10.1016/j.ajog.2006.07.001>.

Wolf, F. Alexander, Fiona K. Hamey, Mireya Plass, Jordi Solana, Joakim S. Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J. Theis. 2019. “PAGA: Graph Abstraction Reconciles Clustering with Trajectory Inference Through a Topology Preserving Map of Single Cells.” *Genome Biology* 20 (1): 59. <https://doi.org/10.1186/s13059-019-1663-x>.

Wolock, Samuel L, Smita Krishnaswamy, and B Jesse Huang. 2019. “Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data.” *Cell Systems* 8 (4): 281–291. e9. <https://pubmed.ncbi.nlm.nih.gov/30954476/>.

Yu, Miao, Gary C Hon, Keith E Szulwach, Chun-Xiao Song, Peng Jin, Bing Ren, and Chuan He. 2012. “Tet-Assisted Bisulfite Sequencing of 5-Hydroxymethylcytosine.” *Nature Protocols* 7 (12): 2159–70. <https://doi.org/10.1038/nprot.2012.137>.

Zhang, Wenqian, Ying Yu, Falk Hertwig, Jean Thierry-Mieg, Wenwei Zhang, Danielle Thierry-Mieg, Jian Wang, et al. 2015. “Comparison of RNA-Seq and Microarray-Based Models for Clinical Endpoint Prediction.” *Genome Biology* 16 (1). <https://doi.org/10.1186/s13059-015-0694-1>.

Zhang, Yong, Tao Liu, Clifford A Meyer, Jie Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, et al. 2008. “Model-Based Analysis of ChIP-Seq (MACS).” *Genome Biology* 9 (9): R137. <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2008-9-9-r137>.

Zheng, Grace XY, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, et al. 2017. “Massively Parallel Digital Transcriptional Profiling of Single Cells.” *Nature Communications* 8 (1): 1–12. <https://www.nature.com/articles/ncomms14049>.

Ziemann, Mark, Yotam Eren, and Assam El-Osta. 2016. “Gene Name Errors Are Widespread in the Scientific Literature.” *Genome Biology* 17 (1). <https://doi.org/10.1186/s13059-016-1044-7>.