Choosing Genomics Tools

October, 2022

Table of Contents

# About this Course

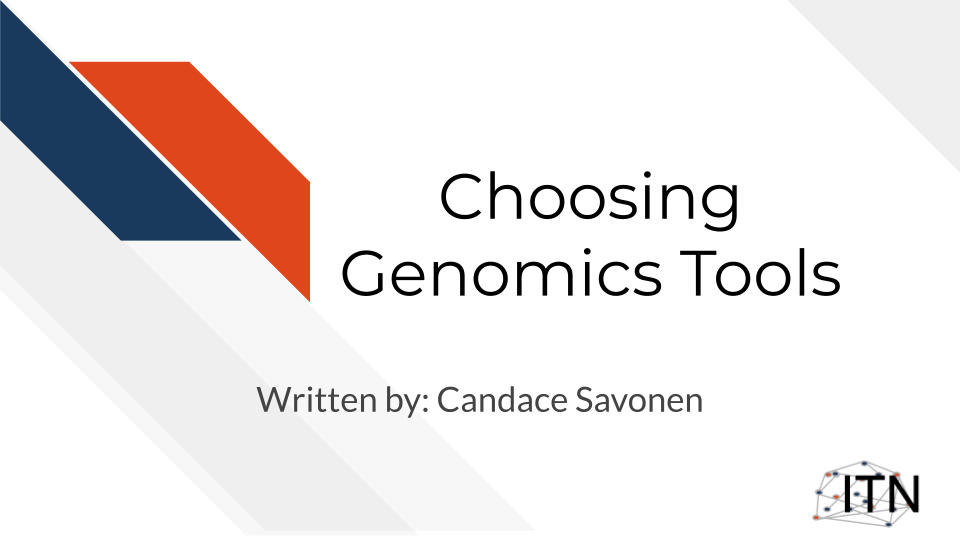
This course is part of a series of courses for the [Informatics Technology for Cancer Research (ITCR)](https://itcr.cancer.gov/) called the Informatics Technology for Cancer Research Education Resource. This material was created by the ITCR Training Network (ITN) which is a collaborative effort of researchers 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

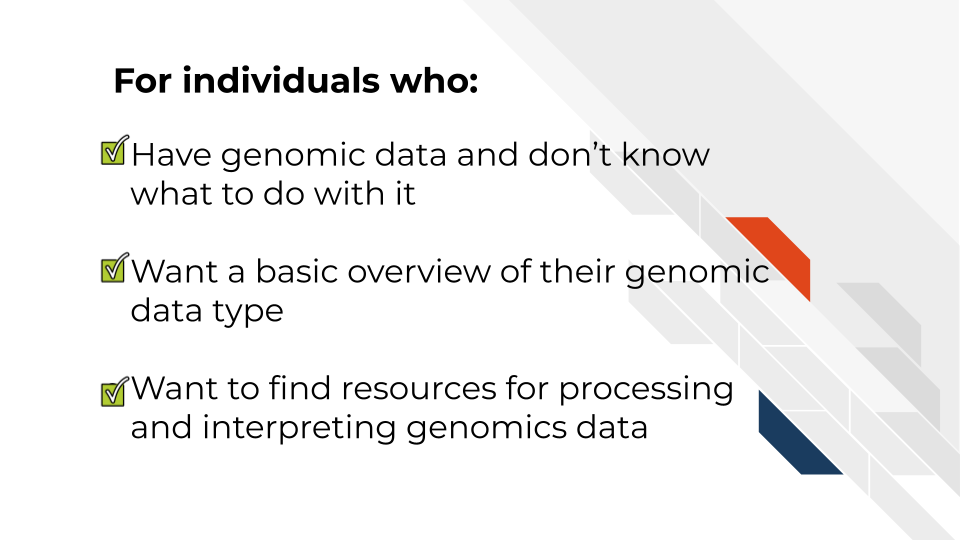


## 1.1 Target Audience

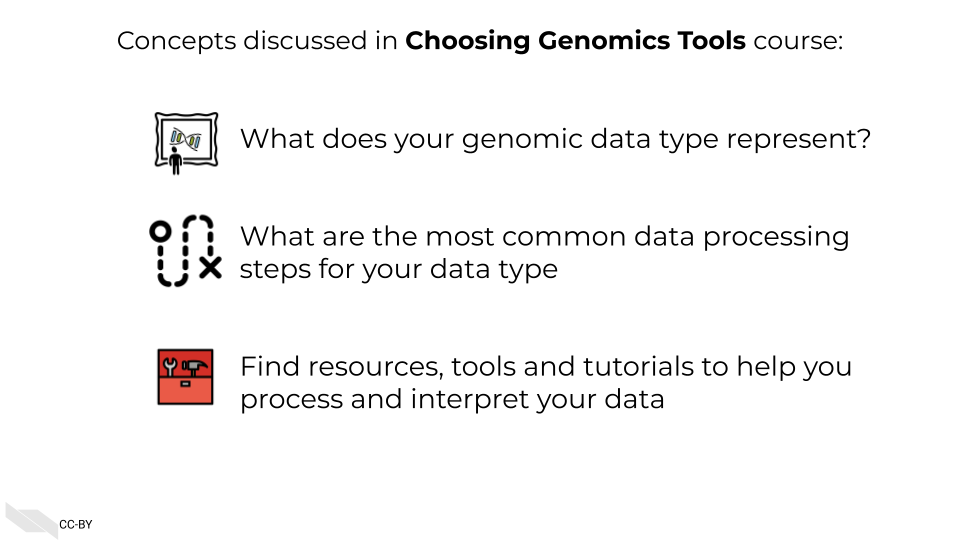
The course is intended for students in the biomedical sciences and researchers who have been given data and don’t know what to do with it or would like an overview of the different genomic data types that are out there.

*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:

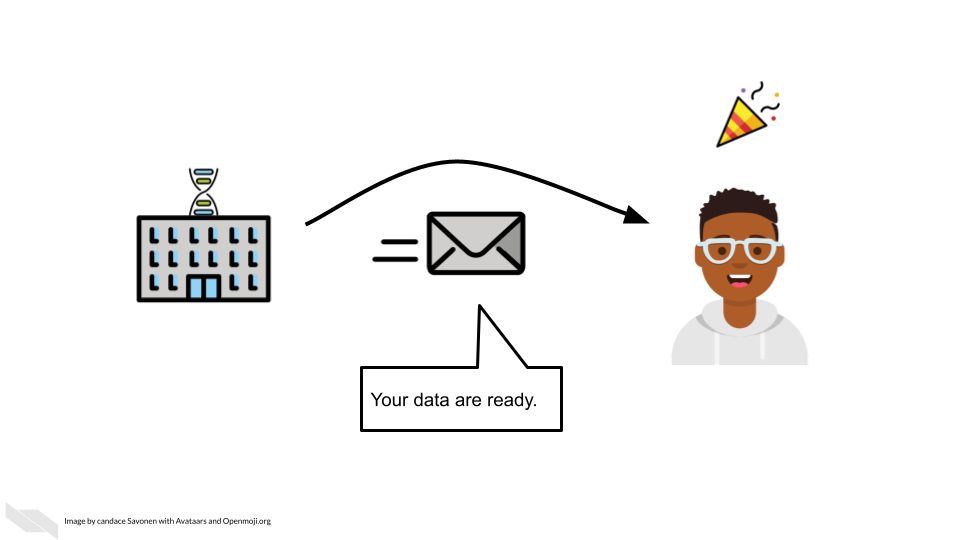


## 1.3 Motivation

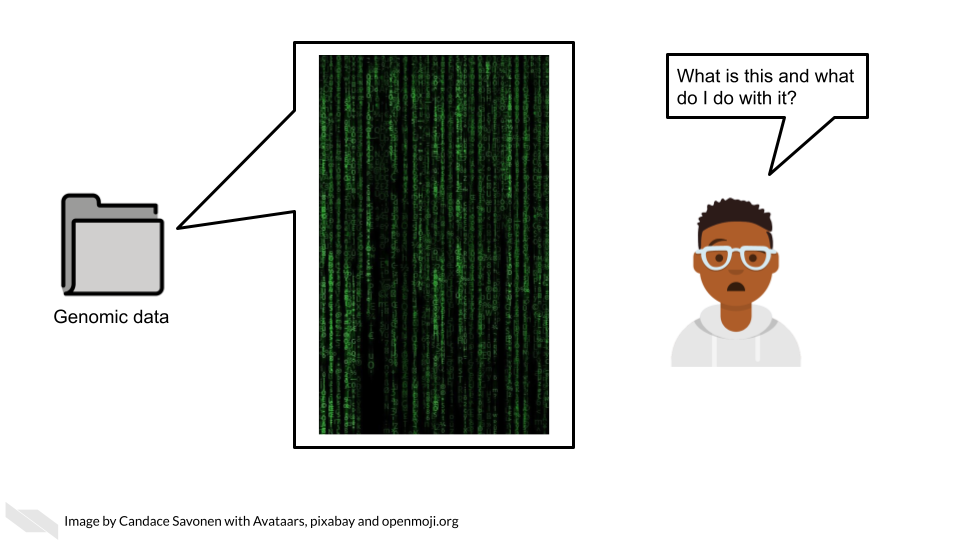
Cancer datasets are plentiful, complicated, and hold untold amounts of information regarding cancer biology. Cancer researchers are working to apply their expertise to the analysis of these vast amounts of data but training opportunities to properly equip them in these efforts can be sparse. This includes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

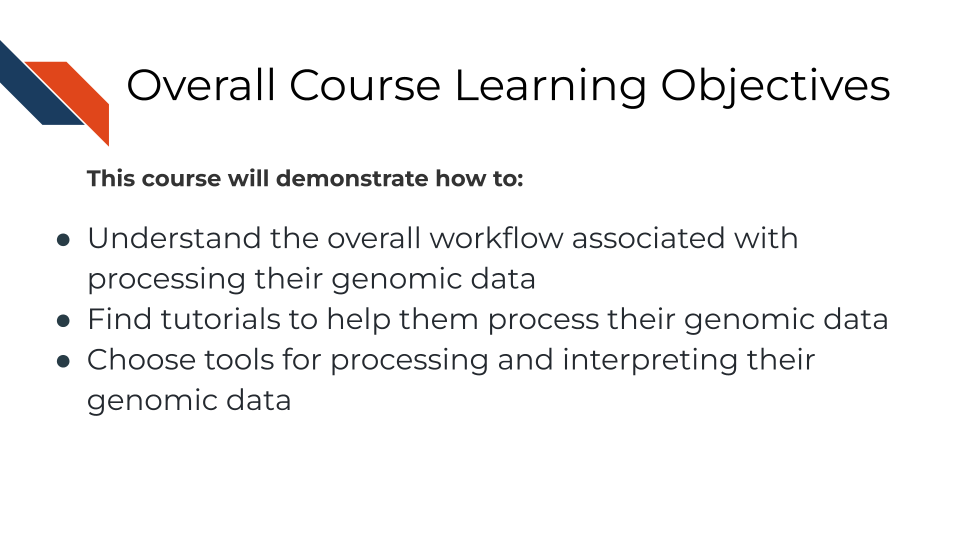


As an example, data file formats may not have been covered in their training, and the data they received seems unintelligible and not as straightforward as they hoped.



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

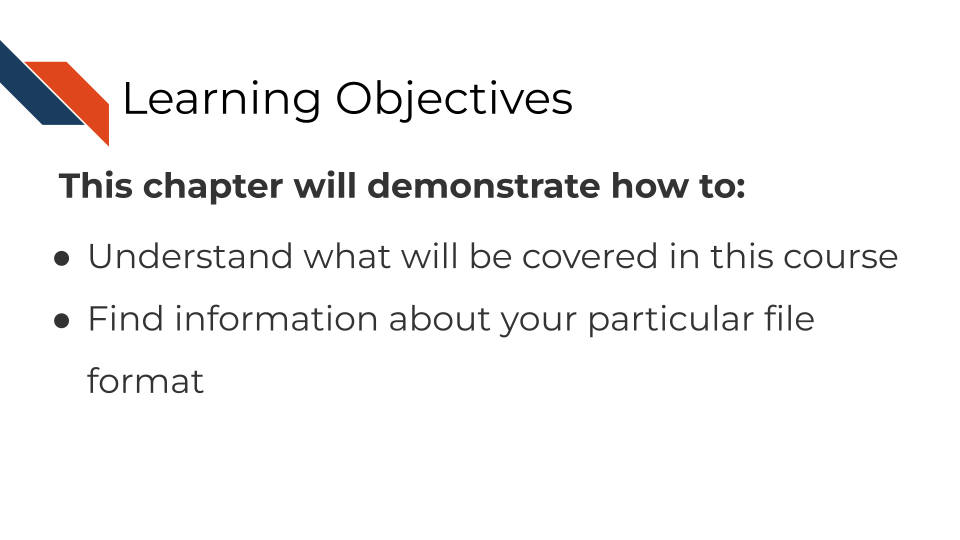
**What is not the goal**  
Teach learners about choosing parameters or about the ins and outs of every genomic tool they might be interested in. This course is meant to connect people to other resources that will help them with the specifics of their genomic data and help learners have more efficient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

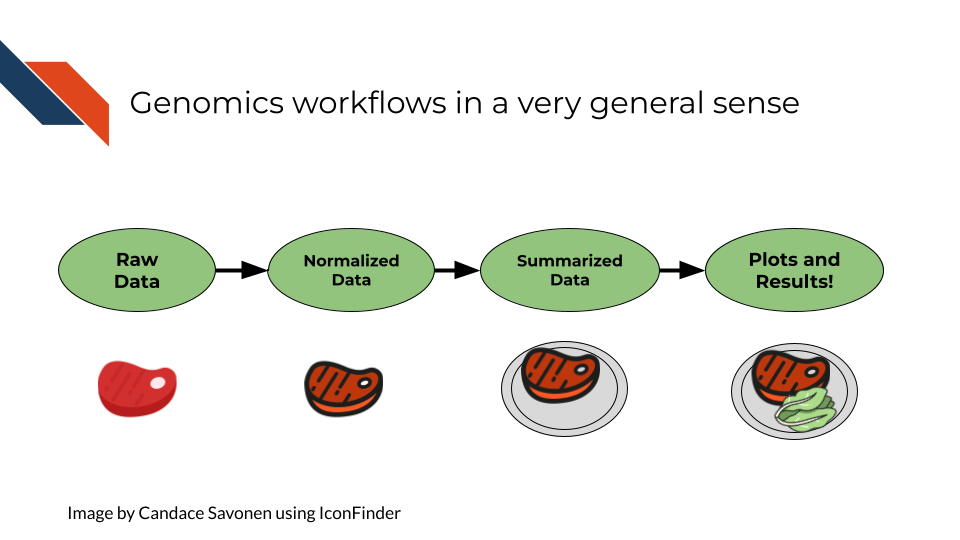
## 2.1 Learning Objectives



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.

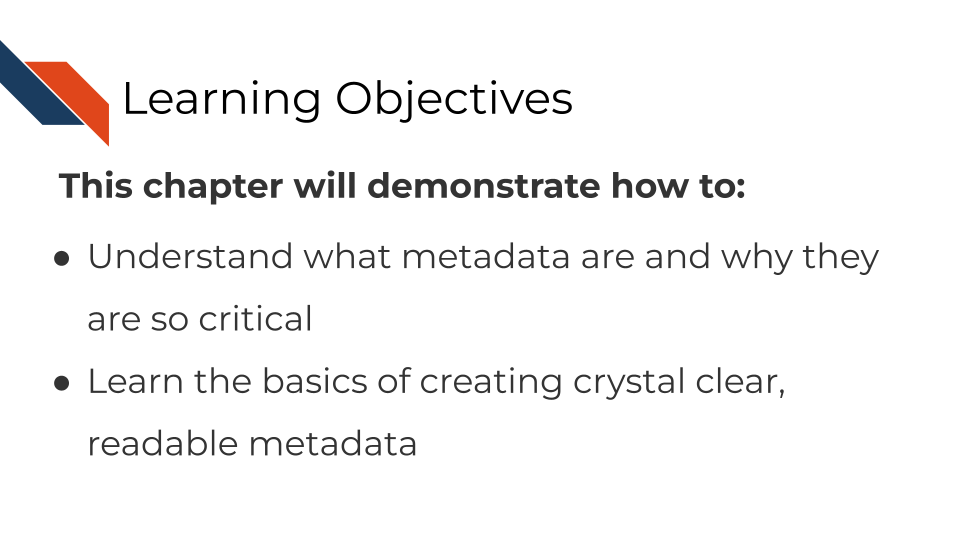
For [more about GTF and GFF files](https://uswest.ensembl.org/info/website/upload/gff.html).

### 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

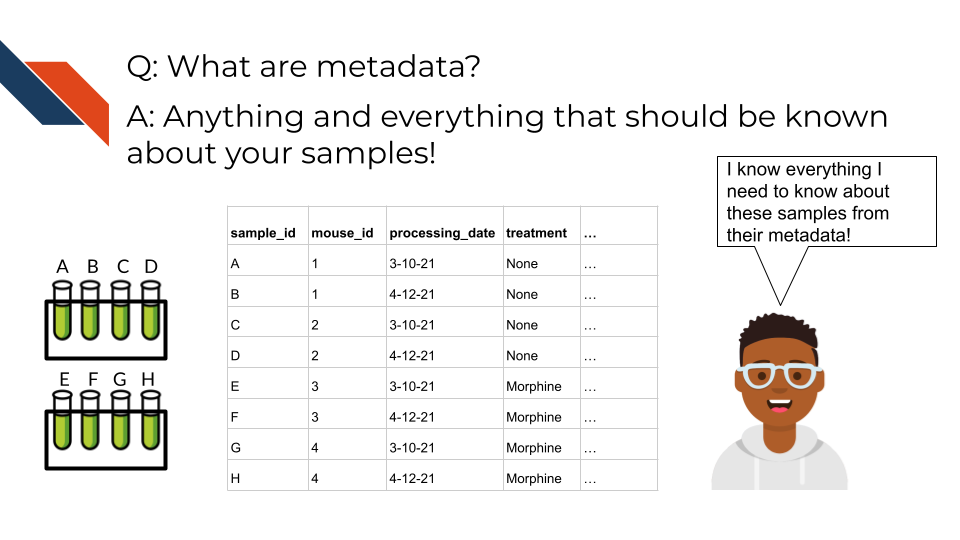


## 3.2 What are metadata?

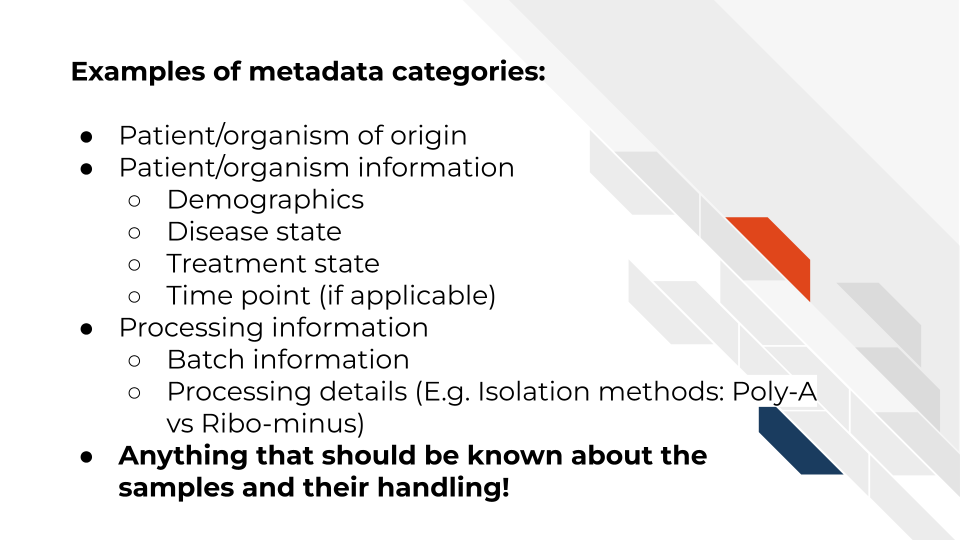
Metadata are critically important descriptive information about your data.

**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!

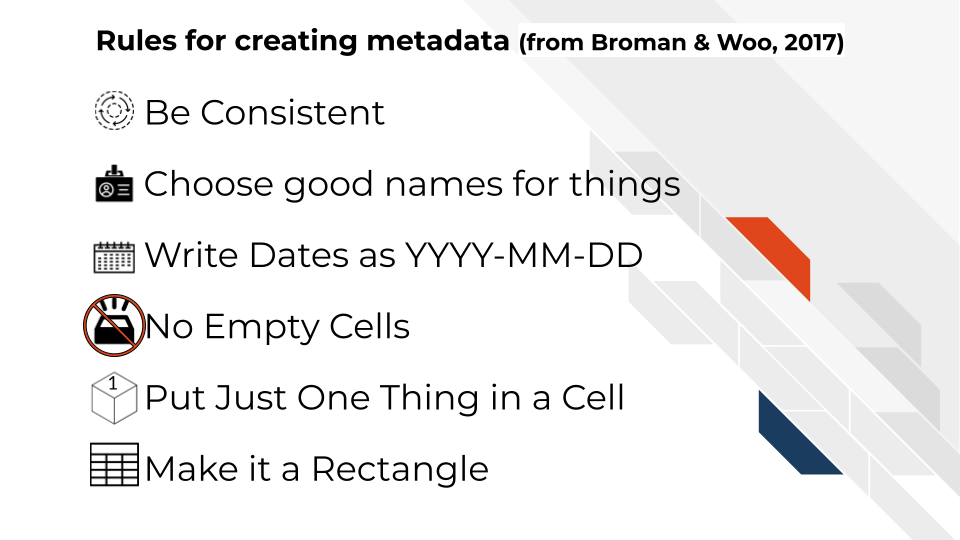
#### 3.3.1.2 Goal B: Avoid introducing errors into your metadata in the future!

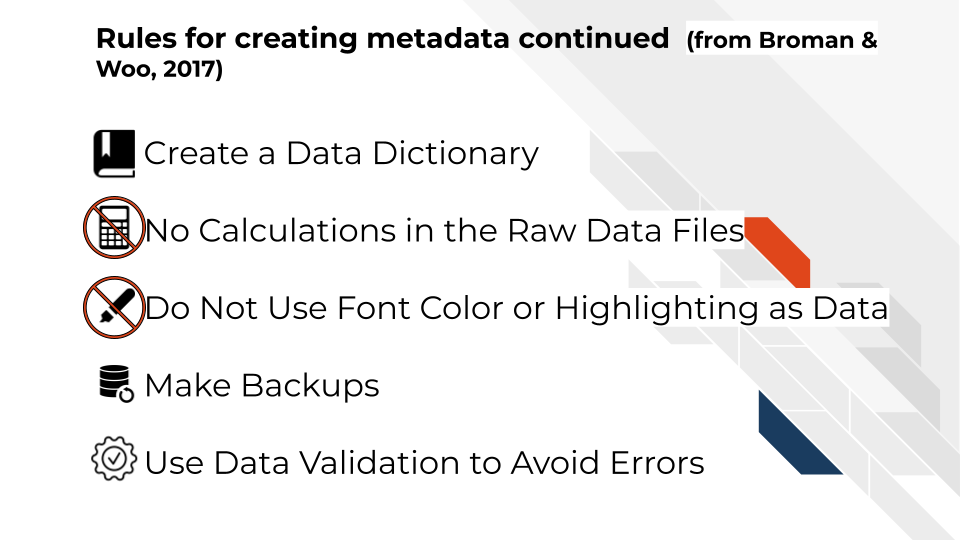
Toward these two goals, [this excellent article](https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1375989) by Broman & Woo discusses metadata design rules. We will very briefly cover the major points here but highly suggest you read the original article.

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:

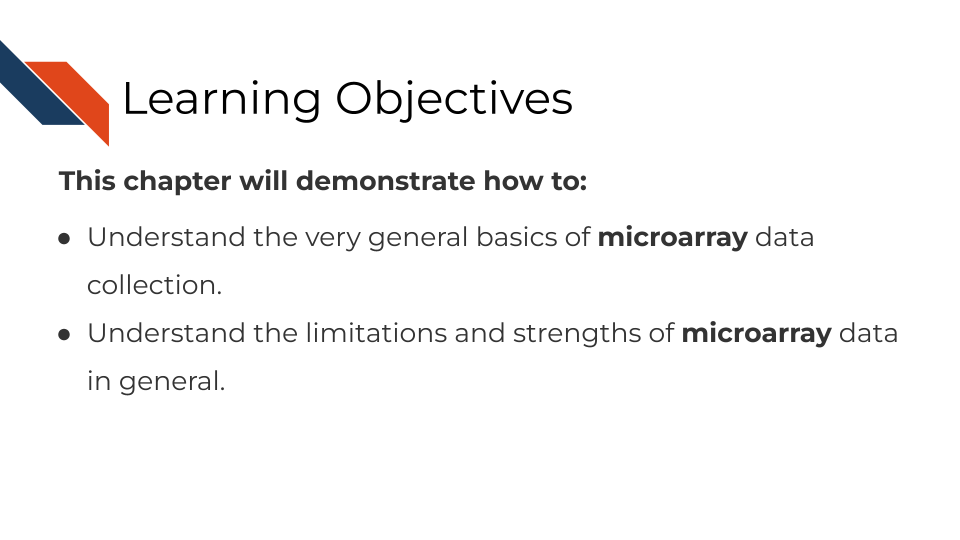




If you are not the person who has the information needed to create metadata, or you believe that another individual already has this information, make sure you get ahold of the metadata that correspond to your data. It will be critical for you to have to do any sort of meaningful analysis!

# 4 Microarray Data

## 4.1 Learning Objectives

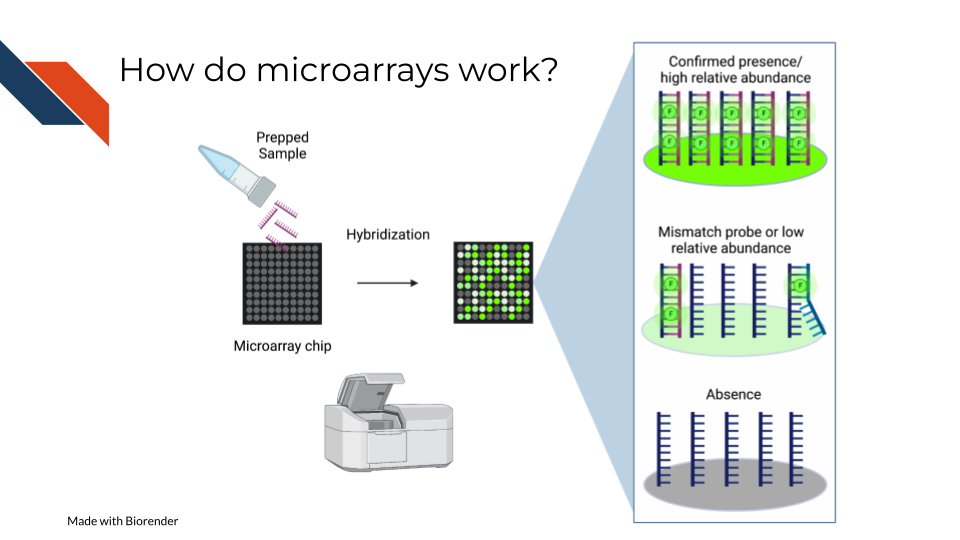


## 4.2 Summary of microarrays

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.

## 4.3 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.

### 4.3.1 Pros:

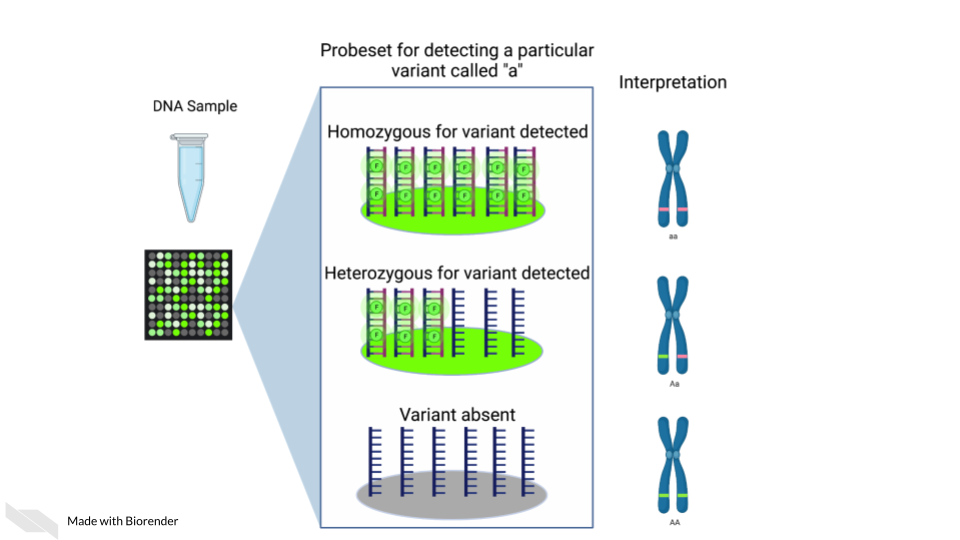
* Microarrays are much more affordable than high throughput sequencing which can allow you to run more samples and have more statistical power ([Tarca, Romero, and Draghici 2006](#ref-Tarca2006); [ALSF 2019](#ref-refinebioexamples2019)).
* Microarrays take less time to process than most high throughput 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 ([Zhang et al. 2015](#ref-Zhang2015)).

### 4.3.2 Cons:

* Microarray chips can only measure the targets they are designed for, and cannot be used for exploratory purposes ([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)).

## 4.4 What types of arrays are there?

### 4.4.1 SNP arrays

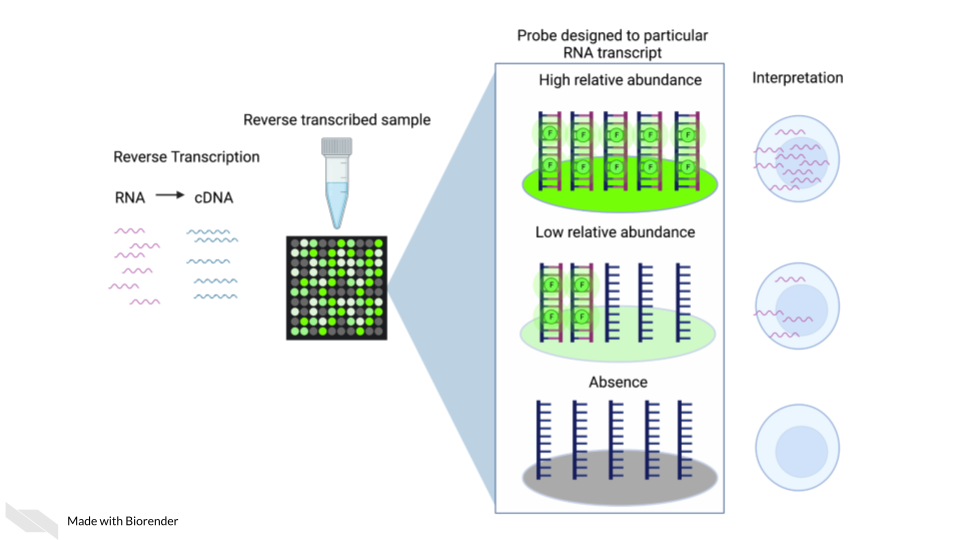


Single nucleotide polymorphism arrays are designed to measure DNA variants. They are designed to target DNA variants. When the sample is hybridized, the amount of fluorescence detected can be interpreted to indicate the presence of the variant and whether the variant is homogeneous or heterogenous. The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

#### 4.4.1.1 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.

### 4.4.2 Gene expression arrays

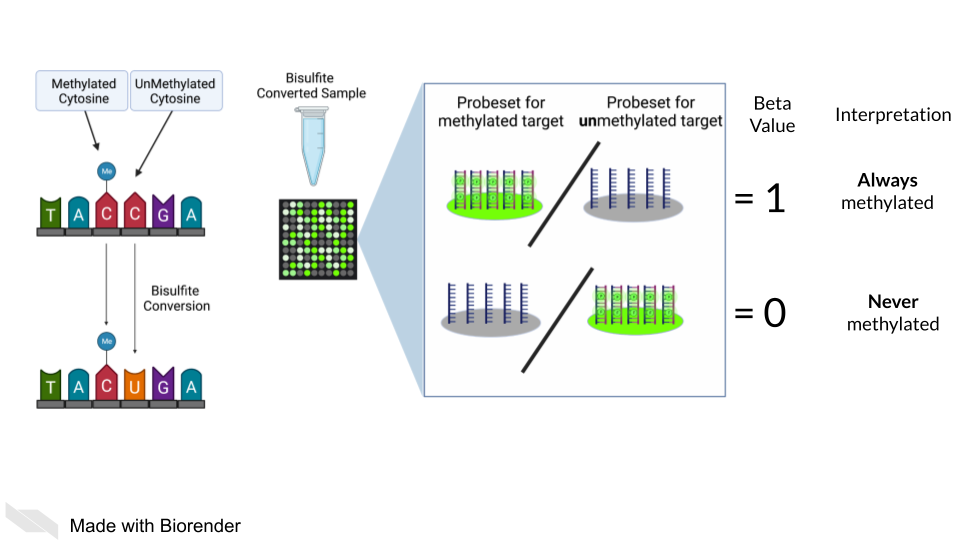


Gene expression arrays are designed to measure gene expression. They are designed to target and measure relative transcript abundance level.

#### 4.4.2.1 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)).

### 4.4.3 DNA methylation arrays

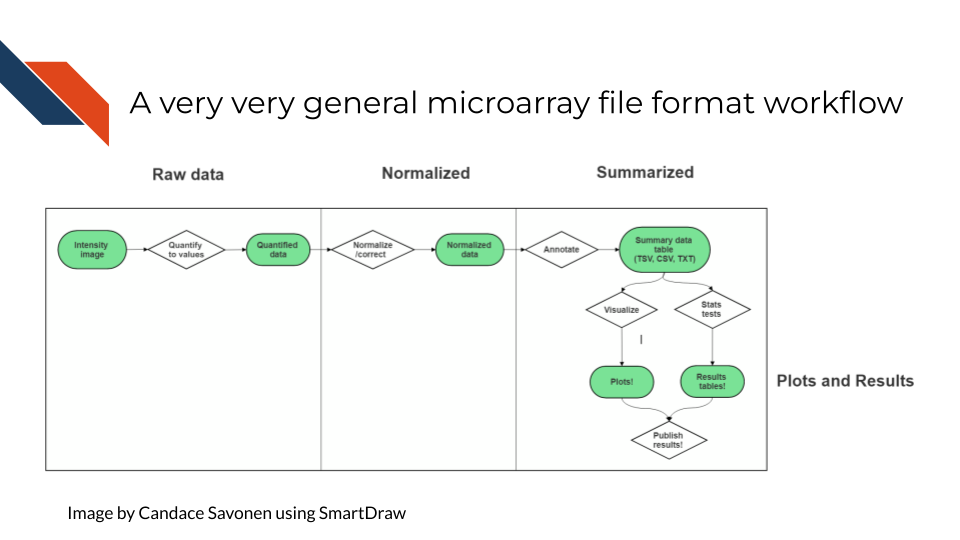


DNA methylation can also be measured by microarray. To detect methylated cytosines (5mC), DNA samples are prepped using bisulfite conversion. This converts unmethylated cytosines into uracils and leaves methylated cytosines untouched. Probes are then designed to bind to either the uracil or the cytosine,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

## 4.5 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.

### 4.5.1 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)

### 4.5.2 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).

## 4.6 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.):

### 4.6.1 Microarray file formats

#### 4.6.1.1 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).

#### 4.6.1.2 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).

#### 4.6.1.3 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)

#### 4.6.1.4 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).

## 4.7 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:

#### 4.7.0.1 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)).

#### 4.7.0.2 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://uswest.ensembl.org/info/website/upload/gff.html).

### 4.7.1 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.

### 4.7.2 Microarray processing tutorials:

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

#### 4.7.2.1 General arrays

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

#### 4.7.2.2 Gene Expression Arrays

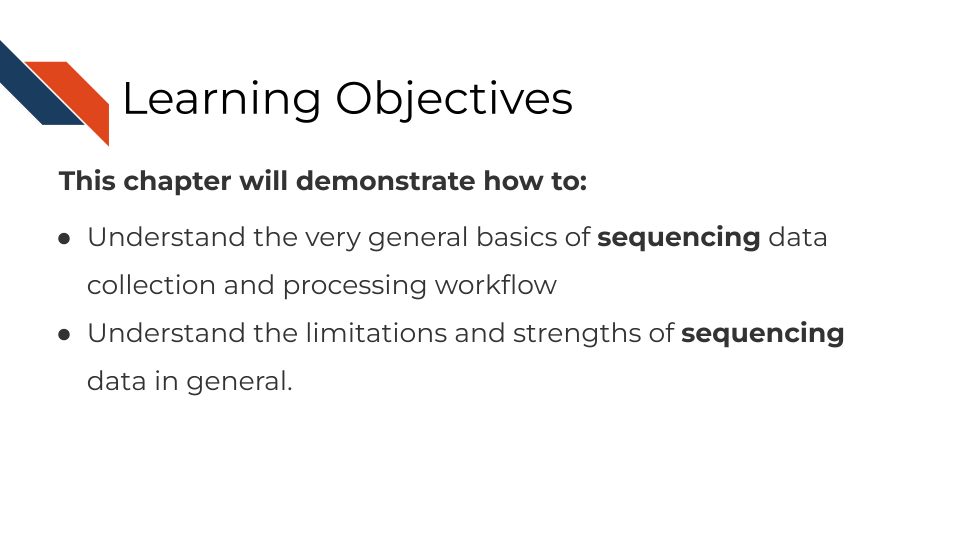
* [An end to end workflow for differential gene expression using Affymetrix microarrays](https://bioconductor.org/packages/devel/workflows/vignettes/maEndToEnd/inst/doc/MA-Workflow.html).

#### 4.7.2.3 DNA Methylation Arrays

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

# 5 Sequencing Data

## 5.1 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.

## 5.2 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.

## 5.3 Sequencing concepts

### 5.3.1 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.

#### 5.3.1.1 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.

#### 5.3.1.2 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.

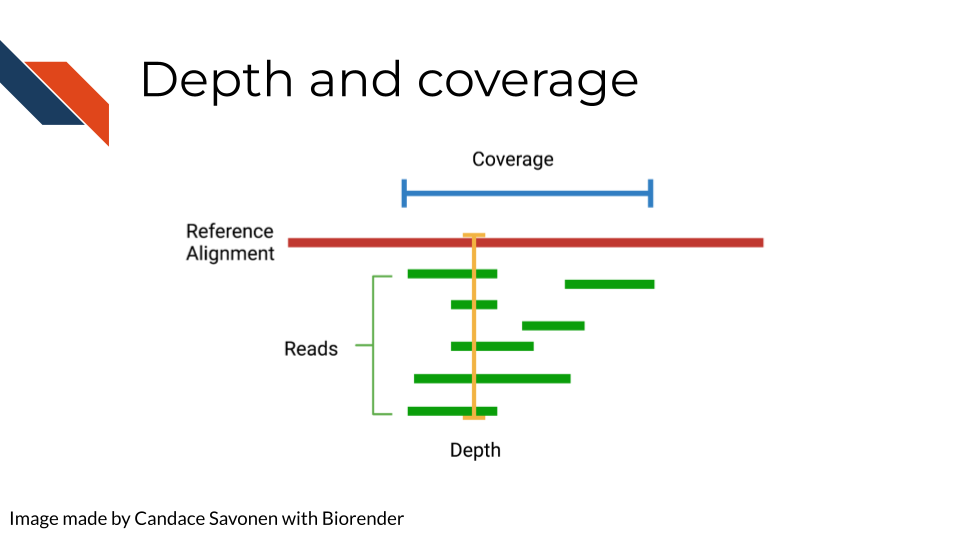
#### 5.3.1.3 Length bias

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.

### 5.3.2 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.

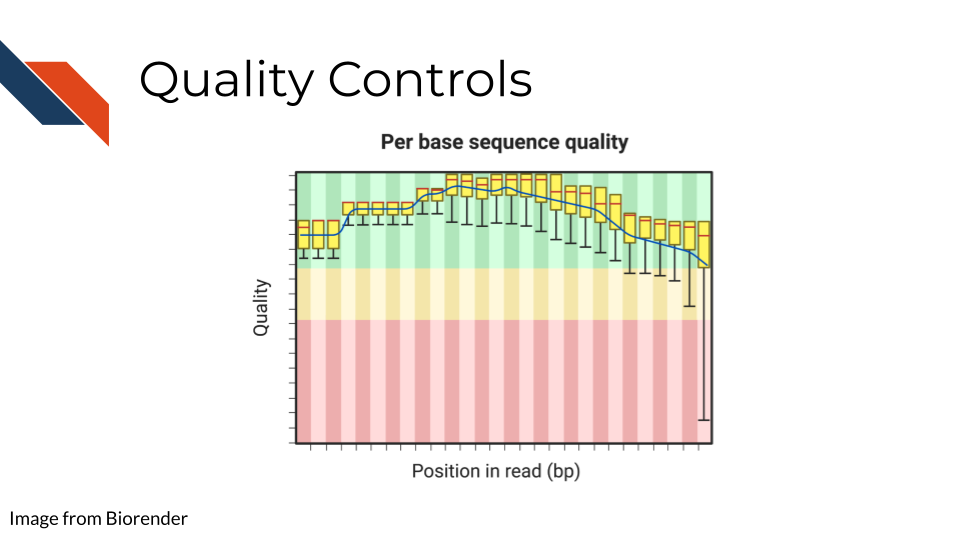
### 5.3.3 Depth of coverage



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.

### 5.3.4 Quality controls

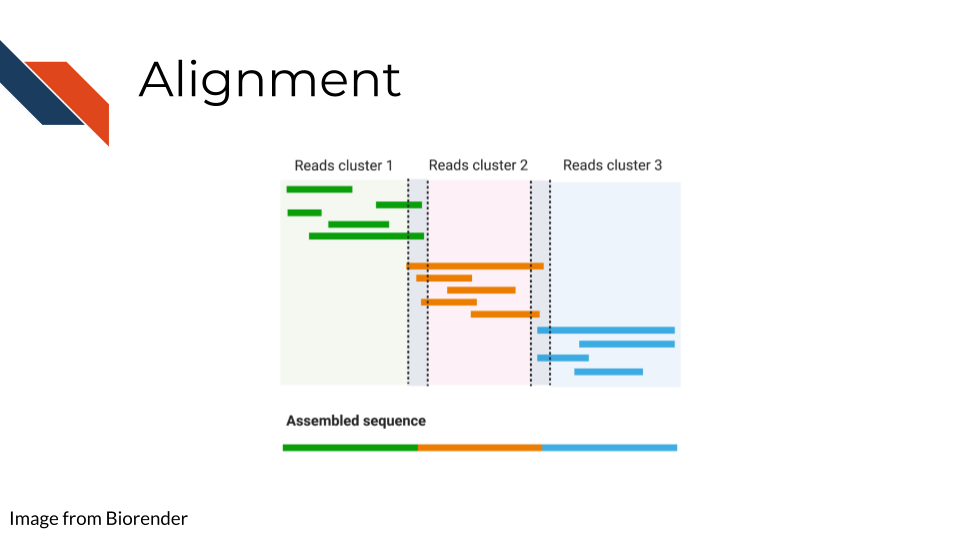


Sequencing bases involves some error/confidence rate. As mentioned, some parts of the genome are harder to read than others. Or, sometimes your sequencing can be influenced by poor quality sample that has degraded. Before you jump in to further analyzing your data, you will want to investigate the quality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

### 5.3.5 Alignment

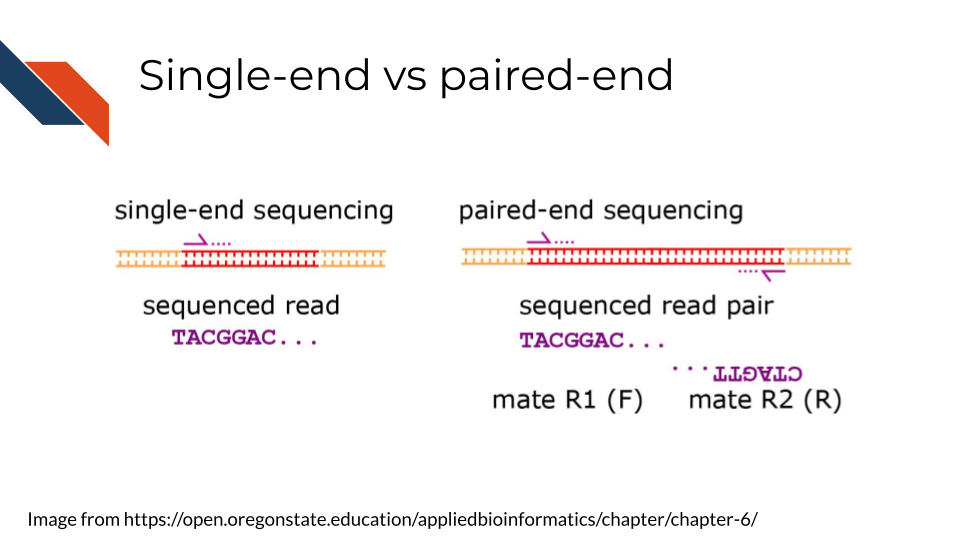


Once you have your reads and you find them reasonably trustworthy through quality control checks, you will want to align them to your reference. The reference you align your sequences to will depend on the data type you have: a reference genome, a reference transcriptome, something else?

* 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.

### 5.3.6 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.

## 5.4 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:

### 5.4.1 Sequencing file formats

#### 5.4.1.1 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).

#### 5.4.1.2 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).

#### 5.4.1.3 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).

#### 5.4.1.4 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.

#### 5.4.1.5 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).

#### 5.4.1.6 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).

#### 5.4.1.7 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).).

### 5.4.2 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.

# 6 DNA Methods

## 6.1 Learning Objectives

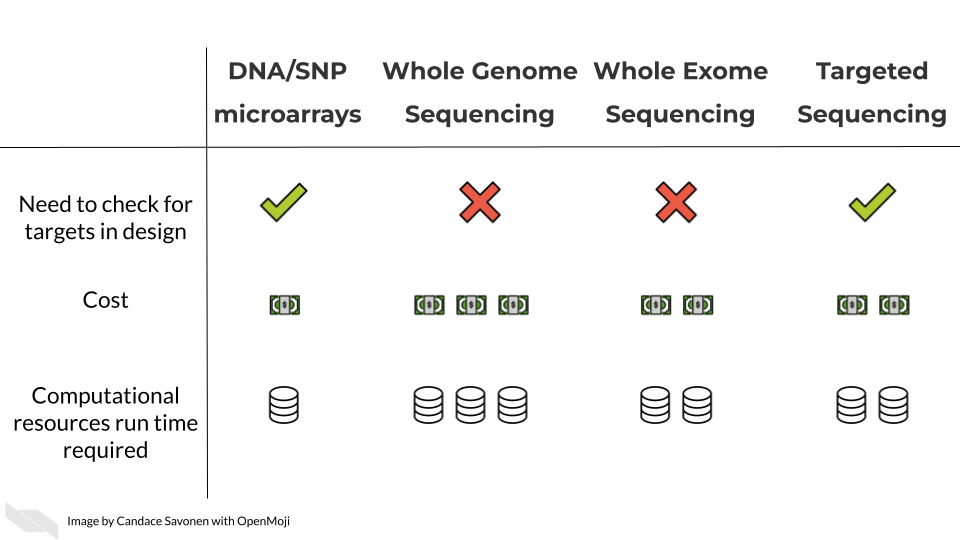


## 6.2 What are the goals of analyzing DNA sequences?



## 6.3 Comparison of DNA methods

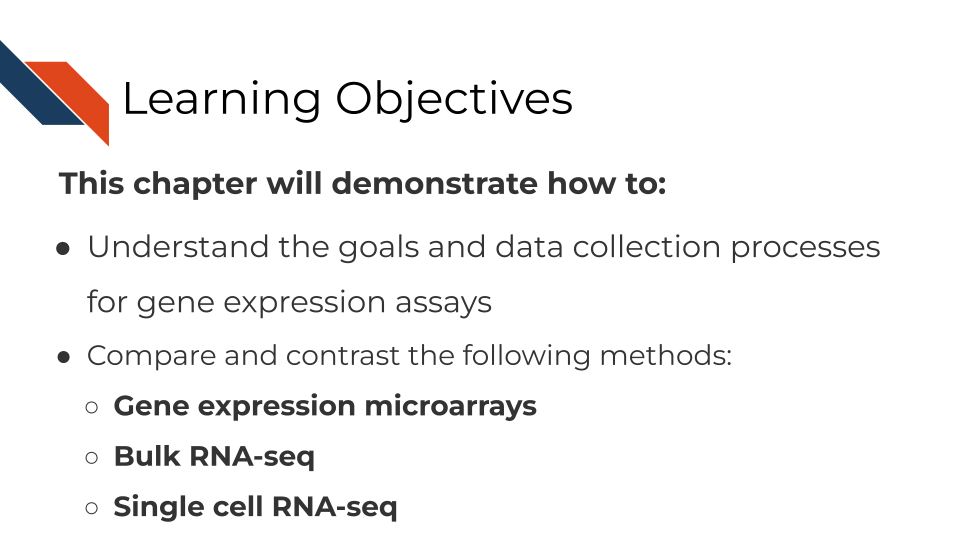
There are three general methods we will discuss for evaluating DNA sequences. Whole Genome Sequencing (WGS) assays more of the genome than other methods but is much more costly and computationally intensive. Depending on your goals WGS may be overkill. SNP microarrays on the other hand, are much more cost effective but are not able to be used for exploratory purposes. Whole Exome Sequencing (WXS or WES) and other targeted sequencing methods allow you to survey regions of the genome in way that is more cost effective and potentially at higher depths.



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.

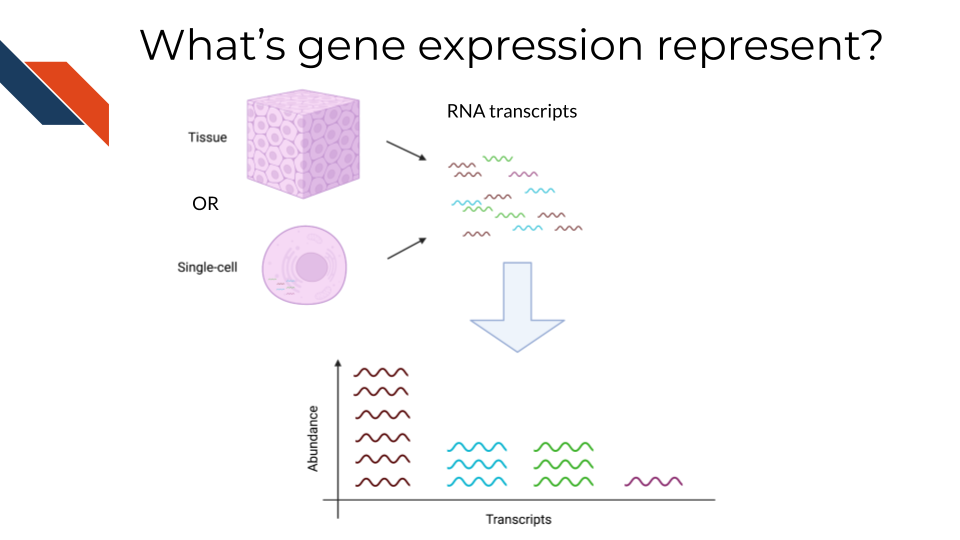
# 7 RNA Methods

## 7.1 Learning Objectives



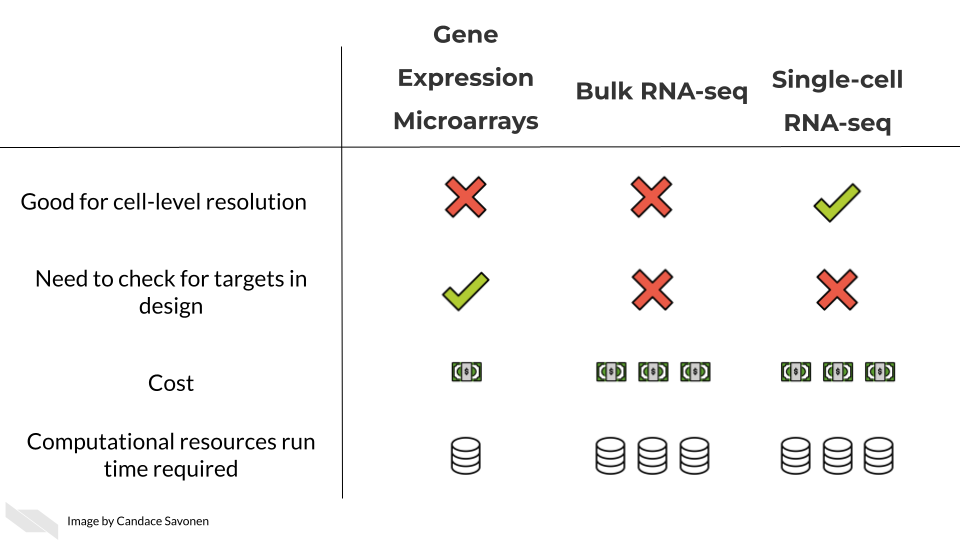
## 7.2 What are the goals of gene expression analysis?

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.



## 7.3 Comparison of RNA methods

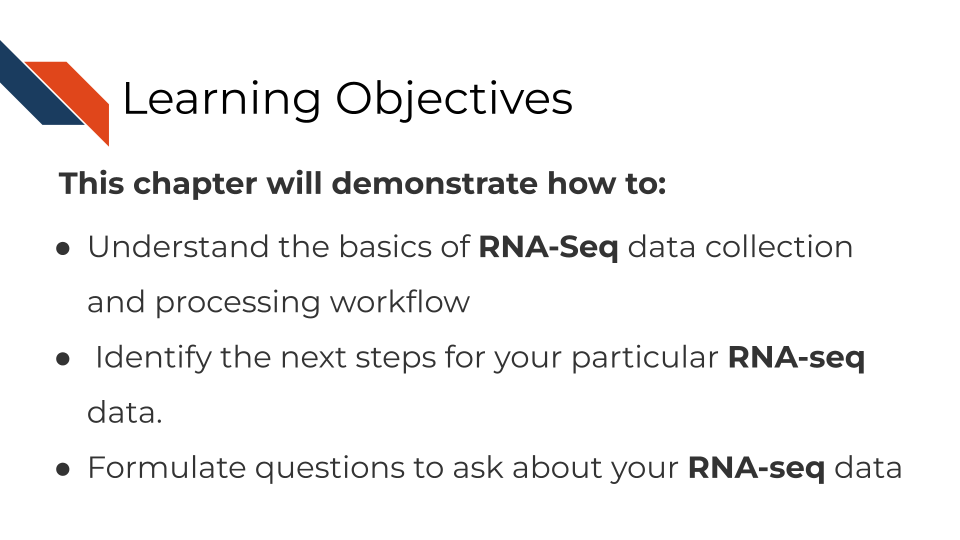
There are three general methods we will discuss for evaluating gene expression. RNA sequencing (whether bulk or single-cell) allows you to catch more targets than gene expression microarrays but is much more costly and computationally intensive. Gene expression microarrays have a lower dynamic range than RNA-seq generally but are much more cost effective.



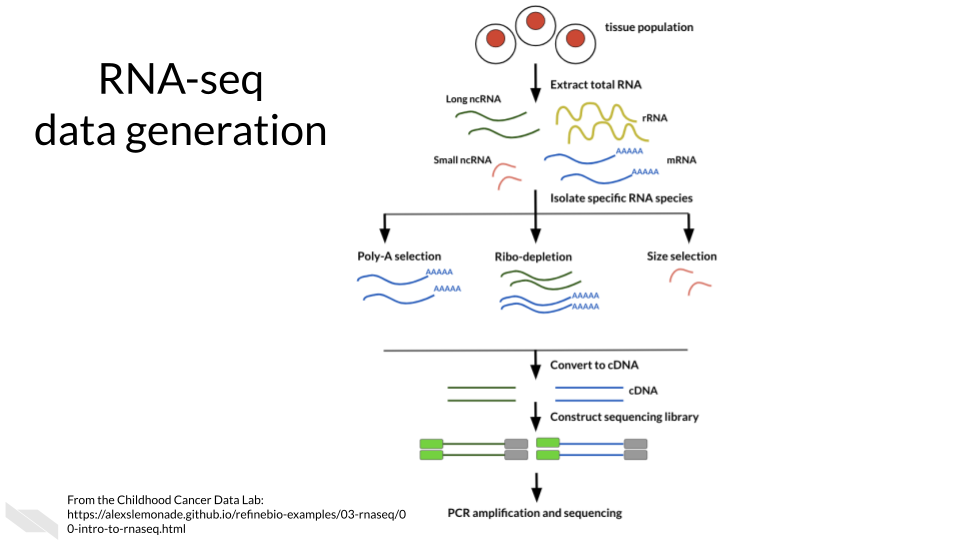
In these upcoming chapters we will discuss in more detail each of these methods, what the data represent, what you need to consider, and what resources you can consult for analyzing your data.

# 8 Bulk RNA-seq

## 8.1 Learning Objectives

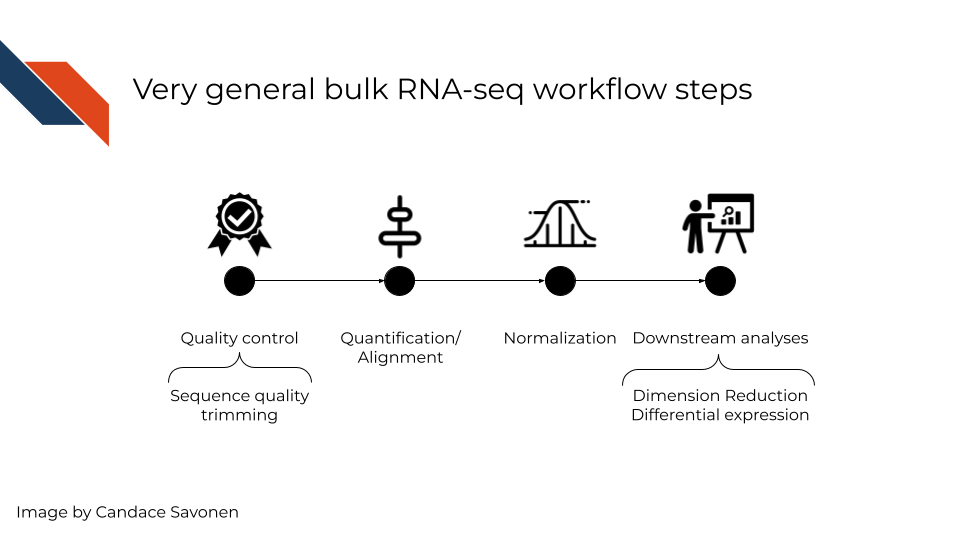


## 8.2 Where RNA-seq data comes from



## 8.3 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.



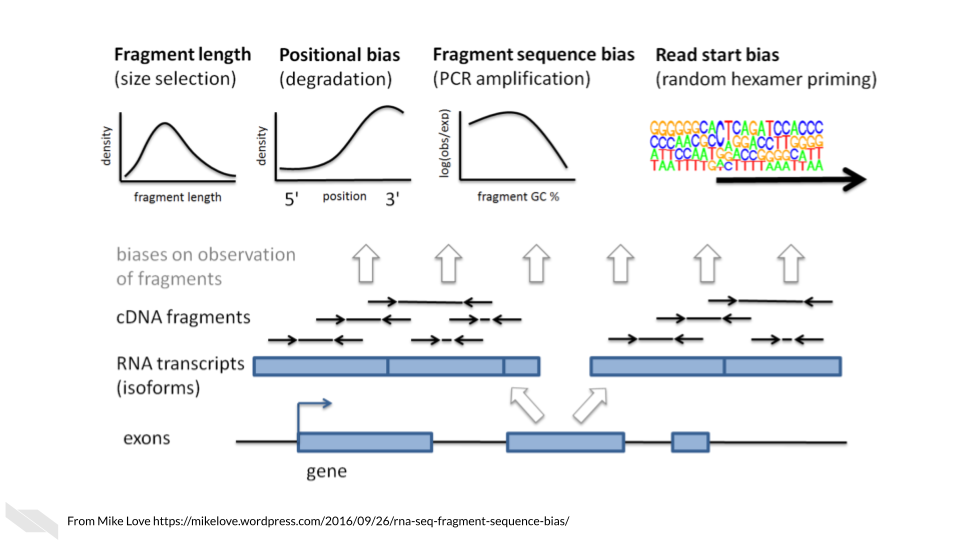
In this chapter we will highlight some of the more popular RNA-seq tools, that are generally suitable for most experiment data but there is no “one size fits all” for computational analysis of RNA-seq data ([Conesa et al. 2016](#ref-Conesa2016)). You may find tools out there that better suit your needs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

## 8.4 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.

## 8.5 RNA-seq data **limitations**

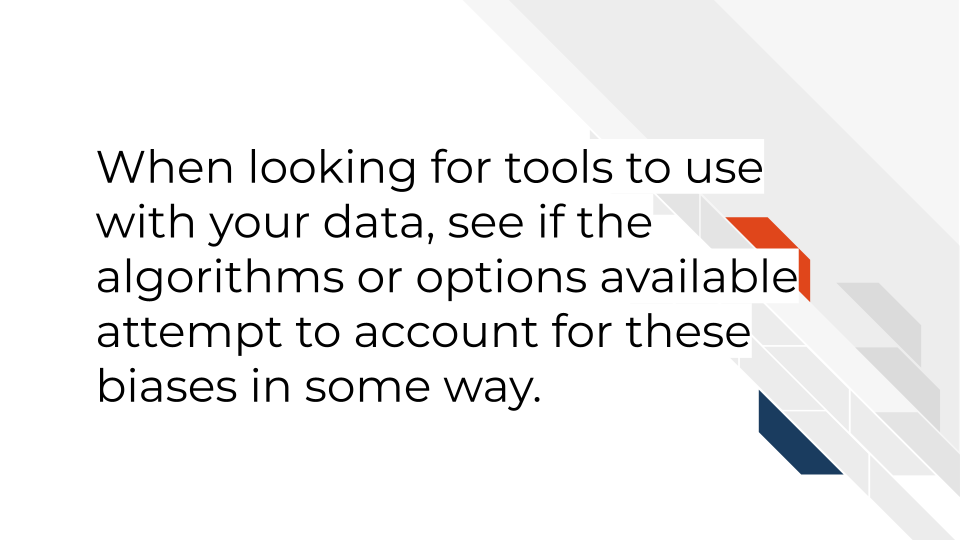
RNA-seq suffers from a lot of the common sequence biases which are further worsened by PCR amplification steps. We discussed some of the sequence biases in the previous sequencing chapter.



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.



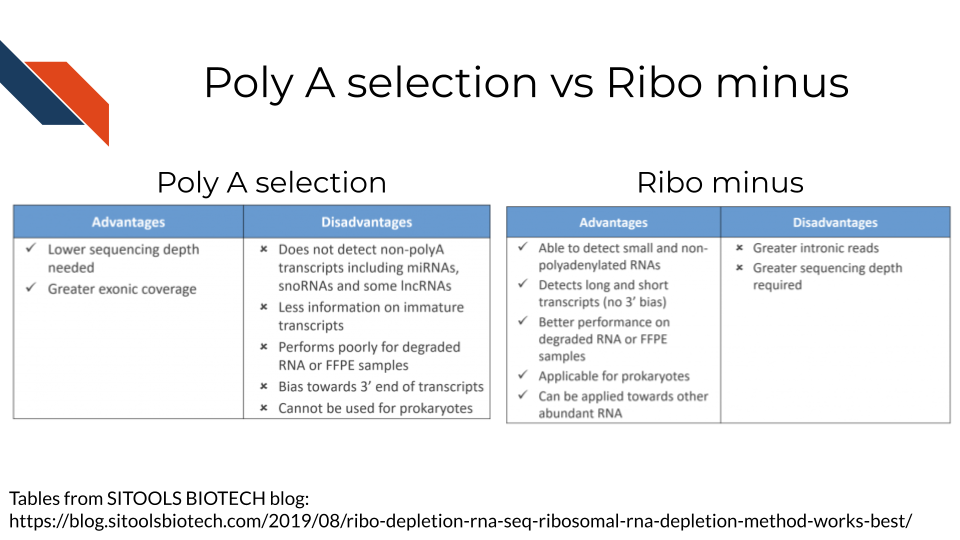
## 8.6 RNA-seq data considerations

### 8.6.1 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.

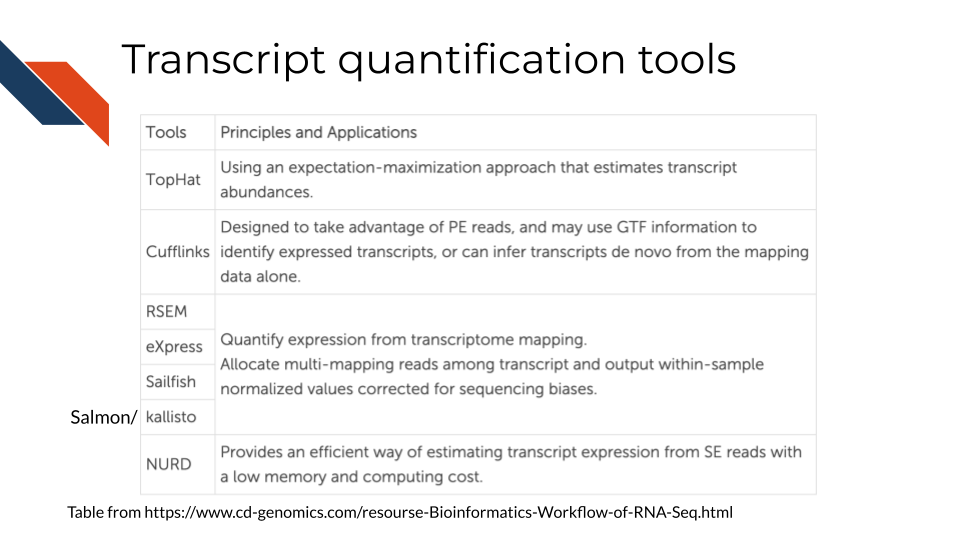


### 8.6.2 Transcriptome mapping

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).



### 8.6.3 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)**

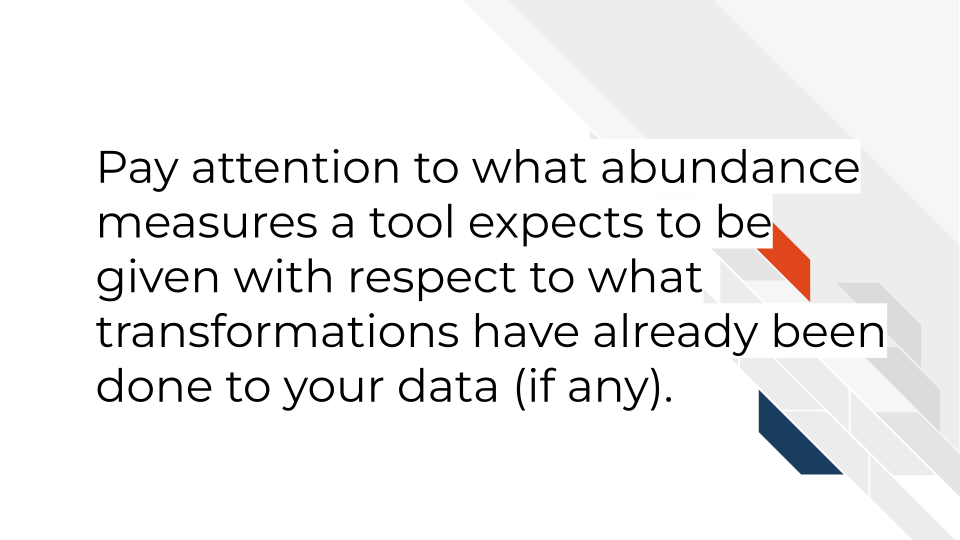
FPKM is very similar to RPKM. RPKM was made for single-end RNA-seq, where every read corresponded to a single fragment that was sequenced. FPKM was made for paired-end RNA-seq. With paired-end RNA-seq, two reads can correspond to a single fragment, or, if one read in the pair did not map, one read can 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)**

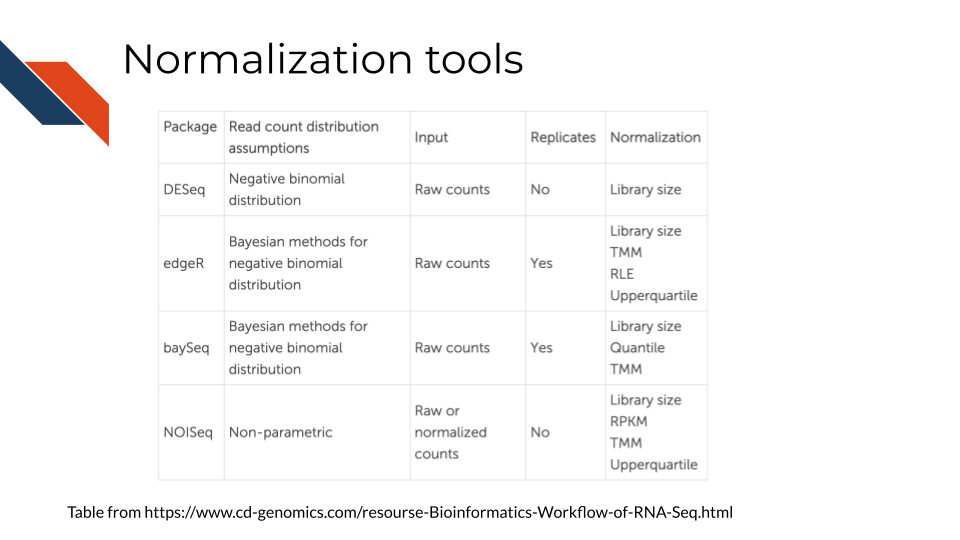
1. Divide the read counts by the length of each gene in kilobases. This gives you reads per kilobase (RPK).
2. Count up all the RPK values in a sample and divide this number by 1,000,000. This is your “per million” scaling factor.
3. Divide the RPK values by the “per million” scaling factor. This gives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



### 8.6.4 RNA-seq downstream analysis tools

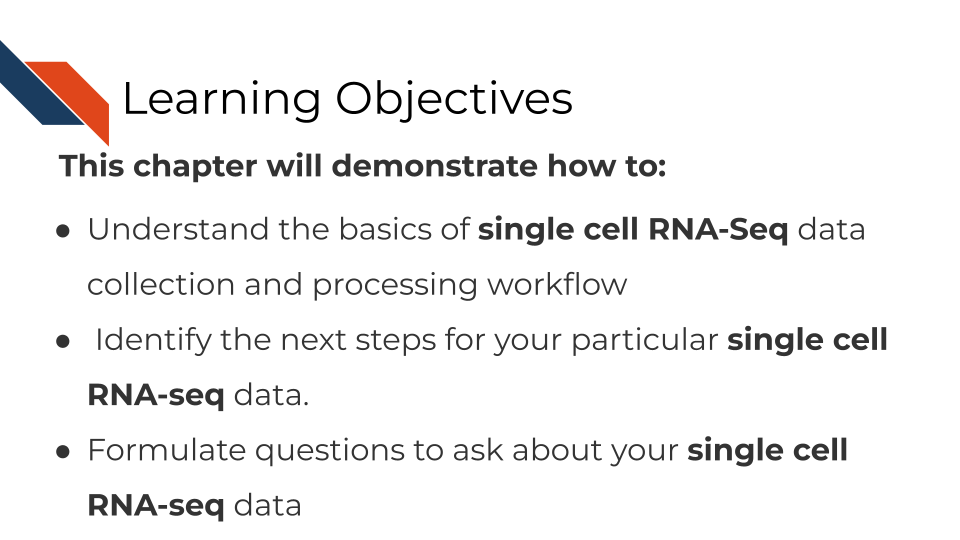


## 8.7 More reading about RNA-seq data

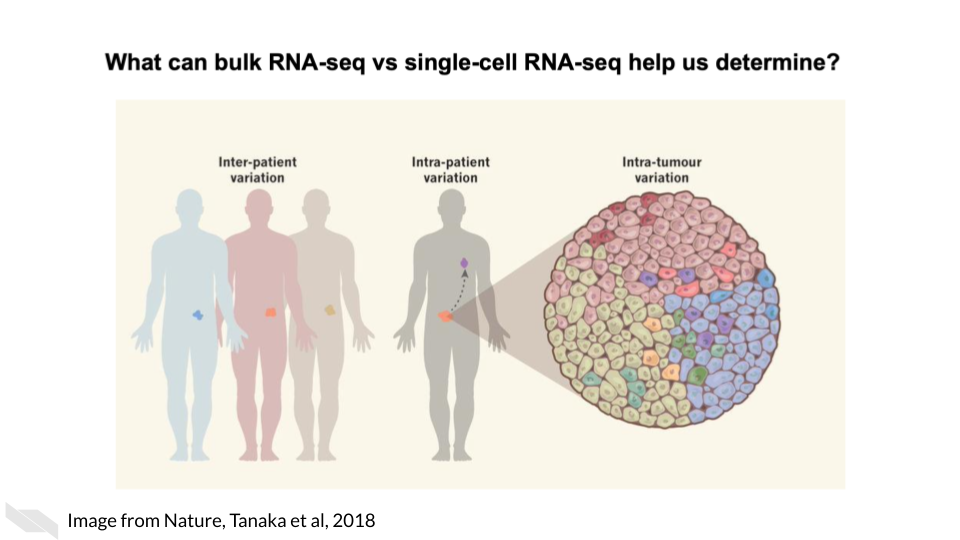
* [Refine.bio’s introduction to RNA-seq](https://alexslemonade.github.io/refinebio-examples/03-rnaseq/00-intro-to-rnaseq.html)
* [StatQuest: A gentle introduction to RNA-seq](https://www.youtube.com/watch?v=tlf6wYJrwKY) ([**Starmer2017-rnaseq?**](#ref-Starmer2017-rnaseq)).
* [A general background on 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)).

# 9 Single-cell RNA-seq

## 9.1 Learning Objectives

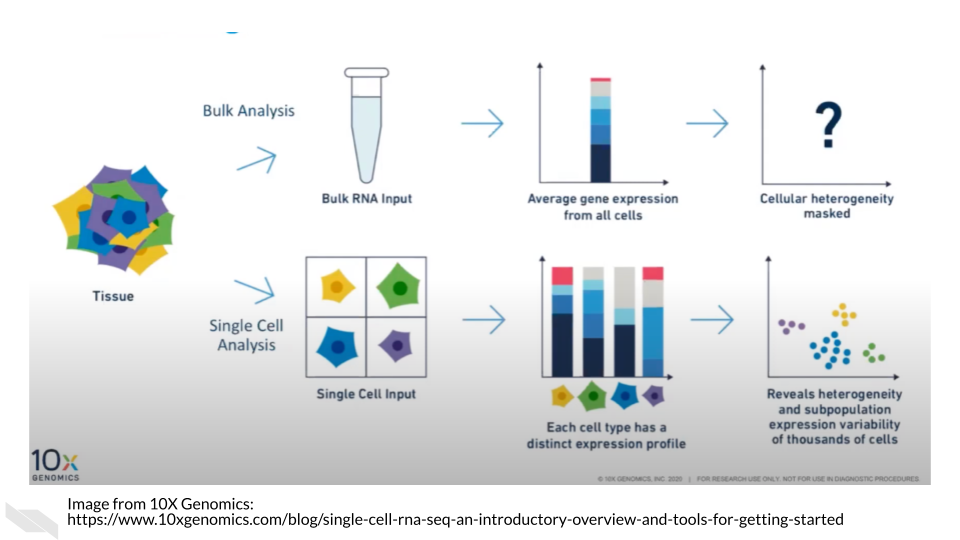


## 9.2 Where single-cell RNA-seq data comes from



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.



## 9.3 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.

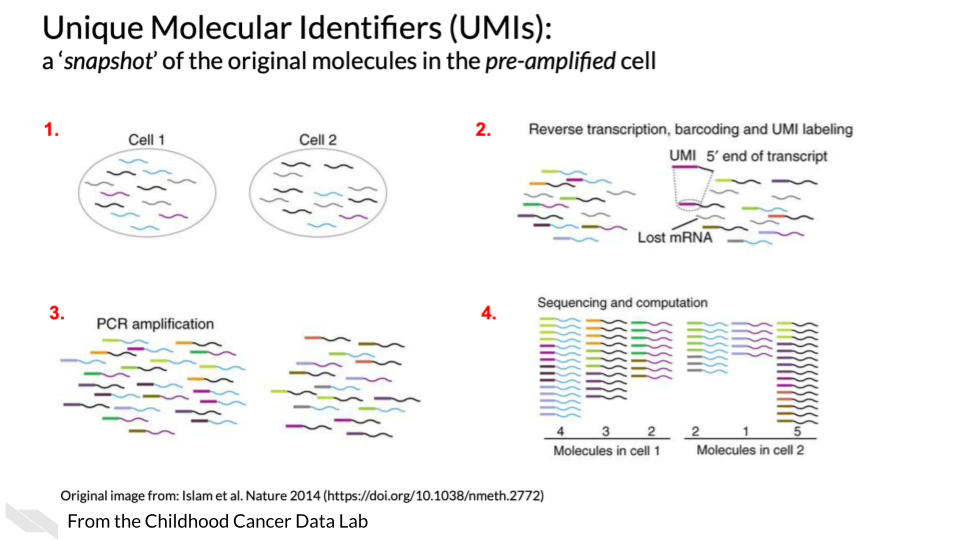




(Material borrowed from ([“Alex’s Lemonade Training Modules” 2022](#ref-AlexsLemonade2022))).

### 9.3.1 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.



## 9.4 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).

## 9.5 Quantification/Alignment

TODO: How to choose between different quantification tools

* [Alevin](https://salmon.readthedocs.io/en/latest/alevin.html)
* [CellRanger](https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger)
* [Kallisto bustools](https://www.kallistobus.tools/)

## 9.6 Quality control

TODO: More about considerations for quality control

#### 9.6.0.1 Checking UMIs

* [AlevinQC](https://bioconductor.org/packages/release/bioc/html/alevinQC.html)

#### 9.6.0.2 Checking for doublets

* [scDblFinder](https://bioconductor.org/packages/release/bioc/html/scDblFinder.html)
* [DoubletDetection](https://doubletdetection.readthedocs.io/en/stable/)

## 9.7 Normalization and downstream analyses

TODO: More about considerations for downstream analysis and normalization

* [Seurat](https://satijalab.org/seurat/)
* [Single Cell Genome Viewer](https://github.com/KrasnitzLab/SCGV)
* For normalization [scater](https://bioconductor.org/packages/devel/bioc/vignettes/scater/inst/doc/overview.html)
* [scanpy](https://scanpy.readthedocs.io/en/stable/)

## 9.8 Useful tutorials

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

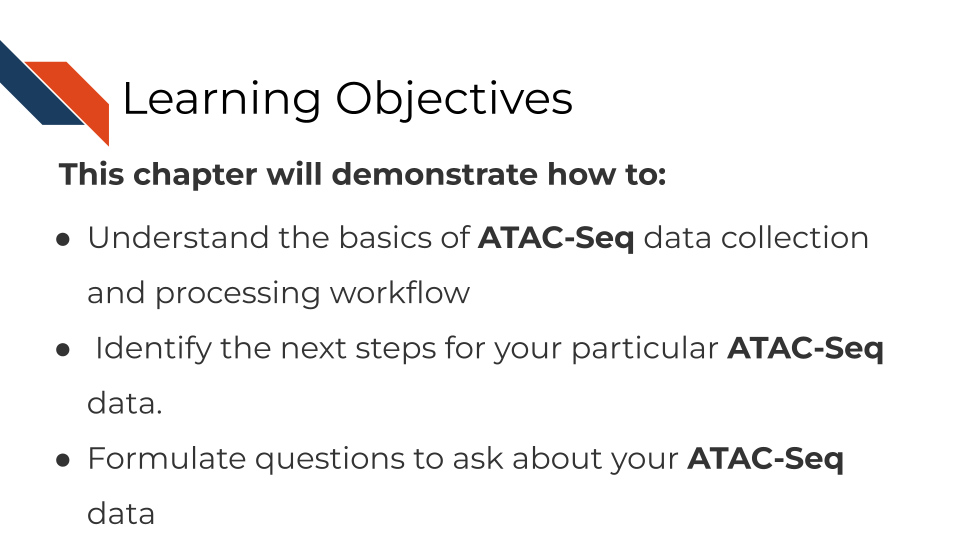
* [Orchestrating Single Cell Analysis with Bioconductor](http://bioconductor.org/books/3.15/OSCA.intro/) - An excellent tutorial for processing single cell data using Bioconductor.
* [Advanced Single Cell Analysis with Bioconductor](http://bioconductor.org/books/3.15/OSCA.advanced/) - a companion book 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.

## 9.9 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)).

# 10 ATAC-Seq

## 10.1 Learning Objectives



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

The goal of ATAC-seq is to….



# 11 ATAC-Seq

## 11.1 Where ATAC-Seq data comes from

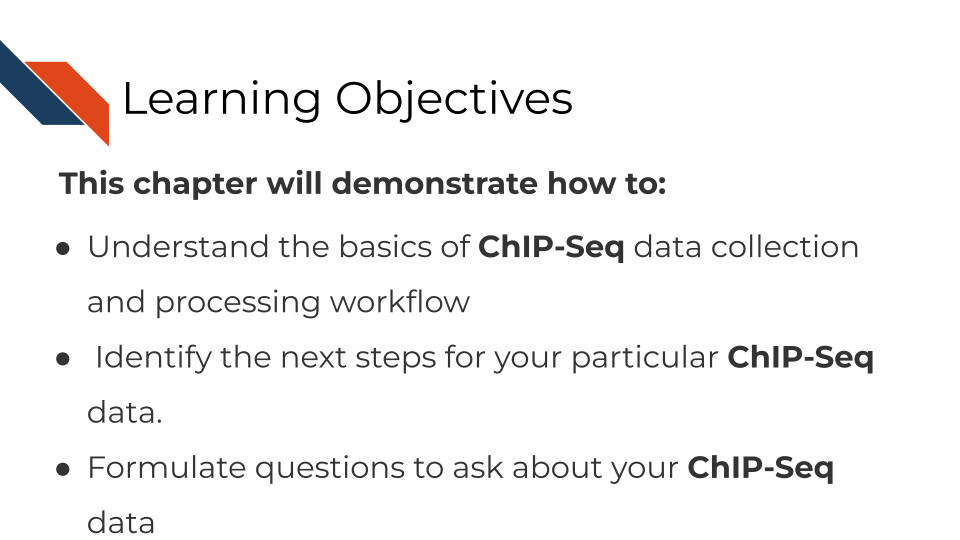
## 11.2 ATAC-Seq data **strengths**:

## 11.3 ATAC-Seqdata **limitations**:

## 11.4 More about ATAC-Seq

# 12 ChIP-Seq

## 12.1 Learning Objectives



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

The goal of ChIP-Seq is to….



## 12.3 Where ChIP-Seq data comes from

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

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

## 12.6 More about ChIP-Seq

# 13 DNA Methylation Sequencing Analysis Methods

## 13.1 Learning Objectives

{r, fig.alt = "This chapter will demonstrate how to: Understand the basics of bisulfite sequencing data collection and processing workflow. Identify the next steps for your particular bisulfite sequencing data. Formulate questions to ask about your bisulfite sequencing data ", out.width = "100%", echo = FALSE} ottrpal::include\_slide("https://docs.google.com/presentation/d/1YwxXy2rnUgbx\_7B7ENH9wpDX-j6JpJz6lGVzOkjo0qY/edit#slide=id.g12890ae15d7\_0\_71")

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



## 13.3 Methylation vs Hydroxymethylation

## 13.4 Bisulfite/Oxidative bisulfite sequencing

## 13.5 Methylation microarrays

# 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 Directors | [Jeff Leek](https://jtleek.com/) |
| Content Consultants |  |
| 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/) |
| 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 | Emily Voeglein, Fallon Bachman |

## ─ 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 2022-10-25   
##   
## ─ 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

“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>.

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>.

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>.

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>.

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>.

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>.

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/>.

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>.

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>.

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>.