Choosing Genomics Tools

May, 2022

Table of Contents

# About this Course

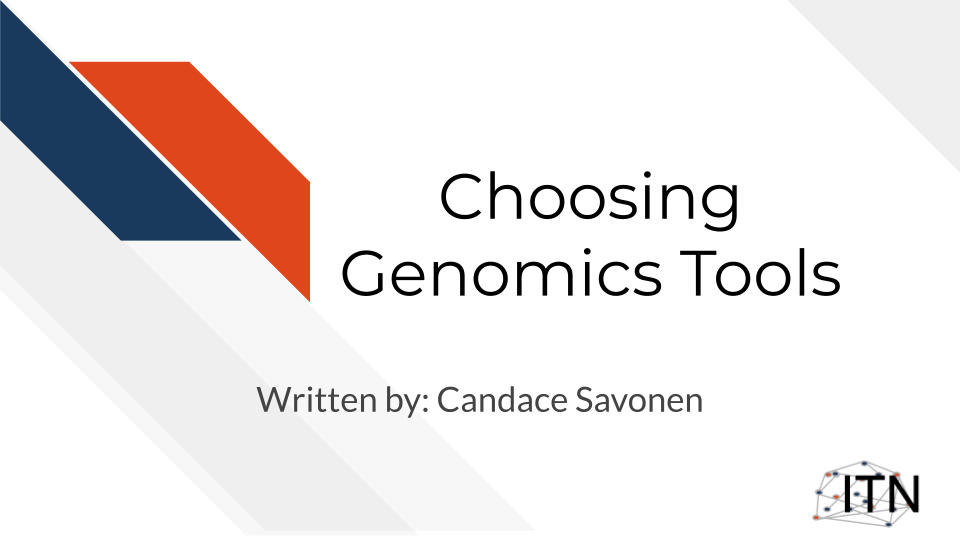
This course is part of a series of courses for the [Informatics Technology for Cancer Research (ITCR)](https://itcr.cancer.gov/) called the Informatics Technology for Cancer Research Education Resource. This material was created by the ITCR Training Network (ITN) which is a collaborative effort of researchers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

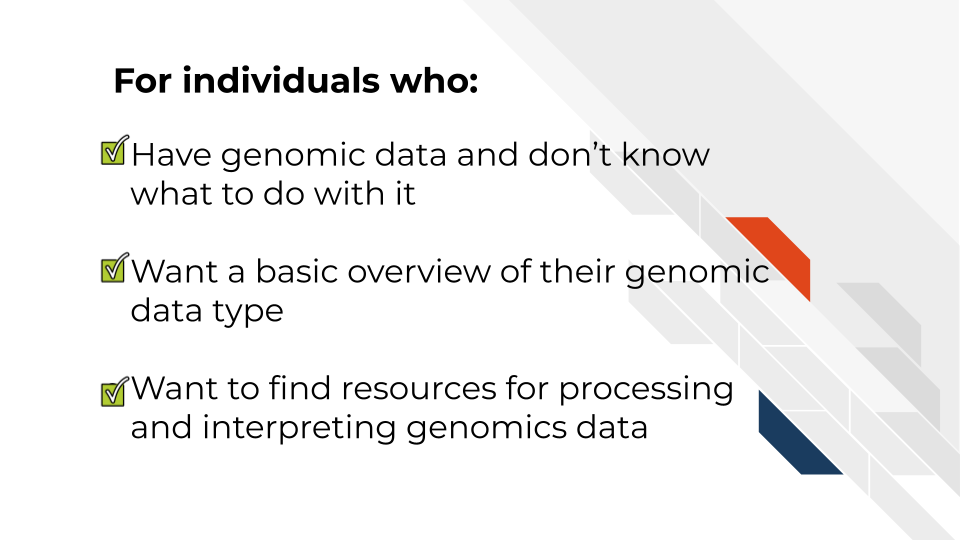


## 1.1 Target Audience

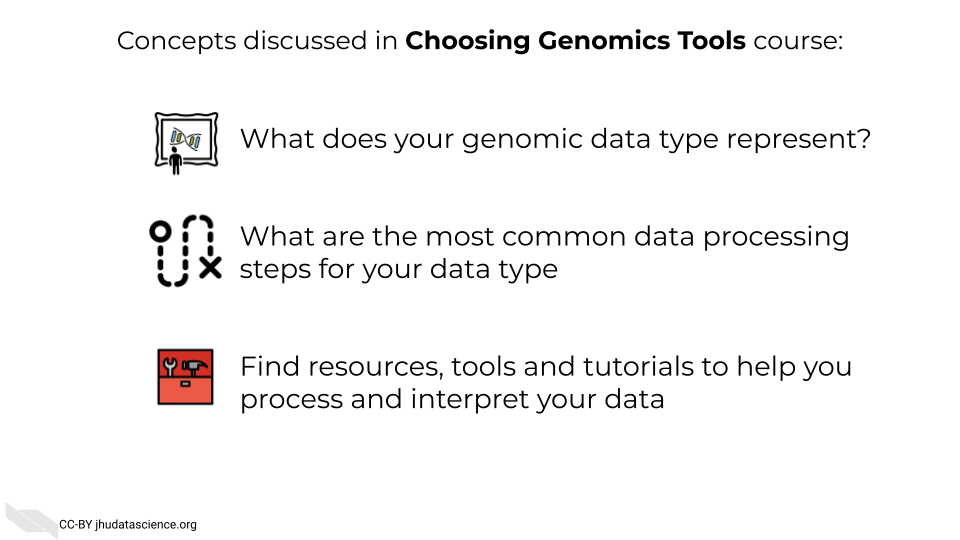
The course is intended for students in the biomedical sciences and researchers who have been given data and don’t know what to do with it or would like an overview of the different genomic data types that are out there.

*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:

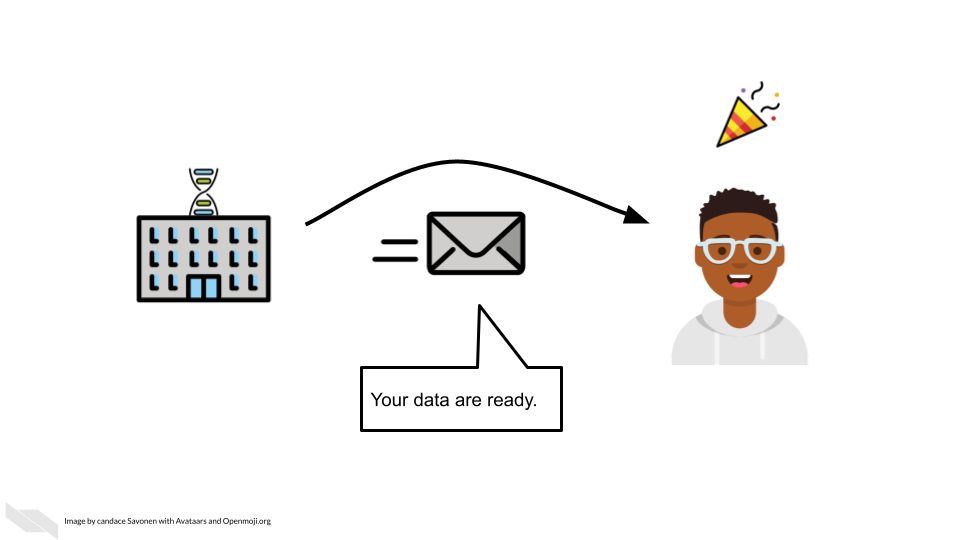


## 1.3 Motivation

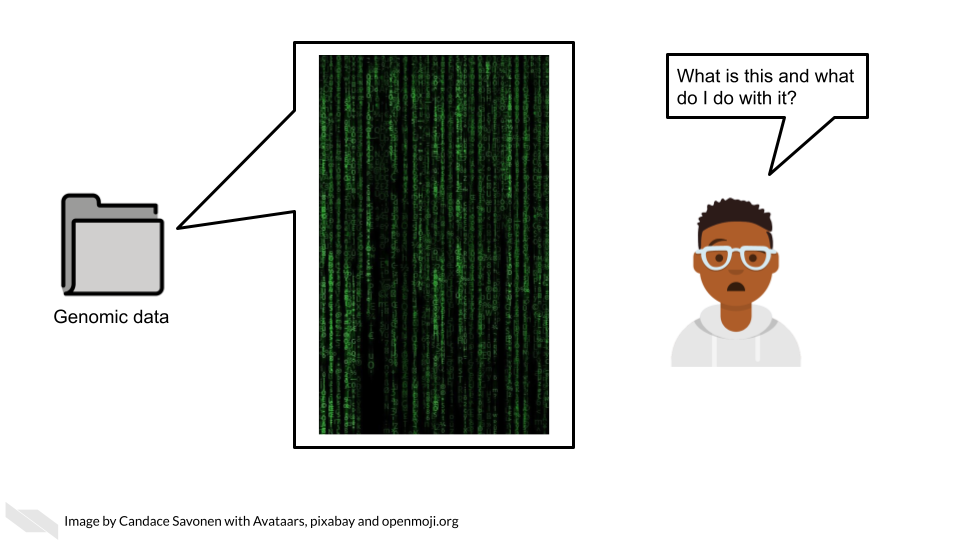
Cancer datasets are plentiful, complicated, and hold untold amounts of information regarding cancer biology. Cancer researchers are working to apply their expertise to the analysis of these vast amounts of data but training opportunities to properly equip them in these efforts can be sparse. This includes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

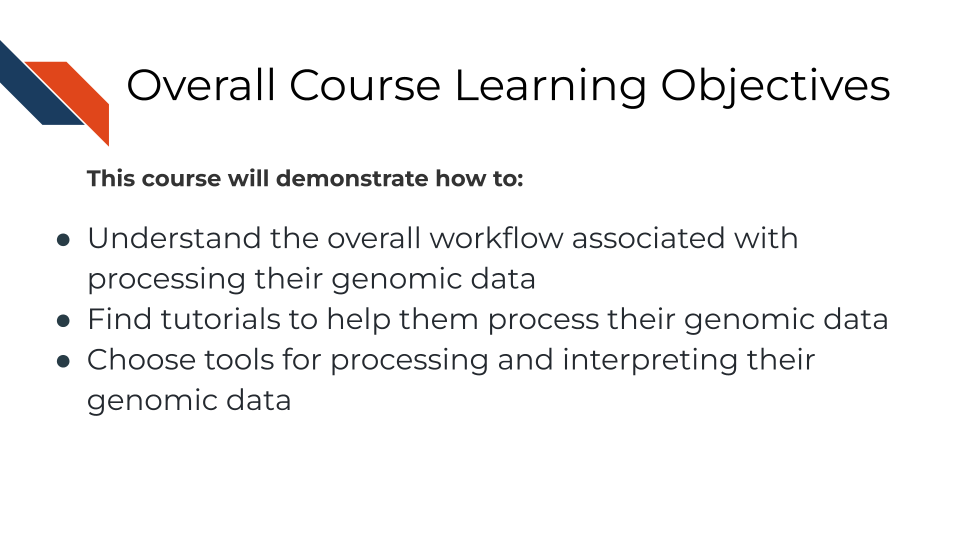


As an example, data file formats may not have been covered in their training, and the data they received seems unintelligible and not as straightforward as they hoped.



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

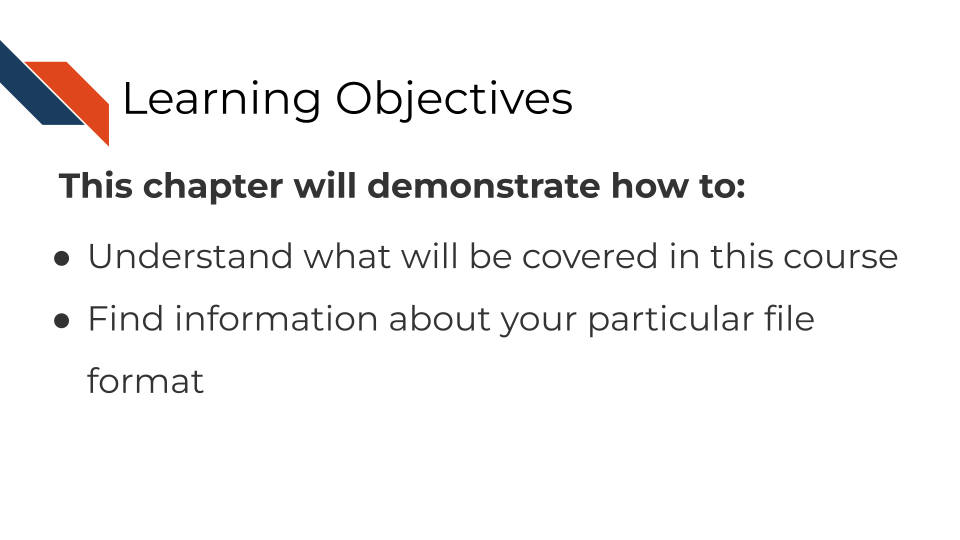
**What is not the goal**  
Teach learners about choosing parameters or about the ins and outs of every genomic tool they might be interested in. This course is meant to connect people to other resources that will help them with the specifics of their genomic data and help learners have more efficient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

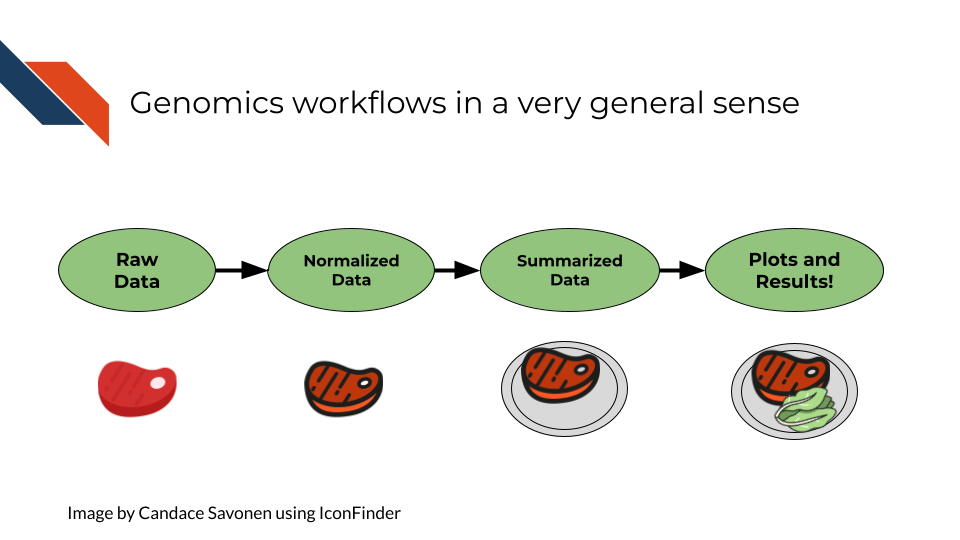
## 2.1 Learning Objectives



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

## 2.2 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:

### 2.2.1 Sequencing file formats

#### 2.2.1.1 SAM - Sequence Alignment Map

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

#### 2.2.1.2 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).

#### 2.2.1.3 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).

#### 2.2.1.4 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.

#### 2.2.1.5 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).

#### 2.2.1.6 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).

#### 2.2.1.7 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).).

## 2.3 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.):

### 2.3.1 Microarray file formats

#### 2.3.1.1 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).

#### 2.3.1.2 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).

#### 2.3.1.3 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)

#### 2.3.1.4 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).

## 2.4 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:

#### 2.4.0.1 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)).

#### 2.4.0.2 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.

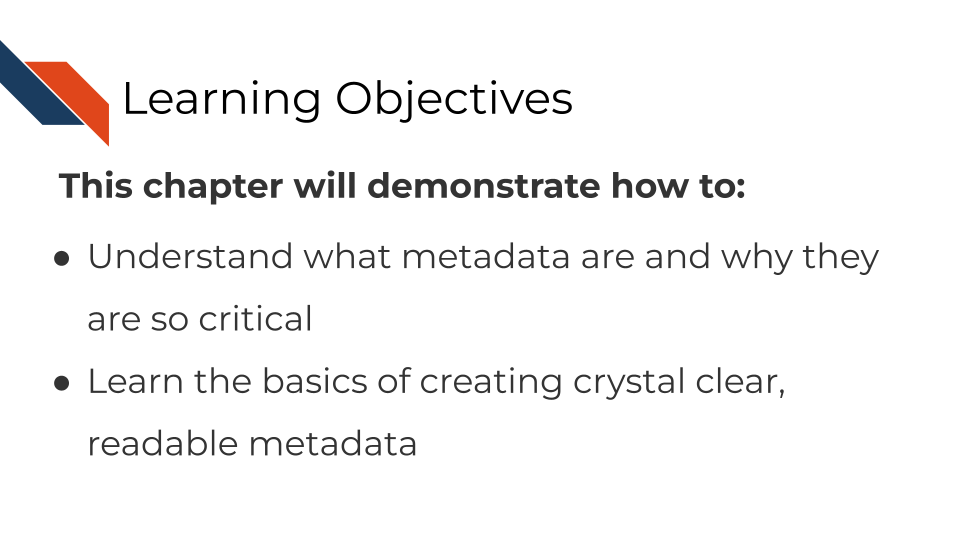
For [more about GTF and GFF files](https://uswest.ensembl.org/info/website/upload/gff.html).

### 2.4.1 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

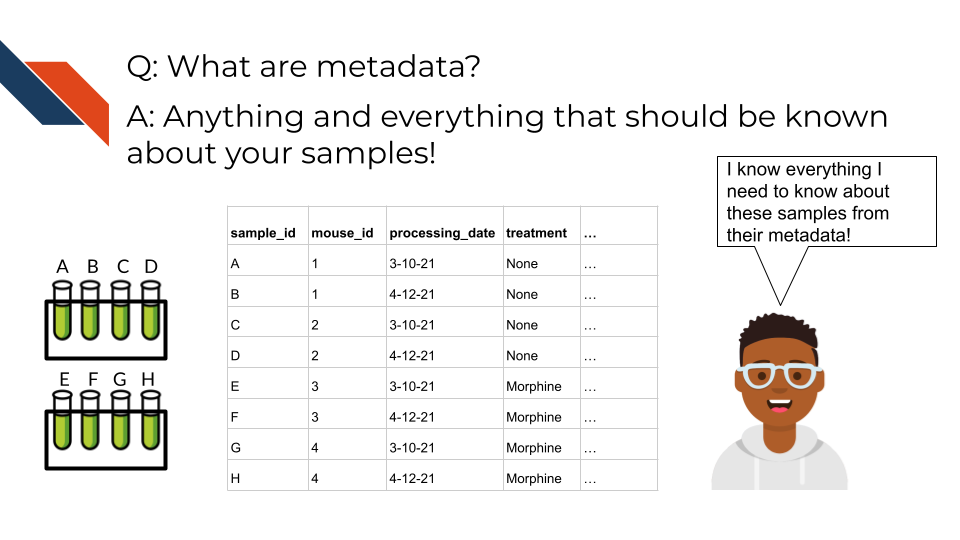


## 3.2 What are metadata?

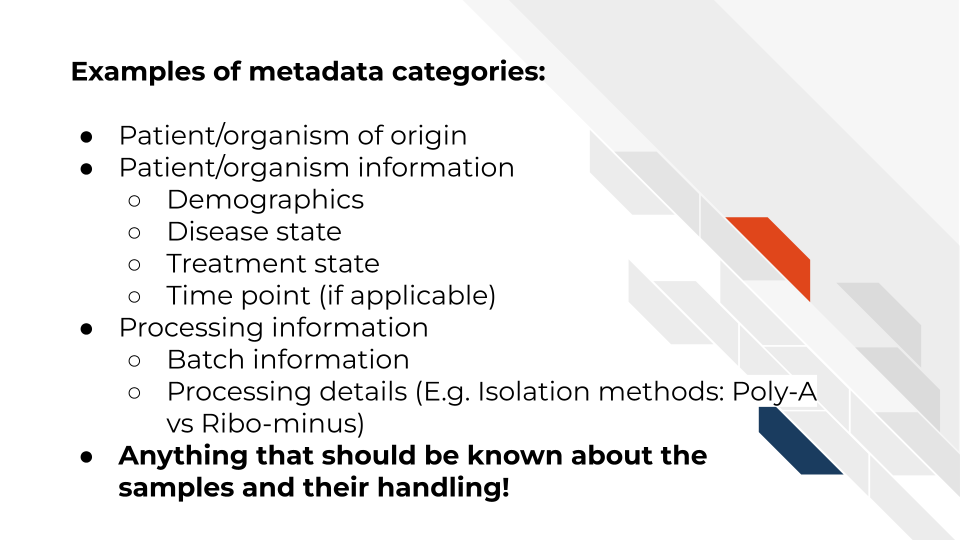
Metadata are critically important descriptive information about your data.

**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/Data_Management_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:

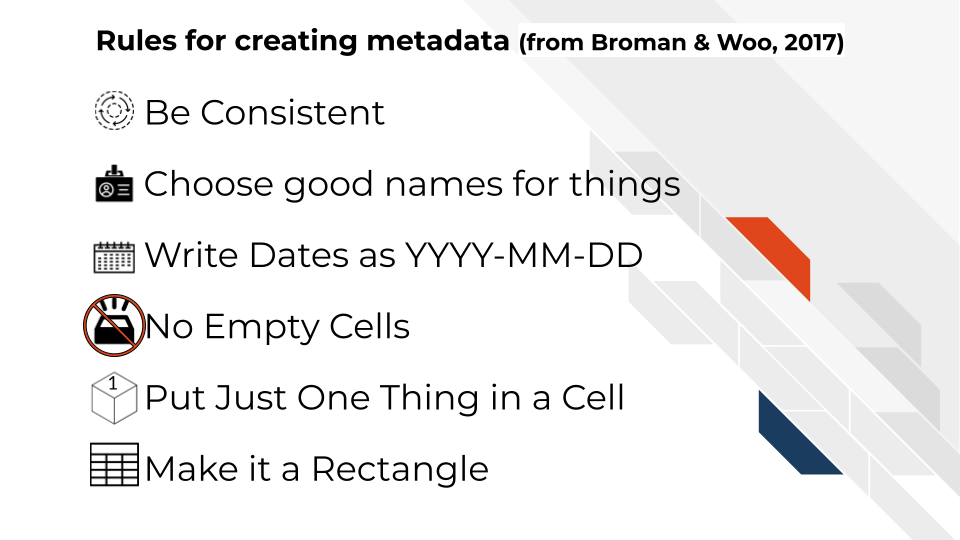
#### 3.3.1.1 Goal A: Make it *crystal clear* and *easily readable* by both humans and computers!

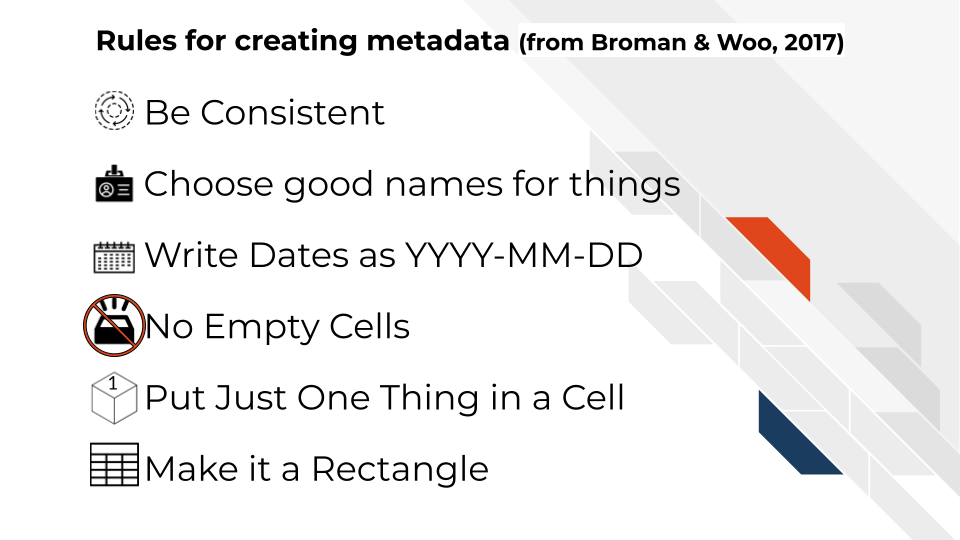
#### 3.3.1.2 Goal B: Avoid introducing errors into your metadata in the future!

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.

### 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!

# About the Authors

These credits are based on our [course contributors table guidelines](https://github.com/jhudsl/ottrpal_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 2022-05-13   
##   
## ─ 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