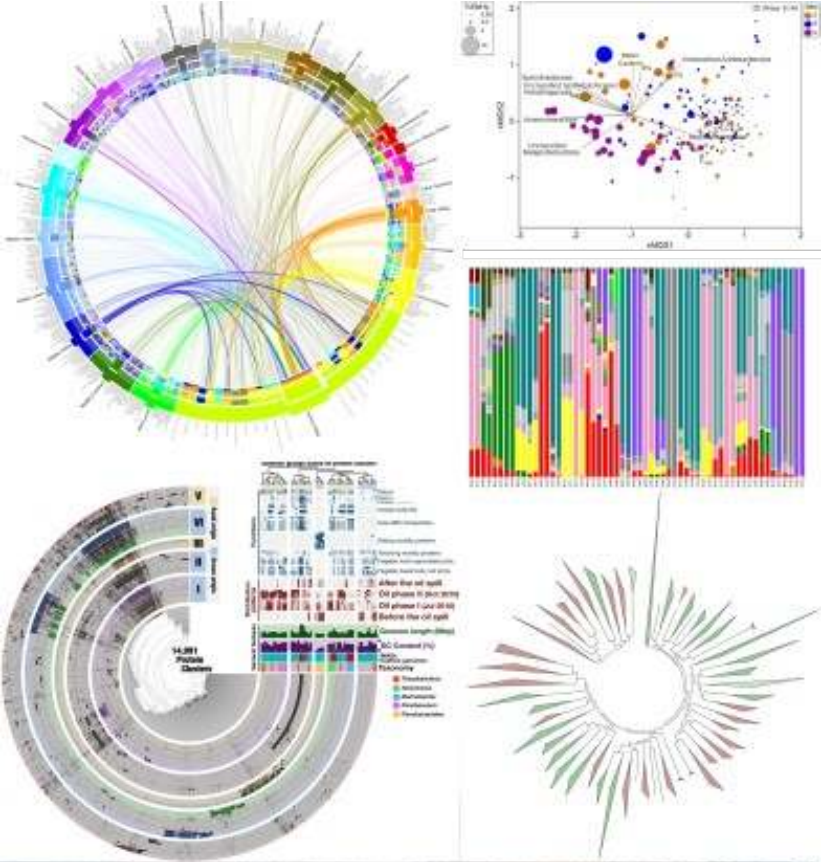


GIOVANNELLI LAB

DIPARTIMENTO DI BIOLOGIA

University of Naples Federico II

Environmental Metagenomic

aa 2020-2021

BIOINFORMATICS AND SEQUENCE HANDLING

Donato Giovannelli

donato.giovannelli@unina.it • @d_giovannelli • http://dgiovannelli.github.io

(bio)informatics

Bioinformatic is the use of informatics to study biology (broadly speaking). In the last two decades this has revolved mainly (but not only) around large scale sequence manipulation.

Today bioinformatic is a fast-moving field of research with far reaching applications that requires a deep knowledge of several different disciplines including (but not limited to) **ecology, biology, genetics, statistics, informatics, mathematics, etc..**

(bio)informatics

Bioinformatic is the use of informatics to study biology (broadly speaking). In the last two decades this has revolved mainly (but not only) around large scale sequence manipulation.

Today bioinformatic is a fast-moving field of research with far reaching applications that requires a deep knowledge of several different disciplines including (but not limited to) **ecology, biology, genetics, statistics, informatics, mathematics, etc..**

Most researchers "doing bioinformatics" are actually merely using the tools of bioinformatics. **Bioinformaticians develop new tools!**

Why it often happens on Linux (any flavor)

- *Linux is free as in speech!*
- *Linux is open source. Anyone can modify any aspect of it!*
- *Linux is very reliable. The* **Internet** *runs on linux servers.* **NASA**, *DOE,* **Interpol**, *DARPA,* **NORAD**, *ISS, and many many other run on linux*
- *It is very stable and blocked programs do not destabilize the session*
- *There are virtually no viruses for Linux (very few)*
- *Supports natively all programming languages*
- *Most commercial software have free linux alternative or stable linux versions*

Linux – Unix - OsX???

UNIX is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center

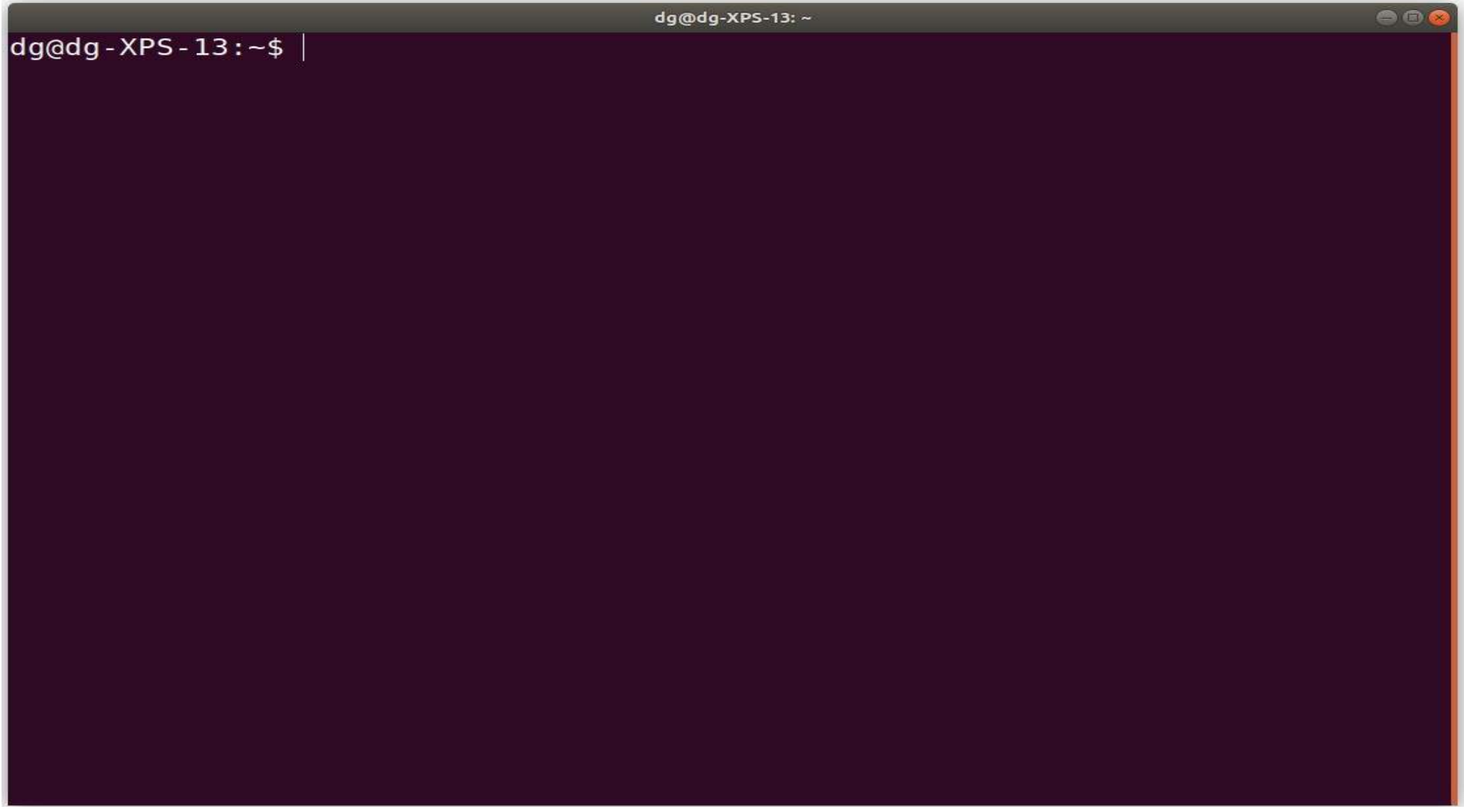
Linux is a type of Unix based operative system (so is OSX by Mac)

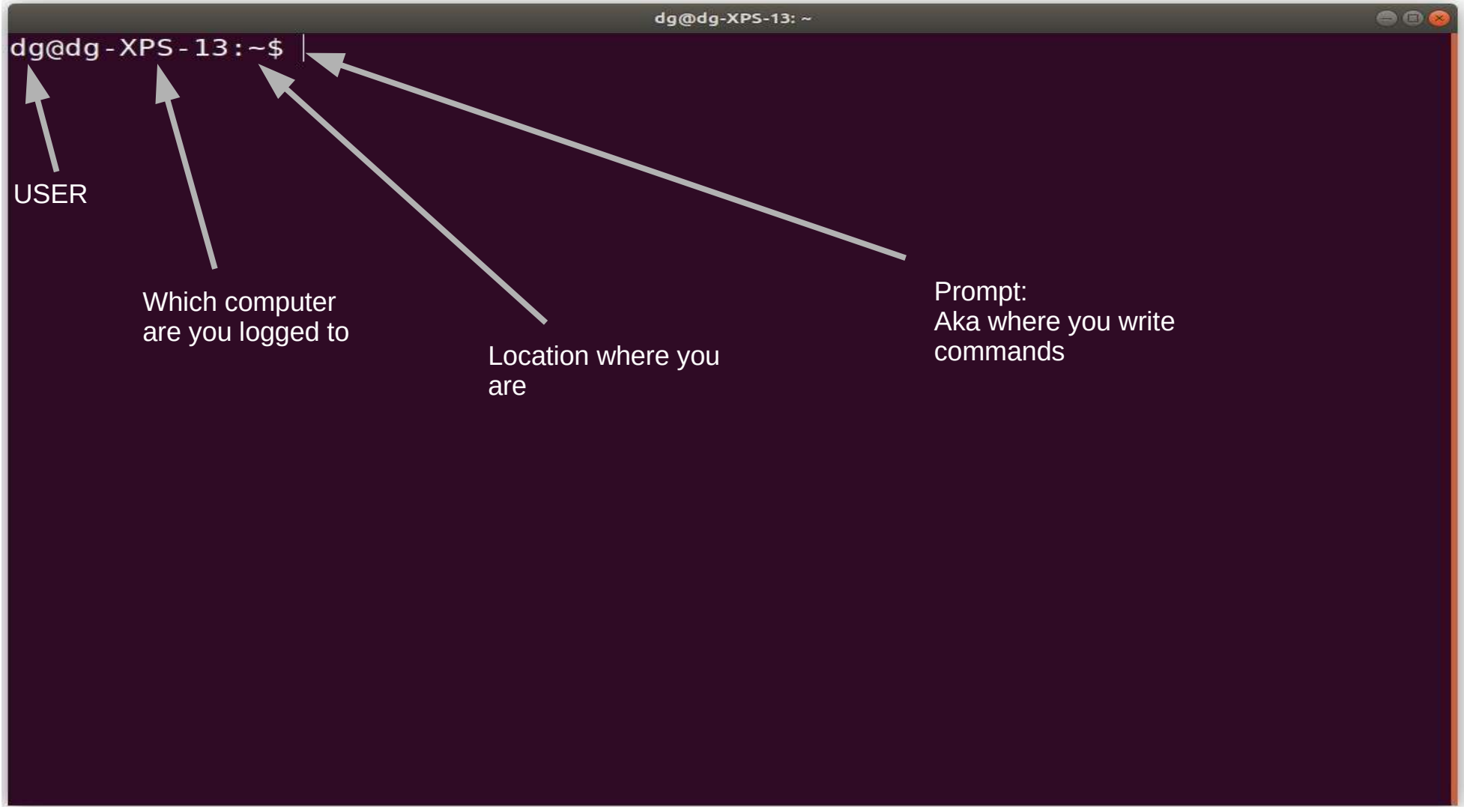
Most of the free and open source software is developed in and for Unix systems

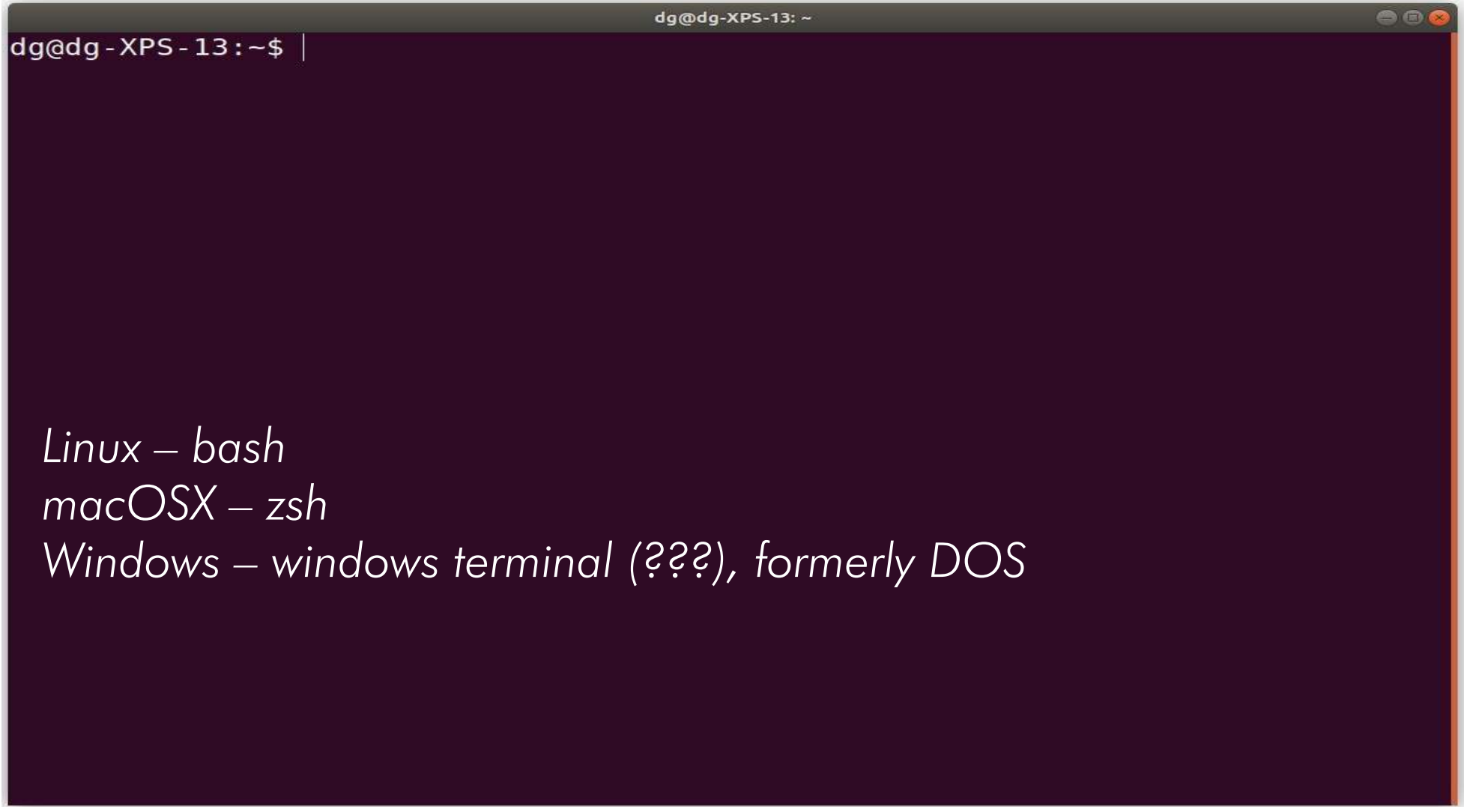
Bioinformatic is also (mostly) possible on Microsoft machine, but you will have to pay with a variable amount of pain…

Don't be afraid of the ***shell****?*

Don't be afraid of the ***shell****?*


dg@dg-XPS-13: ~
dg@dg-XPS-13:~$


dg@dg-XPS-13: ~
dg@dg-XPS-13:~$
USER
Which computer are you logged to
Location where you are
Prompt:
Aka where you write commands



Linux – bash
macOSX – zsh
Windows – windows terminal (???), formerly DOS

Things to know: absolute vs relative path

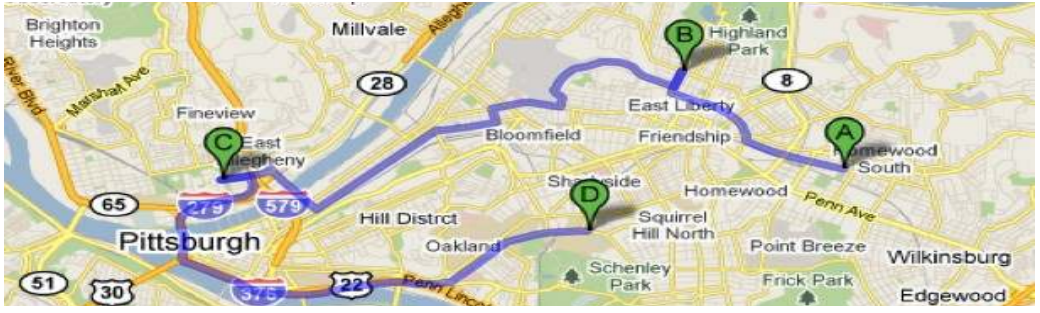
There is a big difference in the use of absolute vs relative paths when calling a command

An ***absolute path*** *refers to the complete details needed to locate a file or folder, starting from the root element and ending with the other subdirectories*

A ***relative path*** *refers to the position of a file or subforlder relative to where the command is being executed*

`/home/dg/Desktop/experiment/draft_dataset.csv` – *Absolute path*

`~/experiment/draft_dataset.csv` – *Relative path*



Things to know: miscellaneous

- *Unix systems are cAsE sensitive*
- *Avoid using sp a c es in files/folders names*
- *Avoid using com,ma (,) dot.s (.) or special chara/cters in any name (e.g. ?|\/!@#$%^&*). They have special meaning and should be avoided*
- *- and _ are allo_wed and en-couraged*
- *Suggestion: use short, unique names, possibly in lowercase*

Things to know: naming suggestions

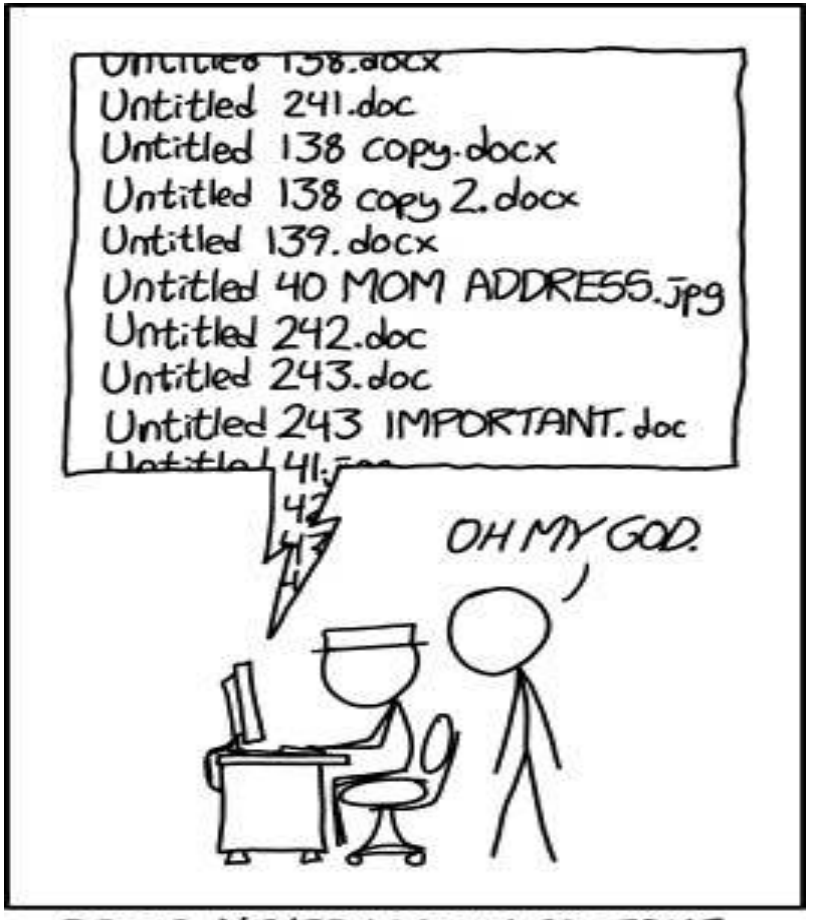
Use names that are **human readable**, **machine readable** *and* **behave well in default sorting and searching**

Human readable: *you can understand its content and context based on the name alone*

Machine readable: *can use regex to search it in different ways, and can make operations on the file name or file groups*

Behaves well: *can be sorted easily*

Any approach goes, if **consistent** *and* **well designed**



Most common names in your computers

Untitled importante.csv

Dati esperimento 1.xlsx

dati esperimenti 2 e 3.xls

Draft paper topi Ultima versione.docx

Draft paper topi Ultima versione/ultima.docx

Draft paper topi Ultima ultima versione con commenti.docx

Draft paper topi nuova versione.docx

File note importante.docx

Fig. 1.jpg

figure2.JPG

File importantissimo (tesi dottorato/anno1).docx

some-how-decent names

EXPERIMENT_TAKE1_2019_aa1.csv
Fig1_scatterplot_temp_ph.jpg
Giovannelli_et_al_new_paper_draft_001.rtf
Fig2_boxplot_antibiotic_resistance.jpg
AWESOME_EXPERIMENT_TAKE4_2019_aa4.csv

some-how-decent names

EXPERIMENT_TAKE1_2019_aa1.csv
Fig1_scatterplot_temp_ph.jpg
Giovannelli_et_al_new_paper_draft_001.rtf
Fig2_boxplot_antibiotic_resistance.jpg
AWESOME_EXPERIMENT_TAKE4_2019_aa4.csv

God-level names!

2019-09-17_MY_AWESOME_EXPERIMENT_TAKE1__aa1.csv
2019-11-22_MY_AWESOME_EXPERIMENT_TAKE2_aa2.csv
2019-02-01_Giovannelli_et_al_new_paper_draft_001.rtf
2019-11-28_experiment1_figure3_scatterplot.csv
2019-12-01_MY_AWESOME_EXPERIMENT_TAKE4_aa4.csv

Naming suggestions

Be consistent in your use of capital letters

Avoid spaces and use instead dash – and underscores _

Absolutely avoid any special chatracter

Add a date if possible

Indent numbers (001 and not 1), otherwise ordering by name gets messy

Use – and _ deliberately in the text name:

`my_experiment1-2019_11_12-testing_multi_drug-mouse1.csv`

Can be easily imported in a sheet and extract info like this

`my_experiment1`	`2019_11_12`	`testing_multi_drug`	`mouse1`

"human readable" *(easy to understand what the heck it is, based on its name)*

```
Jennifers-MacBook-Pro-3:analysis jenny$ ls -1
```

```
01_marshal-data.md
01_marshal-data.r
02_pre-dea-filtering.md
02_pre-dea-filtering.r
03_dea-with-limma-voom.md
03_dea-with-limma-voom.r
04_explore-dea-results.md
04_explore-dea-results.r
90_limma-model-term-name-fiasco.md
90_limma-model-term-name-fiasco.r
Makefile
figure
helper01_load-counts.r
helper02_load-exp-des.r
helper03_load-focus-statinf.r
helper04_extract-and-tidy.r
tmp.txt
```

```
01.md
01.r
02.md
02.r
03.md
03.r
04.md
04.r
90.md
90.r
Makefile
figure
helper01.r
helper02.r
helper03.r
helper04.r
tmp.txt
```

Which set of file(name)s do you want at 3a.m. before a deadline?

How to names files - Jennifer (Jenny) Bryan May 14, 2015
https://speakerdeck.com/jennybc/how-to-name-files

http://xkcd.com/1179/

File formats: they were not all born equal!

File formats can be both ***proprietary*** *or* ***open****.*

Proprietary file formats *are usually program specific and the code used to read them is also proprietary and usually private*

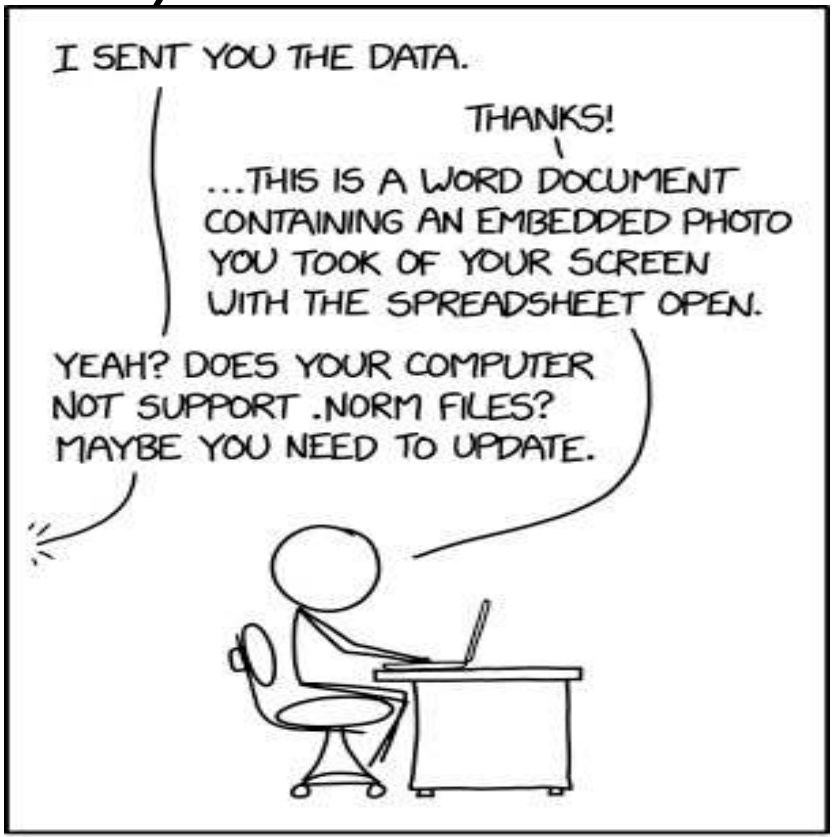
Open formats *have their specs freely available and in many cases are controlled by the International Standard Organization (ISO).*

Examples of proprietary file formats are .doc .docx .xlxs. .key .ai .psd .zip

Examples of open formats are .rtf .odt .svg .csv .png .html .gz .mp3 .pdf

THIS HAS VERY IMPORTANT PRACTICAL IMPLICATIONS!!!

File formats: they were not all born equal!



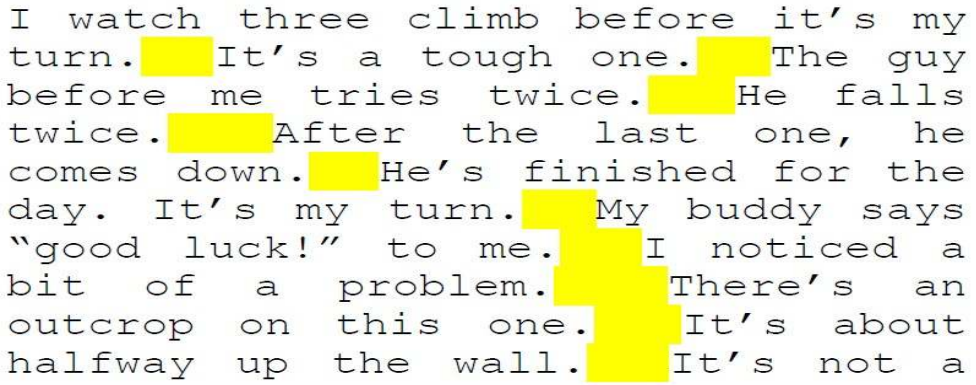
File formats: they were not all born equal!

File type	Common proprietary format	Open format
text	*.doc .docx*	*.txt .rtf .md .odt*
data	*.xls .xlsx*	*.csv .tab .ods*
graphs and line art	*.ai .jpg .psd*	*.svg .eps*
pictures	*.jpg .psd .raw*	*.png .tiff*

Regular expressions (RegEx)

*A **regular expression**, regex or regexp (sometimes called a rational expression) is a sequence of characters that define a search pattern.*

*Regular expressions are used in **search engines, search and replace** dialogs of word processors and text editors, in **text processing utilities** such as **sed** and **AWK** and in lexical analysis. Many programming languages provide regex capabilities either built-in or via libraries.*



The match results of the pattern **(?<=\.) {2,}(?=[A-Z])** *At least two spaces are matched, but only if they occur directly after a period (.) and before an uppercase letter.*

Sequences as text files

Nucleic acid and protein sequences can be visualized using a single character code that uniquely identifies the base or the amino acid present at each position in a sequence:

ATGAAACTCAGTCGTCGTAGCTTTATGAAAGCTAACGCCGTTGCGGCC

MKLSRRSFMKANAVAAAAAAAGLSVPGVARAVVGQQEAIKWDKAPCRFC

Representing sequences as text allows us to use simple text search operations on sequences

Sequences as text files

The sequences presented here are the gene sequence and protein sequences of the Periplasmic nitrate reductase subunit A (NapA) of Escherichia coli

ATGAAACTCAGTCGTCGTAGCTTTATGAAAGCTAACGCCGTTGCGGC

MKLSRRSFMKANAVAAAAAAAGLSVPGVARAVVGQQEAIKWDKAPCRFC

Sequences as text files

The sequences presented here are the gene sequence and protein sequences of the Periplasmic nitrate reductase subunit A (NapA) of Escherichia coli

ATGAAACTCAGTCGTCGTAGCTTTATGAAAGCTAACGCCGTTG

MKLSRRSFMKANAVAAAAAAAGLSVPGVARAVVGQQEAIKWDKAPCRFC

Sequences as text files

The search for promoters, specific portion of a sequence or conserved domains becomes a "simple" text search

CXCXXC ← *A conserved domain*

```
query       1 .[2].LSRRSFMKANAVAAAAAAAGLSVPGV.[3].VVG.[1].QEA     IKWDKAPCRFCGTGCGVLVGTQQGRVVA 66
152991895   1 .[2].LNRREFLKSAAAASAASAVGIAVPSS.[3].AAN.[1].AQK.[1].WRWDKAACRFCGTGCGIMLATKGGRIVA 67
62180834    1 .[2].LSRRSFMKANAVAAAAAAAGLSVPGV.[3].VVG     QQE.[1].IKWDKAPCRFCGTGCGVLVGTQQGRVVA 66
16130143    1 .[2].LSRRSFMKANAVAAAAAAAGLSVPGV.[3].VVG     QQE.[1].IKWDKAPCRFCGTGCGVLVGTQQGRVVA 66
152986752   1 .[2].LTRREFAKANAAAIAAAAAGLPILVR.[4].VTE.[1].DVT.[1].LDWNKAPCRFCGTGCSVMVATRDGQVVA 68
153950634   1 .[2].LSRRDFMKANAAVAAAAAAGMTIPTV.[3].VGE     TTN.[1].IKWDKAPCRFCGTGCGVLVGTQNGRIVA 66
157415051   1      MNRRDFIKNTAIASAASVAGLSVPSS     MLG.[1].QEE.[1].WKWDKAVCRFCGTGCGIMIARKDGKIVA 62
187929912   1 .[2].VSRRAFIKQTAAAATASVAGVTLPAG.[4].VTD.[1].ELT.[1].LKWSKAPCRFCGTGCGVEVAVKDNRVVA 68
224583254   1 .[2].LSRRSFMKANAVAAAAAAAGLSVPGV.[3].VVG     QQE.[1].IKWDKAPCRFCGTGCGVLVGTQQGRVVA 66
269138472   1 .[2].LSRRDFMKANAVAAAAAAAGLTIPTV.[3].VTE.[1].GSD.[1].ITWDKAPCRFCGTGCGVLVGTQNGRVVA 67
```

DNA sequences as text files: fasta

- >HSTDFTR::seq_12343 | HSBSYB | BDHDDNBD
  G**TATTAATA**CCCCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCCTCCTCATCGCTG
  GGCACAGCCCAGAGGGTATAAACAGTGCTGGAGGCTGGCGGGGCAGGCCAGCTGAGT
  CCTGAGCAGCAGCCCAGCGCAGCCACCGAGACACCATGAGAGCCCTCACACTCCTCG
  CCCTATTGGCCCTGGCCGCACTTTGCATCGCTGGCCAGGCAGGTGAGTGCCCCCACCT
  CCCCTCAGGCCGCATTGCAGTGGGGGCTGAGAGGAGGAAGCACCATGGCCCACCTCT
  TCTCACCCCTTTGGCTGGCAGTCCCTTTGCAGTCTAACCACCTTGTTGCAGGCTCAATC
  CATTTGCCCCAGCTCTGCCCTTGCAGAGGGAGAGGAGGGAAGAGCAAGCTGCCCGAG
  ACGCAGGGGAAGGAGGATGAGGGCCCTGGGGATGAGCTGGGGTGAACCAGGCTCCC
  TTTCCTTTGCAGGTGCGAAGCCCAGCGGTGCAGAGTCCAGCAAAGGTGCAGGTATGAG
  GATGGACCTGATGGGTTCCTGGACCCTCCCCTCTCACCCTGGTCCCTCAGTCTCATTCC
  CCCACTCCTGCCACCTCCTGTCTGGCCATCAGGAAGGCCAGCCTGCTCCCCACCTGAT
  CCTCCCAAACCCAGAGCCACCTGATGCCTGCCCCTCTGCTCCACAGCCTTTGTGTCCA
  AGCAGGAGGGCAGCGAGGTAGTGAAGAGACCCAGGCGCTACCTGTATCAATGGCTGG
  GGTGAGAGAAAAGGCAGAGCTGGGCCAAGGCCCTGCCTCTCCGGGATGGTCTGTGGG
  GGAGCTGCAGCAGGGAGTG
-

Fasta file format

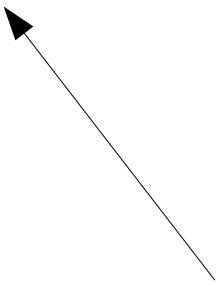
Definition: In bioinformatics, FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. The format also allows for sequence names and comments to precede the sequences.

Fasta file format

General Example:

```
>name comments and extra information

aaNNNCtCANcctgggtagacagaCAGATCGATGCTagatctgat
agctcgTQAGcgaatagCAGAtagCagatagCAagCAAGatagac
ACagaTgaCAgAtCagTAgagAGAcAgatGacaNNctgaANNNtg
acagac
```



Why upper and lower case?

Fasta file format

Example:

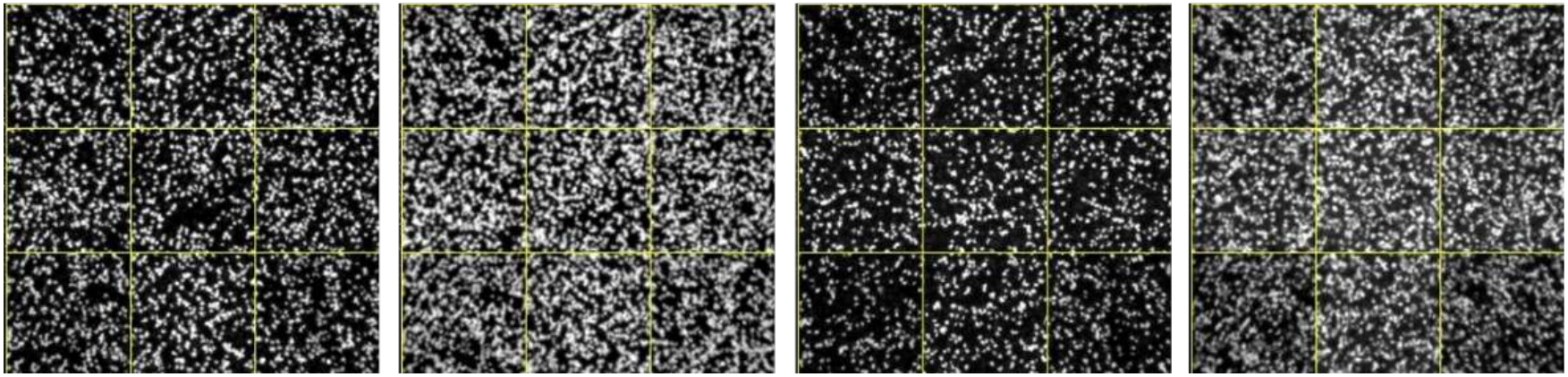
```
>gi|62000046|emb|CAI72603.1| NapA nitrate reductase [Desulfovibrio
desulfuricans]
MSTSRRDFLKYFAMSAAVAAASGAGFGSLALAADNRPEKWVKGVCRYCGTGCGVLVGVKDGKAVAI
QGDPNNHNAGLLCLKGSLLIPVLNSKERVTQPLVRRHKGGKLEPVSWDEALDLMASRFRSSIDMYG
PNSVAWYGSGQCLTEESYVANKIFKGGFGTNNVDGNPRLCMASAVGGYVTSFGKDEPMGTYADIDQ
ATCFFIIGSNTSEAHPVLFRRIARRKQVEPGVKIIVADPRRTNTSRIADMHVAFRPGTDLAFMHSM
AWVIINEELDNPRFWQRYVNFMDAEGKPSDFEGYKAFLENYRPEKVAEICRVPVEQIYGAARAFAE
SAATMSLWCMGINQRVQGVFANNLIHNLHLITGQICRPGATSFSLTGQPNACGGVRDGGALSHLLP
AGRAIPNAKHRAEMEKLWGLPEGRIAPEPGYHTVALFEALGRGDVKCMIICETNPAHTLPNLNKVH
KAMSHPESFIVCIEAFPDAVTLEYADLVLPPAFWCERDGVYGCGERRYSLTEKAVDPPGQCRPTVN
TLVEFARRAGVDPQLVNFRNAEDVWNEWRMVSKGTTYDFWGMTRERLRKESGLIWPCPSEDHPGTS
LRYVRGQDPCVPADHPDRFFFYGKPDGRAVIWMRPAKGAAEEPDAEYPLYLTSMRVIDHWHTATMT
GKVPELQKANPIAFVEINEEDAARTGIKHGDSVIVETRRDAMELPARVSDVCRPGLIAVPFFDPKK
LVNKLFLDATDPVSREPEYKICAARVRKA
```

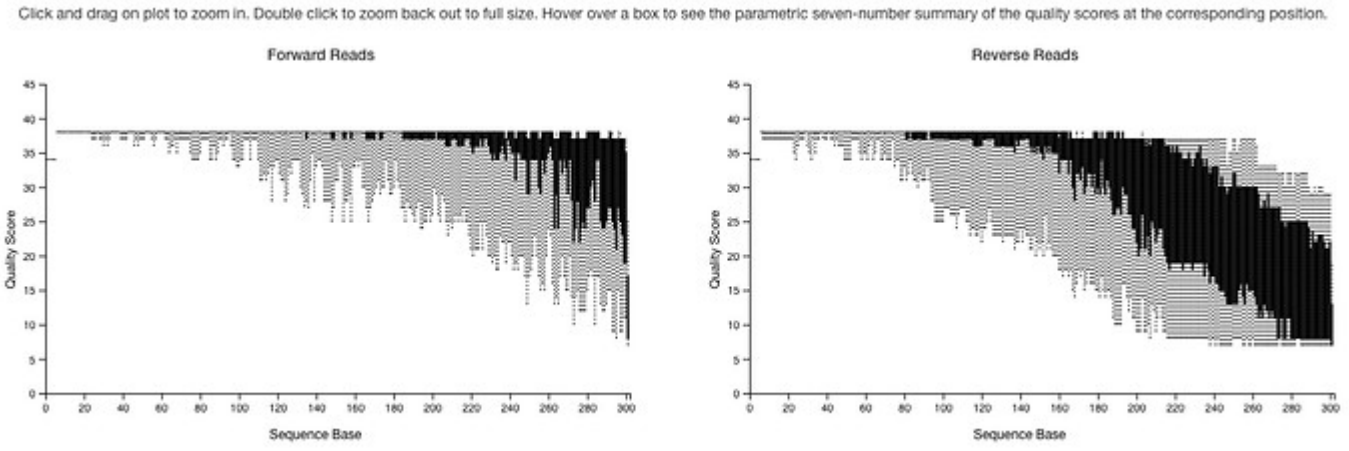
Multi fasta file

```
>Genus_species1 NapA nitrate reductase
MSTSRRDFLKYFAMSAAVAAASGAGFGSLALAADNRPEKWVKGVCRYCGTGCGV
>Genus_species2 NapA nitrate reductase
MSTSAVAAASRRDFLKYFAMSALAADNRPGVCRGAGFGSLAYCGTEKGCGKWVV
>Genus_species3 NapA nitrate reductase
MRDFLKYFSTSRAMSAAVAAASGALALAADGFGSNRPEKWVKGVCRYCGTGCGV
>Genus_species4 NapA nitrate reductase
MDFLKYFASGAGFGSLALAADNRPEKWTSRRMSAAVAAASVKGVCRYCCGVGTG
>Genus_species5 NapA nitrate reductase
MSTSFLKYFAMSAAVRRDAAASGAGFGSLALGVCRYCGTGAADNRPEKWVKCGV
>Genus_species6 NapA nitrate reductase
MSTSRRDFLKYFAMSAAVAAASGAGFGSLALAADNRPEKWVKGVCRYCGTGCGV
```

Join your unique names (_), especially if they are species names
Otherwise you'll get errors in a lot of downstream programs

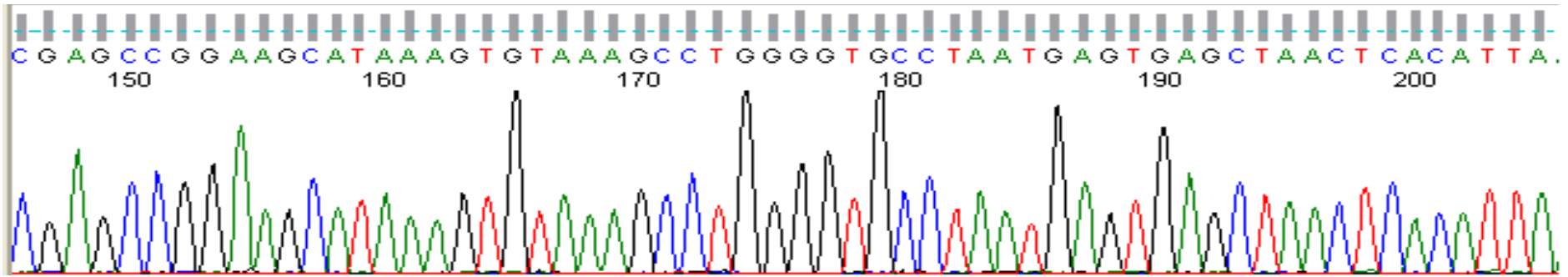
The nature of sequencing and base calling





.ab1 or .abi format (Sanger)

The ABI File Format (.abi or .ab1) is a binary file that is produced by ABI sequencer software. This data file, referred to as a "trace file", and contains the sequence as well the quality score and possible multiple peaks detected.



Single sequences

Windows – ???

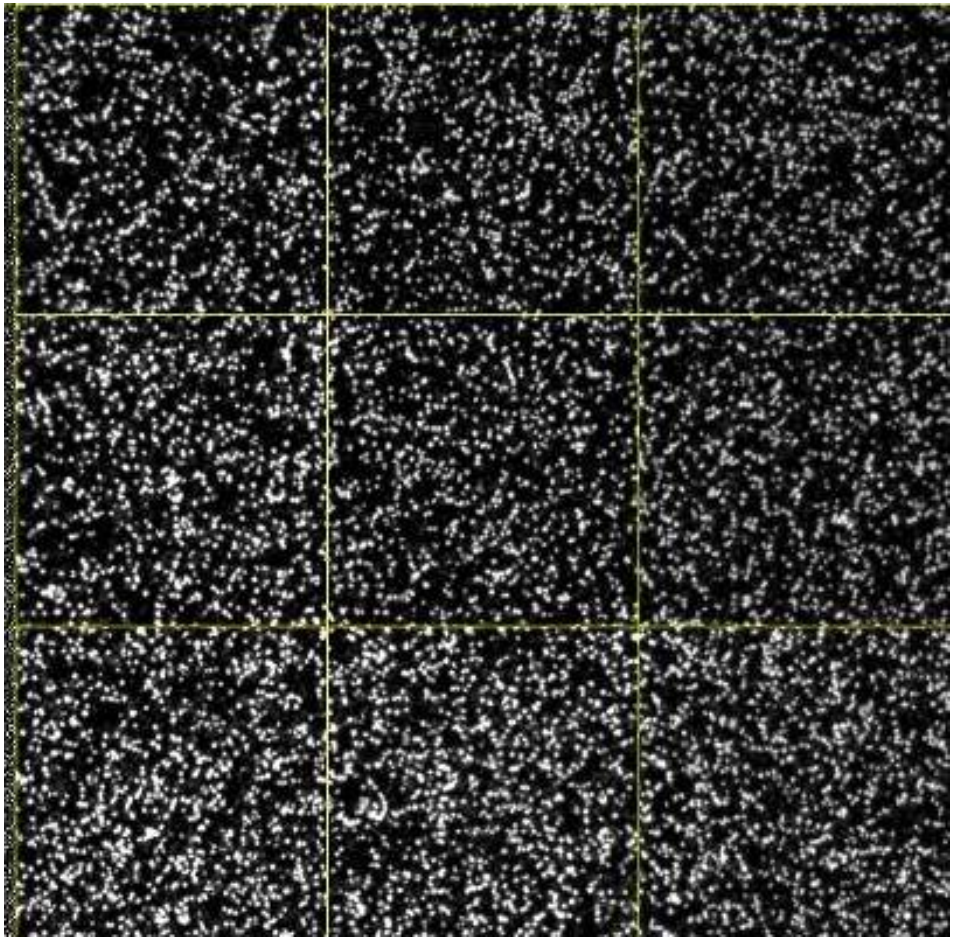
MacOSx – 4Peaks (https://nucleobytes.com/4peaks/index.html)

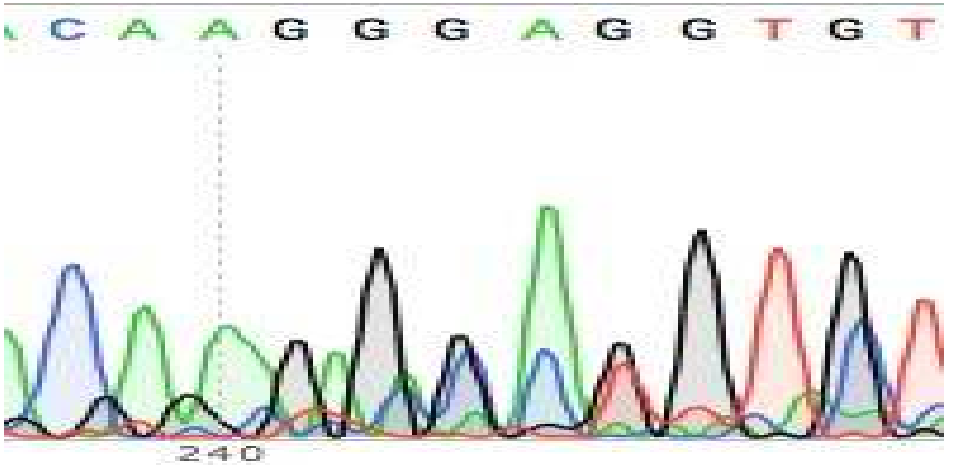
Linux – FinchTV (https://slackbuilds.org/repository/14.2/academic/finchtv/)

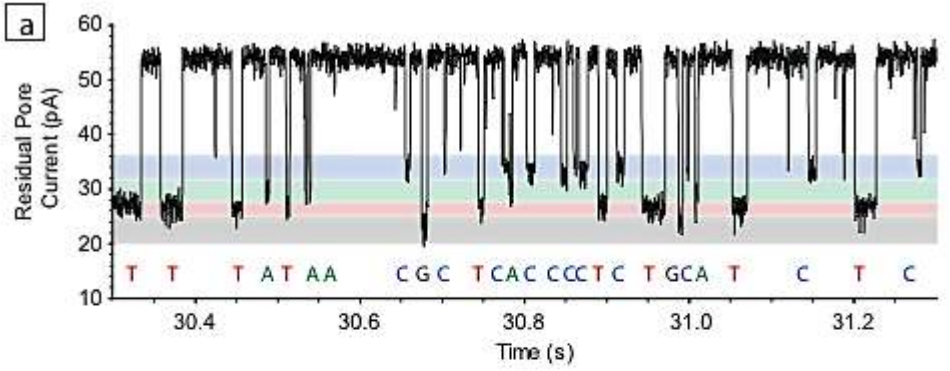
Multiple sequences

Linux – sangeranalyseR (https://github.com/roblanf/sangeranalyseR)

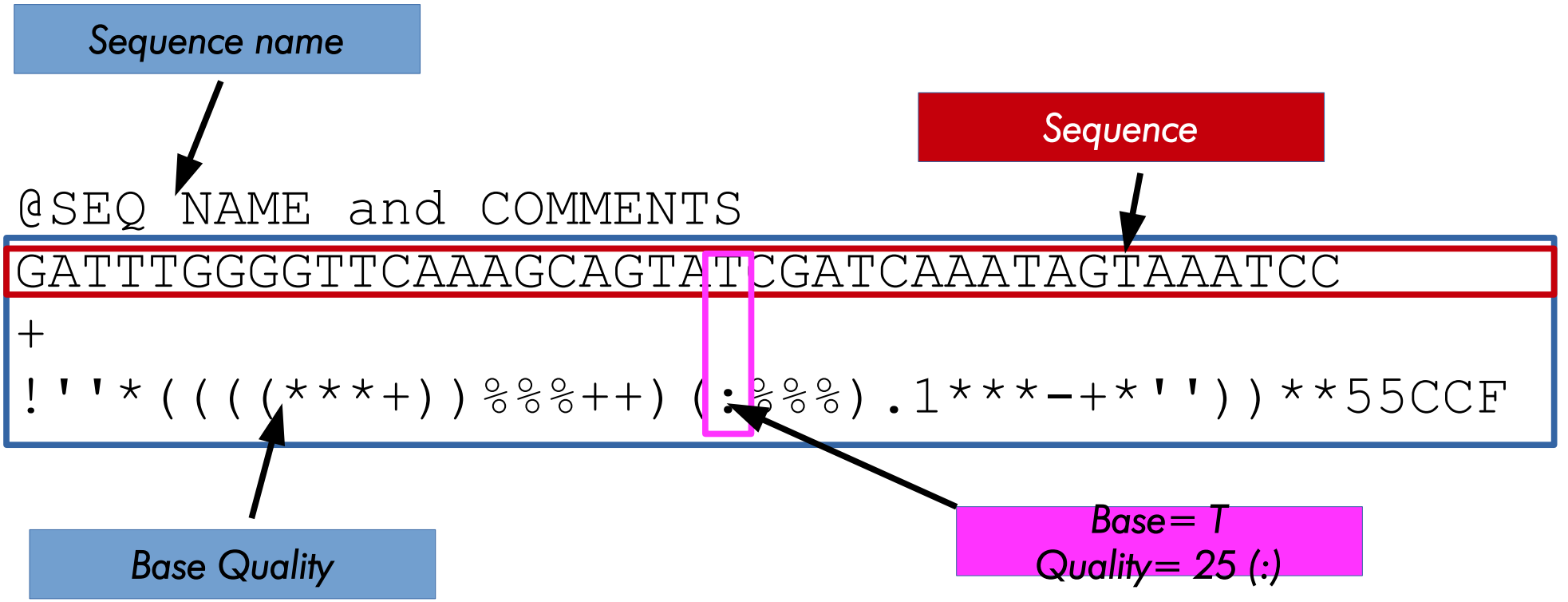
Sequences: fasta and fastq







Fastq files



more info: https://en.wikipedia.org/wiki/FASTQ_format

Fastq files

```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
 ..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
 ..................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                         |    |        |                              |                     |
33                        59   64       73                            104                   126
 0........................26...31.......40
                          -5....0........9.............................40
                               0........9.............................40
                                 3....9.............................41
 0.2......................26...31........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 41)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

Fasta naming convention

To efficiently use command line fasta manipulation is better to use a naming convention for the different type of fasta:

- *.fasta – general fasta file of unspecified content*
- *.fna – nucleotide fasta*
- *.faa – amino acid fasta*
- *.fastq – fasta and quality score combined (more on this later)*

Other important file extension:

- *.aln – alignment file (fasta format recommended)*
- *.tree – tree file (newick format recommended)*
- *.cvs – coma separated file for tabular data*
- *.txt – general text file for notes, data and extra info*

Fasta shell operations

Fasta file are essentially a special case of text file. We can perform text related operations using the shell to investigate or reformat (i.e. parse) fasta files.

Common commands used with fasta files

- `head <name.fasta>` - *show the first sequences*
- `cat <name.fasta>` - *show the entire file*
- `sed <name.fasta>` - *walk through the file content*
- `grep <word> <name.fasta>` - *find the line containing <word>*

Fasta shell operations: examples

Show the first 10 sequences of a name1.fasta file:

- `head -20 name1.fasta`

Join name1.fasta and name2.fasta:

- `cat name1.fasta name2.fasta >> newname.fasta`

substitute ; with a _ in every sequence of the file

- `sed 's/;/_/g' name1.fasta >> newname.fasta`

Get length of each fasta in a multi-fasta file

- `cat file.fa | awk '$0 ~ ">" {print c; c=0;printf substr($0,2,100) "\t"; } $0 !~ ">" {c+=length($0);} END { print c; }`

https://github.com/stephenturner/oneliners

Fasta shell operations: examples (continued...)

Count number of sequences in the file

- `cat name1.fasta | echo $((`wc -l`/2))`
- `grep -c "^>" name1.fasta`

Compress name1.fasta to save disk space:

- `gzip -c name1.fasta >> name1.fasta.gz`

to decompress use (or use while compressed)

- `gzip -d name1.fasta.gz >> name1.fasta`
- `zcat name1.fasta.gz | head -20`

USE GOOGLE TO FIND OUT HOW TO PERFORM OTHER OPERATIONS! THE SHELL IS A POWERFULL TOOL!

Accessing Bioinformatic resources

Webservers. *The beginner best friends are specific portals that allow the users to run specific analysis. Usually there are limitation on the size or type of analysis that can be run on these services*

R Bioconductor. *A collection of tools and libraries within the R framework*

Python – Biopython and Anaconda. *A collection of tools and libraries within the Phyton language*

Perl – Biperl. *A collection of tools and libraries within the Perl language*

This course

For this course we will use ***webservers only****.*

In reality my group uses a combination and webservers and installed tools that include R Bioconductor (the vast majority of the time) and Biopython and Anaconda.

What is R (ahrrrrrrrr!)

R *comes from a programming language named S+, which itself was based on S that was invented in Bell Labs in 1976*

If you do not the Bell Labs, google it. That is how a research place makes a (very long) lasting impact on society

R at its core is a programming language free to download, execute, adapt and redistribute. There's strong community support. It has strong graphing capabilities and suited for interactive data analysis.

It has hundreds of statistical packages (literally), and you can find a package to do any time of analysis you can (or maybe not even) imagine.

And if it does not exsist, you can write your own package

Beyond the terminal

There are several other ways you can interact with R. One of the most popular is the RStudio package. There are several versions of **RStudio,** *including a free one. RStudio bundles code editor, console, command history, debugging, documentation and visualization in a single install*

Another popular choice is to run R inside **Jupiter Notebooks**. *Jupiter Notebooks are an interactive environment were notes, graphs and code can interact and be easily shared. It works wonder for teaching purpose.*

The choice between the two depends on many factors. We will use Rstudio for ease of install, although I personally use a lot more Jupyter Notebooks for my personal research.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Untitled1*

Source on Save | Run | Source

```
1  demo(graphics)
2  ## comments
```

2:12 (Top Level) R Script

Console Terminal

```
Hit <Return> to see next plot:

> polygon(xx, yy, col="gray")

> title("Distance Between Brownian Motions")

> ## Colored plot margins, axis labels and titles.       You do need t
o be
> ## careful with these kinds of effects.       It's easy to go comple
tely
> ## over the top and you can end up with your lunch all over the keyb
oard.
> ## On the other hand, my market research clients love it.
>
> x <- c(0.00, 0.40, 0.86, 0.85, 0.69, 0.48, 0.54, 1.09, 1.11, 1.73, 2
.05, 2.02)

> par(bg="lightgray")

> plot(x, type="n", axes=FALSE, ann=FALSE)
Hit <Return> to see next plot:
```

Environment History Connections

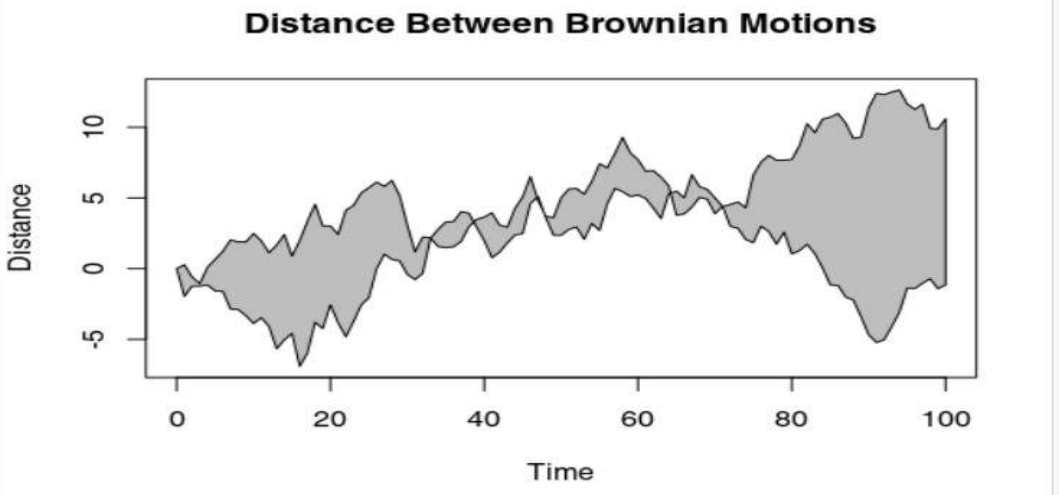
Import Dataset | List

Global Environment

usr	num [1:4] -448 1328 -14 634
x	num [1:12] 0 0.4 0.86 0.85 0.69 0.48 0.54 1.09 1.11 1.…
xadd	392.671782350889
xdelta	860
xscale	0.00761821705426356
xx	int [1:202] 0 1 2 3 4 5 6 7 8 9 ...
y	num [1:101] 0 -1.96 -1.24 -1.24 -1.15 ...
yadd	0
ydelta	600
yscale	0.00398194444444445
yy	num [1:202] 0 0.2783 -0.5827 -1.0614 0.0979 ...

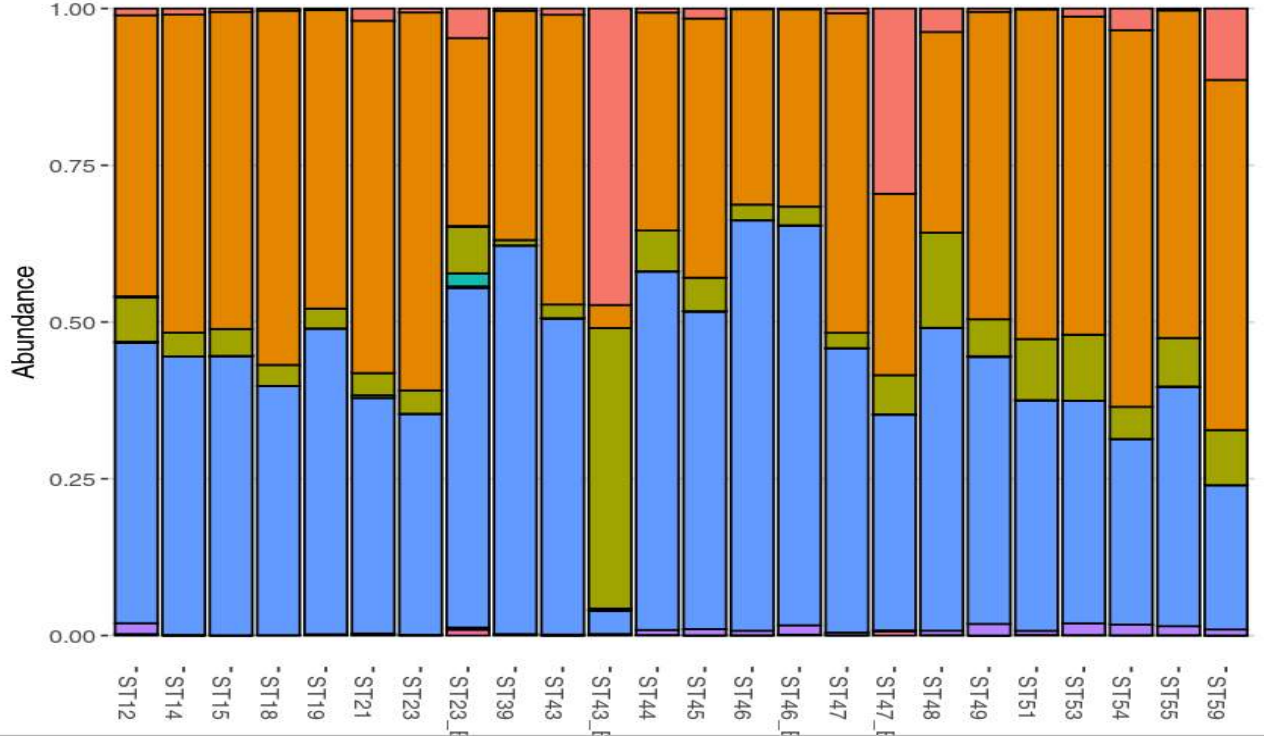
Files Plots Packages Help Viewer

Zoom | Export | Publish



Diversity plots and visualizations

In [103]:
```
# Plot Relative Abundance by Phyla for each station
plot_bar(bac_ra_phyla, fill="Phylum", x="station") +
 # facet_wrap(~depth_level) +
  theme_hc()
```



How I choose

I choose to use **Rstudio** *generally for* **quick and dirty analyses, testing, code development** *or when* **computational efficiency** *is needed*

I choose **Jupyter Notebooks** *every time i'm running* **analyses for a project**, *when I want to* **integrate code**, **plots** *and* **notes/observations**, *when I need to use* **more than one language at the time** *(python, R and bash), for* **long term readability** *and everytime I want to* **share the results with colleagues**

On the wake of publication I generally clean up my jupyter notebook code and upload it both as a **.ipynb** *and* **.r** *file on a* **GitHub** *repository to which I assign a* **DOI using Zenodo** *(more on this the day we speak about* **reproducibility in science**)

R Learning Resources

There are (literally) thousands of tutorials and resources out there. Here a few suggestions:

Data Carpentry "Data Analysis and Visualization in R for Ecologists"
https://datacarpentry.org/R-ecology-lesson/index.html

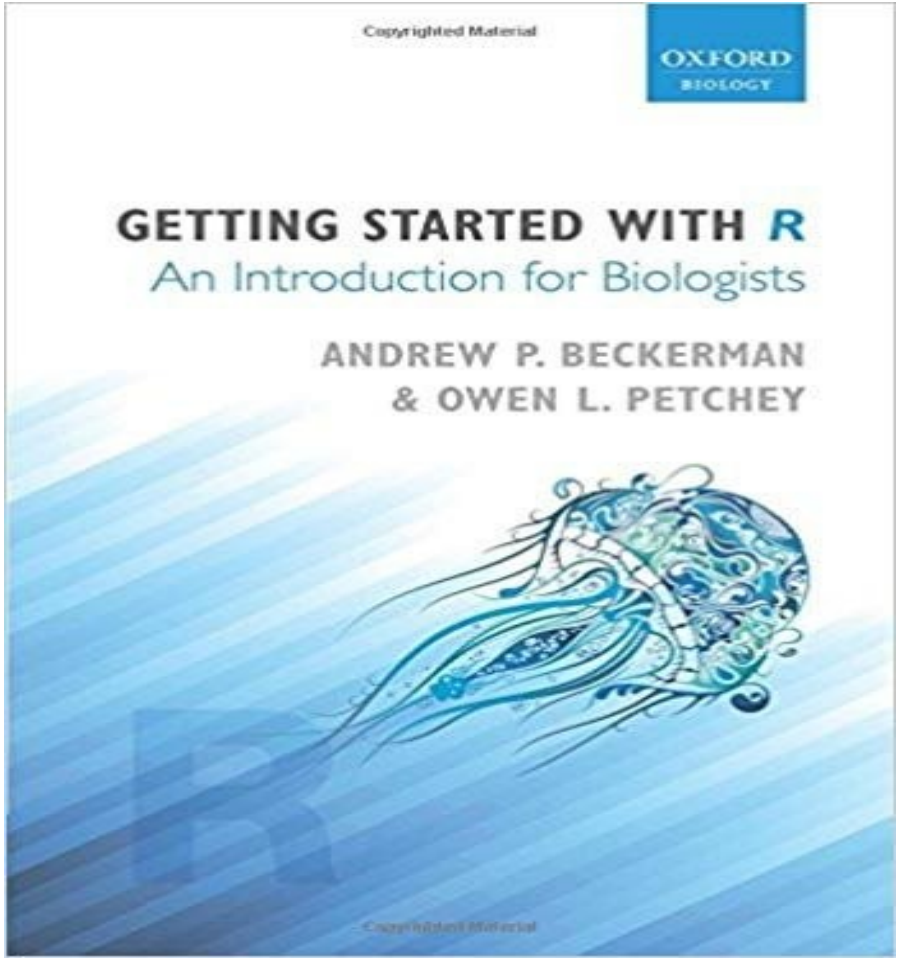
DataCamp "Introduction to R"
https://www.datacamp.com/courses/free-introduction-to-r

DataCamp "Quick-R" website https://www.statmethods.net/input/datatypes.html

Butler MA. 2009. Getting Started in R for Biologists.
http://www2.hawaii.edu/~mbutler/Rquickstart/simpleR.pdf

Venables NM, Smith DM and the R Core Team. 2019. "An Introduction to R"
http://www

R Learning Books



Getting Started with R: An introduction for biologists

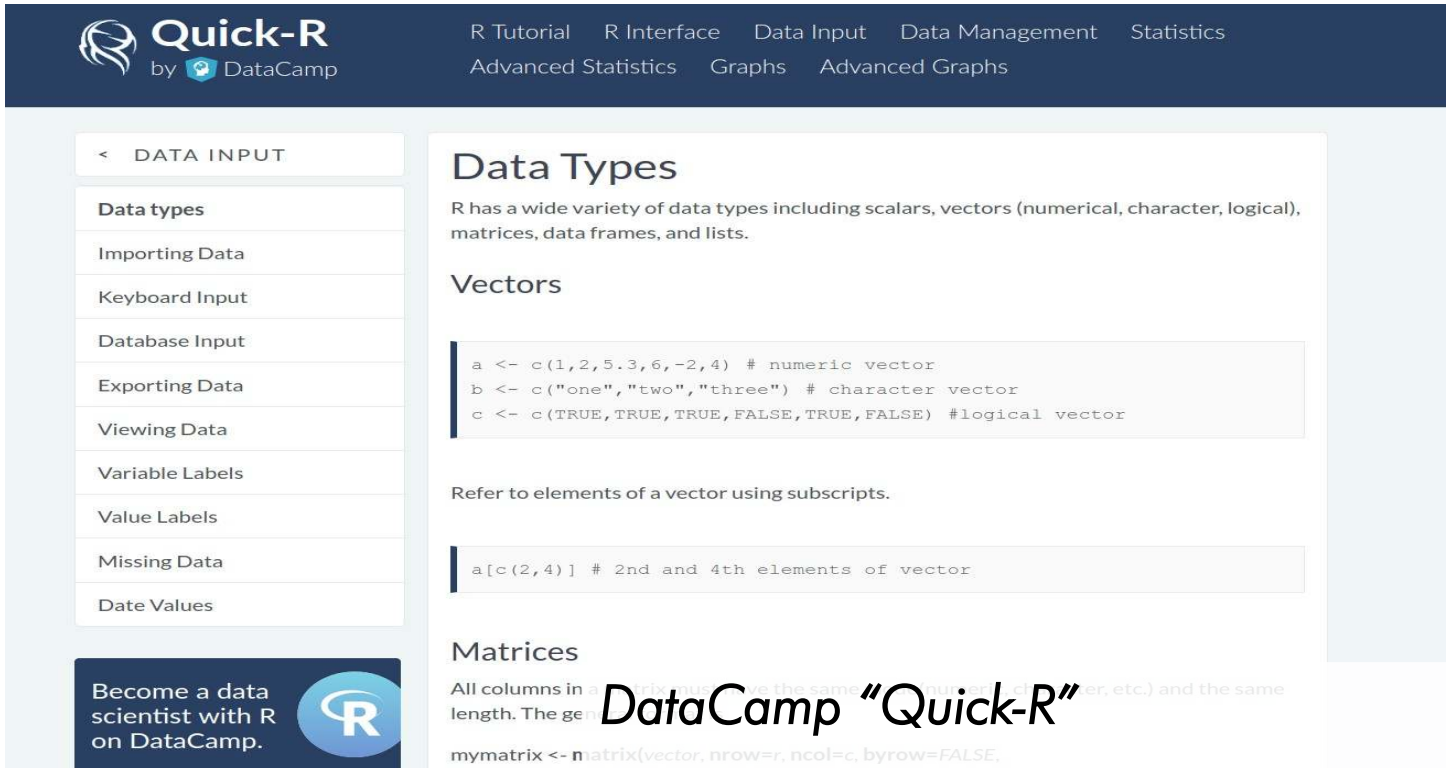
di Andrew P. Beckerman

ed. 2012

Euro 25.67

https://www.amazon.it/Getting-Started-R-introduction-biologists/dp/0199601623

A website I use a lot



DataCamp "Quick-R"

A good quick reference for R basic commands and beyond

https://www.statmethods.net/input/datatypes.html

Anatomy of a -omic project

Any sequencing project is generally composed of two types of data:

- sequencing data (the actual primary data as to speak)

- accessory data (often improperly called metadata)

Often the primary data can be composed of sequences AND something else (like MS spectra for the metaproteomic or similar)

Anatomy of a -omic project: sequencing data

The sequencing data are usually provided in fastq format, mostly compressed:

sample1_001_R1.fastq.gz

sample1_001_R2.fastq.gz

sample2_001_R1.fastq.gz

sample2_001_R2.fastq.gz

...

Anatomy of a -omic project: sequencing data

A lot of information is packed in these names, and is usually used for the processing of the sequences

*sample1_001_**R1**.fastq.gz – the forward read of sample1*

sample1_001_R2.fastq.gz

sample2_001_R1.fastq.gz

sample2_001_R2.fastq.gz

…

Anatomy of a -omic project: sequencing data

A lot of information is packed in these names, and is usually used for the processing of the sequences

*sample1_001_**R1**.fastq.gz – the forward read of sample1*

*sample1_001_**R2**.fastq.gz – the reverse read of sample 1*

sample2_001_R1.fastq.gz

sample2_001_R2.fastq.gz

...

Anatomy of a -omic project: sequencing data

A lot of information is packed in these names, and is usually used for the processing of the sequences

*sample1_**001**_R1.fastq.gz – Sequencing chip lanes used!*

sample1_001_R2.fastq.gz

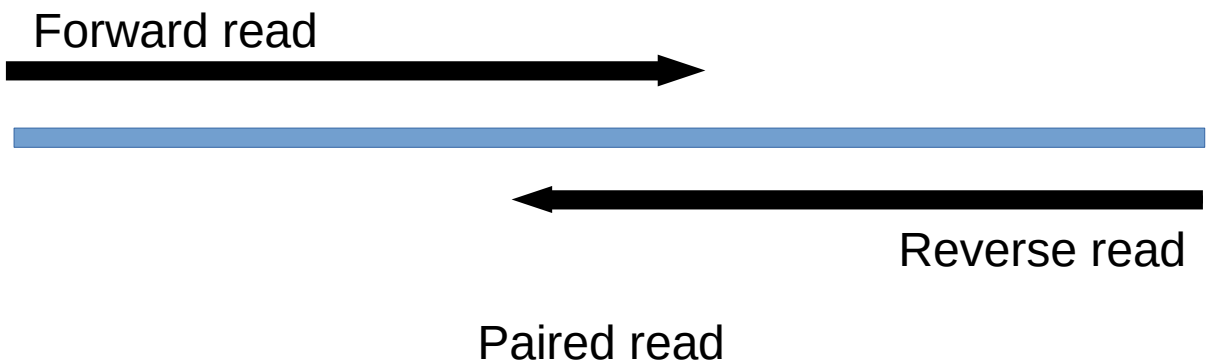
sample2_001_R1.fastq.gz

sample2_001_R2.fastq.gz

...

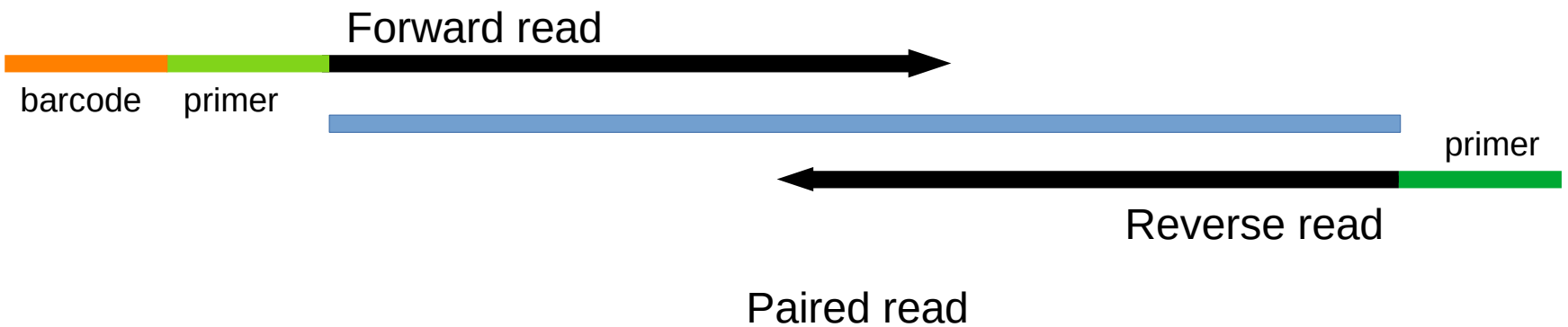
Anatomy of a -omic project: sequencing data

Most sequencing today is Paired-end, that is both the strand of the a DNA fragment are sequenced. The PE reads can be overlapping (most of the times) or not overlapping (mostly in eukaryotic transcriptomic projects)



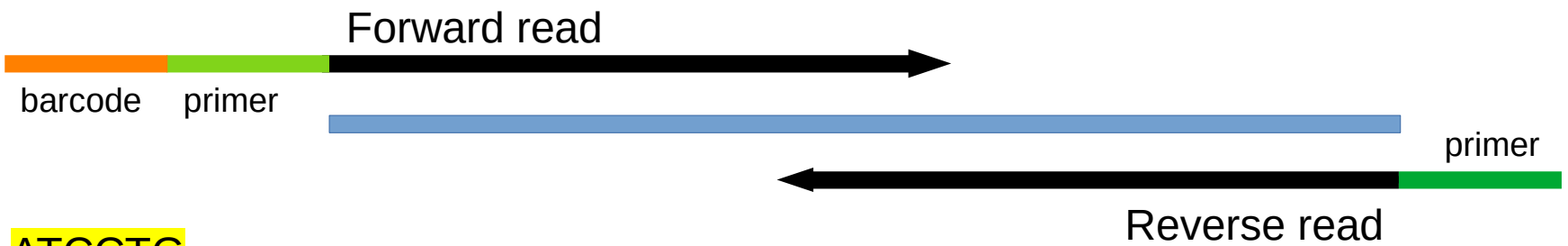
Anatomy of a -omic project: sequencing data

Most sequencing today is Paired-end, that is both the strand of the a DNA fragment are sequenced. The PE reads can be overlapping (most of the times) or not overlapping (mostly in eukaryotic transcriptomic projects)



Anatomy of a -omic project: sequencing data

Most sequencing today is Paired-end, that is both the strand of the a DNA fragment are sequenced. The PE reads can be overlapping (most of the times) or not overlapping (mostly in eukaryotic transcriptomic projects)



ATCCTG
ATCCTG
ATGCGC
ATGCTG

Accessory data

Accessory data usually include metadata (data regarding the collection and processing of all the samples) as well as data regarding the conditions and environmental context of the collected samples.

Things might include coordinates, sample type, time and date of collection, temperature, pH, salinity, soil type, tissue type, etc...

Accessory data

Accessory data usually include metadata (data regarding the collection and processing of all the samples) as well as data regarding the conditions and environmental context of the collected samples.

Things might include coordinates, sample type, time and date of collection, temperature, pH, salinity, soil type, tissue type, etc...

What makes sense depends on the study and should be assessed carefully ***AT THE SAMPLING DESIGN*** *stage*

Accessory data

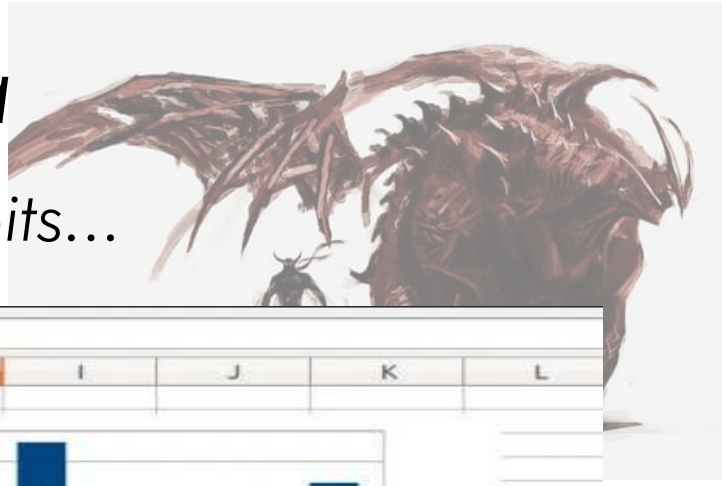
Accessory data usually include metadata (data regarding the collection and processing of all the samples) as well as data regarding the conditions and environmental context of the collected samples:

code	station	site_name	xpeditio	lat	long	trench	lat_proj	province	alt	temp
CY	CY170214	Rio Cayuco	CR17	10.287497	-84.955524	140.90712169	.121202033	Forearc	184	72
EP	EP170215	Espabel	CR17	9.901885	-85.454327	74.291955579	.287061523	Outer Forearc	NA	26.4
ES1	ES170215_1	Estrada	CR17	9.899005	-85.453514	74.082237819	.285917438	Outer Forearc	122	27.9
ES2	ES170215_2	Estrada (river water control)	CR17	9.899005	-85.453514	74.082237819	.285917438	Outer Forearc	122	26.6
ET1	ET170220_1	Eco Thermales	CR17	10.484006	-84.675853	173.292858	9.049541842	Arc	368	40
ET2	ET170220_2	Eco Thermales	CR17	10.484006	-84.675853	173.292858	9.049541842	Arc	368	40.1
FA1	FA170219_1	Finca Ande	CR17	10.336843	-85.069499	139.599114	9.181375807	Forearc	109	55.2
FA2	FA170219_2	Finca Ande	CR17	10.336843	-85.069499	139.599114	9.181375807	Forearc	109	50.8
HN	HN170219	Hornillas	CR17	10.712822	-85.177404	167.413396	9.327031411	Arc	765	87.9
IZ	IZ170223	Irazu Volcano	CR17	9.978333	-83.85222	160.513339	8.649715947	Arc	3325	NA
MT	MT170219	Termales Salitral Mouse Trap	CR17	10.595774	-85.238451	152.554048	9.333031789	Forearc	166	59.1
PA	PA170214	Pueblo Antiguo	CR17	10.283109	-84.929298	141.888606	9.108688042	Forearc	262	45
PB	PB170224	Poas Volcano background soil	CR17	10.196777	-84.229892	164.522409	8.834952797	Arc	2335	NA
PF	PF170222	Pompilo's finca	CR17	10.518466	-84.11518	202.082141	8.845586826	Backarc	53	28.7

Accessory data

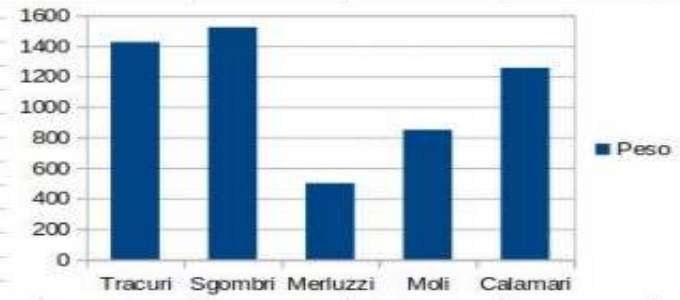
Absolute evil *lays in your spreadsheet habits…*

Accessory data



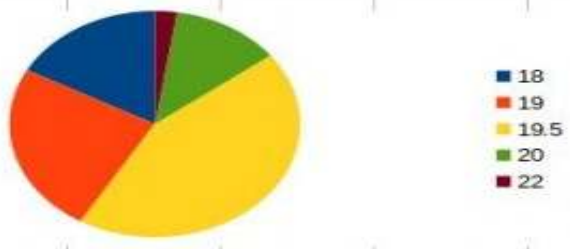
Absolute evil *lays in your spreadsheet habits…*

	A	B	C	D	E	F
1	**Cala n. 22**					
2	Data	22/10/19				
3	Inizio	LAT	42° 14,31			
4		LON	13° 15.12			
5	Fine	LAT	42° 15,21			
6		LON	13° 15,6			
7	Ora inizio	16:21:00				
8	Ora fine	17:21:00				
9						
10	Peso coffe	22,1	17,5	18,12	**57,72**	
11						
12			**Campione**			
13	**Specie**	Tracuri	Sgombri	Merluzzi	Moli	Calamari
14	**Peso**	1425	1521	500	850	1256
15	**Numero**	41	34	8	13	67



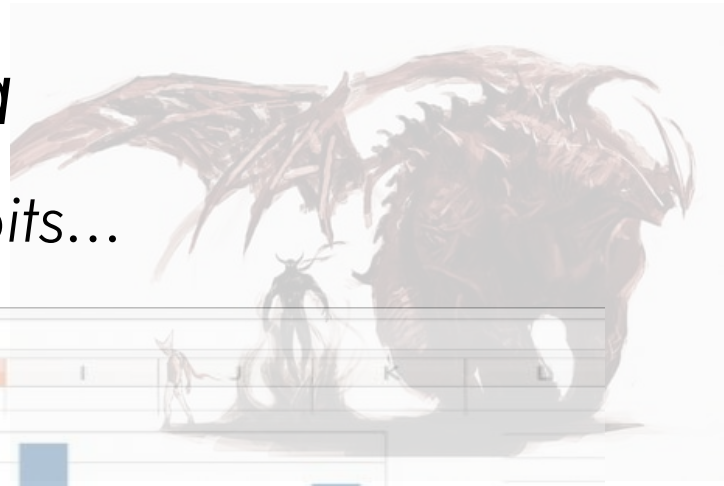
TRACURI

TAGLIA	NUMERO
18	7
19	10
19.5	18
20	5
22	1



Accessory data

Absolute evil *lays in your spreadsheet habits…*

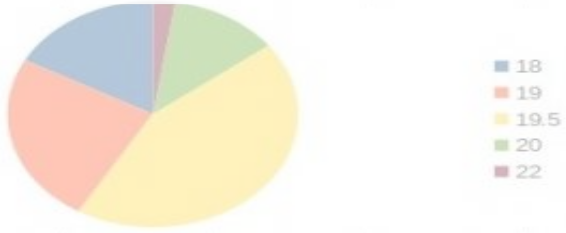


	A	B	C
1	Cala n. 22		
2	Data	22/10/19	
3	Inizio	LAT	42° 14,31
4		LON	13° 15.12
5	Fine	LAT	42° 15,21
6		LON	13° 15,6
7	Ora inizio	16:21:00	
8	Ora fine	17:21:00	

MALE ASSOLUTO!!!

TRACURI

TAGLIA	NUMERO
18	7
19	10
19.5	18
20	5
22	1



Tidy Data

***Tidy data refers** to how computer can interact with data*

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

Table 1: Typical presentation dataset.

Wickham, 2013 Journal of Statistical Software, https://bit.ly/383H8rY

Tidy Data

***Tidy data refers** to how computer can interact with data*

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

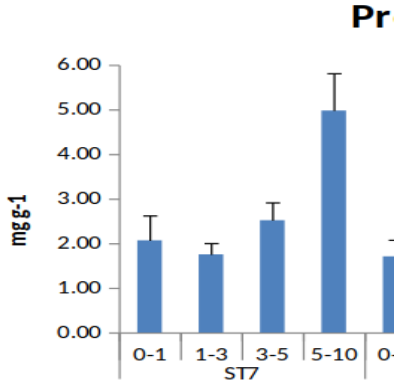
Table 1: Typical presentation dataset.

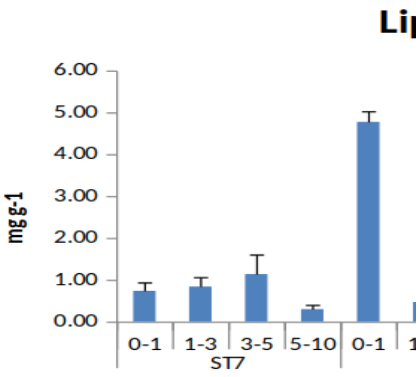
person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Wickham, 2013 Journal of Statistical Software, https://bit.ly/383H8rY

How colleagues send me data

Stazione	Layer	PRT (mg/g)	std	CHO (mg/g)	std	LIP (mg/g)	std	BPC (mg/g)	std	PRT:CHO
ST7	0-1	2.07	0.56	0.37	0.10	0.74	0.20	1.71	0.46	5.65
	1-3	1.75	0.25	0.49	0.07	0.84	0.22	1.68	0.32	3.61
	3-5	2.52	0.40	0.33	0.04	1.14	0.46	2.22	0.56	7.59
	5-10	4.98	0.83	0.27	0.08	0.30	0.10	2.77	0.51	18.20
ST3	0-1	1.71	0.37	0.27	0.07	4.78	0.25	4.53	0.40	6.44
	1-3	2.41	0.39	0.25	0.03	0.47	0.16	1.64	0.32	9.56
	3-5	3.03	1.15	0.19	0.04	0.15	0.07	1.67	0.63	16.27
	5-10	3.96	1.00	0.12	0.01	0.38	0.07	2.27	0.54	34.46
ST4	0-1	0.16	0.02	0.69	0.21	0.20	0.01	0.51	0.10	0.23
	1-3	1.36	0.19	0.12	0.05	0.28	0.08	0.92	0.17	11.68
	3-5	1.35	0.24	0.29	0.01	0.32	0.11	1.02	0.20	4.74
	5-10	0.91	0.50	0.13	0.06	0.12	0.04	0.59	0.30	6.79
		59%		**9%**		**32%**				
		51%		12%		37%				
		56%		6%		38%				
		88%		4%		8%				
		18%		**2%**		**79%**				
		72%		6%		22%				
		89%		4%		7%				
		86%		2%		12%				
		15%		**55%**		**30%**				
		72%		5%		23%				
		65%		11%		24%				
		76%		9%		15%				





OM | Proteine | Carboidrati | Lipidi | Clorofilla | lipidi_BLK | Proteine BLK | Carboidrati BLK

How I need the data to be

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	station	layer	chla	feo	cpe	idp	prt	cho	lip	bpc	prt_cho	tpn	pbm	eub	arc	bar	
2	ST7	0-1	11.43	218.90	230.33	4.98	2.07	0.37	0.74	1.71	6.65	2.73	5.47	151519820	62386732	0.42	
3	ST7	0-1	13.13	230.69	240.43	5.94	2.62	0.47	0.94	1.83	11.19	2.97	5.94	175992858	74032050	0.41	
4	ST7	0-1	9.74	207.11	220.23	4.03	1.51	0.27	0.54	1.59	2.12	2.50	5.00	127046781	50741414	0.43	
5	ST7	1-3	4.44	114.28	118.73	3.80	1.75	0.49	0.84	1.68	3.64	NA	NA	NA	NA	NA	
6	ST7	1-3	5.19	146.09	151.28	4.21	2.00	0.56	1.06	1.76	3.94	NA	NA	NA	NA	NA	
7	ST7	1-3	3.70	82.47	86.17	3.39	1.50	0.41	0.62	1.61	3.35	NA	NA	NA	NA	NA	
8	ST7	3-5	3.74	77.58	81.31	4.61	2.52	0.33	1.14	2.22	7.68	0.40	0.80	22934613	12801425	0.28	
9	ST7	3-5	3.89	86.10	90.00	4.90	2.92	0.37	1.60	2.40	9.27	0.46	0.92	23747831	14400604	0.25	
10	ST7	3-5	3.58	69.05	72.63	4.33	2.12	0.29	0.67	2.04	6.10	0.34	0.68	22121395	11202246	0.33	
11	ST7	5-10	3.23	62.47	65.69	4.92	4.98	0.27	0.30	2.77	19.07	0.14	0.39	10877676	1977100	0.69	
12	ST7	5-10	3.91	76.99	80.90	5.03	5.81	0.35	0.40	3.09	22.92	0.16	0.47	12160277	2210475	0.69	
13	ST7	5-10	2.55	47.94	50.48	4.82	4.14	0.20	0.20	2.46	15.22	0.11	0.32	9595074	1743726	0.69	
14	ST4	0-1	2.91	58.64	61.54	5.02	0.16	0.27	4.78	0.51	0.62	3.00	6.00	178464006	58364649	0.51	
15	ST4	0-1	4.00	72.52	74.34	7.84	0.17	0.33	5.03	0.60	0.77	3.24	6.47	212767897	67094891	0.52	
16	ST4	0-1	1.81	44.75	48.75	2.20	0.14	0.20	4.52	0.41	0.47	2.77	5.53	144160115	49634407	0.49	
17	ST4	1-3	8.94	121.11	130.06	6.90	1.36	0.25	0.47	0.92	5.47	NA	NA	NA	NA	NA	
18	ST4	1-3	9.54	135.93	145.47	7.26	1.55	0.28	0.63	1.01	6.45	NA	NA	NA	NA	NA	
19	ST4	1-3	8.35	106.30	114.64	6.54	1.17	0.22	0.31	0.83	4.49	NA	NA	NA	NA	NA	
20	ST4	3-5	6.29	97.66	103.96	6.07	1.35	0.19	0.15	1.02	7.51	1.65	3.30	94323186	24209857	0.59	
21	ST4	3-5	6.66	107.37	114.03	6.31	1.59	0.22	0.22	1.17	9.37	2.19	4.38	95457206	34416038	0.47	
22	ST4	3-5	5.93	87.95	93.88	5.83	1.12	0.15	0.08	0.86	5.66	1.11	2.22	93189166	14003677	0.74	
23	ST4	5-10	1.56	26.86	28.42	5.51	0.91	0.12	0.38	0.59	7.76	0.27	0.54	19117139	6335276	0.50	
24	ST4	5-10	1.81	29.29	30.95	6.31	1.41	0.13	0.44	0.72	9.79	0.28	0.57	21346712	6656073	0.52	
25	ST4	5-10	1.32	24.43	25.89	4.71	0.41	0.10	0.31	0.46	5.73	0.26	0.52	16887567	6014479	0.47	
26	ST3	0-1	9.18	172.05	181.23	5.05	1.71	0.69	0.20	4.53	2.61	4.37	8.73	240939533	127464027	0.31	
27	ST3	0-1	10.61	186.87	197.48	5.39	2.08	0.90	0.21	4.77	3.30	4.79	9.58	288166367	156163469	0.30	

dataset | Dataset medie | Data set Repliche | data set per statistica | OM | TPN | CHLA | Sheet8

How I **WE** need the data to be

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	station	layer	chla	feo	cpe	idp	prt	cho	lip	bpc	prt_cho	tpn	pbm	eub	arc	bar	
2	ST7	0-1	11.43	218.90	230.33	4.98	2.07	0.37	0.74	1.71	6.65	2.73	5.47	151519820	62386732	0.42	
3	ST7	0-1	13.13	230.69	240.43	5.94	2.62	0.47	0.94	1.83	11.19	2.97	5.94	175992858	74032050	0.41	
4	ST7	0-1	9.74	207.11	220.23	4.03	1.51	0.27	0.54	1.59	2.12	2.50	5.00	127046781	50741414	0.43	
5	ST7	1-3	4.44	114.28	118.73	3.80	1.75	0.49	0.84	1.68	3.64	NA	NA	NA	NA	NA	
6	ST7	1-3	5.19	146.09	151.28	4.21	2.00	0.56	1.06	1.76	3.94	NA	NA	NA	NA	NA	
7	ST7	1-3	3.70	82.47	86.17	3.39	1.50	0.41	0.62	1.61	3.35	NA	NA	NA	NA	NA	
8	ST7	3-5	3.74	77.58	81.31	4.61	2.52	0.33	1.14	2.22	7.68	0.40	0.80	22934613	12801425	0.28	
9	ST7	3-5	3.89	86.10	90.00	4.90	2.92	0.37	1.60	2.40	9.27	0.46	0.92	23747831	14400604	0.25	
10	ST7	3-5	3.58	69.05	72.63	4.33	2.12	0.29	0.67	2.04	6.10	0.34	0.68	22121395	11202246	0.33	
11	ST7	5-10	3.23	62.47	65.69	4.92	4.98	0.27	0.30	2.77	19.07	0.14	0.39	10877676	1977100	0.69	
12	ST7	5-10	3.91	76.99	80.90	5.03	5.81	0.35	0.40	3.09	22.92	0.16	0.47	12160277	2210475	0.69	
13	ST7	5-10	2.55	47.94	50.48	4.82	4.14	0.20	0.20	2.46	15.22	0.11	0.32	9595074	1743726	0.69	
14	ST4	0-1	2.91	58.64	61.54	5.02	0.16	0.27	4.78	0.51	0.62	3.00	6.00	178464006	58364649	0.51	
15	ST4	0-1	4.00	72.52	74.34	7.84	0.17	0.33	5.03	0.60	0.77	3.24	6.47	212767897	67094891	0.52	
16	ST4	0-1	1.81	44.75	48.75	2.20	0.14	0.20	4.52	0.41	0.47	2.77	5.53	144160115	49634407	0.49	
17	ST4	1-3	8.94	121.11	130.06	6.90	1.36	0.25	0.47	0.92	5.47	NA	NA	NA	NA	NA	
18	ST4	1-3	9.54	135.93	145.47	7.26	1.55	0.28	0.63	1.01	6.45	NA	NA	NA	NA	NA	
19	ST4	1-3	8.35	106.30	114.64	6.54	1.17	0.22	0.31	0.83	4.49	NA	NA	NA	NA	NA	
20	ST4	3-5	6.29	97.66	103.96	6.07	1.35	0.19	0.15	1.02	7.51	1.65	3.30	94323186	24209857	0.59	
21	ST4	3-5	6.66	107.37	114.03	6.31	1.59	0.22	0.22	1.17	9.37	2.19	4.38	95457206	34416038	0.47	
22	ST4	3-5	5.93	87.95	93.88	5.83	1.12	0.15	0.08	0.86	5.66	1.11	2.22	93189166	14003677	0.74	
23	ST4	5-10	1.56	26.86	28.42	5.51	0.91	0.12	0.38	0.59	7.76	0.27	0.54	19117139	6335276	0.50	
24	ST4	5-10	1.81	29.29	30.95	6.31	1.41	0.13	0.44	0.72	9.79	0.28	0.57	21346712	6656073	0.52	
25	ST4	5-10	1.32	24.43	25.89	4.71	0.41	0.10	0.31	0.46	5.73	0.26	0.52	16887567	6014479	0.47	
26	ST3	0-1	9.18	172.05	181.23	5.05	1.71	0.69	0.20	4.53	2.61	4.37	8.73	240939533	127464027	0.31	
27	ST3	0-1	10.61	186.87	197.48	5.39	2.08	0.90	0.21	4.77	3.30	4.79	9.58	288166367	156163469	0.30	

dataset | Dataset medie | Data set Repliche | data set per statistica | OM | TPN | CHLA | Sheet8

Data wrangling: *getting data in the right format*

General rules to have data in a usable format:

- one observation per row

- one variable per column

Keep in mind the experimental design!

	Var 1	Var 2	Var ...
Obs 1			
Obs 2			
Obs 3			
...			

something worthwhile learning about

TidyVerse

A collection of packages focused on importing, manipulating, transforming and visualizing data

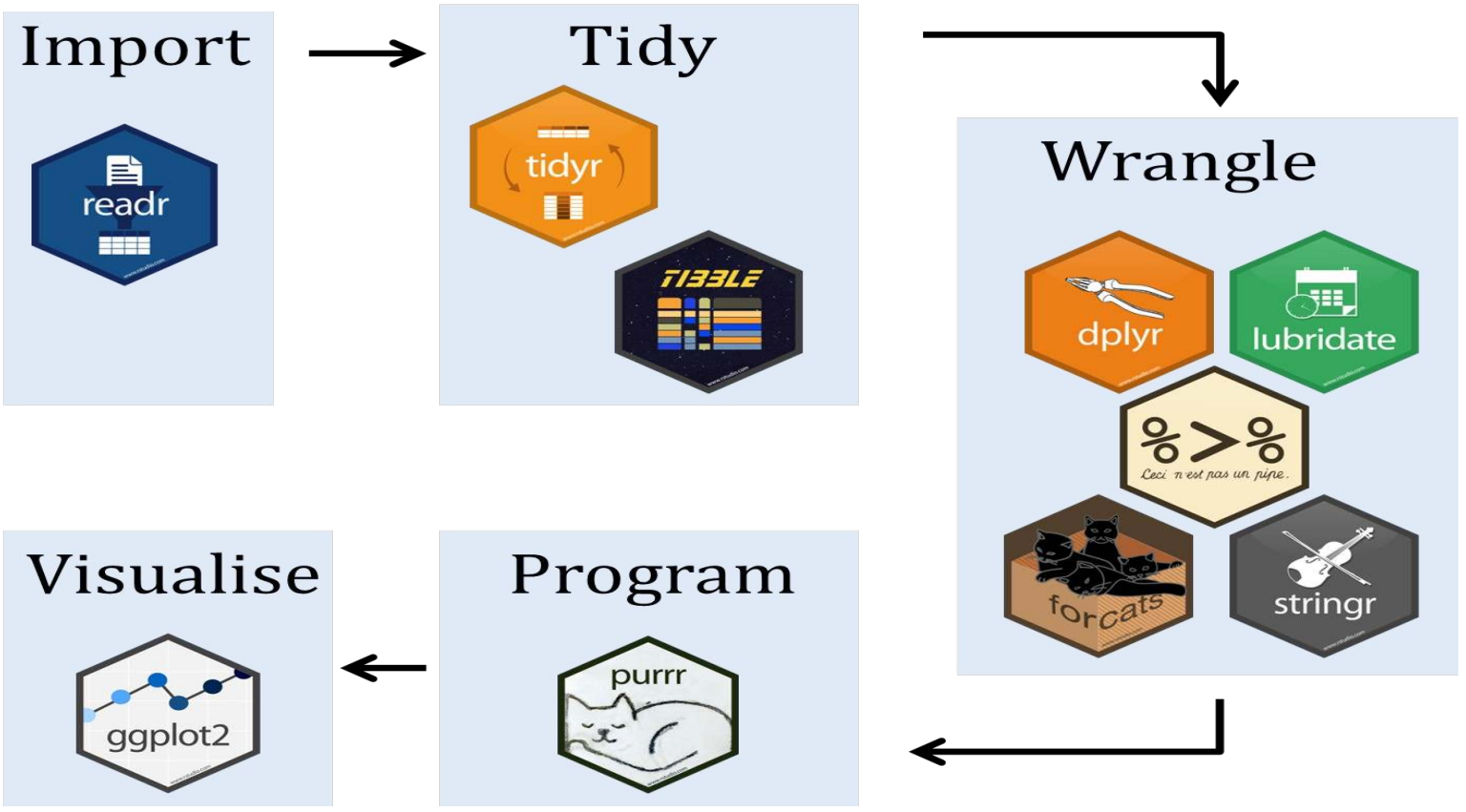
It is composed of several different packages with focus on different aspects of data handling

Powerful and versatile can help to "tidy" messy dataset and get them ready for "serious work"

Basci cheat sheet can be found at

https://rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

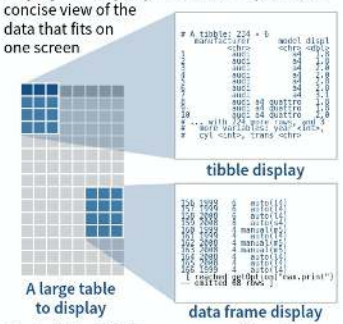
TidyVerse


Import
readr
Tidy
tidyr
tibble
Wrangle
dplyr
lubridate
%>%
Ceci n'est pas un pipe.
forcats
stringr
Program
purrr
Visualise
ggplot2

Tibbles - an enhanced data frame

The **tibble** package provides a new S3 class for storing tabular data, the tibble. Tibbles inherit the data frame class, but improve three behaviors:

- **Subsetting** - [always returns a new tibble, [[and $ always return a vector.
- **No partial matching** - You must use full column names when subsetting
- **Display** - When you print a tibble, R provides a concise view of the data that fits on one screen



- Control the default appearance with options:

  options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf)

- View full data set with **View()** or **glimpse()**
- Revert to data frame with **as.data.frame()**

CONSTRUCT A TIBBLE IN TWO WAYS

tibble(…)
Construct by columns.
tibble(x = 1:3, y = c("a", "b", "c"))

tribble(…)
Construct by rows.

```
tribble(~x,  ~y,
        1,   "a",
        2,   "b",
        3,   "c")
```

Both make this tibble

```
A tibble: 3 × 2
      x     y
  <int> <chr>
1     1     a
2     2     b
3     3     c
```

as_tibble(x, …) Convert data frame to tibble.

enframe(x, name = "name", value = "value")
Convert named vector to a tibble

is_tibble(x) Test whether x is a tibble.

Tidy Data with tidyr

Tidy data is a way to organize tabular data. It provides a consistent data structure across packages.

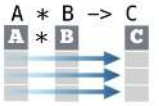
A table is tidy if:

Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**

Tidy data:

Makes variables easy to access as vectors



Preserves cases during vectorized operations

Reshape Data - change the layout of values in a table

Use **pivot_longer()** and **pivot_wider()** to reorganize the values of a table into a new layout.

pivot_longer(data, cols, names_to = "name", names_prefix = NULL, names_sep = NULL, names_pattern = NULL, names_ptypes = list(), names_transform = list(), names_repair = "check_unique", values_to = "value", values_drop_na = FALSE, values_ptypes = list(), values_transform = list(), …)

pivot_longer() pivots **cols** columns, moving column names into a **names_to** column, and column values into a **values_to** column.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

pivot_longer(table4a, cols = 2:3, names_to = "year", values_to = "cases")

pivot_wider(data, id_cols = NULL, names_from = name, names_prefix = "", names_sep = "_", names_glue = NULL, names_sort = FALSE, names_repair = "check_unique", values_from = value, values_fill = NULL, values_fn = NULL, …)

pivot_wider() pivots a **names_from** and a **values_from** column into a rectangular field of cells.

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	NA	NA

pivot_wider(table2, names_from = type, values_from = count)

Handle Missing Values

drop_na(data, …)
Drop rows containing NA's in … columns.

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
D	3

drop_na(x, x2)

fill(data, …, .direction = c("down", "up"))
Fill in NA's in … columns with most recent non-NA values.

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
B	1
C	1
D	3
E	3

fill(x, x2)

replace_na(data, replace = list(), …)
Replace NA's by column.

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
B	2
C	2
D	3
E	2

replace_na(x, list(x2 = 2))

Expand Tables - quickly create tables with combinations of values

complete(data, …, fill = list())
Adds to the data missing combinations of the values of the variables listed in …
complete(mtcars, cyl, gear, carb)

expand(data, …)
Create new tibble with all possible combinations of the values of the variables listed in …
expand(mtcars, cyl, gear, carb)

Split Cells

Use these functions to split or combine cells into individual, isolated values.

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", …)

Separate each cell in a column to make several columns.

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M
C	1999	212K/1T
C	2000	213K/1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172
B	2000	80K	174
C	1999	212K	1T
C	2000	213K	1T

separate(table3, rate, sep = "/", into = c("cases", "pop"))

separate_rows(data, …, sep = "[^[:alnum:].]+", convert = FALSE)

Separate each cell in a column to make several rows.

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M
C	1999	212K/1T
C	2000	213K/1T

→

country	year	rate
A	1999	0.7K
A	1999	19M
A	2000	2K
A	2000	20M
B	1999	37K
B	1999	172M
B	2000	80K
B	2000	174M
C	1999	212K
C	1999	1T
C	2000	213K
C	2000	1T

separate_rows(table3, rate, sep = "/")

unite(data, col, …, sep = "_", remove = TRUE)

Collapse cells across several columns to make a single column.

table5

country	century	year
Afghan	19	99
Afghan	20	00
Brazil	19	99
Brazil	20	00
China	19	99
China	20	00

→

country	year
Afghan	1999
Afghan	2000
Brazil	1999
Brazil	2000
China	1999
China	2000

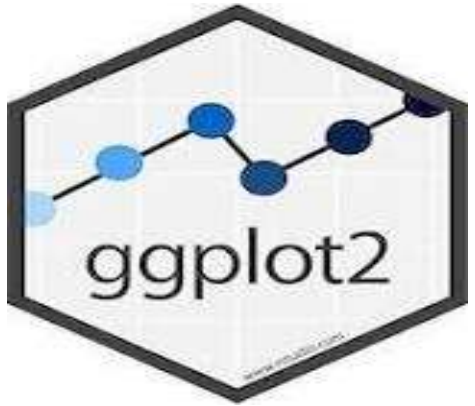
unite(table5, century, year, col = "year", sep = "")

something worthwhile learning about

ggplot2

ggplot2 is a data visualization package for R

ggplot2 is an implementation of Leland Wilkinson's Grammar of Graphics—a general scheme for data visualization which breaks up graphs into semantic components such as scales and layers



Very powerful for data visualization and plotting it has several additional packages built on it

Basic cheat sheet can be found at

https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

RETRIEVING SEQUENCES!

Public databases

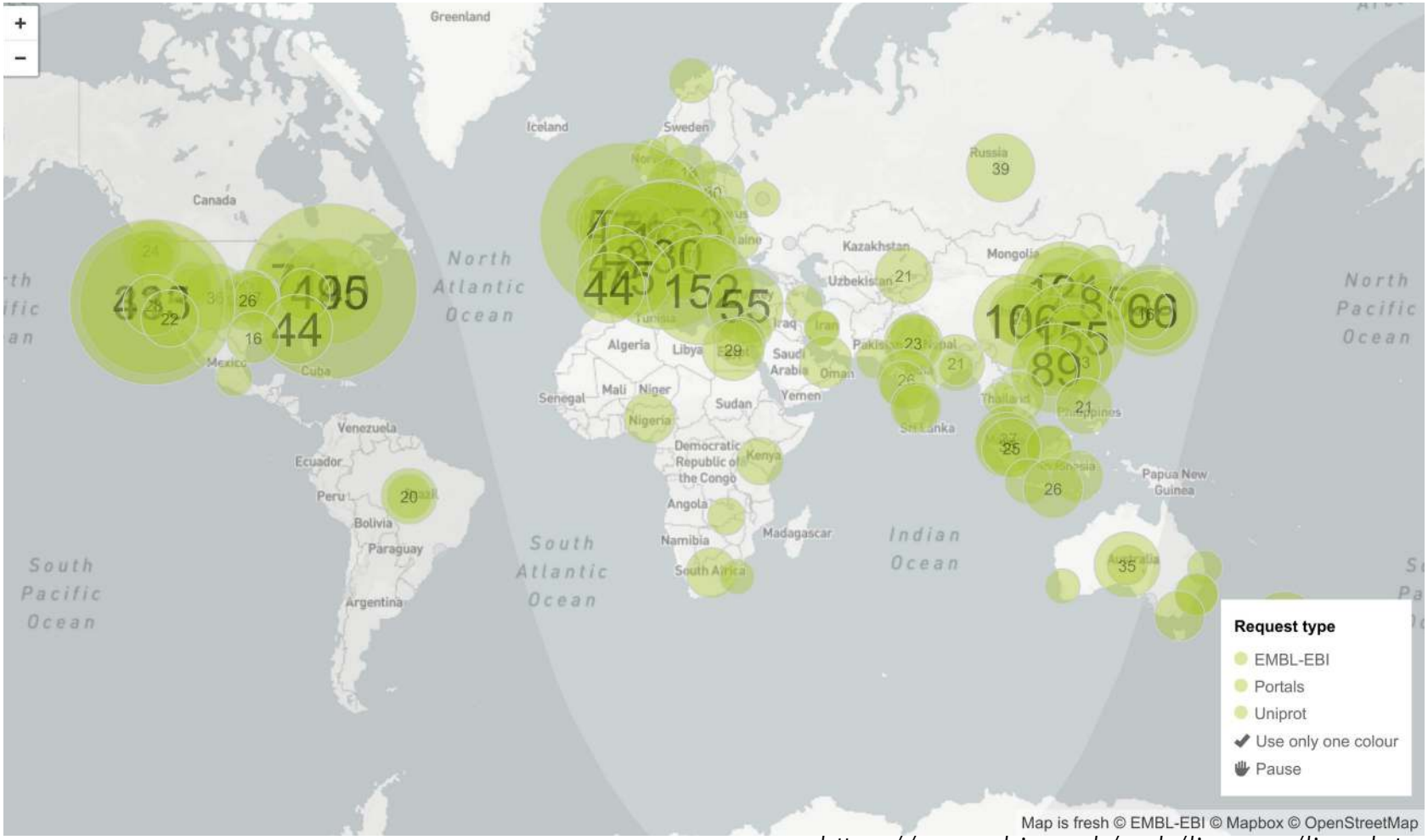
- *There are a large number of public databases containing sequences.*
- *The database to be used largely depends on the type of sequence(s) we are looking for*
- *Some of the database are synced together to have identical non-redundant information*
- *Some database are specific*
- ***Do your database homework before downloading sequences!***

International Nucleotide Sequence Database Collaboration (INSDC)

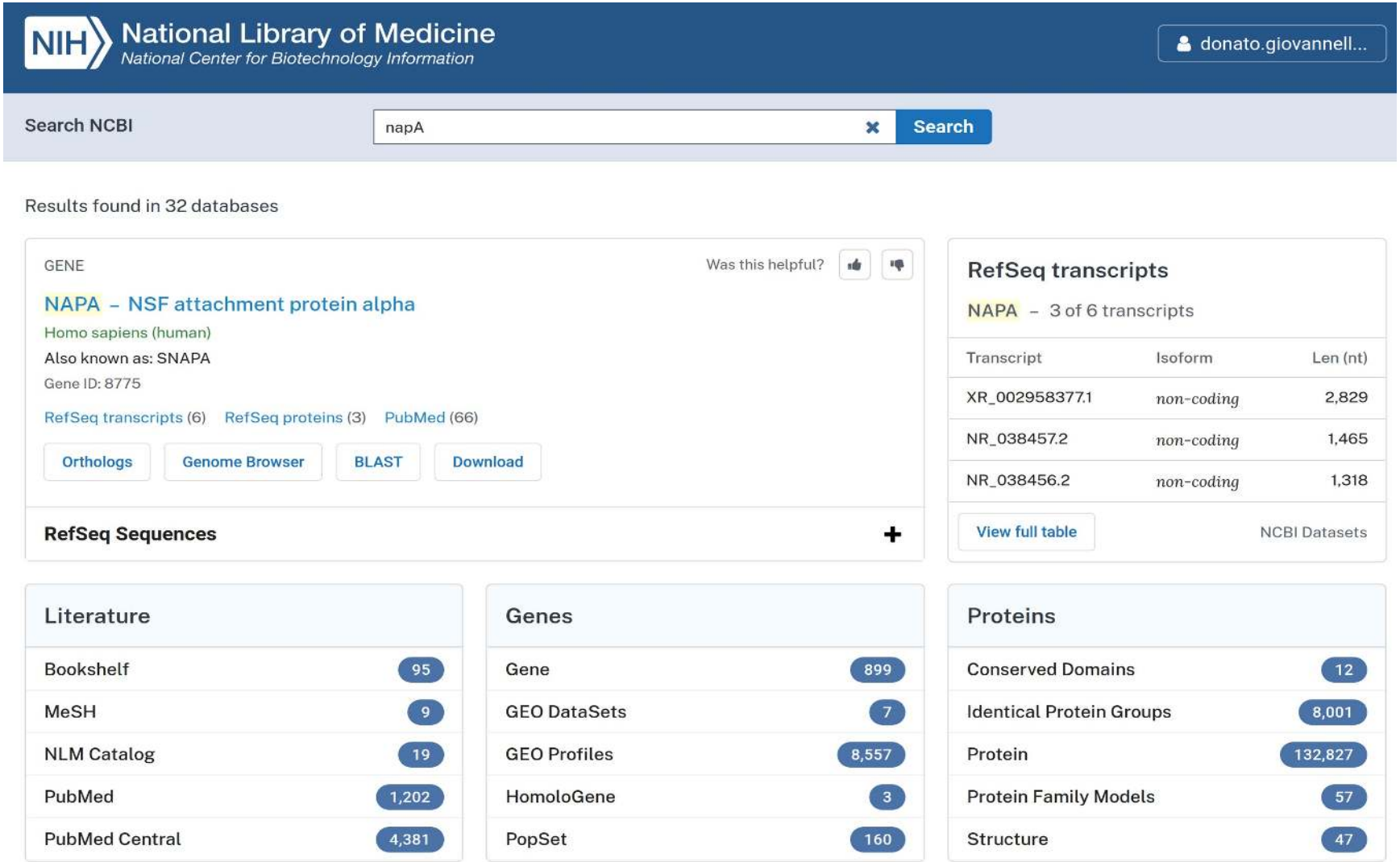
- *established early 1980s*
- *open access for all*
- *globally comprehensive*
- *spanning life science domains*
- *permanent database of record*
- *major ongoing investment*
- *mandatory submission agreement*
- *services and software (node-level)*

http://www.sectordialogues.org/documentos/noticias/adjuntos/dbf7f7_13.%20Cochrane.pdf

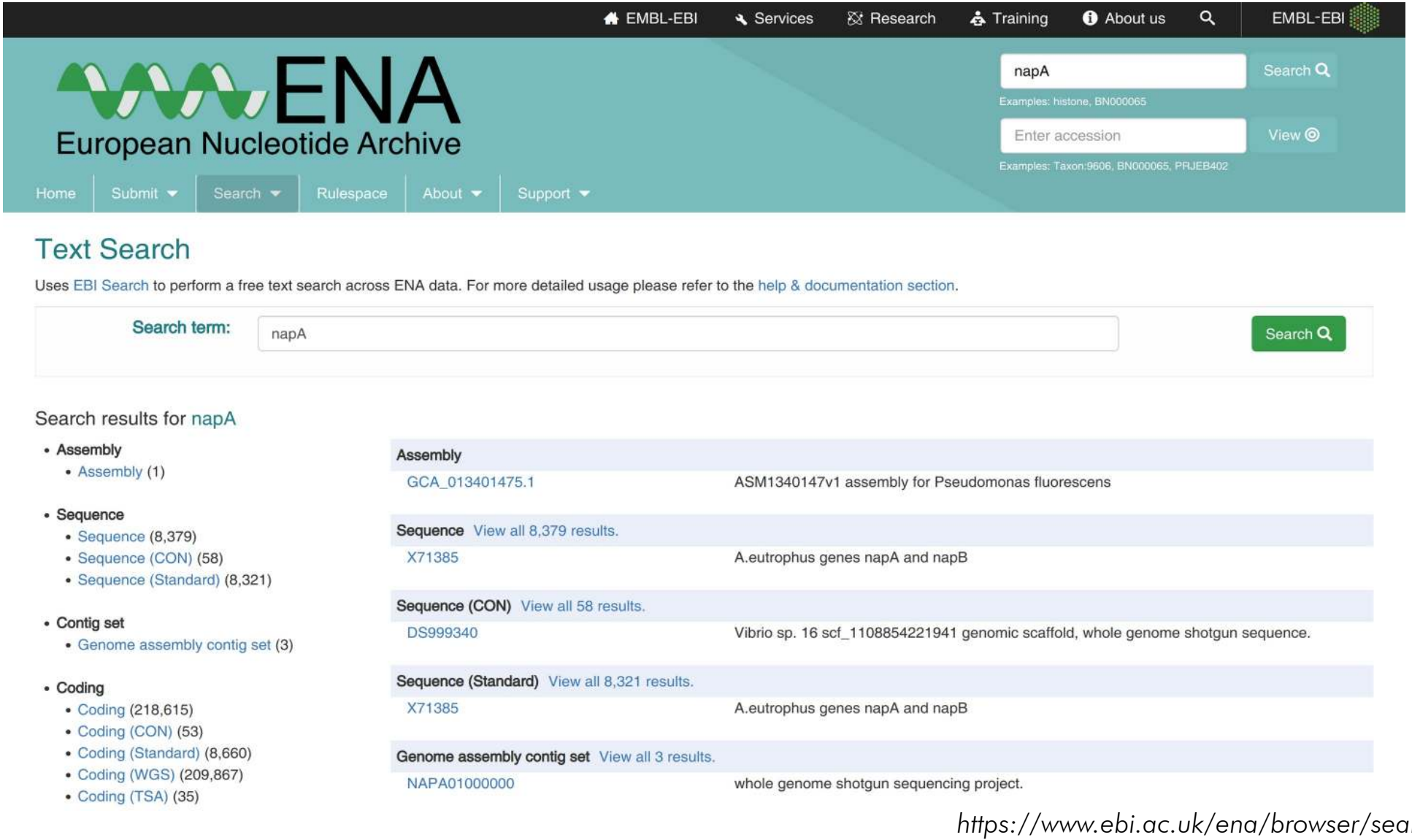
Real time requests



https://www.ebi.ac.uk/web/livemap/live-data-map.html



https://www.ncbi.nlm.nih.gov/search/



https://www.ebi.ac.uk/ena/browser/search

A (very) brief list of databases

- ***InterProt*** *– The database of annotated protein families (https://www.ebi.ac.uk/interpro/)*
- ***UniProt*** *– The database of protein sequence and functions (https://www.uniprot.org/)*
- ***KEGG*** *– The database of genes and genomes with annotated metabolic pathways (https://www.genome.jp/kegg/)*
- ***IMG/JGI*** *– The database of the Department of Energy genomic resources and projects (https://img.jgi.doe.gov/)*
- ***BioCyc*** *– The database of genomes and metabolic pathways (https://biocyc.org/)*
- ***PDB*** *– The database of protein structure (https://www.rcsb.org/)*

SEARCHING FOR METAGENOMIC PROJECTS

Searching metagenomic data

There are three major entry point to search for metagenomic projects data:

- ***Research publications*** *on specific project*
- *Using the search engine in the* ***NCBI*** *or* ***ENA*** *website*
- *Using the search engine of the* ***JGI/IMG***

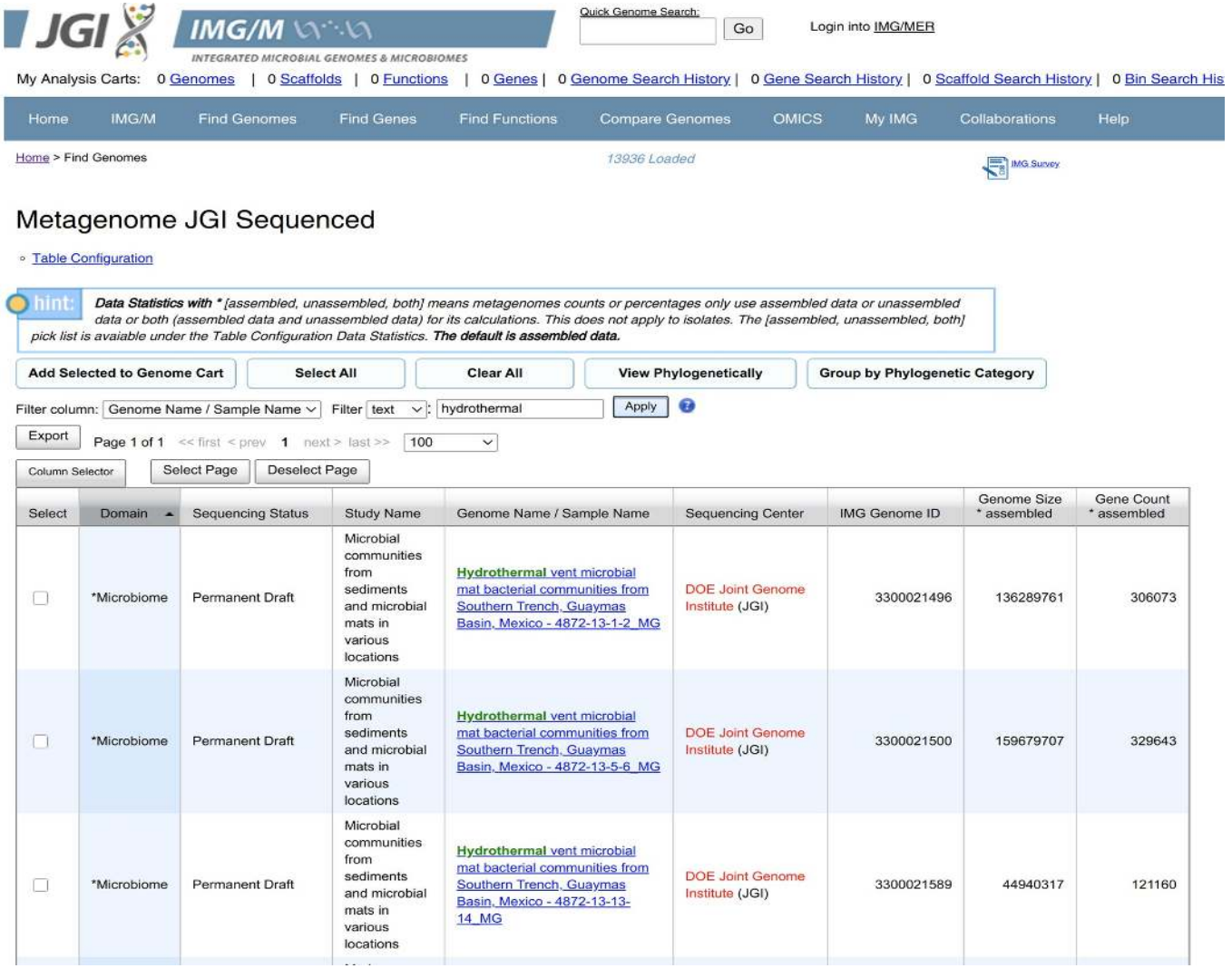
Searching metagenomic data

Research publications on specific project. *All publication that include sequencing data **MUST** report the accession number of their deposition in the paper and make the sequences publicly available*

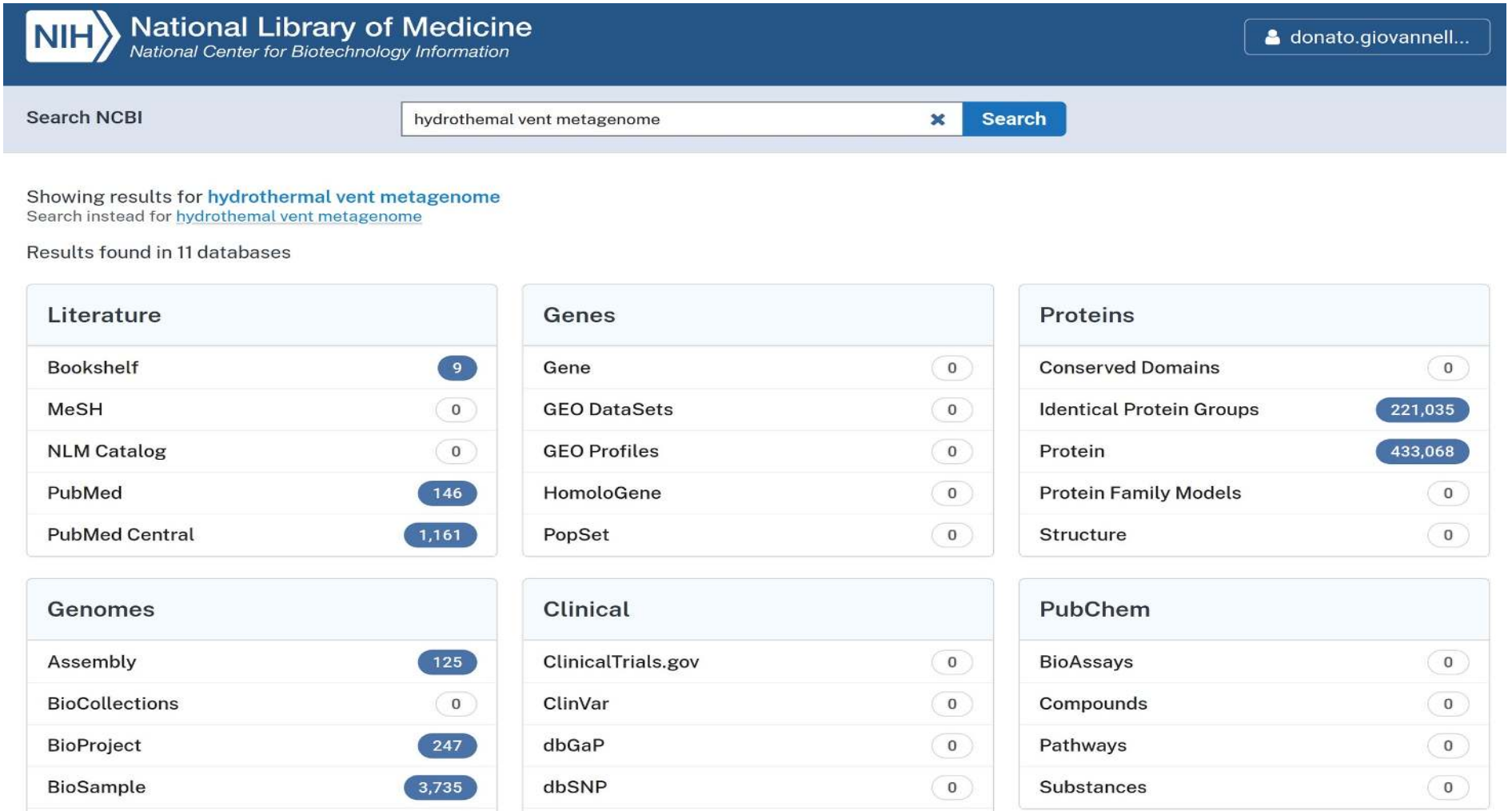
Data availability

This Targeted Locus Study project has been deposited at DDBJ/EMBL/GenBank under the accession KEBJ00000000, with project ID PRJNA579365. The version described in this paper is the first version, KEBJ01000000. Metagenomic data are in the NCBI SRA with project ID PRJNA627197. The full environmental dataset is available at https://github.com/dgiovannelli/SubductCR_16S-diversity.git and released as a permanent version (v1.0) using Zenodo under https://doi.org/10.5281/zenodo.4553845. Source data are provided with this paper.

*Using the search engine of the **JGI/IMG***



*Using the search engine in the **NCBI** or **ENA** website*



Getting the raw data: short-read archives

Data from metagenomic (and tag amplicon and any other sequences project really) can be downloaded in two basic formats:

- the raw reads in fastq format, providing access to the unprocessed data

- the processed fasta files, that may assume different forms depending on the specifics of the project

Depending on the question you are trying to answer and the workflow you are planning to use one, the other or both formats might be required.

Getting the raw data: short-read archives

Data from metagenomic (and tag amplicon and any other sequences project really) can be downloaded in two basic formats:

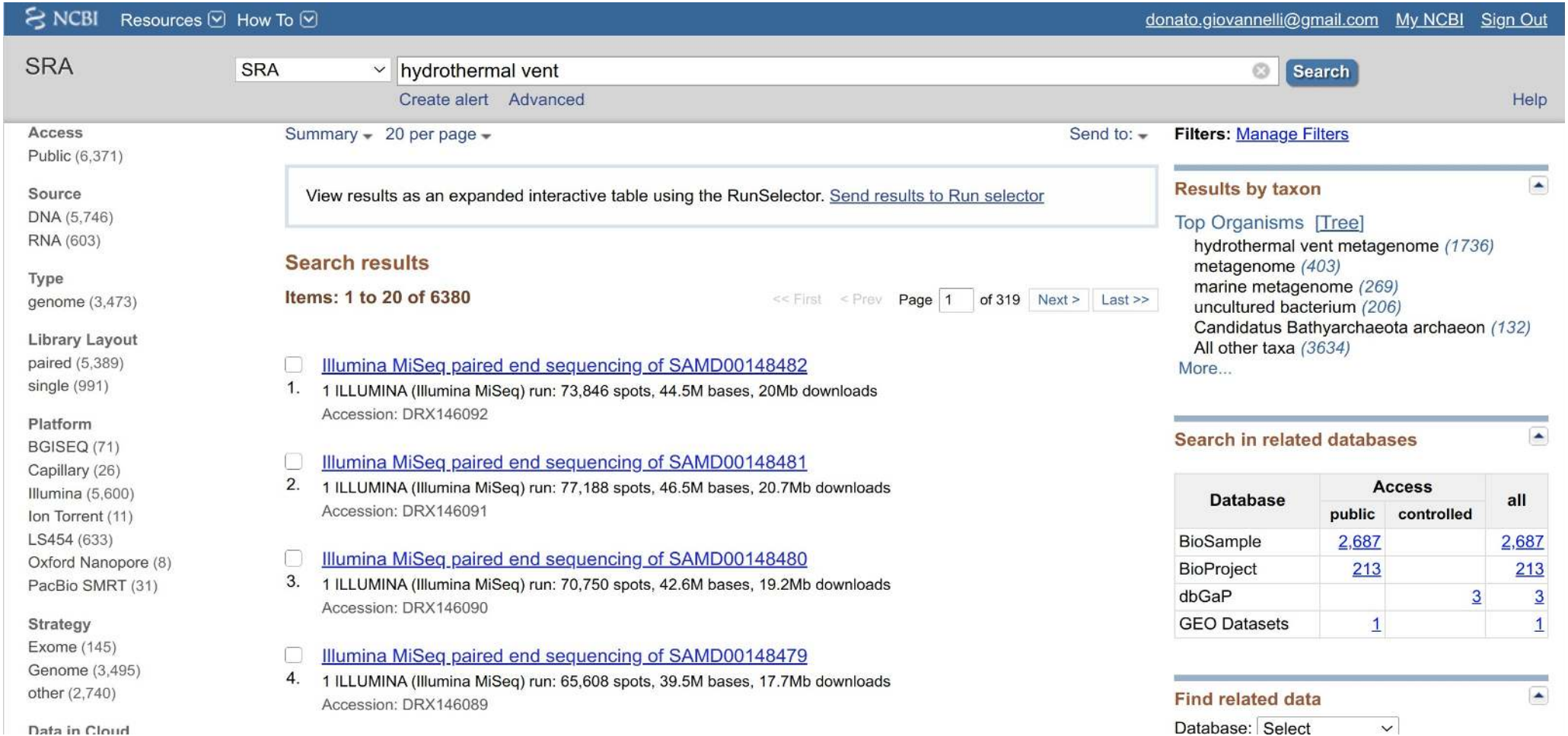
- the raw reads in fastq format, providing access to the unprocessed data

- the processed fasta files, that may assume different forms depending on the specifics of the project

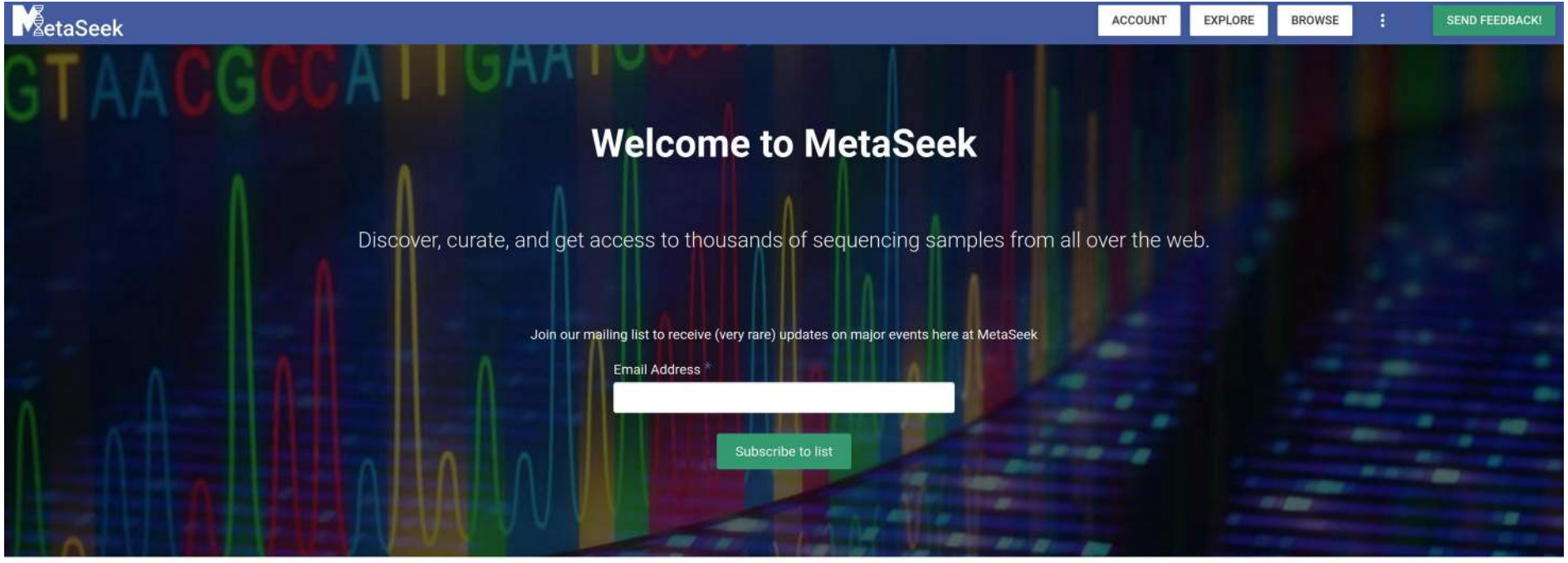
Depending on the question you are trying to answer and the workflow you are planning to use one, the other or both formats might be required.

For this course we will only use the raw reads directly, and I will show you how to process them.

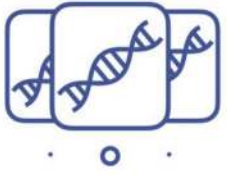
Getting the raw data: short-read archives



A useful tool to mine the SRA: metaseek



Explore



Discover

Create

https://www.metaseek.cloud/

MetaSeek

ACCOUNT | EXPLORE | BROWSE | SEND FEEDBACK!

RESET FILTERS

General Sample Info

Investigation Type

All

Environmental Package

All

Library Source

All

Study Type

All

Sequencing Info

Environmental/Contextual Info

Explore

SAVE DISCOVERY

Number of Datasets

currently showing

7521033 datasets

out of 7521033 total datasets

Estimated Total Download Size

6.0PB

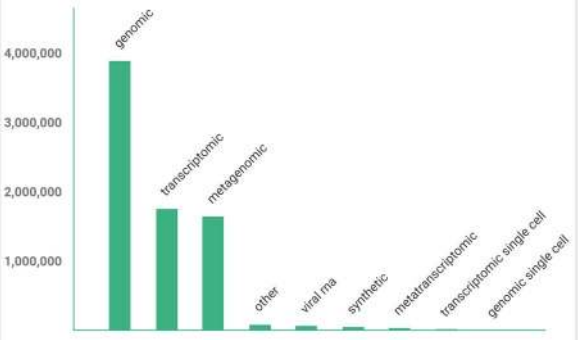
User

Create an account or log in with Google to save your discoveries.

SIGN UP/LOG IN WITH GOOGLE

General Sample Info

library_source_summary



Sequencing Info

avg_read_length_summary