

# Introduction to Bioinformatics

Alejandro Rojas

PLPA504V/CEMB590V

# About today

- Get to know each other
- Course organization
- Course outline and goals

# Course goals

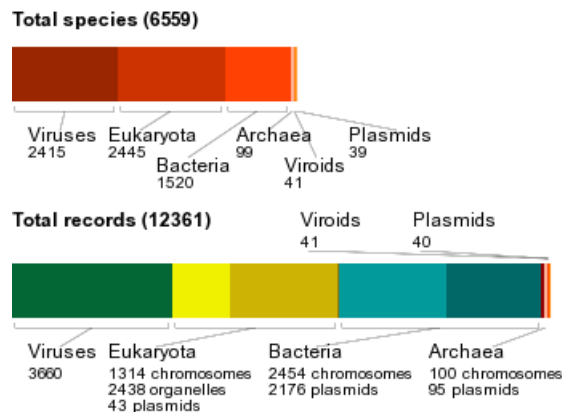
- After this class you should be able to:
  - Use some basic bioinformatics tools and choose the right parameters.
  - Handle large-scale datasets with computer programming.
  - Resolve biological problems with bioinformatics.

# Two important questions

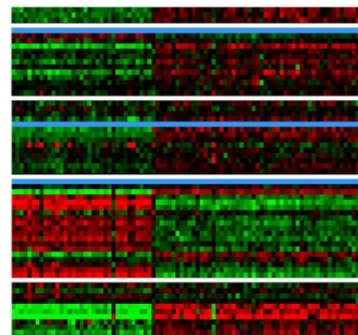
- Why is it a good thing that you are learning bioinformatics?
- What is bioinformatics anyway?
  - To answer these questions, we need to first talk about biological data.
- What kinds of biological data are available?

# Avalanche of biological data

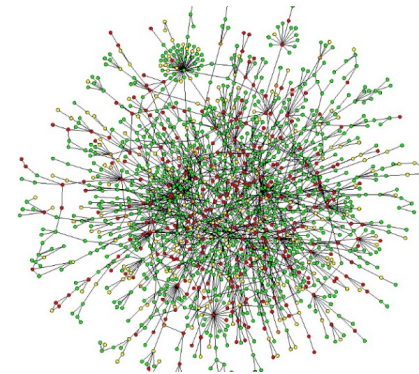
- You need to know how to play with these data with *computation and bioinformatics*.



>6000 genome projects with some records in NCBI.



469,860 expression experiments in NCBI GEO



At least one interactions for 2.5 million proteins from 630 organisms in the String database

# Bioinformatics

- Research, development, or application of
- Computational tools and approaches for
- Expanding the use of biological, medical, behavioral, health data,
- Including those to acquire, store, organize, archive, analyze, or visualize such data.

# Algorithm, Data Structure, Programming

- Two independent stress-treatments
  - Lists of differentially expressed genes. How will you find out the common genes?

## ***List 1***

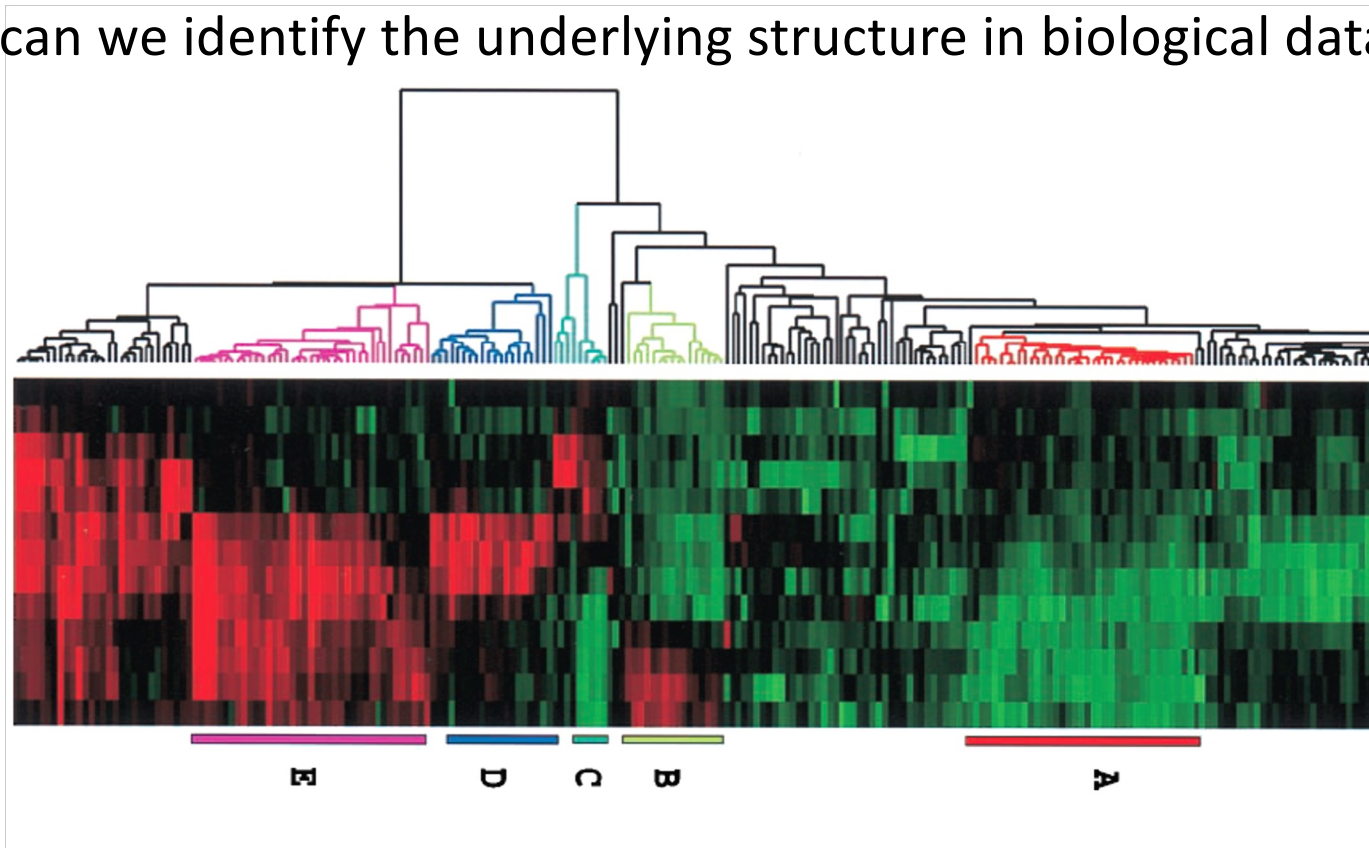
Thx  
Phh1  
Lok4  
Jij3  
Arg25  
Arg2  
Qrt3  
Kww4

## ***List 2***

Lnn1  
Thx  
Thy  
Phh1  
Kwe40  
Arg2  
Arg3  
Jij3  
Arg25

# Artificial Intelligence & Robotics

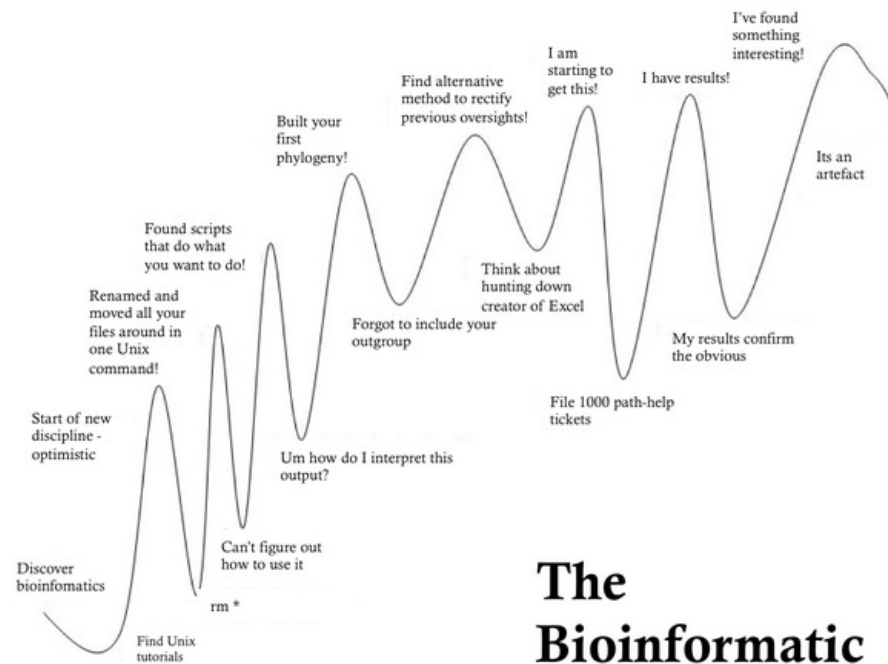
- How can we identify the underlying structure in biological data?



Eisen et al., 1998. PNAS 95:14863



# Steep learning curve



**The  
Bioinformatic  
learning curve**

# Class schedule

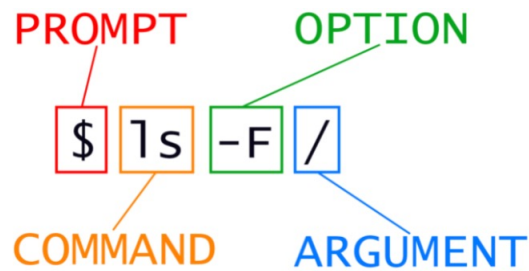
Date	Lecture Topic	Notes
Week 1	Course Intro / UNIX I: Cmdline, GitHub	
Week 2	Project Organization and UNIX I continuation	
Week 3	UNIX II: Biocluster HPCC, Running programs Tools for data processing	Homework 1 Due
Week 4	UNIX IV: Advanced UNIX and data processing	
Week 5	Working with sequence data	Homework 2 Due
Week 6	Python - Variables, running, cmdline, strings, math	
Week 7	Review Bioinformatics and Databases	Homework 3 Due
Week 8	Alignment and Bioinformatics Algorithms; BLAST, cmdline	
Week 9	Bioinformatics I - Aligning short reads, coverage	Homework 4 Due
Week 10	Bioinformatics II - Genome Assembly	
Week 11	Bioinformatics III - Protein Sequence analyses (HMMER, InterPro, SignalP)	
Week 12	Bioinformatics IV - Orthology, Phylogenetics and automation	
Week 13	Bioinformatics V - RNASeq analyses	Homework 5 Due
Week 14	Data visualization in R and python	
Week 15	Automation and workflows	
Week 16	Final Project Reports Due	

# Review Shell and Unix



OnDemand provides an integrated, single access point for all of your HPC resources.

<https://pinnacle-portal.uark.edu/>



# Getting help!

- \$ man ls
- Understand syntax of the command

```
LS(1)                                BSD General Commands Manual                                LS(1)

NAME
  ls -- list directory contents

SYNOPSIS
  ls [-ABCFGHLOPRSTUW@abcdfghiklmnopqrstuwX1] [file ...]

DESCRIPTION
  For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information. For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any requested, associated information.

  If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

  The following options are available:

  -@      Display extended attribute keys and sizes in long (-l) output.

  -1      (The numeric digit 'one'.) Force output to be one entry per line. This is the default when output is not to a terminal.

  -A      List all entries except for . and .. Always set for the super-user.

  -a      Include directory entries whose names begin with a dot (.).

  -B      Force printing of non-printable characters (as defined by ctype(3) and current locale settings) in file names as \xxx, where xxx is the numeric value of the character in octal.

  -b      As -B, but use C escape codes whenever possible.

  -C      Force multi-column output; this is the default when output is to a terminal.

  -c      Use time when file status was last changed for sorting (-t) or long printing (-l).

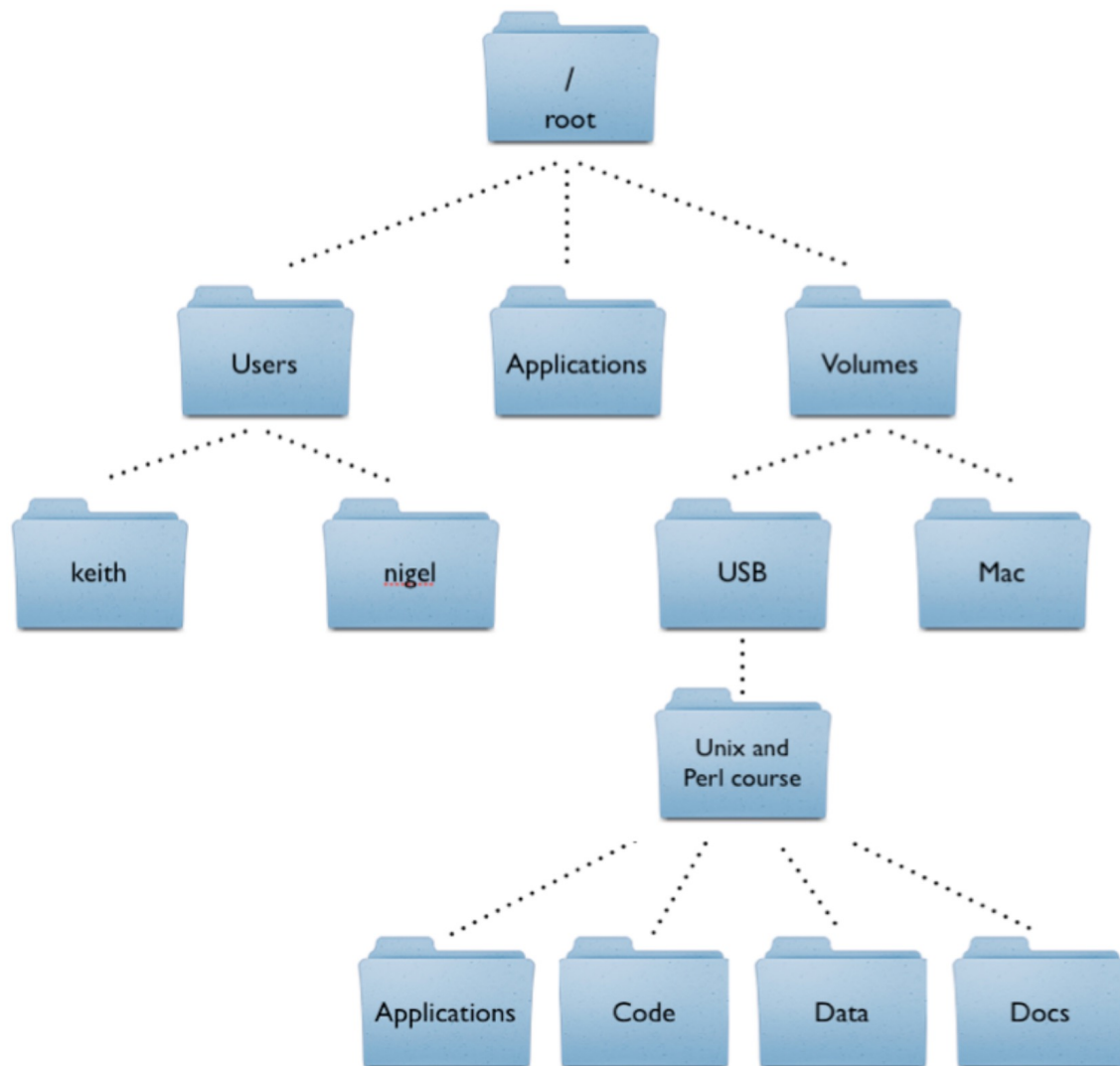
  -d      Directories are listed as plain files (not searched recursively).

  -e      Print the Access Control List (ACL) associated with the file, if present, in long (-l) output.

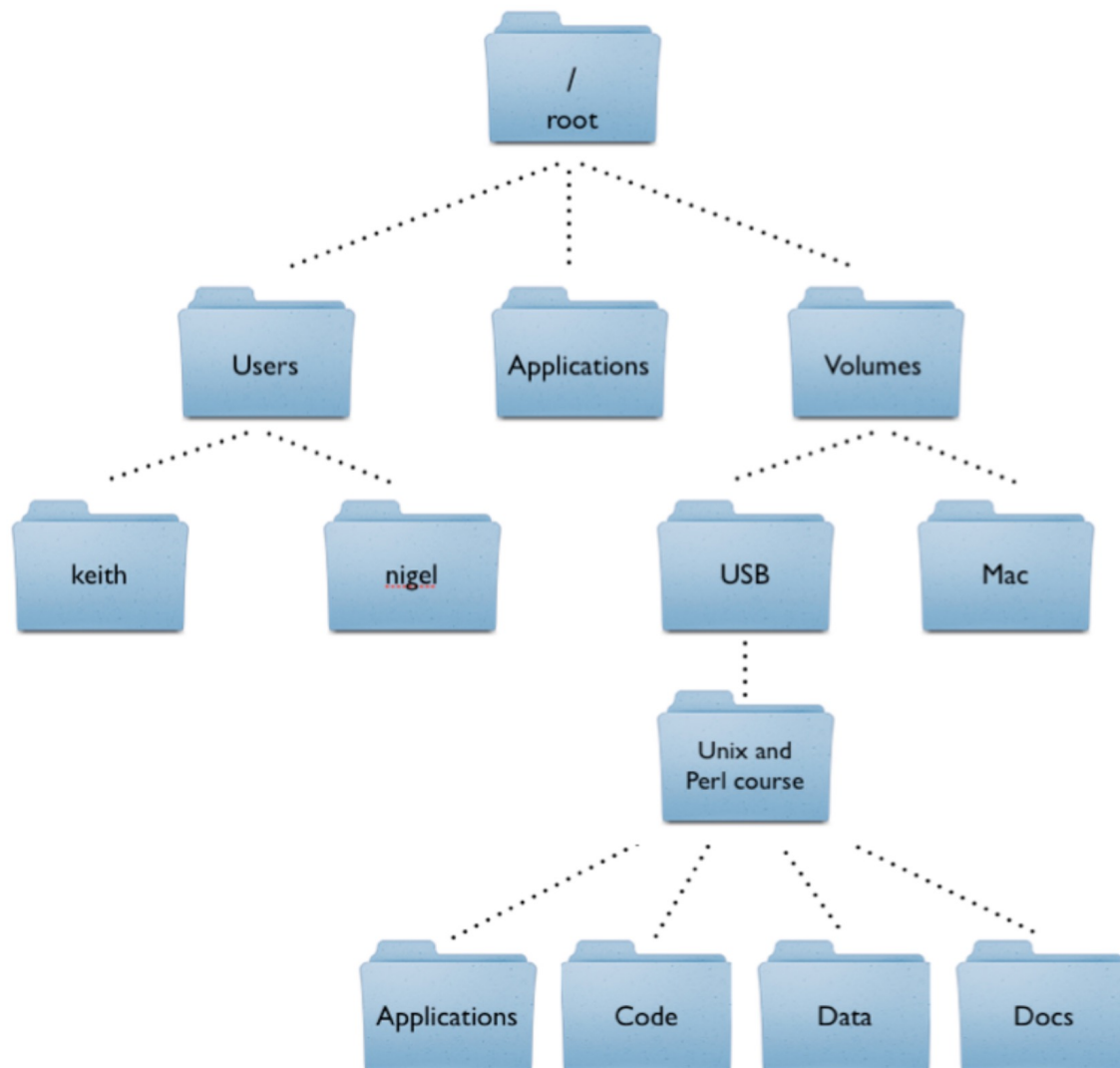
  -F      Display a slash (/) immediately after each pathname that is a directory, an asterisk (*) after each that is executable, an at sign (@) after each symbolic link, an equals sign (=) after each socket, a percent sign (%) after each whiteout, and a vertical bar (|) after each that is a FIFO.

  -f      Output is not sorted. This option turns on the -a option.

  -G      Enable colored output. This option is equivalent to defining CLICOLOR in the environment. (See below.)
```



Example directory structure




Example directory structure

- **pwd** - present working directory
- **ls** - list files
  - **ls -l** (list files as a list)
  - **ls -lht** (list files in a list sorted by most recent and humans readable file size)
- **cd** - change directory
  - **cd .** - current directory
  - **cd ..** - previous directory
  - **cd ~** - go to home folder
- **mkdir** - make directory
- **mv** - move or rename files
- **cp** - copy files
- **rm** - remove or delete files

# Text editor

- Nano, Vim, Emacs...
- \$ nano draft.txt

- $\wedge$  = 

- Write or save



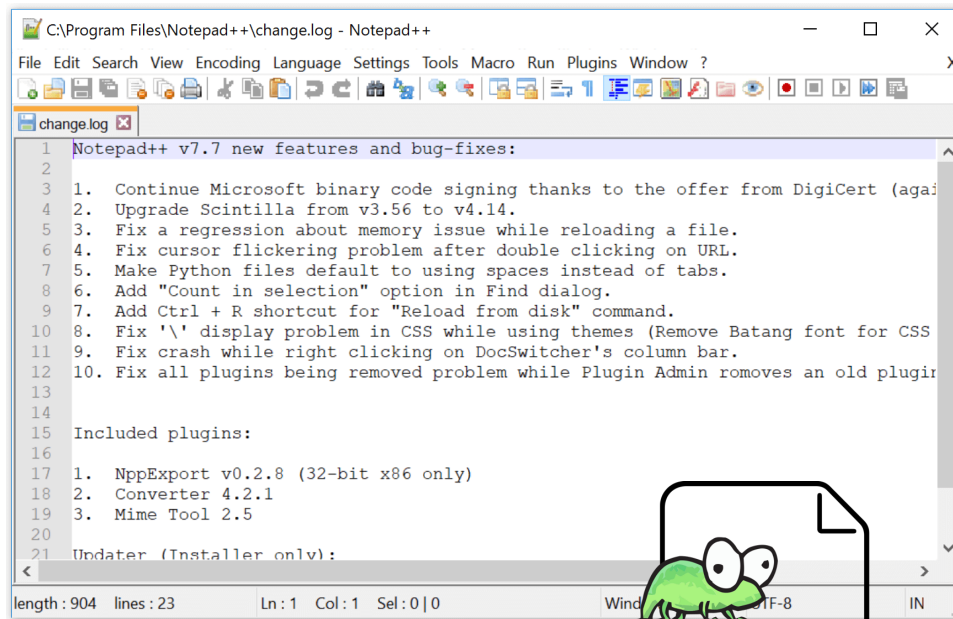
```
GNU nano 2.0.6           File: draft.txt           Modified

It's not "publish or perish" any more,
it's "share and thrive".
█

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```



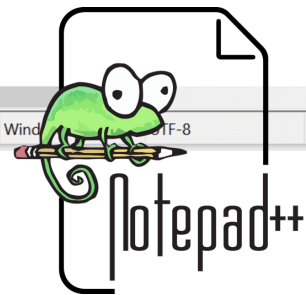
# Text editor



A screenshot of the Notepad++ v7.7.0 application window on a Windows operating system. The title bar reads "C:\Program Files\Notepad++\change.log - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar contains various icons for file operations and editing. The main text area displays the "change.log" file, which lists new features and bug fixes for version 7.7.0. The status bar at the bottom shows "length: 904 lines: 23 Ln: 1 Col: 1 Sel: 0|0".

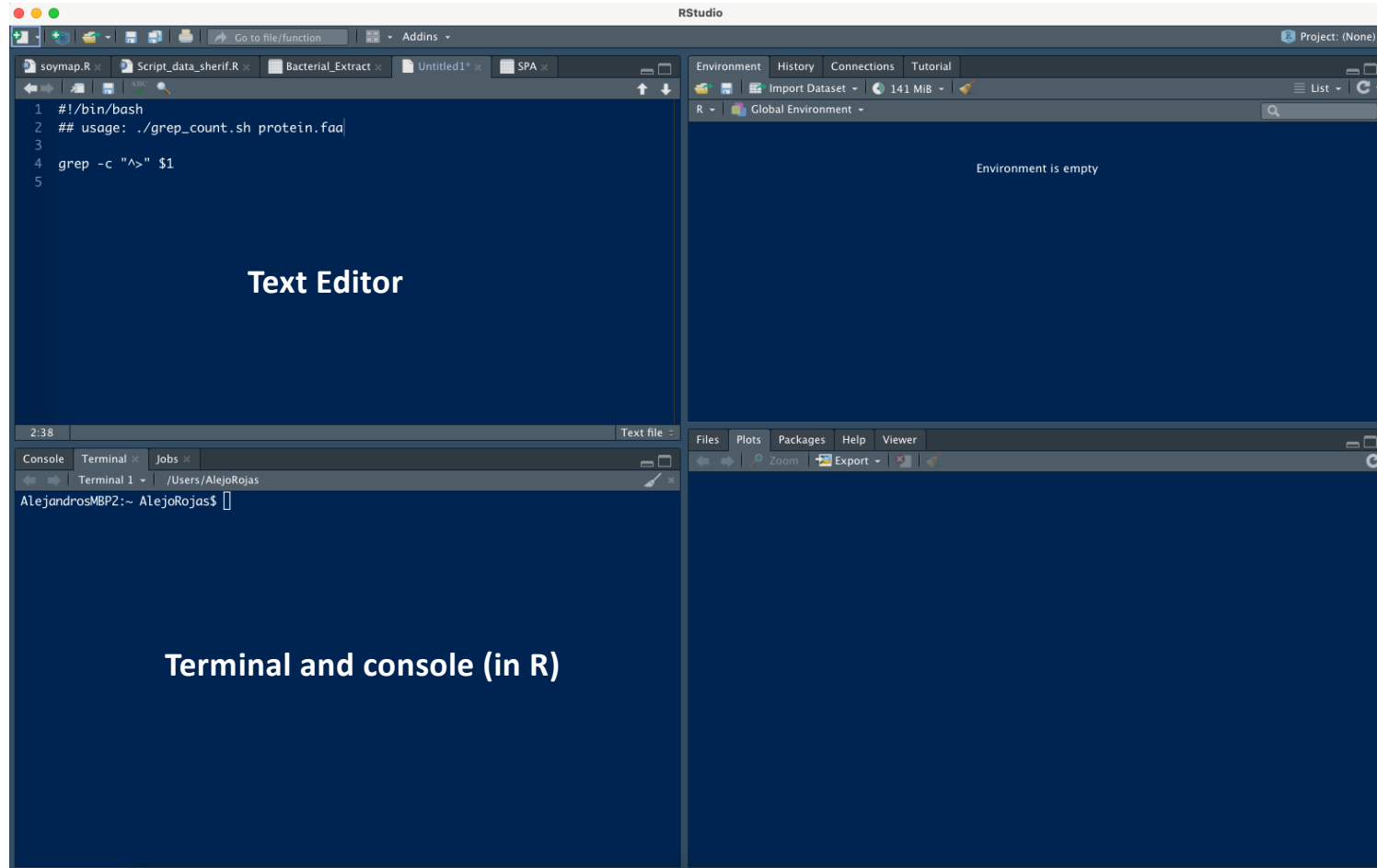
```
1 Notepad++ v7.7 new features and bug-fixes:
2
3 1. Continue Microsoft binary code signing thanks to the offer from DigiCert (again)
4 2. Upgrade Scintilla from v3.56 to v4.14.
5 3. Fix a regression about memory issue while reloading a file.
6 4. Fix cursor flickering problem after double clicking on URL.
7 5. Make Python files default to using spaces instead of tabs.
8 6. Add "Count in selection" option in Find dialog.
9 7. Add Ctrl + R shortcut for "Reload from disk" command.
10 8. Fix '\n' display problem in CSS while using themes (Remove Batang font for CSS)
11 9. Fix crash while right clicking on DocSwitcher's column bar.
12 10. Fix all plugins being removed problem while Plugin Admin removes an old plugin
13
14 Included plugins:
15
16 1. NppExport v0.2.8 (32-bit x86 only)
17 2. Converter 4.2.1
18 3. Mime Tool 2.5
19
20 Updater (Installer only):
21
```

Windows



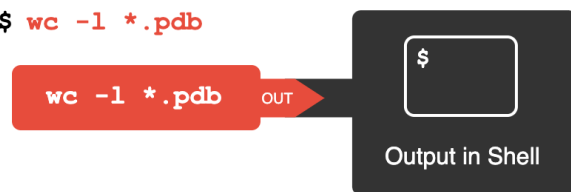
Mac

# IDEs (Interactive Development Environment)

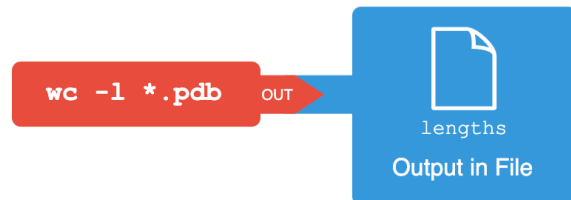


# Merging commands and std output

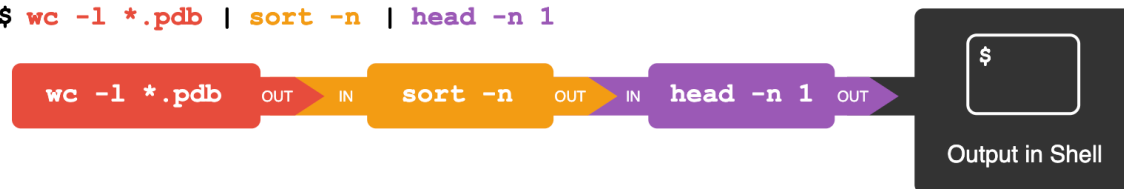
```
$ wc -l *.pdb
```



```
$ wc -l *.pdb > lengths
```

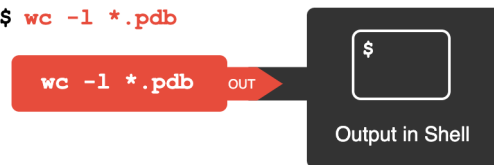


```
$ wc -l *.pdb | sort -n | head -n 1
```

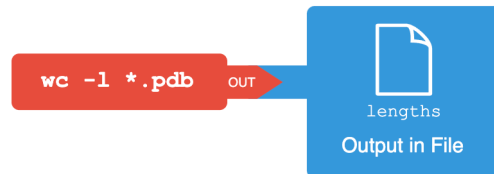


# Merging commands and std output

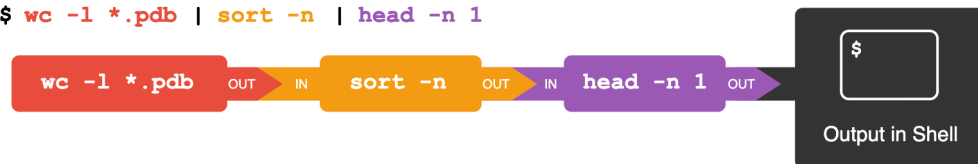
```
$ wc -l *.pdb
```



```
$ wc -l *.pdb > lengths
```



```
$ wc -l *.pdb | sort -n | head -n 1
```



- **wc** - counts lines, words, and characters in its inputs.
- **cat** - displays the contents of its inputs.
- **sort** - sorts its inputs.
- **head** - displays the first 10 lines of its input.
- **tail** - displays the last 10 lines of its input.
- **cut** - command is used to remove or 'cut out' certain sections of each line in the file
- **command > [file]** - redirects a command's output to a file (overwriting any existing content).
- **command >> [file]** - appends a command's output to a file.
- **[first] | [second]** is a pipeline: the output of the first command is used as the input to the second.

**Protein Advanced Search Builder**

((("prolactin"[Protein Name]) AND frogs) AND srcdb refseq[Properties])

[Edit](#) [Clear](#)

**Builder**

Protein Name   [Hide index list](#)

- prolactin (641)
- prolactin 1 (19)
- prolactin 1 like (9)
- prolactin 2 (28)
- prolactin 2 like (2)
- prolactin 2 precursor (1)
- prolactin 2a1 (4)
- prolactin 2a1 like (9)
- prolactin 2a1 like protein (2)
- prolactin 2a1 precursor (2)

[Previous 200](#)  
[Next 200](#)  
[Refresh index](#)

AND  All Fields  frogs  [Show index list](#)

AND  All Fields  srcdb refseq[Properties]  [Show index list](#)

AND  All Fields     [Show index list](#)

or [Add to history](#)

---

**History** [Download history](#) [Clear history](#)

Search	Add to builder	Query	Items found	Time
<a href="#">#30</a>	<a href="#">Add</a>	Search ((("prolactin"[Protein Name]) AND frogs) AND srcdb refseq[Properties])	1	16:18:17
<a href="#">#29</a>	<a href="#">Add</a>	Search prolactin	9897	16:14:08
<a href="#">#28</a>	<a href="#">Add</a>	Search frogs	196522	16:13:49
<a href="#">#27</a>	<a href="#">Add</a>	Search creatine kinase	3273	16:13:33
<a href="#">#26</a>	<a href="#">Add</a>	Search srcdb refseq[Properties]	64565536	16:13:10

<https://www.ncbi.nlm.nih.gov/genbank/samplerecord/>