Lab #2

grep and sed

BINF 2111, Fall 2023

Housekeeping

- Check-In
  - Last week's material
  - grep and sed
- Attendance
- Academic integrity
- Bonuses
- Lab grades
- Personal computer permission

Terminology

String Sequence of characters and can contain letters, numbers, symbols and even spaces.

RegEx Regular Expression, a pattern (or filter) that describes a set of strings that matches the pattern.

grep A command-line utility for searching plain-text data sets for lines that match a regular expression.

FASTA A text-based file format for representing either nucleotide sequences or amino acid sequences.

sed A stream editor that can perform lots of functions on file like searching, find and replace, insertion, or deletion.

Commands To Know

Commands are case sensitive !!

● Command

● Options

● Input (like a file or folder)

● RegEx

Command	Meaning	Usage
rm	Remove, delete a file. Use -r to delete a directory and all of its contents.	rm [file] rm -r [directory]
clear	Clears all the text on the screen.	clear
grep	Finds text that matches a pattern and returns the lines containing that text.	grep [options] "[regex]" [file]
sed	Stream editor. Diverse command that can manipulate text/files.	sed [options] '[regex]' [file]
tr	Translate, change or delete characters in a file.	tr [options] [old format] [new format]

Operators To Know

Operator	Meaning	Usage
&&	And. Use between two commands to do both commands	echo "hi" && echo "bye"
^	Find matches that are at the beginning of a line.	grep "^ATG" example.fasta
$	Find matches that are at the end of a line.	grep "TAG$" example.fasta
|	Pipe, or. Use between two regex matches to match both.	grep "hi|bye" file.txt
.	Match any single character	grep "wh." file.txt

Command Breakdown - grep

- **grep:** finds text that matches a pattern and returns the lines containing that text
  - Useful Options
    - -c This prints only a count of the lines that match a pattern
    - -n Display the matched lines and their line numbers.
    - -v This prints out all the lines that do not matches the pattern
    - -E Treats pattern as an extended regular expression (ERE)
    - -o Print only the matched parts of a matching line, with each such part on a separate output line.
  - Usage
    - `grep -cv "hello" file.txt file1.txt file2.txt`
    - `egrep -o "hi|bye" file.txt` OR `grep -Eo "hi|bye" file.txt`
    - `grep -n "hello" file.txt`

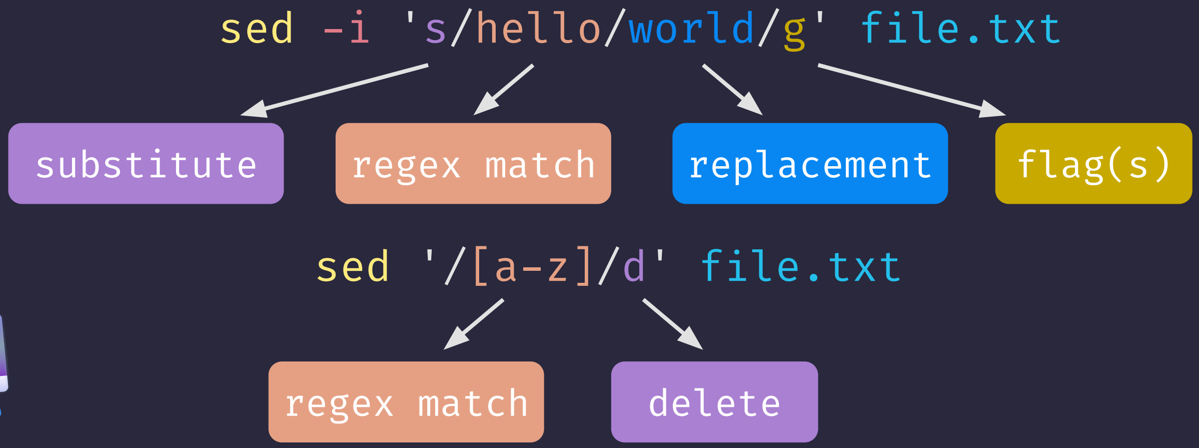
Full List of grep Options

- -c This prints only a count of the lines that match a pattern
- -h Display the matched lines, but do not display the filenames.
- -i Ignores, case for matching
- -l Displays list of a filenames only.
- -n Display the matched lines and their line numbers.
- -v This prints out all the lines that do not matches the pattern
- -e Specifies expression with this option. Can use multiple times.
- -f Takes patterns from file, one per line.
- -E Treats pattern as an extended regular expression (ERE)
- -w Match whole word
- -o Print only the matched parts of a matching line, with each such part on a separate output line.
- -A n Prints searched line and n lines after the result.
- -B n Prints searched line and n line before the result.
- -C n Prints searched line and n lines after before the result.

Command Breakdown - sed

- **sed**: stream editor. Diverse command that can manipulate text/files
    - Mac users need to use gsed

- Command Breakdown:



```
sed -i 's/hello/world/g' file.txt
```

substitute

regex match

replacement

flag(s)

```
sed '/[a-z]/d' file.txt
```

regex match

delete

sed's Useful Options, Commands, & Flags

- **Options**
  - -i In-place, files are edited rather than printing output to the terminal
  - -E or -r Use extended regular expressions rather than basic regular expressions (similar to using egrep)

- **Commands**
  - s/[m]/[r]/[flags] Substitute, match the regex ([m]) in a file and replace matched string with [r]
  - /[m]/d Delete, delete the line containing the regex match [m]
  - [#]a [text] Append, appending text after line [#] (use i for before)

- **Flags**
  - /g Global, apply the replacement to all matches, not just the first
  - /I Case Insensitive, match the regex case insensitively
  - /[#] Nth, only replace the Nth match of the regex

sed Examples

- **`sed -i '/hello/d' file.txt`**
  - Edit file.txt so that lines with hello are deleted
- **`sed -E 's/[Hh]ello/world/g' file.txt`**
  - Find all occurrences of Hello and hello and replace them with world
- **`sed '2a hello' file.txt`**
  - Insert hello after the second line
- **`sed '2i hello' file.txt`**
  - Insert hello before the second line
- **`sed 's/hello/world/Ig' file.txt`**
  - Find all occurrences of hello, ignoring case, and replace with world
- **`sed 's/hello/world/3' file.txt`**
  - Find the third occurrence of hello and replace with world

Command Breakdown - tr

- **tr**: translate, change or delete characters in a file
  - Options
    - -c Complement, apply other option to characters not in the given string
    - -d Delete character(s)
    - -s Squeeze, replace repeated characters with a single occurrence
    - -t Truncate, if set 1 is larger than set 2, the size of set 1 will be matched to the size of set 2 (unavailable on Mac)
  - Usage
    - `tr [options] [set 1] [set 2]`

tr Examples

- **`cat file.txt | tr "[a-z]" "[A-Z]"`**
  - Turns all lowercase letters in file.txt into uppercase letters
- **`tr "[:lower:]" "[:upper:]" < file.txt`**
  - Turns all lowercase characters in file.txt into uppercase characters
- **`echo "This    Is    Lab2" | tr -s " "`**
  - Removes repeated spaces, output: `This Is Lab2`
- **`tr -d T <<< "This    Is    Lab2"`**
  - Deletes T, output: `his    Is    Lab2`
- **`echo "this is course number 199384" | tr -cd "[:digit:]"`**
  - Removes everything except digits, output: `199384`
- **`tr -t 'isef' '12' <<< "This is Lab2"`**
  - Replace is with 12, ignore replacing ef, output: `Th12 12 Lab2`

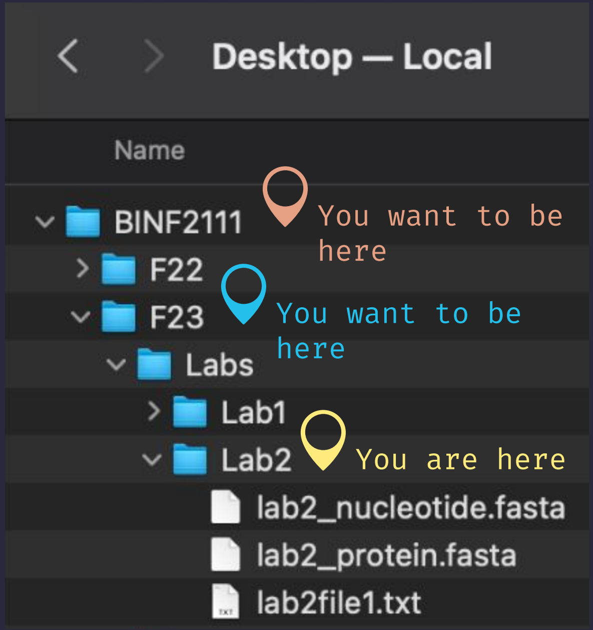
Navigating Your Files Easily

- If you are in Lab2, how do you get to BINF2111?
  `cd ~/Desktop/BINF2111`

- If you are in Lab2, how do you get to F23?
  `cd ../../`

- If you moved to F23 from Lab2, how do you get back to Lab2?
  `cd -`

```
- is the previous directory
~/ is the home directory
../ moves up one directory
./ is the current directory
```



What's A FASTA File?

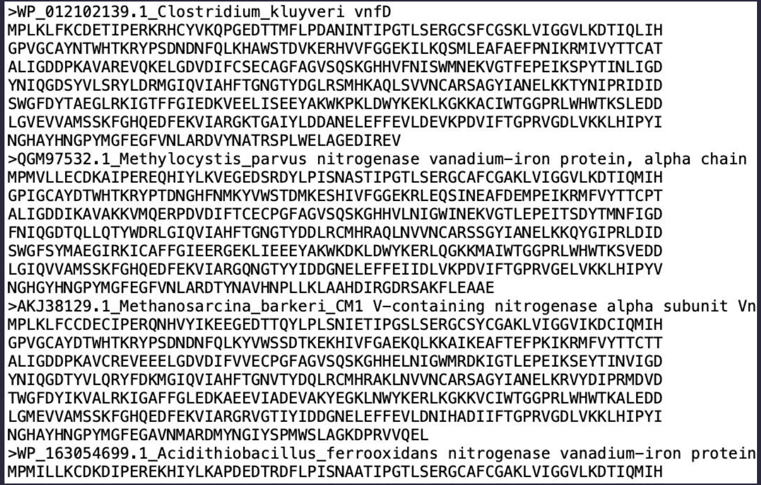
- **FASTA**: A text-based file format for representing either nucleotide sequences or amino acid sequences
    - Amino acids are represented by their single letter code
    - Nucleotides and amino acids cannot be in the same file
    - Files can end in **.fasta**, **.fna** (nucleotide only), or **.faa** (amino acid only)

- The Basic Format:

```
>Sequence_ID Sequence Information (organism, gene name, GC
content, description of sequence)
ATGTTAGCTASGTCTAAGTCGATCGAT…
```

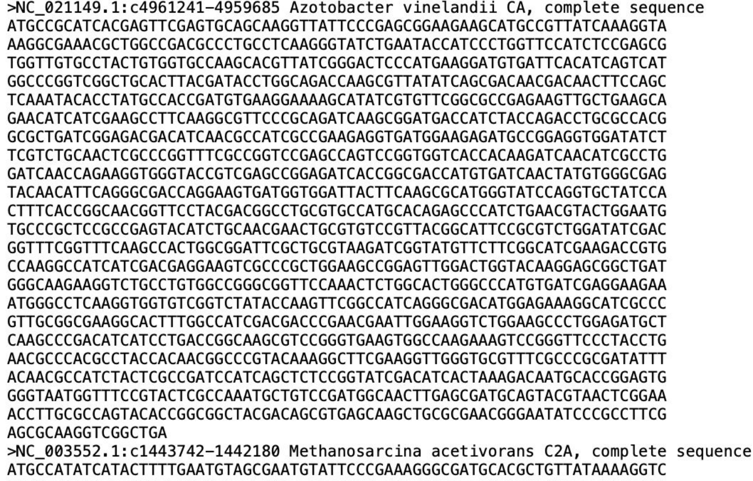
FASTA File Examples

An .faa Example:



```
>WP_012102139.1_Clostridium_kluyveri vnfD
MPLKLFKCDETIPERKRHCYVKQPGEDTTMFLPDANINTIPGTLSERGCSFCGSKLVIGGVLKDTIQLIH
GPVGCAYNTWHTKRYPSDNDNFQLKHAWSTDVKERHVVFGGEKILKQSMLEAFAEFPNIKRMIVYTTCAT
ALIGDDPKAVAREVQKELGDVDIFCSECAGFAGVSQSKGHHVFNISWMNEKVGTFEPEIKSPYTINLIGD
YNIQGDSYVLSRYLDRMGIQVIAHFTGNGTYDGLRSMHKAQLSVVNCARSAGYIANELKKTYNIPRIDID
SWGFDYTAEGLRKIGTFFGIEDKVEELISEEYAKWKPKLDWYKEKLKGKKACIWTGGPRLWHWTKSLEDD
LGVEVVAMSSKFGHQEDFEKVIARGKTGAIYLDDANELEFFEVLDEVKPDVIFTGPRVGDLVKKLHIPYI
NGHAYHNGPYMGFEGFVNLARDVYNATRSPLWELAGEDIREV
>QGM97532.1_Methylocystis_parvus nitrogenase vanadium-iron protein, alpha chain
MPMVLLECDKAIPEREQHIYLKVEGEDSRDYLPISNASTIPGTLSERGCAFCGAKLVIGGVLKDTIQMIH
GPIGCAYDTWHTKRYPTDNGHFNMKYVWSTDMKESHIVFGGEKRLEQSINEAFDEMPEIKRMFVYTTCPT
ALIGDDIKAVAKKVMQERPDVDIFTCECPGFAGVSQSKGHHVLNIGWINEKVGTLEPEITSDYTMNFIGD
FNIQGDTQLLQTYWDRLGIQVIAHFTGNGTYDDLRCMHRAQLNVVNCARSSGYIANELKKQYGIPRLDID
SWGFSYMAEGIRKICAFFGIEERGEKLIEEEYAKWKDKLDWYKERLQGKKMAIWTGGPRLWHWTKSVEDD
LGIQVVAMSSKFGHQEDFEKVIARGQNGTYYIDDGNELEFFEIIDLVKPDVIFTGPRVGELVKKLHIPYV
NGHGYHNGPYMGFEGFVNLARDTYNAVHNPLLKLAAHDIRGDRSAKFLEAAE
>AKJ38129.1_Methanosarcina_barkeri_CM1 V-containing nitrogenase alpha subunit Vn
MPLKLFCCDECIPERQNHVYIKEEGEDTTQYLPLSNIETIPGSLSERGCSYCGAKLVIGGVIKDCIQMIH
GPVGCAYDTWHTKRYPSDNDNFQLKYVWSSDTKEKHIVFGAEKQLKKAIKEAFTEFPKIKRMFVYTTCTT
ALIGDDPKAVCREVEEELGDVDIFVVECPGFAGVSQSKGHHELNIGWMRDKIGTLEPEIKSEYTINVIGD
YNIQGDTYVLQRYFDKMGIQVIAHFTGNVTYDQLRCMHRAKLNVVNCARSAGYIANELKRVYDIPRMDVD
TWGFDYIKVALRKIGAFFGLEDKAEEVIADEVAKYEGKLNWYKERLKGKKVCIWTGGPRLWHWTKALEDD
LGMEVVAMSSKFGHQEDFEKVIARGRVGTIYIDDGNELEFFEVLDNIHADIIFTGPRVGDLVKKLHIPYI
NGHAYHNGPYMGFEGAVNMARDMYNGIYSPMWSLAGKDPRVVQEL
>WP_163054699.1_Acidithiobacillus_ferrooxidans nitrogenase vanadium-iron protein
MPMILLKCDKDIPEREKHIYLKAPDEDTRDFLPISNAATIPGTLSERGCAFCGAKLVIGGVLKDTIQMIH
```

An .fna Example:



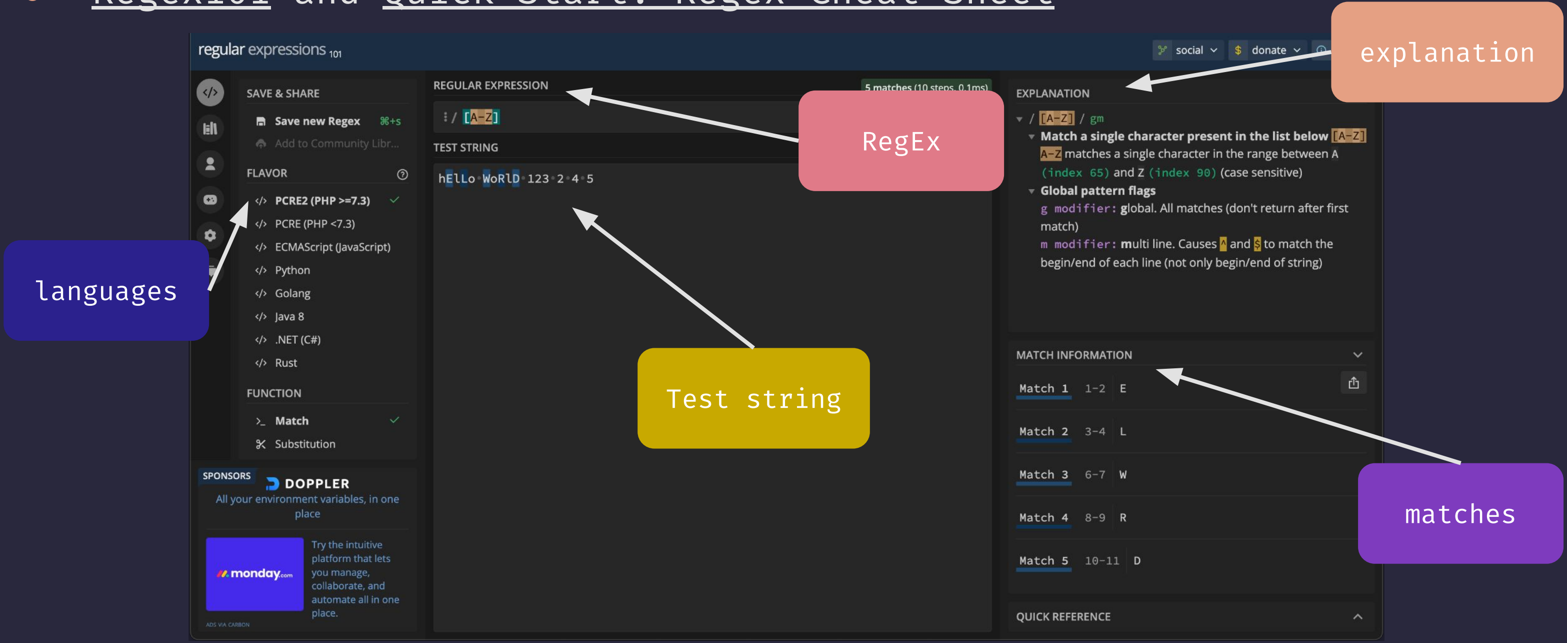
```
>NC_021149.1:c4961241-4959685 Azotobacter vinelandii CA, complete sequence
ATGCCGCATCACGAGTTCGAGTGCAGCAAGGTTATTCCCGAGCGGAAGAAGCATGCCGTTATCAAAGGTA
AAGGCGAAACGCTGGCCGACGCCCTGCCTCAAGGGTATCTGAATACCATCCCTGGTTCCATCTCCGAGCG
TGGTTGTGCCTACTGTGGTGCCAAGCACGTTATCGGGACTCCCATGAAGGATGTGATTCACATCAGTCAT
GGCCCGGTCGGCTGCACTTACGATACCTGGCAGACCAAGCGTTATATCAGCGACAACGACAACTTCCAGC
TCAAATACACCTATGCCACCGATGTGAAGGAAAAGCATATCGTGTTCGGCGCCGAGAAGTTGCTGAAGCA
GAACATCATCGAAGCCTTCAAGGCGTTCCCGCAGATCAAGCGGATGACCATCTACCAGACCTGCGCCACG
GCGCTGATCGGAGACGACATCAACGCCATCGCCGAAGAGGTGATGGAAGAGATGCCGGAGGTGGATATCT
TCGTCTGCAACTCGCCCGGTTTCGCCGGTCCGAGCCAGTCCGGTGGTCACCACAAGATCAACATCGCCTG
GATCAACCAGAAGGTGGGTACCGTCGAGCCGGAGATCACCGGCGACCATGTGATCAACTATGTGGGCGAG
TACAACATTCAGGGCGACCAGGAAGTGATGGTGGATTACTTCAAGCGCATGGGTATCCAGGTGCTATCCA
CTTTCACCGGCAACGGTTCCTACGACGGCCTGCGTGCCATGCACAGAGCCCATCTGAACGTACTGGAATG
TGCCCGCTCCGCCGAGTACATCTGCAACGAACTGCGTGTCCGTTACGGCATTCCGCGTCTGGATATCGAC
GGTTTCGGTTTCAAGCCACTGGCGGATTCGCTGCGTAAGATCGGTATGTTCTTCGGCATCGAAGACCGTG
CCAAGGCCATCATCGACGAGGAAGTCGCCCGCTGGAAGCCGGAGTTGGACTGGTACAAGGAGCGGCTGAT
GGGCAAGAAGGTCTGCCTGTGGCCGGGCGGTTCCAAACTCTGGCACTGGGCCCATGTGATCGAGGAAGAA
ATGGGCCTCAAGGTGGTGTCGGTCTATACCAAGTTCGGCCATCAGGGCGACATGGAGAAAGGCATCGCCC
GTTGCGGCGAAGGCACTTTGGCCATCGACGACCCGAACGAATTGGAAGGTCTGGAAGCCCTGGAGATGCT
CAAGCCCGACATCATCCTGACCGGCAAGCGTCCGGGTGAAGTGGCCAAGAAAGTCCGGGTTCCCTACCTG
AACGCCCACGCCTACCACAACGGCCCGTACAAAGGCTTCGAAGGTTGGGTGCGTTTCGCCCGCGATATTT
ACAACGCCATCTACTCGCCGATCCATCAGCTCTCCGGTATCGACATCACTAAAGACAATGCACCGGAGTG
GGGTAATGGTTTCCGTACTCGCCAAATGCTGTCCGATGGCAACTTGAGCGATGCAGTACGTAACTCGGAA
ACCTTGCGCCAGTACACCGGCGGCTACGACAGCGTGAGCAAGCTGCGCGAACGGGAATATCCCGCCTTCG
AGCGCAAGGTCGGCTGA
>NC_003552.1:c1443742-1442180 Methanosarcina acetivorans C2A, complete sequence
ATGCCATATCATACTTTTGAATGTAGCGAATGTATTCCCGAAAGGGCGATGCACGCTGTTATAAAAGGTC
```

RegEx Breakdown

- **RegEx**: regular expression, a pattern (or filter) that describes a set of strings that matches the pattern
  - Commonly Used Patterns:
    - \d or [[:digit:]] or [0-9] Any number from 0 to 9
    - \w or [[:alpha:]] or [A-Za-z] Any letter, regardless of capitalization. \w includes numbers
    - \s or [[:space:]] Any whitespace character (space, tab, newline, carriage return, form feed, and vertical tab)
    - [A-Z] or [[:upper:]] Any uppercase character
    - [a-z] or [[:lower:]] Any lowercase character
    - \n New line
    - \t Tab
    - [^match] Anything but match, [^a-z] any non-lowercase character

RegEx Resources

- Regex101 and Quick-Start: Regex Cheat Sheet



REGULAR EXPRESSION

5 matches (10 steps, 0.1ms)

```
/ [A-Z]
```

TEST STRING

```
hEllo WoRlD 123 2 4 5
```

EXPLANATION

/ [A-Z] / gm

Match a single character present in the list below [A-Z]

A-Z matches a single character in the range between A (index 65) and Z (index 90) (case sensitive)

Global pattern flags

g modifier: global. All matches (don't return after first match)

m modifier: multi line. Causes ^ and $ to match the begin/end of each line (not only begin/end of string)

MATCH INFORMATION

```
Match 1   1-2   E
Match 2   3-4   L
Match 3   6-7   W
Match 4   8-9   R
Match 5   10-11 D
```