Lecture 15 - String Manipulations with Regular Expressions

Anchors

^ Indicates the beginning of a line

$ Indicates the end of a line

Anchors

```
@0_V10.F.030_FCC638CACXX:5:1101:1193:1928#ATCNCGATC/1
TCATGTATAAAAATGCCGTATGTGTCTGTTCGTTTGCCATTCATAGACTCGAAAACT
+
efhggfhfhhhdggXdfffcgcfhh_e_cedfddhhhhhbcfadbgeg]ddbZ^a]_
```

Anchors

```
@0_V10.F.030_FCC638CACXX:5:1101:1193:1928#ATCNCGATC/1
TCATGTATAAAAATGCCGTATGTGTCTGTTCGTTTGCCATTCATAGACTCGAAAACT
+
efhggfhfhhhdggXdfffcgcfhh_e_cedfddhhhhhbcfadbgeg]ddbZ^a]_
```

```
^@.+/1$
```

Conditional

|	Indicates **or** when placed between two strings

Conditional

```
grep '(B|b)iology' myCV.txt
```

BIOS 101: Biology for non-majors

BIOS 185: Introduction to biology for majors

Backreferencing

`( )` Groups the string within

`\1` References the string within the group

Backreferencing

2139.Rpomonella.**haw**thorn.**Dow**agiac.MI.m

2140.Rpomonella.**haw**.**Dow**agiac.MI.m

2000.Rpomonella.**Haw**.**Urb**ana.IL.f

2001.Rpomonella.**Haw**thorn.**Urb**ana.IL.f

Haw**_SiteAbbrev_ID#**

```
[0-9]{4}\.Rpomonella\.[Hh]aw(thorn)?\.[A-Z][a-z]+
\.[A-Z]{2}\.[mf]
```

Backreferencing

2139.Rpomonella.**haw**thorn.**Dow**agiac.MI.m

2140.Rpomonella.**haw**.**Dow**agiac.MI.m

2000.Rpomonella.**Haw**.**Urb**ana.IL.f

2001.Rpomonella.**Haw**thorn.**Urb**ana.IL.f

([0-9]{4})\.Rpomonella\.[Hh]aw(thorn)?\.**([A-Z][a-z]{2})**[a-z]+\.[A-Z]{2}\.[mf]

Backreferencing

2139.Rpomonella.**haw**thorn.**Dow**agiac.MI.m

2140.Rpomonella.**haw**.**Dow**agiac.MI.m

2000.Rpomonella.**Haw**.**Urb**ana.IL.f

2001.Rpomonella.**Haw**thorn.**Urb**ana.IL.f

([0-9]{4})`\.Rpomonella\.[Hh]aw(thorn)?\.`**([A-Z][a-z]{2})**
`[a-z]+\.[A-Z]{2}\.[mf]`

Haw_\3_\1

Thanks for catching this!!

Grep Exercise (#12)

Challenge: Utilizing `grep`, print to standard out the open reading frames in `R.mendax.1.fasta`.

(start codon: ATG, stop codon: TAA,TAG,TGA)

Grep Exercise (#12)

Challenge: Utilizing `grep`, print to standard out the open reading frames in `R.mendax.1.fasta`.

(start codon: ATG, stop codon: TAA,TAG,TGA)

```
grep -Eo 'ATG([ATCG]{3})+(TAA|TAG|TGA)'
R.mendax.1.fasta
```

Sed Exercise

Challenge: Utilizing `sed` and `grep`, rearrange the columns in `Fall2017MaggotCounts.csv` to list `Host,Location,DateCollected,Number`. Ignore the column headings and include only maggots collected from an `apple` host in September. Print the output to a file named `rearranged.csv`

Sed Exercise

Challenge: Utilizing `sed` and `grep`, rearrange the columns in `Fall2017MaggotCounts.csv` to list `Host,Location,DateCollected,Number`. Ignore the column headings and include only maggots collected from an `apple` host in September. Print the output to a file named `rearranged.csv`

```
cat Fall2017MaggotCounts.csv | sed -E
's/(9[0-9/]+),([A-Za-z&.]+),
(apple),([0-9]+)/\3,\2,\1,\4/g' | grep -E '^a' >
rearranged.csv
```

Regex in R and Python

`R`:

```
install.packages('stringr')
```

```
library('stringr')
```

`Python`:

```
import re
```

Regex in R and Python

R :

result = str_**extract**(searchString,regexString)

May have to escape some metacharacters!

http://stringr.tidyverse.org/articles/regular-expressions.html

Python:

result = re.**search**(regexString,searchString)

r"regexString" passes raw string to function

https://docs.python.org/2/library/re.html

R and Python Exercise

Challenge: Utilizing `R` or `Python`, print to standard out the open reading frames in `R.mendax.1.fasta`.

(start codon: ATG, stop codon: TAA,TAG,TGA)