Biological file formats

Each file format represents

1. **Information** – types of knowledge that are stored in the file

2. **Optimization** – types of operations that are easy/efficient to perform

The above implies that some information may not be present or cannot be easily extracted from a certain file format.

Download your first data file
Saccharomyces Cerevisae Feature File

You can use the a browser or the command line:

```
$ curl http://url.goes.here -o sc.gff
```

(The URL is on the course webpage)

http://downloads.yeastgenome.org/curation/chromosomal_feature/saccharomyces_cerevisiae.gff

What is this?
Look at the extension → a GFF file.

• Due to a historical limitation (20 years or so ago) and only on Windows → files ended up having a three character extension → `.txt`, `.exe` etc.

• This limitation also turned out to be a blessing and it stood the test of time. Makes it easy to see the file type.

• Note: the file extension can be incorrect → mind-bogglingly confusing errors may arise then.
Use unix command to “poke” at the file

```
# what folder are we in
pwd

# get the data, no flag stores the result in the sc.gff
curl http://downloads.yeastgenome.org/curation/chromosome1_feature/sacchromys

# check to see what files we have here
tar -tvf sc.gff

# check to see how many lines does the file have
wc -l sc.gff

# look at the first ten lines
head sc.gff

# look at the last ten lines
tail sc.gff

# page through the file
more sc.gff
```

Tabular formats

- Many common bioinformatics data formats are column based and tab-separated
- First format we deal with will be the

**GFF3 – Generic Feature Format**

(search for GFF3 to see the specification for version 3)


**The GFF3 specification**

Tab separated with 9 columns. Missing attributes may be replaced with a dot .

1. **Seqid** (usually chromosome)
2. **Source** (where is the data coming from)
3. **Type** (usually a term from the sequence ontology)
4. **Start** (interval start relative to the seqid)
5. **End** (interval end relative to the seqid)
6. **Score** (the score of the feature, a floating point number)
7. **Strand** (+/-)
8. **Phase** (used to indicate reading frame for coding sequences)
9. **Attributes** (semicolon separated attributes \( \text{Name}=	ext{ABC}; \text{ID}=1 \))

Example attribute specification: \( \text{name}=\text{REB1}; \text{id}=\text{YP33546} \)
What do the terms mean?

Sequence ontology browser

Unix commands in this lecture

• wc, cat, head, tail, sort, cut, grep, more, clear

Handy Tips

CTRL-C or ESC → interrupts any process that may be running

clear → clears the screen
cursor keys allow you to recall past commands
auto-complete → write part of the filename then press TAB
Check head/tail of the file

Paging data with: less (more)

- q or ESC → quits the pager
- SPACE or f → go forward, next page
- b → go backward
- / word → search for a word
- / → repeats the search for the last word

Find patterns in the file with GREP

Connecting streams

- Input streams: data from the keyboard or files
- Output streams: print on screen, into files

Stream redirection the symbols of “arrows” <, >

Input stream redirection from file: < filename
Output stream redirection to a file: > filename
Redirecting to a file creates/overwrites that file

```bash
ealbert@porthos:~/work
$ grep VAL606W sc.gff > match.gff
```

ealbert@porthos:~/work
$ ls
match.gff sc.gff

---

Piping streams

- The pipe character `|` channels the output of one command into the other

  (located above the ENTER key)

You can pipe multiple commands together

---

Piping commands

```bash
# how many genes in the data
$ grep gene.sc.gff | wc -l
7546

# how many genes on chromosome 6
$ grep gene.sc.gff | grep chr6 | wc -l
1119

# how many non-dubious genes on chromosome 6
$ grep gene.sc.gff | grep chr6 | grep -v Dubious | wc -l
997
```

---

Break the file into two – before the the word FASTA and after the word

```bash
# file out at what line does the word FASTA appear in the file
$ awk -v OFS="\t" '{if ($3~/FASTA/) print $1}' test.gff | tail -n 1
54

# isolate only the lines that contain tabular features
$ awk -v OFS="\t" '{if ($3~/\t/) print $1}' test.gff | tail -n 1
54
```
Homework 2

• Create a file that show all ontology terms that are present in the provided GFF file with a count of how many times this element occurs in the yeast genome.

  What is the top-ten list of terms that occur most frequently (provide counts)

• Pick an ontology term that is unfamiliar to you and look it up in the Sequence Ontology, paste the explanation into the homework