Lecture 4, Regular expressions continued

Greg Caporaso
gregcaporaso@gmail.com

Steps for building a regular expression

1. Copy the text you want to match to a new file
2. Mark the areas you'd like to capture
3. Add wildcards (maybe include spaces, if that helps)
4. Remove any extra spaces that you added.
5. Define the replacement string.

In class exercise

- Reformat sequence headers in a fasta file
  – Rewrite each identifier as the portion of the identifier preceding the . character, followed by an underscore, followed by the genus name

Defining custom wildcards

- You'll eventually want to match only certain characters, for example the letters in the nucleic acid or protein alphabet. This is accomplished using the `[ ]` characters:

  `[ACGT]` will match a single A, C, G or T

Wildcard ranges

- `[A-Z]` will match any uppercase letter
- `[a-z]` will match any lowercase letter
- `[A-Za-z]` will match either upper or lower case letters
- `[0-9]` will match any digit

In class exercise

- Start with LatLong.txt
- Make Lat/Long pairs tab-separated on a single line
- Remove *trailing* N and E, and replace with *leading* +
- Remove *trailing* S and W, and replace with *leading* -

Negative character sets

- To match all characters except a certain set, precede your set with a ^
  - [^ACTGactg] : match characters not corresponding to standard nucleotide abbreviations
- Separating tab-delimited text (i.e., match each tab-separated entry)
  - ?

Negative character sets

- To match all characters except a certain set, precede your set with a ^
  - [^ACTGactg] : match characters not corresponding to standard nucleotide abbreviations
- Separating tab-delimited text (i.e., match each tab-separated entry)
  - ([^\t]+)\t

Boundaries

- Some regular expression terms don’t match characters, but boundaries between characters
  - ^ : beginning of a line
  - $: end of a line
- Warning: ^ has different meanings depending on whether it is in square brackets or not.
- Search/replace on boundary characters inserts text at the boundary

In class exercise

- Reformat “blast9” output
  - Remove header (i.e. comment) lines
  - Format each line to contain the subject id, the query id, the e-value, the percent identity, and the alignment length, in that order!
  - Format as comma-separated text

Quantifiers

- Specify how many times a pattern should occur to be matched.

  + : match one or more occurrences

  * : match zero or more occurrences

  ~~? : match zero or one occurrences~~ [This is python regular expression syntax]

Greediness

- Quantifiers will match the *longest* string of text that matches a certain pattern
- Example: removing a polyA tail:
  - Search: (\w+)A*   Replace: \1
  - Search: (\w+)A+   Replace: \1

Greediness

- Quantifiers will match the *longest* string of text that matches a certain pattern
- Example: removing a polyA tail:
  - Search: (\w+)A*   Replace: \1
  - Search: (\w+)A+   Replace: \1
- ? Specifies to match the shortest string of text that matches the pattern
  - Search: (\w+?)A+   Replace: \1 (but this still doesn't work)

Greediness

- Quantifiers will match the *longest* string of text that matches a certain pattern
- Example: removing a polyA tail:
  - Search: (\w+)A*   Replace: \1
  - Search: (\w+)A+   Replace: \1
- ? Specifies to match the shortest string of text that matches the pattern
  - Search: (\w+?)A+   Replace: \1 (but this still doesn't work)
- Next, specify that the A+ must appear at the end of a line
  - Search: (\w+?)A+$ Replace: \1

Custom quantifiers

- To define custom numbers of matches, use {}
  - A{3} : match exactly 3 A characters
  - A{3,5} : match at least 3 and at most 5 A characters
  - A{3,} : match at least 3 A characters

Robust searches

- Sometimes your queries will fail
  - Won’t produce output (good)
  - Will produce incorrect output (bad)
- Fail loudly! Produce a (useful) error message on failure.

Designing robust searches

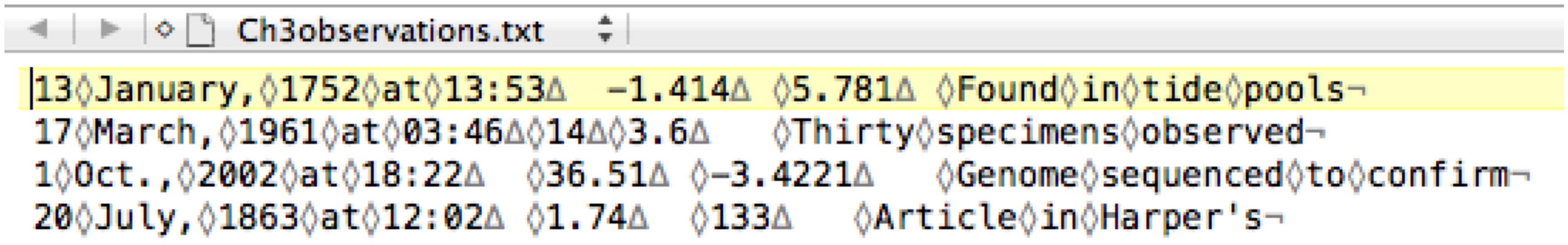
- Make assumptions explicit
  - If you're assuming that your records start with '>', search for '^>' to avoid matching '>' characters that show up in other places
- Match full lines by including ^ and $ in your search query
- Check the number of matches that were made: is it reasonable?

Testing of software

- Start thinking about what positive and negative controls for these terms might look like. Software testing is something we'll be discussing regularly through-out the semester.

In class exercise

- Start with Ch3observations.txt



```
13◊January,◊1752◊at◊13:53∆  -1.414∆ ◊5.781∆ ◊Found◊in◊tide◊pools¬
17◊March,◊1961◊at◊03:46∆◊14∆◊3.6∆    ◊Thirty◊specimens◊observed¬
1◊Oct.,◊2002◊at◊18:22∆  ◊36.51∆ ◊-3.4221∆    ◊Genome◊sequenced◊to◊confirm¬
20◊July,◊1863◊at◊12:02∆ ◊1.74∆   ◊133∆    ◊Article◊in◊Harper's¬
```

- Rearrange to look like:

  Year \t Mon. \t Day \t Hour \t Minute \t X \t Y