Regular expressions

Python – matching vs searching

- Matching - Looks only at the start of the string
- Searching - Looks for pattern everywhere in the target
- `re` module has multiple functions
  - `re.match(pattern, target, flags)`
  - `re.search(pattern, target, flags)`
  - `re.findall(pattern, target, flags)`
  - `re.split(pattern, target, maxsplit, flags)`
  - `re.sub(pattern, replacement, target,count, flags)`
  - `re.subn(pattern, replacement, target,count, flags)`

Literal matches (Exact matches)

Pattern	Target	Match Position	Search position
GAATTC	GAATTC		
	TTGAATTC		
	AATGTGAATTC		

Literal matches (Exact matches)

Pattern	Target	Match Position	Search position
GAATTC	GAATTC	0	0
	TTGAATTC	None	2
	AATGTGAATTC	None	5

Character sets

Pattern	Matches
[ATCG]	One DNA base character
[A-Za-z_]	One underscore or letter
[^0-9]	Any character except a digit
[-+/*^]	Any of the +, -, /, * or ^
[0-9\t]	Any digit or a tab
.	Any character

Some examples

- DsaI site - CCGCGG, CCGTGG, CCACGG, or CCATGG
  - CC[GA][TC]GG
- SecI site – CCNNGG
  - CC[ATCG][ATCG]GG
- CjuI - CjuI recognizes CA, followed by C or T, followed by any five bases, followed by a G or an A, followed by TG
  - CA[CT][ATCG][ATCG][ATCG][ATCG][ATCG][AG]TG

Character classes

Character	Matches
\d	Any digit
\D	Any nondigit
\s	Any whitespace character
\S	Any nonwhitespace character
\w	Any character considered part of a word
\W	Any character not considered part of a word

Boundaries

Character	Matches
^	Start of a line or pattern
$	End of a line or pattern
\A	Start of the pattern only
\Z	End of the pattern only

Variable length matching

Character	Matches
?	Zero or one repetitions of the preceding regex
*	Zero or more repetitions of the preceding regex
+	One or more repetitions of the preceding regex
{n}	Exactly n repetitions of the preceding regex
{m,n}	Between m and n (inclusive) repetitions of the preceding regex

Grep

Courtesy wikipedia

- `grep` is a command-line utility for searching plain-text data sets for lines matching a regular expression
- g/re/p - **g**lobally search a **r**egular **e**xpression and **p**rint
- `grep pattern filename`
- Several flags available `(-x, -v, -e, -i)`
- Matches regular expressions
- Very fast
- `grep` uses *Boyer-Moore Algorithm*

Simple exercises

(Try them with `grep/re`)

- Find words with all 5 vowels – any order
- Find words with all 5 vowels in alphabetical order
- Words with no vowels
- Words with one or more ‘z’
- Beginning with *micro*
- With *micro* somewhere in the middle of the word
- Ending with *tion*