CHEATSHEET LESSON 8: Merging Python and Bash

Python sys module

Module	Command	Description	Example
sys	sys.argv	Contains a list of command-line arguments (all strings!)	save_me = sys.argv[1]
sys	sys.path	Contains a list of directories in the python path	sys.path.append("another_path_i_want_to_access")

Python os and shutil modules

Module	Command	Description	Unix equivalent	Example
os	os.system	Run a UNIX command from python	anything you would type into a command line	os.system("pwd")
os	os.listdir	List all items in a given directory	ls	os.system("/directory/of/interest/")
os	os.remove	Remove a file	rm	os.remove("i_hate_this_file.txt")
os	os.rmdir	Remove a directory	rm -r	os.rmdir("/i/hate/this/directory/")
os	os.mkdir	Create a new directory	mkdir	os.mkdir("/path/to/brand/new/directory/")
os	os.mkdirs	Create many new directories	mkdir	os.mkdir("/path/to/a/brand/new/directory/", "/path/to/another/brand/new/directory/")
os	os.chdir	Change directory where python is running	cd	os.chdir("/another/directory/where/i/want/to/be/")
shutil	shutil.copy	Copy a file	cp	shutil.copy("old_file.txt", "new_file.txt")
shutil	shutil.move	Move a file	mv	shutil.move("old_file.txt", "new_file.txt")

Bash sort and uniq

Command	Meaning	Example
sort -b	ignore leading blanks	sort -b filename > filename.sorted
sort -r	reverse	sort -r filename > filename.sortedr
sort -k POS1	sort by field/character indicated by POS1	sort by field 2: sort -k 2 filename sort by second character sort in field 2: sort -k 2.2 filename
sort -k POS1,POS2	sort based on the characters from POS1 to POS2	sort by characters in fields 2 and 3: sort -k 2,3 filename sort starting with second character in field 2 up to and including field 3: sort -k 2.2,3 filename
uniq -c	prefixes lines with the number of times they occur	uniq -c filename
uniq -d	prints only repeated lines	uniq -d filename
uniq -u	prints only unique lines	uniq -u filename > filename.unique
uniq -f N	skips N number of lines	uniq -f 30 filename

Bash useful oneish-liners:

```
# replace XX and YY with AA and ZZ for every instance in the file
sed -e s/XX/AA/g -e/YY/ZZ/g filename > newfile

# sort and count sequences starting with barcodeY by size and alphabetically
gunzip -c yourzippedfastqfile.fastq.gz | grep ^[barcodeY] | sort -r | uniq -c

# print columns 2 and 3 from a .csv file
`cat yourfile.csv | awk -F, '{ print ($2,$3) }'

# move all files that end in .txt to .text
ls *.txt > list
sed s/"\(.*\).txt"/"mv \1.txt \1.text"/g list > command
bash command
```