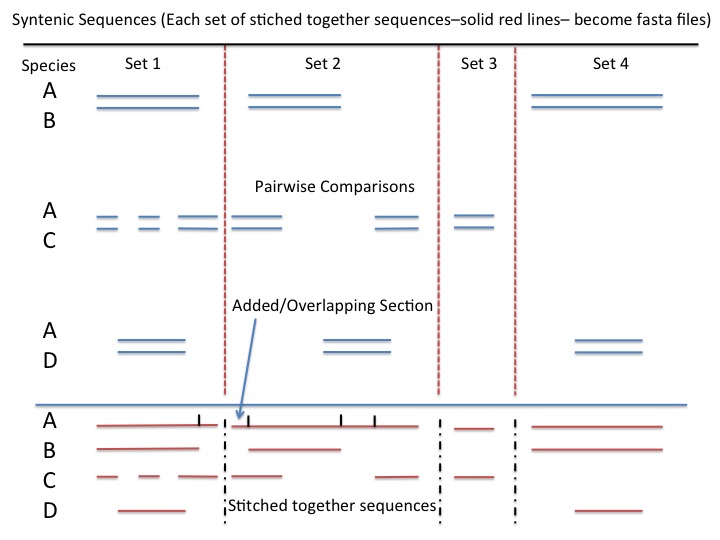
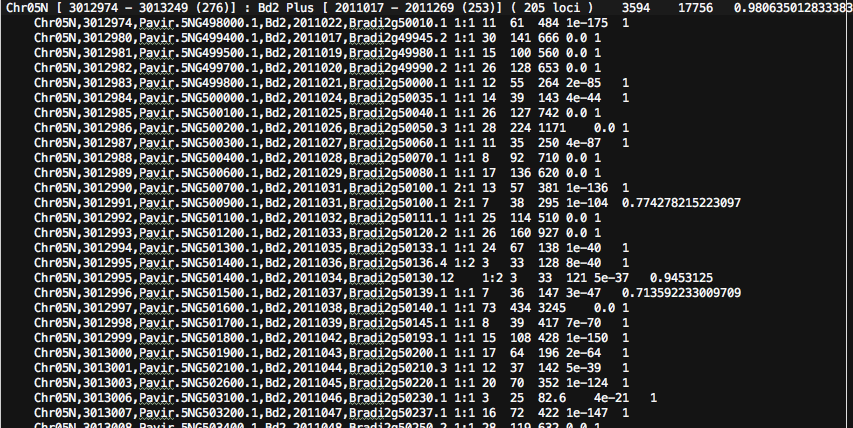
**Synteny Analysis on Pairwise Comparisons**

Description:

SyntenyFinal.py/SyntenyStructure(function) generates Fasta files that essentially stitch together pairwise comparisons of syntenic sequences across multiple species/comparisons.

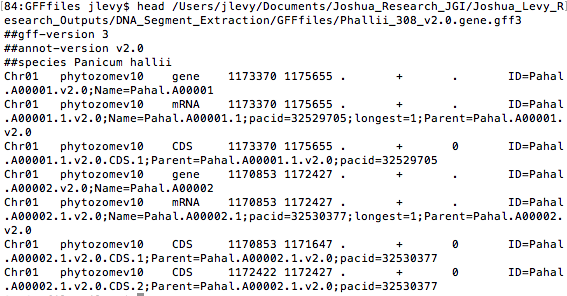
**Types of Input Files necessary for analysis:**

.unout files- unout files describe a synteny between query and target species. Typical format looks like

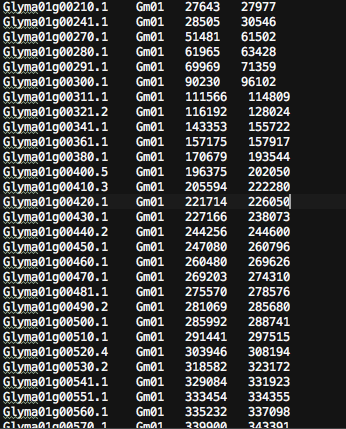


These files contain multiple headers. Each header specifies start/end genes for a syntenic sequence (eg. 3012974 – 3013249) for a particular chromosome of a species (in this case Chr05N). These start and end genes will be matched up with geneIDs/names in the following lines (eg. 3012974 corresponds to geneID Pavir.5NG498000.1).

.gff3/.gff files- The gff files match up gene ID’s to their corresponding coordinates/positions in a particular chromosome in the .fa genome file. Typical formatted as such:



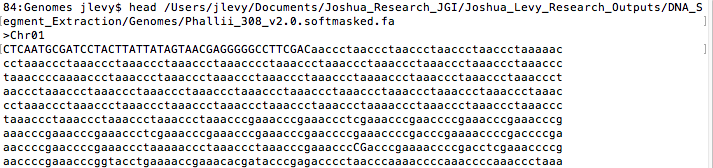
.sort2 files- The gff files **MUST** be parsed using gff2sort to generate .sort2 files. These files look like:



This is a simplified version of the .gff file and is used for a fast and speedy final analysis. Each line is organized by

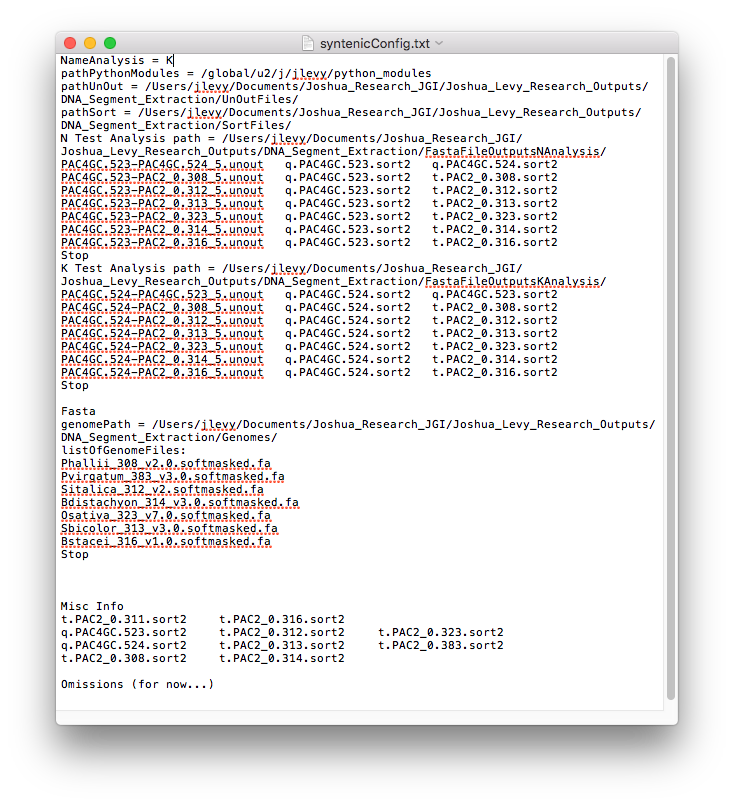
geneID ChromosomeName startCoordinate endCoordinate of gene. The genes are referenced in the unout file and found in the sort2 files during the analysis. More info on sort2 to come…

.fa files- These are the files that contain the entire sequence of the genome for particular species. Looks like:



These files will be turned into python Fasta objects during the analysis and their syntenic sequences will be exported into another fasta file

configuration text .txt file- File used to set up entire analysis. I have named my file syntenicConfig.txt and if you wish to have a different file name, please edit it within the python script for syntenyFinal.



More info on setting up config file to come…

.py- Run 2 scripts on python 2.7 to set up analysis.

Scripts are syntenyFinal and gff2sort.

**Types of Files Output from Analysis:**

.fasta- Similar to .fa, but these are exported syntenic sequences. These will be run through a Multiple Sequence Aligner for final analysis. 

Format may be different in future script, but its headers contain

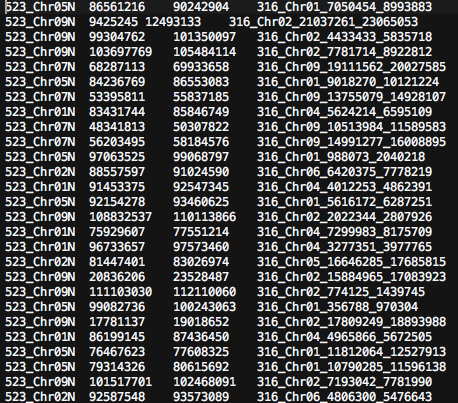
SpeciesName Chromosome Start\_Coordinate EndCoord

Followed by the actual syntenic sequence itself. Each fasta file contains similar syntenic sequences from many different species. Each file specifies one particular “type” of synteny, and the pairwise comparisons are combined to form these files…

.bed- See Bedtools for more info on how it works. <http://bedtools.readthedocs.io/en/latest/>

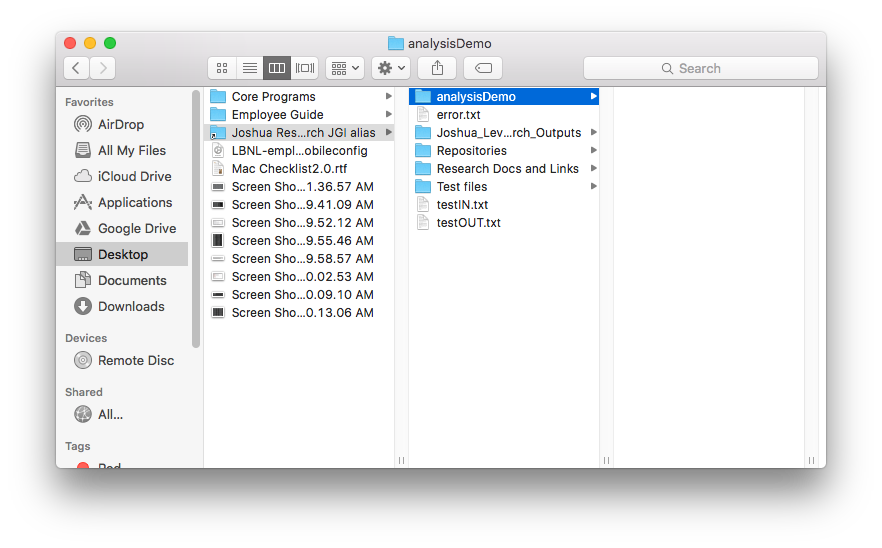
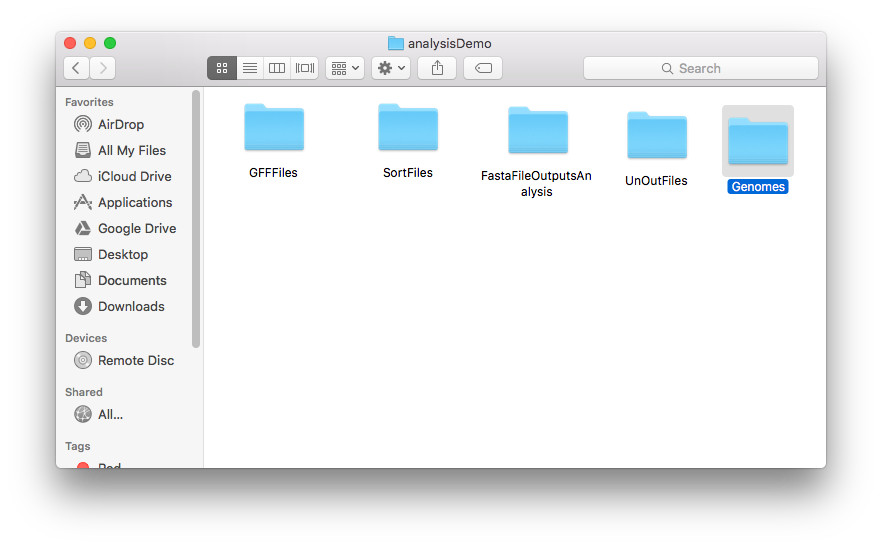
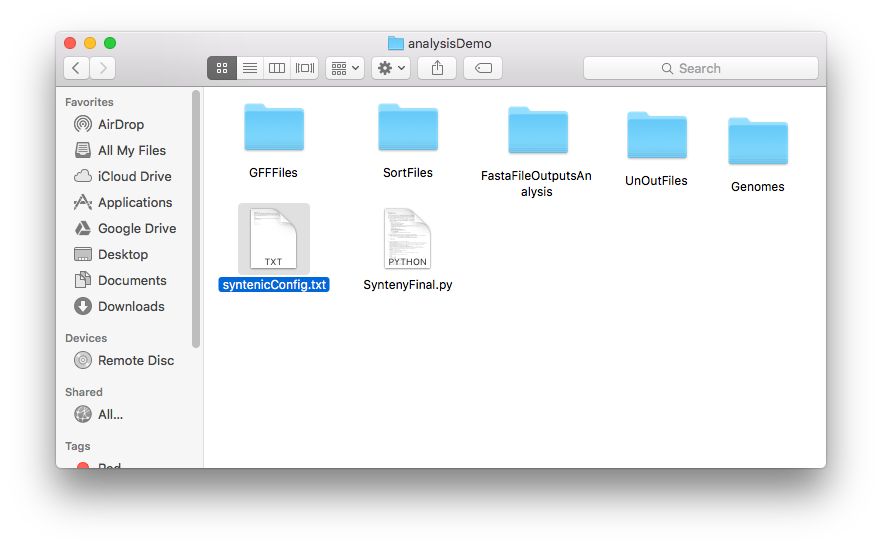
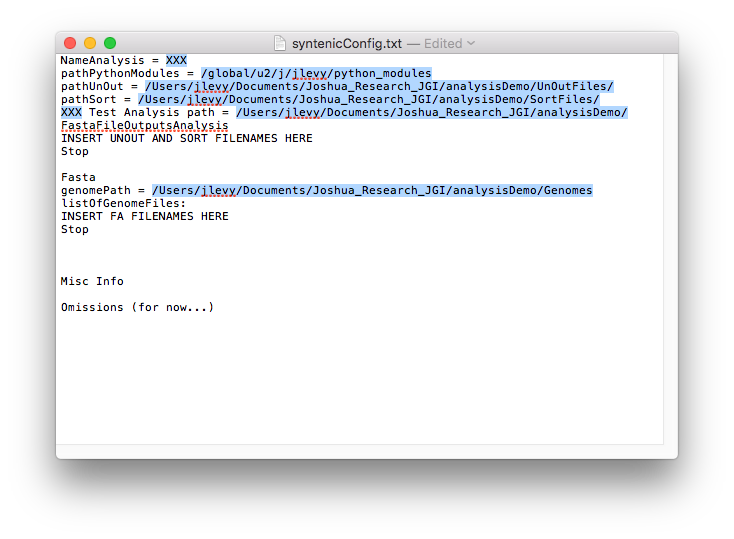
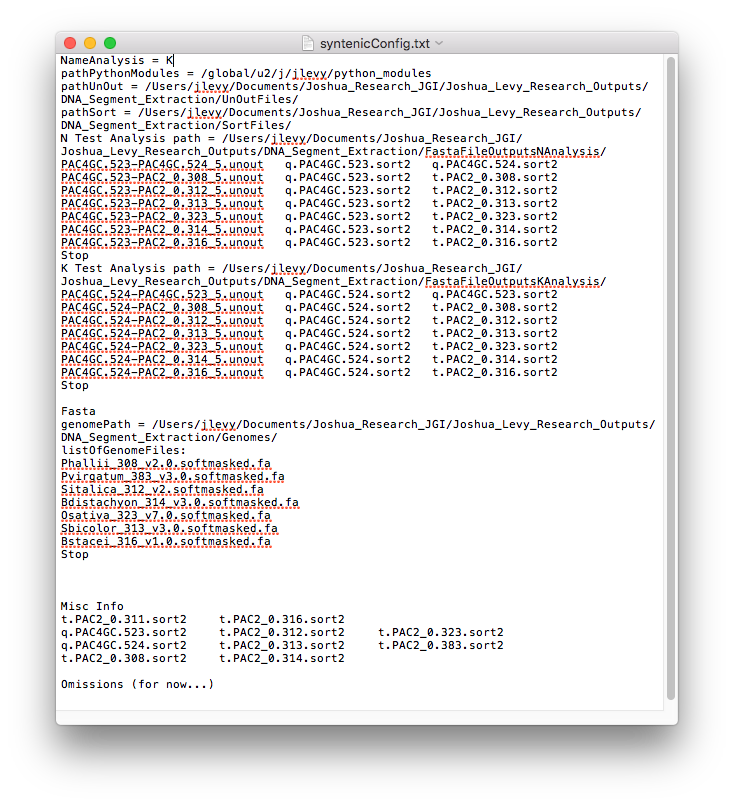
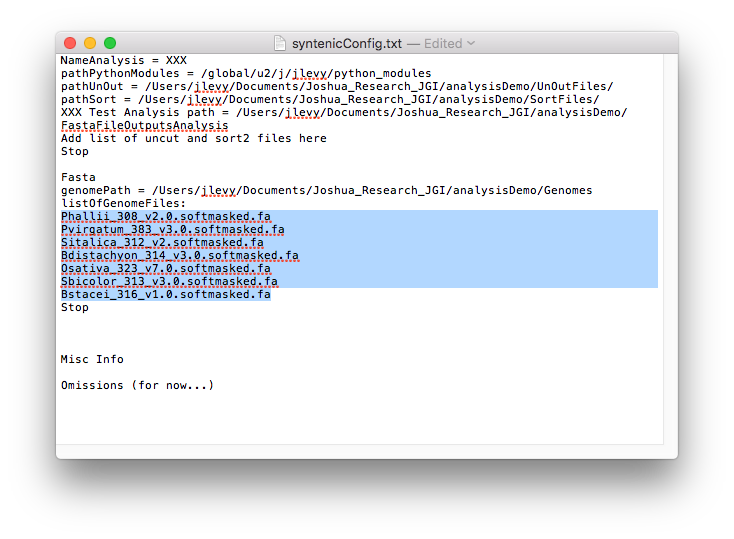
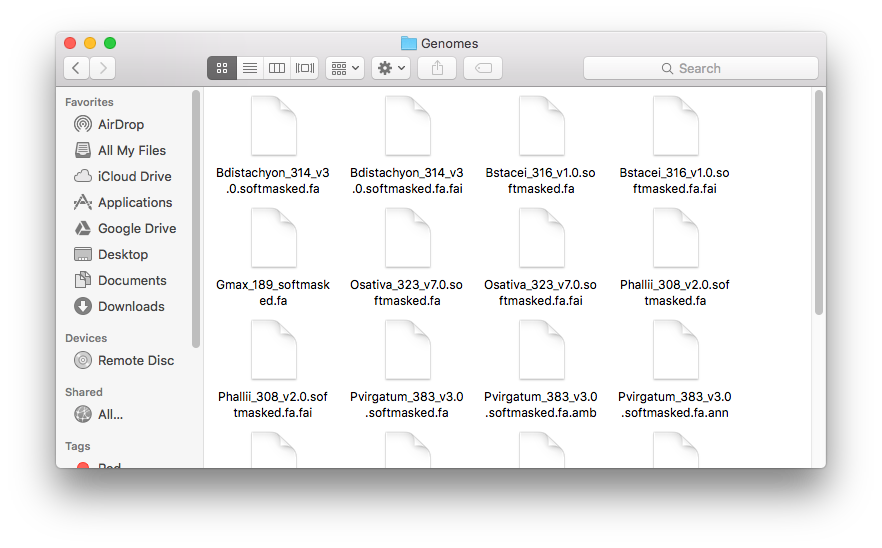
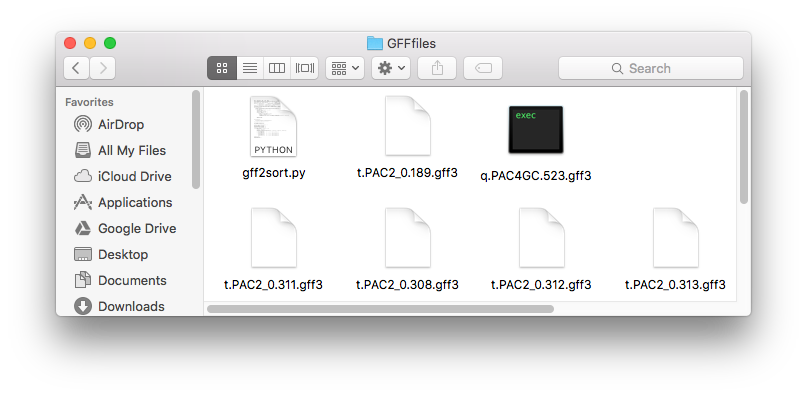
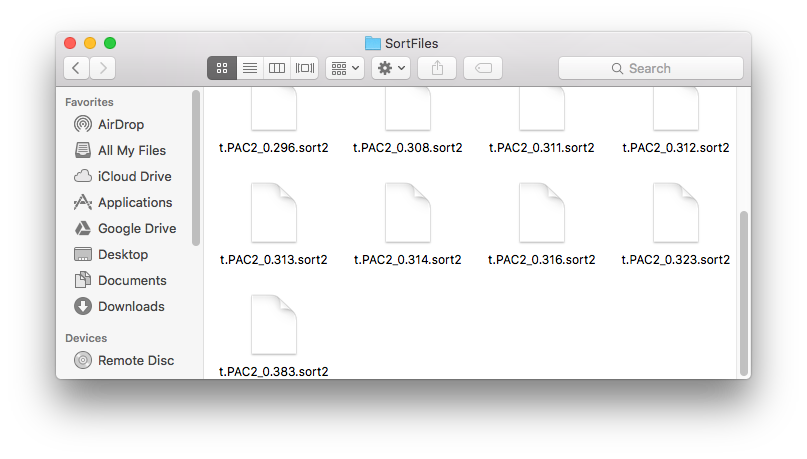
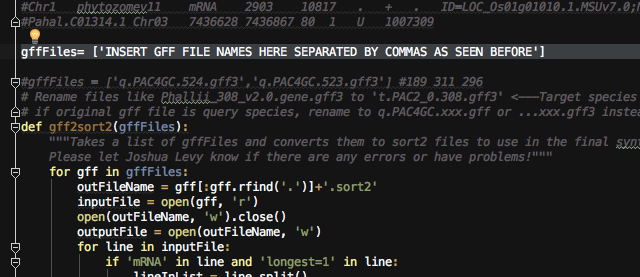
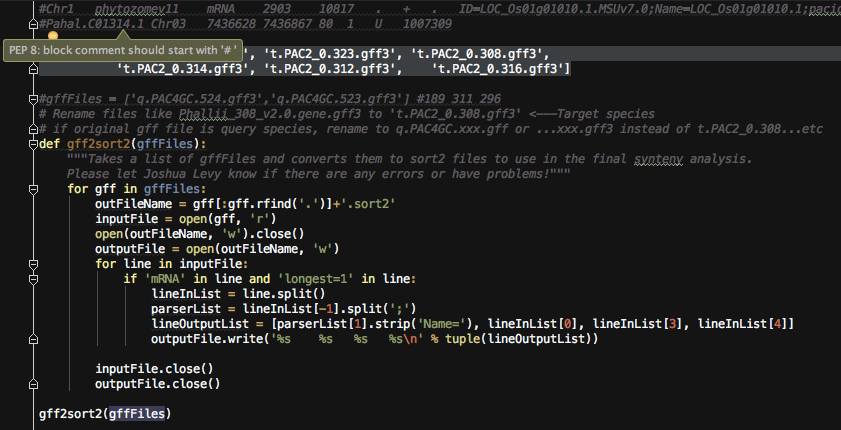
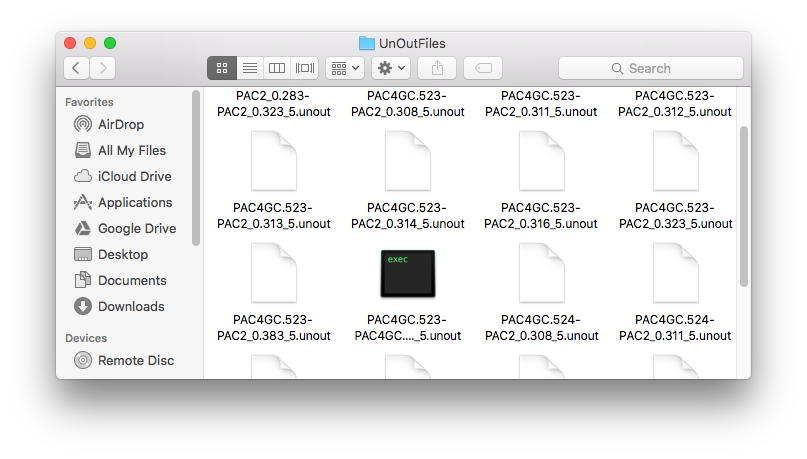
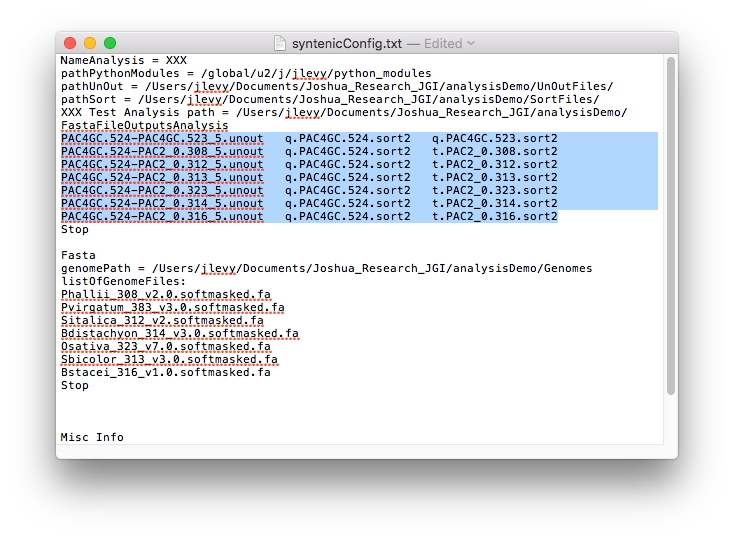
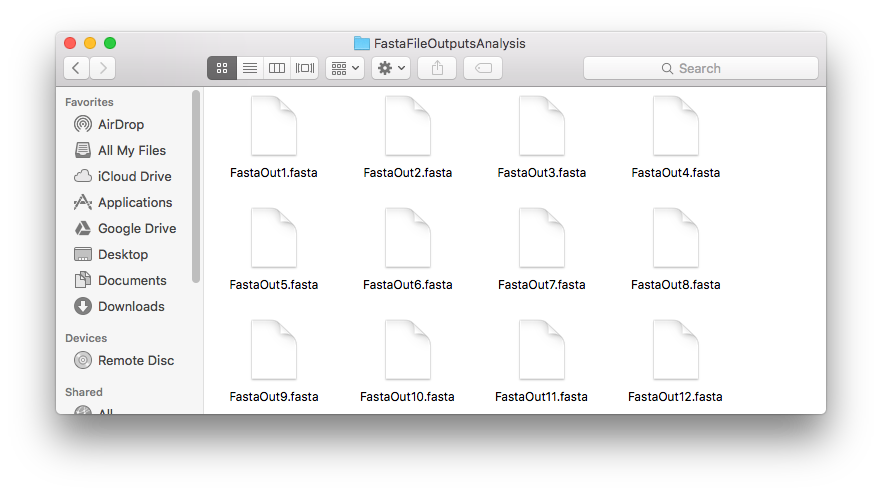
pybedtools was used for analysis… Used for outputting fasta files above..

First three items on line describe query species, fourth describes target..



errorFile (ERRTEST.txt)- error file generated if cannot generate some of the fasta files. Will list species\_chromosome start and end coords of bad output.

**How to set up file structure, folders and config for analysis:**

1. Setup main folder where analysis will be performed. In this case lets call it analysisDemo. 
2. Inside analysisDemo, create the following folders… 
3. Copy syntenicConfig.txt, syntenyFinal.py over to analysisDemo and open up syntenicConfig.txt (config file)…
4. Edit the paths listed to reflect your analysis. Edit pathPythonModule (the path where to find modules you have installed so it can run the right libraries), pathUnOut, pathSort, XXX Test Analysis path, genomePath… Also, change name from N Test Analysis to XXX(your analysisname) Test Analysis and replace N with XXX for analysisName= XXX. XXX will also be input argument to the main function in syntenyFinal. You can erase all other info from any other analysis (N or K in this case)…
5. Drag .fa genome files to the genomes folder and add names of genome .fa files to syntenicConfig.txt (the .fai files in the picture will be generated from the analysis Fasta structure). Species number (eg. 383 should be surrounded by two ‘\_’ for file name)
   1. Use softmasked genome assemblies
6. Drag/copy gff2sort.py and all .gff files to folder GFFfiles and rename by the following conventions:
   1. Rename file with either ‘q.’ or ‘t.’ followed by ‘PAC4GC.’ Or ‘PAC2\_0.’ Followed by species name/number (eg. 383, needs to correspond to species number on .fa) followed by .gff or .gff3.
   2. Example: t.PAC2\_0.383.gff3 corresponds to Pvirgatum\_383\_v3.0.softmasked.fa
   3. 524 and 523 are both special cases… Try to follow usual naming convention
7. Edit gff2sort.py by changing the list variable gffFiles. You will erase the contents of the variable gffFiles and type in the name of the gff files you need to convert to sort2 files. Save and run this script from the terminal. .sort2 files will be created, and copy all of them over to the SortFiles folder.
8. Copy all unout pairwise Synteny comparisons files to the UnOutFiles folder.
9. Finally, edit syntenicConfig.txt once again. For each pairwise synteny, write the name of the unout file, followed by 3 spaces ‘ ‘ followed by the query species sort2 filename followed by 3 spaces ‘ ‘ followed by the target species sort2 filename (eg. PAC4GC.524-PAC2\_0.323\_5.unout q.PAC4GC.524.sort2 t.PAC2\_0.323.sort2).
   1. There should be one synteny per line, and no lines that do not have any content.
   2. There should be lines of these unout and 2 sort2 files from the line containing the analysis output path all the way down to the line containing Stop.
   3. Order the lines such that the target species that is the closest relative to the query species is highest on the list.
   4. Notive how in the previous example the unout is from 524-323\_5.unout (Please keep the \_5 in the name). The first sort filename should contain 524 and the second should contain 323 in this case for query and target.
10. Go to the main directory, analysisDemo, and run syntenyFinal.py. All fasta exports will be found in the FastaFileOutputsAnalysis folder.
11. NOTE: See the comments in the code for SyntenyFinal for an idea of how it all works and the logic behind it. Email Joshua Levy ([joshualevy44@berkeley.edu](mailto:joshualevy44@berkeley.edu)) for any questions. Please ask if going to do an analysis on a polyploid…
12. NOTE: Please install python modules numpy, pandas, pybedtools, pyfaidx and install a C compiler, fortrann 77 compiler, and BedTools for full functionality.