Overview:

1. Get sequencing data
2. Get the loci were interested in
3. Extract CDS
4. Generate multiple sequence alignment
5. Get predicted common ancestors
6. Realign to predicted common ancestors
7. Run MKT

Getting setup:

Organization:

1. Create a file system to keep the data systematically organized.
2. Create readme files to keep track of what you did for each step and to keep track of software versions.

Software to install:

1. A text editor - I use sublime, it's free and you can just say no when the window pops up asking you to buy it occasionally - https://www.sublimetext.com
2. Install anaconda if you haven't yet - https://www.anaconda.com/products/distribution
3. Use anaconda to download the software we will need
    a. Bwa (https://github.com/lh3/bwa) - `conda install -c bioconda bwa`
    b. PRANK (http://wasabiapp.org/software/prank/) - `conda install -c bioconda prank`
    There are extra dependencies to download for prank – exonerate, MAFFT, and BppAncestor - conda has exonerate and MAFFT
    Follow the instructions on the prank website to download Bpp scroll down to Optional: installation of MAFFT, Exonerate, and BppAncestor– http://wasabiapp.org/software/prank/prank_installation/
    Follow what they say in that code block for installing and compiling the bppsuite. Open terminal, go to that programs folder you created in your home directory and then start where is says "mkdir bppsuite"
    c. Biopython (http://biopython.org) - `conda install -c conda-forge biopython`
    d. R - `conda install -c r r`
    e. Rstudio - `conda install -c r rstudio`
    f. X Bazam (https://github.com/ssadedin/bazam) - `conda install -c bioconda bazam`
    g. Genome Analysis Toolkit (GATK) - `conda install -c bioconda gatk`
    h. Picard (http://broadinstitute.github.io/picard/) - `conda install -c bioconda picard`
    i. Paml with conda - https://anaconda.org/bioconda/paml

Analyses:

***bam* from *D. americana* Ahmed lab**

Yasir sent .bam files which are raw sequencing files that have already been aligned to a reference genome. Therefore, these alignments contain genome coordinate information specific to the genome he aligned them to, and not the *bam* reference he sent us. I think the easiest thing to do is realign these reads to just the *D. americana bam* region he sent. There are also other species in there for which we can get the *bam* region from ncbi and align as well.

Alignment and variant calling pipeline:

https://gatk.broadinstitute.org/hc/en-us/articles/360035890411?id=3893

1. Create a folder for this analysis and put the reference files and compressed .bam files here.
2. Extract the compressed .bam files with tar - `tar -xvf archive.tar.gz`
3. Use bazam to go from .bam to .fastq (raw read format that comes off of the sequencing machine).
4. Realign to *bam* region with bwa mem

** At this point, can choose to use geneious to visualize the alignment, call variants, and generate fasta files for each individual sequence instead of continuing below with GATK. For a few loci from genomes that have already been analyzed and quality controlled (like this one) this is a nice option.**

5. Sort and Mark Duplicates with Picard (this will have to be run either individually for each file, or you can look at the amr_bam_realign.sh script I wrote and the readme file to create a for loop that sorts and marks duplicates over the files in a directory - https://gatk.broadinstitute.org/hc/en-us/articles/360036510732-SortSam-Picard- https://gatk.broadinstitute.org/hc/en-us/articles/360037052812-MarkDuplicates-Picard-
6. ~~base recalibration gatk~~ we dont do this step because our sample is too small (single gene) https://gatk.broadinstitute.org/hc/en-us/articles/360035890531-Base-Quality-Score-Recalibration-BQSR-
7. Call variants with gatk HaplotypeCaller - https://gatk.broadinstitute.org/hc/en-us/articles/360037225632-HaplotypeCaller
8. Aggregate the individual vcfs together for joint genotyping - https://gatk.broadinstitute.org/hc/en-us/articles/360035889971

9. ~~Variant recalibration with gatk https://gatk.broadinstitute.org/hc/en-us/articles/360035531612-Variant-Quality-Score-Recalibration-VQSR~~ also not going to do this step because we dont have a training set of true variants. Because we are going to filter out low frequency polymorphisms from our sample I think this isn't a problem.

Create individual fasta files from joint vcf file:

1. Install bcftools from conda
2. Install tabix from conda
3. We're using a modified version of this pipeline - https://samtools.github.io/bcftools/howtos/consensus-sequence.html
    a. Index the gziped VCF file with tabix -
        i. `tabix -p vcf yourfilename.vcf.gz`
    b. Use your reference genome file you've been using and the text file youve been using with your sample names to get individual fasta files with alternate alleles. Remember that both the reference file and your vcf file must be in the same directory with all of their indexed files -
        i. Example for a single sample -

```
cat reference.fa | bcftools consensus yourfilename.vcf.gz -s samplename -o samplename.bam.fasta
```

        ii. You can write a for loop to run this across all of your samples i.e.

```
For f in `cat samplenames.txt`
do
   cat reference.fa |
   bcftools consensus yourfilename.vcf.gz \
   -s $f \
   -o $f.bam.fasta
done
```

4. That should get you individual fasta files for each americana sample!

Get CDS and align:

I put the whole gene region fasta, the CDS fastas and a single fasta file with all of the sequences for ame-nov-vir-lum bam in google drive. So you can use the combined file to move forward with step three here if you want, but everything is there in case you want to go through the individual sequences too

1. Get CDS
    a. option 1: open your files in geneious, and use the americana gff file with the annotations to annotate the sequences. Then extract only the CDS for each one.
    b. option 2: use the gff parser script to extract the CDS from each file. If you do this, its still a good idea to look at the files in geneious and make sure that everything looks okay (the CDS looks correct, no early stop codons etc.)
2. Create a single fasta file containing the sequences you want to align for the MK test and to get the predicted common ancestor sequences. Typically, you need at least one population sample, and then at least two outgroups to generate a predicted common ancestor. You can do this by just highlighting all of the sequences you want in geneious and then choosing file>export>documents>fasta or if you already have individual files in a directory you can use the unix command cat i.e
   `cat fasta1 fasta2 fasta3 > multifasta`

3. Get a file with the unique sequences only - In order to use codeML to get the predicted common ancestor sequence, we need to input an alignment and a tree file. We can only use unique sequences, because codeml will throw an error if the tree has a branch length = 0.
    a. Use get_unique_codeML10-05-22.py to input your multifasta file and a fasta file with only the unique sequences.
4. Align the sequences in the unique fasta file with prank. If you type prank in your terminal you can see all of the options. These are the ones I usually use for a coding sequence alignment –

```
prank -codon -F -showtree -d=input.fasta -o=outputname
```

Prank will take a while to run, but you will get a file with your alignment, and a treefile. This is your input for codeml

5. Run codeml to get the predicted common ancestor sequences:
    a. Codeml takes a control file as input – i put a sample of one I used in the codeml folder https://drive.google.com/open?id=1HWRXuZQXdlX7o8r-JiTrfHlYj2mBXbDk&authuser=jeb486%40cornell.edu&usp=drive_fs
    b. You can leave all of the settings as they are (you can read more about them, most of them are not very important here because we are only running codeml to get the predicted common ancestor, not to look for evidence of positive selection), but change the sequence, tree, and output for your alignment, alignment tree, and the output name you want.
    c. You can include a path name with the sequence and tree file in teh control file, or put the control file in the same directory with the tree and sequence. Then, in terminal, cd to the directory the control file is in, and just type in codeml. Codeml will automatically look for the control file and the sequence and tree files you directed it to in the control file.
    d. Codeml also takes a while to run!
6. Youll get a file “rst” as output, along with a bunch of other stuff that we dont need. I put an example of the one generated with the control file in google drive. Getting what we need out of this file is kind of clunky and annoying at the moment –
    a. Open the rst file in sublime, its large so itll take a minute to open
    b. Scroll down to where it says “tree with node labels for Rod Page's TreeView” and copy and paste the text of the tree into a new text file. I put an example of this from the rst output example in the drive folder as well so you can take a look.
    c. Then use geneious to visualize the tree and find the node name of the common ancestor(s) you want to use for the MK test. In this case, I think getting the common ancestor of lummei and americana/nov would be good because it has virilis as an outgroup and lummei and ame have been diverged for about 5 million years (similar ot mel and sim).
    d. Then go back to the rst file and do command F to find “node # [your number]”.

This should get you to the common ancestor sequence. Then, copy and paste this into a new text file and save it. I also like to either open this text file in geneious or paste the sequence into geneious and save it from there to fix the weird formatting in the rst file.

e. Now we want to realign the common ancestor with all of our data (not just the unique sequences) using prank. So make a new multifasta file that contains all of the americana sequences and the new predicted common ancestor (you can include the other species if you want, but we dont necessarily need an alignment with them for the MK test) and run prank again on this file.
f. Now we can use this as input to the MK test – http://mkt.uab.es/mkt/MKT.asp
    i. Eventually well use the iMKT in Rstudio, but this is the test ive been using.
    ii. Go to the main parameters tab, check “exclude low frequency variants” and then adjust the number to 12.
    iii. Uncheck “align sequences”
    iv. Then go back to the main tab, and using the alignment output from prank - copy and paste your prank aligned americana sequences into one box, and the single predicted common ancestor in the other box and then hit analyze! (>node34_Dame_Dlum_bam_anc, Dame_OC.bam)