Small Group Exercise #5:

Annotation of genomic and transcriptomic assemblies and parsing of annotation reports

1. In this exercise, annotation of both a genomic and transcriptomic assembly from Lamellibrachia luymesi will be conducted with the Trinotate pipeline. First, create two working directories in your home directory:

    i. `cd` (to take you to the top level of your home directory)
    ii. `mkdir genomic_trinotate` (create the first directory)
    iii. `mkdir transcriptomic_trinotate` (create the second directory)

2. Now copy already finished assemblies from Ray (genomic) and Trinity (transcriptomic) for *Lamellibrachia luymesi* to each of the directories. We will be utilizing these "canned" assemblies rather than the ones assembled yourself so that everyone is working with a standardized set (remember the variation in the assembly assessments from exercise #3?)

    i. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_finished_genomic_transcriptomic_assemblies/Lamellibrachia_luymesi_all_genomic_RAY_05_2015.fasta ./genomic_trinotate/` (to copy the genomic assembly to its working directory)
    ii. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_finished_genomic_transcriptomic_assemblies/Lamellibrachia_luymesi_sub1M_NON_NORM_TRI_05_2015.fasta ./transcriptomic_trinotate/` (to copy the transcriptomic assembly to its working directory)

3. You will need to make a copy of the `Trinotate_example.sh` script to each of the working directories. Go ahead and do that now:

    i. `cp /home/shared/biobootcamp/data/example_ASC_queue_scripts/Trinotate_example.sh ./genomic_trinotate` (to copy the script to the first working directory)
    ii. `cp /home/shared/biobootcamp/data/example_ASC_queue_scripts/Trinotate_example.sh ./transcriptomic_trinotate/` (to copy the script to the second working directory)
    iii. `cd genomic_trinotate/` (change into the genomic working directory)
    iv. `nano Trinotate_example.sh` (to view the contents of the script)
    v. Note that while comments in the script indicate that nothing needs to be tweaked, make note of the parameters that you will supply to the queue system for running the script.

4. Now submit your `Trinotate_example.sh` script for the genomic assembly to the ASC queue system.

5. Check that the job is in the queue using `squeue`. Once it starts, do a “long listing” within the directory. You’ll see that a new directory as well as a `*.log` file have been created. View the contents of the log file:

    i. `cat *.log`
    ii. What information is in the log file?

6. Now change directories to the one with the transcriptomic assembly (using `cd ../transcriptomic_trinotate/`) and submit your `Trinotate_example.sh` script for that assembly to the ASC queue system just as you did for the genomic one.

7. Check the status of both jobs in the queue using `squeue`.

8. NOTE: you might want to know what the `auto_Trinotate.sh` pipeline looks like. To view the contents of the script:

    i. `cat /home/shared/biobootcamp/bin/auto_Trinotate.sh`
    ii. Spend a few minutes reading the content of the script line-by-line. While it looks complicated, the comments included in the script (which start with ###) should guide you through the process. Its good practice to always include such comments in scripts that you write either for your own sake or that of others.

9. Since the Trinotate pipeline takes some time (~12-24 hours) to complete, we will move on to using pre-built, but “raw”, results from the pipeline.

    i. `cd` (to take you to the top level of your home directory)
    ii. `mkdir trinotate_genomic_transcriptomic_results` (create a working directory for the results)
    iii. `cd trinotate_genomic_transcriptomic_results` (change into the working directory)
    iv. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_Trinotate_annotation/Lamellibrachia_luymesi_Trinotate_annotation_raw/Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_May2015.tar.gz .` (to copy the raw Trinotate results for the genomic assembly to the current directory)
    v. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_Trinotate_annotation/Lamellibrachia_luymesi_Trinotate_annotation_raw/Lamellibrachia_luymesi_sub1M_NON_NORM_Trinotate_May2015.tar.gz .` (to copy the raw Trinotate results for the transcriptomic assembly to the current directory)

10. Based on the pathname of the files, the two (2) `*.tar.gz` files are considered “raw” since they each represent a compressed archive containing the individual output files from the Trinotate pipeline. To see what this means, unpack one of them:

    i. `tar –xvzf Lamellibrachia_luymesi_sub1M_NON_NORM_Trinotate_May2015.tar.gz` (to unpack any .tar.gz archive)
    ii. Now do a “long listing”. You see that a new directory has been created from unpacking the archive. Do a “long listing” to see what is in there:
    iii. `ls –al Lamellibrachia_luymesi_sub1M_NON_NORM_Trinotate_May2015/`
    iv. There should be ~10 files listed, with a variety of file extensions. Those with “transdecoder” in the file names are output from extracting the potential open reading frames (i.e., ORFs or protein-coding regions) such as the amino acid sequences themselves and their coordinates within respective contigs. You’ll also see that there are copies of the BLAST+ reports and PFAM output. Feel free to explore the contents of these files using the `head` command to see the first 10 lines in a file:
        a. `head FILENAME`

11. Now let’s generate the final Trinotate reports from the archives. Within the directory containing the two (2) `*.tar.gz files`, execute the following command:

    i. `generate_Trinotate_report.sh`
    ii. It will take 2-4 minutes to complete, with text scrolling across your screen during the process (try to follow the process by reading the text as it goes by). Once complete, the command line prompt will return.

12. Do a “long listing” in the directory and note that there are now two (2) files ending in a `*.tab` extension. These are our finished Trinotate reports containing annotation in tab-delimited fields/columns for the genomic and transcriptomic assemblies of Lamellibrachia luymesi. While you could open these files in Microsoft Excel, use the `head` command to see the first 10 lines of each file and note that the first line (starting with a #) is a header detailing what data are in each field/column.

Now that we have the two reports as plain text files, we can utilize our arsenal of Linux commands to begin analyzing them towards identifying potentially interesting biological trends.

13. Remember the concept of “controls”. In this case, one logical control is to check the number of entries in our final Trinotate reports against the number of sequences in the FASTA files that were annotated, the hypothesis being that they are equal to each other (i.e., one annotation entry to one FASTA entry). For this, we will need copies of the FASTA files that were submitted to Trinotate. Get them by:

    i. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_finished_genomic_transcri`

`ptomic_assemblies/Lamellibrachia_luymesi_500bp_plus_genomic_RAY_05_2015.fasta .`

ii. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_finished_genomic_transcriptomic_assemblies/Lamellibrachia_luymesi_sub1M_NON_NORM_TRI_05_2015.fasta .`

14. Next, quantify the number of FASTA entries in each of the files that you just copied using the same technique from your pre-Bootcamp assignment and write those numbers down. HINT: you'll use "grep" to count a regular expression that is unique to each of the FASTA identifiers.

15. Now that we have those, quantify the number of lines in each of your Trinotate reports using the `wc -l` command, which you should remember from the on-line tutorial that you did as part of the pre-Bootcamp assignment. Subtract one from the number that you get due to the header also being counted.

    i. Do the number of entries in FASTA files exactly match the number of lines in their respective Trinotate reports?

16. Let's dig into this a little deeper, focusing on the smaller of the two datasets, the genomic assembly with file names starting with `Lamellibrachia_luymesi_500bp_plus_genomic_*`.

    i. `head -1 Lamellibrachia_luymesi_500bp_plus_genomic_RAY_05_2015.fasta`

    ii. `head -2 Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab`

        a. What did you do in the above with the option flag to `head`?

    iii. From the header and first entry of the Trinotate report, identify which of the first two (2) fields/columns match the first entry in the FASTA file EXACTLY. Once you have that number, insert it in place of the "X" in the command below and execute it in your Terminal (the command should be written as one single line):

        a. `awk -F"\t" 'NR>1 {print $X}' Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | head -1`
        b. Does the output EXACTLY MATCH the descriptor from the first entry of the FASTA file (expect for the ">")? If so, good. But:
        c. What is the "NR" in the awk command doing? Consult your awk cheat sheet for details on "NR". HINT: pesky header.
        d. Why did you pipe the output of the awk command to head? If you are not sure, bring up the previous command and rerun it without the pipe and head (i.e., "| head -1"). What happened? Why was `head` useful in this case?

    iv. Now lets quantify how many unique entries are in the column from above. We will "recycle" most of our previous command and "tweak" it further (again, the command should be written as one single line):

    a. `awk -F"\t" 'NR>1 {print $X}' Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | sort | uniq | wc -l`
    b. What additions did we add to pipeline and more importantly, what are these additions doing? For more details on each of the commands in this pipeline, consult `man cmd_name`.
    c. Now check the number returned from this latest pipeline against the number you received from quantifying the number of descriptors in the FASTA file. Do they now match?
    d. So what lead to the discrepancy in the numbers from each file? What does this tell you biologically regarding the protein-coding information within your assembled genomic contigs? HINT: According to the header, what are the titles of $5 and $6 in the Trinotate report?

The above provided insight regarding how the Trinotate (or any other annotation) report might be structured. Now let's examine the information in the report itself. Having good annotation provides one "control" in regards to what was actually sequenced. Remember, sequencing technologies like Illumina are non-specific, meaning that any viable nucleic acids that were extracted could have been sequenced and might appear in downstream analyses. This includes nucleic acids from non-targeted organisms, such as prey items that were consumed or ones living on or in the organism itself. Let's see if we can find potential cases of this in the transcriptomic and genomic annotation of Lamellibrachia luymesi.

17. Let's do a quantification of the regular expression "bacteria" across our Trinotate transcriptomic report. Since we are not sure if the spelling is "bacteria" or "Bacteria" in the report, let's err on the side of caution and make our search case-insensitive. So, replace the "X" in the command below with the appropriate flag for such a case-insensitive search (ask among your group if you aren't sure what it is):
    i. `grep –c –X bacteria Lamellibrachia_luymesi_sub1M_NON_NORM_Trinotate_Annotation_Report_May2015.tab`
    ii. Now quantify the total number of entries in the Lamellibrachia_luymesi_sub1M_NON_NORM_Trinotate_Annotation_Report_May2015.tab report using the `wc –l` command (remember to subtract one from the number that you get due to the (pesky) header also being counted)
    iii. What is the ratio of "bacteria" to total annotation entries in the Trinotate transcriptomic report?
    iv. Now repeat the above search and ratio calculation for the Trinotate genomic report (i.e., Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab).
    v. Once you have your numbers, discuss among the group the following questions:
        a. How similar or different are the ratios calculated from the transcriptomic vs. genomic assemblies and their annotation reports?

    b. Why might the ratios be so different and what does this tell you about the methodological approaches? HINT: when generating genomic vs. transcriptomic data
    c. Lastly, is this difference in ratios of significant concern when viewed in light of the biology of the targeted organism Lamellibrachia luymesi? HINT: think back to the opening lecture of the Bootcamp….

18. Let's see if we can get a finer-level taxonomic identification for these "bacteria". What does an entry containing this regular expression look like?:
    i. `grep –i bacteria Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | head -1`
    ii. Examine the entry that is returned, looking for 1) instances of "bacteria" and 2) what field(s)/column(s) they occur in. Unfortunately, this output is a little hard to read. So let's fix that:
    iii. `grep –i bacteria Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | head -1 | xargs -d "\t" -n 1`
        a. What is `xargs`? Explore the man page. HINT: doesn't it look like we broke a single line into a single column? Now it easier to read (pretty cool, huh, and that is just one of the many things xargs can do).
    iv. It looks like there are two (2) columns with our regular expression (why are there two columns in the first place? HINT: check the header names for those columns). Let's focus on the first of them. We will also be making our search more specific by restricting it to 1) "Bacteria" (note the capital "B" and removal of the "–i" option below) and 2) adding filters to remove any potential "bleed-through" of "Eukaryota, Viruses and Archaea". Insert the number of the field/column identified above in Section 18.iii in place of the "X" in the command below and execute it in your Terminal:
        a. `grep Bacteria Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | egrep -v "Eukaryota|Viruses|Archaea" | awk -F"\t" '{print $X}' | awk -F"^" '{print $7}' | less`
            a. As you scroll through the output in "less", did we accomplish what we set out to do? Press "q" to exit "less" when ready
            b. Discuss among the group 1) what the `egrep` command is doing and 2) why there is, and what is different about, the second awk statement. To better understand "egrep", "man egrep" and read the "DESCRIPTION" and "OPTIONS" sections of the manual page. Also try repeating the above command, but without the inclusion of "egrep" portion of the pipeline. As for the second awk statement, compare the –F option to that of the first awk statement for clues.
        b. Now add a `wc –l` in place of the `less` command to quantify the number of entries matching our regular expression. Compare this to the number you got from your

previous `grep –i –c bacteria` quantification. What do the differences in the numbers represent? Lastly, remember that you can always redirect information like this to a file for later use if you needed to (How?).

v. Let's rerun the command in Section 18.iv.a above and look at the output. The information in field/column #3 seems to be informative from a taxonomic perspective. How about extracting that field/column and quantifying it:
   a. `grep -i Bacteria Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | egrep -v "Eukaryota|Viruses|Archaea" | awk -F"\t" '{print $X}' | awk -F"^" '{print $7}' | awk '{print $3}' | sort | uniq -c`
   b. Talk among your group about the results and how you got to this point. Specifically:
      a. Why was `sort` required before using `uniq –c`?
      b. What did `uniq –c` actually do in this case?
      c. What about the blank entry in the output? Where did that come from and how do we get rid of it? HINT: try piping the output from the third awk command through `sed '/^$/d'` before piping it to `sort | uniq –c`. The command would look like: `grep -i Bacteria Lamellibrachia_luymesi_500bp_plus_genomic_Trinotate_Annotation_Report_May2015.tab | egrep -v "Eukaryota|Viruses|Archaea" | awk -F"\t" '{print $3}' | awk -F"^" '{print $7}' | awk '{print $3}' | sed '/^$/d' | sort | uniq -c|sort -n`
      d. What did sed do to that blank entry and why (i.e., what does the command mean)?
      e. Lastly (and most importantly): do the results make biological sense given what is known regarding Lamellibrachia luymesi? Spend a few minutes in Google researching the biology of the species and discuss your findings with the group.

This short exercise should serve as an example of how powerful Linux and its tools can be when dealing with thousands to millions of data points. However, other take home messages from this exercise are:

1. Analyses should be conducted in a human-logical manner. While the computer does the heavy lifting, it's the human that needs to direct where the analyses are going (and why).
2. Check, double-check and triple-check what you are doing, especially when applying filters to data. While the filter might look perfectly fine to you, the computer may interpret it in a very different way. In that case, the computer is 99.999% right since it's doing exactly what you told it to do and you, the operator, is the error. Don't blame the computer if you have to retract a dataset that you didn't carefully quality control.
3. Most importantly, don't lose sight of the importance regarding understanding the biology of your organism. Otherwise, no amount of data or analyses will make sense at the end of the day.

NOW, IF TIME PERMITS, IS "FREE-PLAY". PRACTICE WITH THE TOOLS THAT YOU HAVE

BEEN INTRODUCED TO IN THIS SECTION BY SEARCHING THE TRINOTATE REPORTS FOR INFORMATION YOU ARE INTERESTED IN. THIS COULD BE YOUR FAVORITE GENE(S) OR OTHER DATA. AS MENTIONED BEFORE, THE EASIEST WAY TO LEARN THESE MATERIALS ARE BY PRACTICING THEM. BOUNCE IDEAS AMONG THE GROUP TO SPARKE NEW IDEAS. FOR EXAMPLE, ONE COULD COMBINE ASPECTS OF THIS EXERCISE WITH THE LAST PART OF EXERCISE #3 TO EXTRACT ALL OF THE BACTERIAL CONTIGS FROM THE GENOMIC ASSEMBLY BASED ON THE ANNOTATION REPORTS. HINT: COMBINE #13-16 ABOVE WITH select_contigs.pl USED IN THE LAST PART OF EXERCISE #3.