Mini-Mouse RNA-seq Tutorial – Jax Mouse Project-Based Learning

The items in BLUE are action items, with focus points in green. In **#hashtag black text is background information** to help you interpret what you are doing. Depending on the timeframe, it may make the most sense to work steadily through the blue action items, and later return to the hashtags – for example during downtime when waiting for data to run.

EXERCISE 1: SMALL DATASET ANALYSIS

Preparation:

A. **Galaxy Public: Setting up a Galaxy Account on Public Server: Go to https://usegalaxy.org/. Click on User in the top row and register an account.**
B. **IGV: Download IGV – Integrative Genomics Viewer: http://software.broadinstitute.org/software/igv/**
C. **Text Editor: You will need a text-editing program such as TextWranger (Mac) or Notepad (PC) or similar program; these are available as free downloads.**
D. **Excel: You will need Excel or similar program.**

# **Galaxy** is an open-source workspace for genomics/bioinformatics work. You have access to a free, limited-capacity (250 GB), online account (you've just created this - you will choose this option for today). Other options: you can download it on your computer (not recommended for novices), you can use your institution's computing cluster's instance of Galaxy if available (or have it installed there), or you can run Galaxy in the cloud. The latter 3 options are ideal for running large datasets as your wait-time will be less and the data-storing capacity will be greater. Both this tutorial and the follow-up Exercise 2 can be run using your **free online account**.

1. Galaxy General Navigation: Click on Analyze Data (top) to get to the main page. Note that you have no History in the History panel on the right. If you ever navigate away from this main page and need to get back, click on Analyze Data again.

# **Navigation**: On the left is a series of tools already embedded into Galaxy; you will select among these to perform various operations and analyses. On the right is your History bar where you'll find all of your analyses. You should create a new history every time you begin with a new data set. Once you load data into your history, you can name it by clicking on the title, writing a name, and hitting **enter/return**. The center space is for visualizing data, working with tools, setting up workflows, and viewing results.

# Once you begin to accumulate more histories, you can click on the **open-book icon** in the upper right to see all of them. To get back to the workspace, hit **Analyze Data** (top row). The **wheel** in the History panel, to the left of the open-book icon, has a drop-down menu with numerous options including **Create New**, which you'll use the next time you generate an analysis.

2. Find Data for RNA-seq analysis: https://usegalaxy.org/u/andrea-tilden/h/mini-mouse-data-jax-1. Upload History (upper right) to move it to your own Galaxy history.

# **Experiment – Source of Data:** These data were taken from an unpublished mouse study in which the Eya2 gene was knocked out. The data are publicly available in ENA-SRA (more details later). Tissue was taken from the retina at midnight in a control and a knockout adult mouse. Later in this tutorial, you will explore the gene and the study further. These data represent a small fraction of the full dataset, curated for this tutorial in a manner such that they can be analyzed from start to finish in the course of one or two lab periods.

# **Experimental Details:** Each tissue dataset pair represents **paired-end reads (forward and reverse)**. Recall that in RNA preparation, the RNA is fragmented – in this particular case, to a mean **fragment length of around 300**. Illumina sequencing parameters: **reads in this dataset are all ~76 bps long** with **a gap (insert) of around 148 bases** in the middle. Knowing the mean fragment length, the length of the read, and thereby the length of the insert, is important in setting parameters downstream in Galaxy. While this information is provided for you here, and is the same for Exercise 2, you would need to find this information when you venture into working independently with other archived datasets; for example, in future studies you may need to track down the accompanying journal article to find this and other important information.

3. Visualize Data: Click on the eye icon of one of the Control or Eya2-KO datasets in the History panel. In the workspace, you'll see a portion of the data (1 MB), representing sequence reads.

# Sequence reads: How many bases long should each be? Are they forward or reverse? This file type is the output of Illumina sequencing and is called a **Fastq file**. It contains the identifying information for the read on the top line, the sequence on the next line, and quality scores for each base on the bottle line below the +, coded by different characters (more info: https://en.wikipedia.org/wiki/FASTQ_format). Remind yourself why these RNA-seq reads contain T instead of U.

4. Visualize Annotations: Click on the eye icon on the iGenomes annotation file in your history (at the bottom). It contains all of the coordinates (start and end) for each individual exon for every known gene on mouse chromosome 5. Note the gene IDs in the Attributes column. This will be used later to attach gene names to all of your reads.

# Gene Annotations: Find **the first gene on chr5**. What is it? How many exons does it have? What is its position on the chromosome? Is it coded by the plus or minus strand? What is the **first gene coded on the *other* strand**? How many exons does it have?

5. Generate Quality Report: In the left-hand tool panel, find the NSG set of tools (Next Generation Sequencing). Click on NGS QC and manipulation. Click on FastQC: read quality reports. In the center workspace, select the multiple doc icon in the upper left under short read data from your current history. All 4 data files should now appear. Highlight all and click Execute. You'll now see gray blocks in your history column, indicating they are cued up. They'll turn to yellow when they're running, and green when finished. Hit the browser reset button after a minute to check on the progress, and keep occasionally doing this until they are yellow and then green.
(Run Time: a few minutes)

# **FastQC:** This tool allows you to look at the aggregate quality of your data. Most Illumina data are quite good, but sometimes the quality of the ends is poor and they need to be trimmed. Recall that the quality score of individual reads is coded by the Fastq format, so the FastQC program allows you to **visualize** the quality without having to comprehend codes and calculate aggregate scores.

6. Visualize Quality Report: Click on the eye icon of the first dataset: x: FastQC on data x: Webpage. Scroll down to the Per base sequence quality bar plot. See below for details and observe the general quality of each dataset.

# Data Quality Bar Plot: Each column represents the range of quality of one of the 76 bases in your reads; green is good/orange is reasonable/red is poor. The y-axis is the phred score, which is coded in your data by the top row as you saw in **step 3**. (for more on phred scores: https://en.wikipedia.org/wiki/Phred_quality_score). The red line is the median, the blue line is the mean, the yellow box is the 25-75% quartile range, and the black whiskers are the 10-90% range. For your purposes, you'll consider anything in which the **median is below 20 (red)** to be poor quality, for which trimming will be necessary.

7. Trim Poor-Quality Data: Go to the Tool bar, to NGS: QC and manipulation, and find Trimmomatic. Select Paired end data (two separate input files), and select your first data pair of forward and reverse reads (R1 and R2 *paired*, not unpaired). Leave all default parameters, and Execute. This will eliminate all of the regions of individual reads that reduce the aggregate score along with their pairs in the other dataset. Go back to Trimmomatic and do the same for the second data pair.
(Run Time: a few minutes)

# ILLUMINACLIP: This step in Trimmomatic is not necessary for this particular dataset, but note the purpose for this step and the options. You will likely use this step when processing large datasets. Most newer data are likely to be MiSeq or HiSeq.

8. Visualize Trimming Results: When the trimmer is finished, go back to FastQC in the Tool panel and now run all of your trimmed data to check on the results of your trimming.
(Run Time: a few minutes)

9. Map Reads: In the Tool panel, go to NGS: RNA Analysis. Find HISAT2. In the workspace, under Single end or paired reads? select Individual paired reads. Next, select a forward and reverse pair for Controls. Use a built-in genome: in Select a reference genome, type mm10 and select Mouse (Mus musculus): mm10. Then Execute. Now do this again for the Eya2-KO dataset pair.
(Run Time: 30+ minutes – a good time to review what you've done and read the #s)

10. Check Progress: During the wait, refresh your browser from time to time. You are likely to get an orange error signal for each of the 2 sets of data ("An error occurred… Set it manually or retry auto-detection"); this is a common occurrence. Click on this, and click Auto-detect; that should quickly fix the problem.

# **HISAT:** This step takes all of the reads in your paired-end datasets, forward and reverse, and aligns them to the mm10 mouse genome build. Each read is ~76 bases long, and the entire mouse genome is around 2.7 billion base pairs (the human is around 3.2 billion). Traditional Sanger sequencing produces reads of 300-1000. What are the tradeoffs regarding both shorter- and longer-read technology? How does paired-end sequencing help to address a problem with Illumina sequencing?

# **HISAT is specialized for RNA-seq alignment to the genome**. If you were looking at Illumina *genome or exome* sequencing data, you would use *Bowtie or BWA* (found in the Tool panel under NGS: Mapping). **The additional challenge with mapping RNA reads against a genome is introns**; can you explain why this is a problem? HISAT uses its parent program **Bowtie** initially to identify reads that do vs. don't align contiguously to the genome. The reads are then split into smaller fragments within the program and re-aligned to identify whether they are likely to span splice junctions.

11. View Mapping Summary: For each HISAT dataset, click on the name to expand the panel, observe the mapping statistics for each dataset, and answer the following questions for each set: How many left and right (forward and reverse) reads were in the input data file? How many (and what percent) were mapped? How many aligned at more than one location in the genome – and why would this happen? How many pairs aligned together? Discordant alignments occur when pairs do not map together as expected (ex. wrong sequential orientation, or not within expected distance).

# **Mapping to mm10:** Keep in mind that while you will only be looking more closely at chr5 (because the dataset was carved from a small region mapping to chr5), you mapped these reads to the entire genome, which is readily available in most Galaxy instances. Next, you'll proceed to **looking at your data mapped to chr5**. In the follow-up advanced exercise, you'll expand to a bigger dataset, *and* you'll look at all chromosomes.

12. Visualize Read Mapping: Now that you've looked at summary data, it is useful to visualize what the tools have actually done with your data. Open IGV (you downloaded this earlier). Select Mouse (mm10), and select chr5. Next, go back to Galaxy, click on the label of *one* of your HISAT datasets to open the dropdown panel. Select display with IGV local. Now go back to IGV. It may take a moment for your data to load. Initially, you won't be able to see anything at the whole-chromosome resolution. Then add the HISAT accepted hits for your other dataset (go back to Galaxy to do this as described above).

# **Chromosome 5:** Chr5 is a larger chromosome; chromosomes are numbered more-or-less in order of size. Can you find more information online about it such as the number of known **protein-coding genes**? How does its **overall size** and **gene density** compare to other chromosomes?

13. Find Data Region: Zoom in to the region of chr5:112200107-113086162 (type the coordinates in, no spaces). This is the section representing the complete block of data in the tutorial, which altogether occupy a small region with a handful of genes. The top track is chromosome coordinates, the center two tracks are your data (which are not yet visible), and the bottom track is annotations (RefSeq Genes) superimposed onto the chromosome coordinates. You will see the names of some but probably not all genes in this region (it is still too compressed).

14. Find Mapped Reads: One of the genes that should be in your current IGV view is Tfip11 in the bottom track (recall that these are gene names and locations, not your data). You can scroll right and left within the view by dragging left/right within any track. Drag the genome such that Tfip11 is at the center of your view. Go to the -||||||+ zoom feature in the upper right and click + once to zoom in. Re-center Tfip11 if necessary. Keep zooming by one and re-centering until your data *first appear* in the two center tracks. This will be a block of around 24 kb (24,000 bases).

15. Interpret IGV Format: In the refseq gene annotations track, how are exons and introns represented? How are forward and reverse strand represented? Hover over a Tfip11 exon. What type of information is represented in the popup? What observations can you make about the HISAT mapping of your reads in this view?

16. Detailed IGV View: Zoom in three more times. Now you will start to see more details in your mapped reads. Can you tell which are forward and reverse reads? Hover over one of your reads to get a popup of the details and note the information contained here.

# Color Coding in IGV: You'll note colors and lines in your reads; **you won't focus on these in this exercise**, but if you click on **View** (top of screen) → **Color Legends**, and select **Mutations** → **Edit**, you will see a listing of the colors and their meanings. This is important when sequencing genomes and exomes to visualize SNVs and other variants, for example.

17. More Detailed IGV View: Center the view on an exon that has mapped reads, and zoom in about 5 more times until you see the individual nucleotides at the top of the bottom track. Beneath this is the amino acid translations in all 3 frames. Do any frames contain stop codons? Click on the arrow in the lower left next to Sequence to see the translation for the other strand. How can you determine which translation is correct? (you may want to zoom out a few times to get the entire exon into view)

18. Differential Expression in IGV: Return to the ~24 kb zoom-out where your data first appeared. Scroll to the right and left to see if you can observe any apparent differences in the number of mapped reads between the controls and experimentals for any of the genes (remember that it takes a moment for data to load when you scroll to a new region). You won't use IGV to quantify these differences; this is for visual observation only. You'll now return to Galaxy.

19. Assemble Transcripts: In Galaxy, go to Tools → NGS: RNA Analysis → Cufflinks. Select both your HISAT datasets to be run in parallel, with the iGenomes annotations. Leave all default parameters as they are. Each dataset will produce 4 sets of results: gene expression, transcript expression, assembled transcripts, and skipped transcripts. Click on the eye icon of assembled transcripts in either dataset; scroll over to see FPKM values (see below).
(Run Time: a few minutes)

# Cufflinks: This tool takes HISAT-processed short reads and assembles them into as many mRNA transcripts as it can, and as complete as possible; then Cufflinks measures the abundance of each transcript. **Transcript abundance is quantified as FPKM** (fragments per kilobase of transcript per million mapped reads). It will identify the various expressed **isoforms (alternative splice variants)** as well – both known and novel. You may want to review the parameters of alternative splicing.

20. Quick Preview of Cufflinks Results: Click on the eye icon of Cufflinks ...: gene expression for one of the two datasets. In the left-most column, you'll see a list of all chr5 genes; when you scroll to the right, you'll find the FPKM columns where you'll likely see zeros for most of the data. (It will be a challenge to find your expressed genes here, but if you want to find them, keep scrolling down and they'll appear as a cluster of FPKM numbers. You'll find them more directly later...)

21. Quick Preview of Alternative Splice Variants: Click on the eye icon of Cufflinks ...: transcript expression. Look at the gene names. Are there any genes for which the name shows up more than once? These are known alternative spice variants for the chr5 genes. Some of your transcripts may be spice variants (again, it will be a challenge to find your data here, and you'll find them later).

22. Create a Transcriptome: In Galaxy, in NGS RNA Analysis, select Cuffmerge. Here, select your 2 Cufflinks assembled transcript files. Under Use Reference Annotation, select yes, and select your Gene Annotation File. Then Execute.
(Run Time: a few minutes)

# Cuffmerge: This tool pulls together all of your assembled transcripts from both datasets, plus the reference annotation, into one pool, a final collective **transcriptome**. It also filters out fragments that are likely artifacts. In other words, it creates a collective index of all of the different ***known*** genes and isoforms that have been expressed in the experiment, plus any ***new*** expressed isoforms (transcripts) from this experiment, under all tested conditions (in this case, the 2 different tissues). If you click on the eye icon for the output, you'll see a list of all the transcripts: their coordinates on chr5 (exon by exon) and names. You will not see quantitative expression data here (FPKM), as that is not the purpose of creating the transcriptome.

23. Measure Differential Gene Expression (DGE): In NGS: RNA Analysis, select Cuffdiff. In your workspace, under Trancripts, select the Cuffmerge data. Under Generate SQLite, select Yes. Under Conditions 1 and 2, select your 2 HISAT datasets, first the Control and second the Eya2-KO. Name them Control and Eya2-KO. Click Execute. You'll see numerous different actions launched in the History panel.
(Run Time: a few minutes)

# **Cuffdiff:** This tool uses the transcriptome produced with Cuffmerge as a new – and maybe improved - map upon which to look at gene expression. You've gone from **mapping reads to a genome** (in our case just one chromosome) to determine which genes were expressed, to **quantifying that expression**, and finally to **quantifying that expression with respect to all *known* and all *novel* isoforms**.

24. Download DGE Results: Among the Cuffdiff results, find Cuffdiff ...: gene differential expression testing. Click on the eye icon to see the data. The Galaxy interface is not the most convenient in which to see and work with the data. Click on the name to expand the green Cuffdiff panel, and click on the download icon. This will download a text file called GalaxyXX... (XX = the number of the green panel) to your downloads folder (or wherever you've configured your browser to download files).

25. Import DGE Results into Excel: Open the downloaded Galaxy file: two-finger-click (Mac) or right-click (PC) on the file and select Open File in TextWrangler (or analogous text-editing program). Then Edit → Select All → Copy → and paste this into an Excel spreadsheet. This should produce a column-delineated spreadsheet.

26. Reduce Spreadsheet Clutter: In the top row, replace value_1 with Control FPKM and value_2 with Eya2-KO FPKM. Delete columns test_id, gene_id, sample_1, sample_2, status, and test_stat. Highlight Row 2; go to Window (top of screen) → Freeze Panes to fix the top row as headers.

27. Find Data in Spreadsheet: Click on Row 2 to highlight. Click shift + command + down-arrow to select all data beneath the header row. Click Data (top of screen) → Sort → Column D → Largest to Smallest. Now all of the tutorial dataset should appear in the first 15 rows. (If data appear in scientific notation, you can change this to number by clicking Format → Category: Number). You can delete all downstream zero data (shift + command + down arrow, Edit (top of screen) → Delete).

28. Create a Fold-Change Column: Highlight the log2(fold_change) column. Go to Insert (top of screen) → Column. Name the column fold-change. In cell E2, type =d2/c2, and copy and paste the formula for the rest of the column. Log2(fold-change) is a standardized "official" way of reporting differences in expression levels, but the human brain processes simple fold-change ratios better.

29. Make Fold-Change Symmetrical: In the fold-change column, up-regulated genes have values greater than 1, and down-regulated are lower than 1. A doubling of a value in column D over C results in a fold-change of 2, a doubling of a value in C over D results in a fold-change of 0.5. You can correct the values to make them symmetrical around ± 1 by calculating the log2 of fold-change (this is effectively what the the log2(fold_change) conversion does). Make the data in the fold-change column symmetrical around zero by switching the formula in the less-than-1 cells to =-1*(CX/DX).

30. Interpret Data – FPKM Meaning: FPKM is fragments per kilobase of exon per million fragments mapped, a method that normalizes mapping data (number of reads mapped per region) to the size of the library, and to the size of the gene. For example, full-length mRNAs in our 14 gene set range from ~800 to ~8000 bases. Without this normalization, a single assembled transcript for the largest RNA would register as 10-fold greater expression than that of a single smallest transcript.

31. Interpret Data – FPKM Levels: There is no universal standard for determining which FPKM values are "meaningful." We do not know what minimal level of expression of a given gene is inherently necessary to have an effect, and this likely varies among genes. One approach is to consider anything above 1 FPKM as an expressed gene. But there is obviously a

massive range of expression even within our dataset. With large datasets, the tendency is to select a cutoff FPKM level for downstream analyses (later in this tutorial) that provides us with a list of the most highly-expressed genes. The assumption is that a gene with an FPKM of 100,000 will have a much larger impact than one with an FPKM of 100. For our data, for example, you'll select an FPKM cutoff of 100 (must be at least 100 in either control or Eya2-KO). Which of your genes do not pass this threshold? Highlight those that do not meet this threshold in Red in your Excel spreadsheet.

32. Interpret Data - Fold-Change: Just as there is no way to know what level of expression is meaningful, there is no way to know what fold-change is biologically meaningful above 1 or below -1. For all genes that pass our selected FPKM threshold, you'll adopt a fold-change of ±1.30 as the threshold. Highlight in Orange all genes for which fold-change is between +1.30 and -1.30. In Black will be all genes that meet the selected thresholds for both expression levels and fold-change; we will further study this set in downstream analyses. Note the log2(fold change) that accompanies each fold-change – this is the value you'll report in a writeup/publication.

33. Interpret Data - p-value and q-value: The p-value is the result of standard pairwise comparisons, with p < 0.05 registering as significant. When conducting numerous such comparisons on the same dataset, as in a whole-transcriptome DGE study with 10,000+ genes, you need to correct for *multiple* comparisons. This secondary FDR (false discovery rate) test assumes that 5% of significant data in very large datasets with multiple tests will be false positives. This test winnows our data further to yield a smaller number of significantly differentially-expressed genes. (*For our data, where n = 1, the p- and q-values are not particularly meaningful, nor is the q value; but you will proceed as though they are*.) Which genes register as significantly differentially-expressed according to q-value? How many/which of the genes in Black in the Excel spreadsheet register as significantly differentially-expressed?

34. Visualize DGE with a Heatmap: In Galaxy, in the Tool panel, select NGS: RNA Analysis → cummeRbund. Your SQLite data will be the only option for this tool. Click on Insert Plots. Reduce the width and height to about half the defaults. Under Plot Type, select Heatmap. This particular heatmap will not be particularly informative. Typically, data are represented as individual replicates, with hundreds/thousands of genes, that can be clustered in groups to determine general parameters of expression data. Note that your two samples are on the x-axis, and the 14 genes in the dataset are on the y-axis, with general expression levels coded by color. Other plot-types in cummerBund are commonly seen in DGE analysis literature and are worth exploring with larger datasets.

35. Cause-and-Effect: It is worth pausing here to clarify for yourself what the data are telling you so far (this is particularly important if the topic is new to you). Write a phrase in your Excel spreadsheet that describes the results, such as: "When Eya2 is knocked out, xxx, xxx, xxx, ... genes are down-regulated; therefore, normal expression levels of these genes are dependent on Eya2. When Eya2 is knocked out, yyy, yyy, yyy, ... genes are up-regulated; therefore, Eya2 keeps expression of these genes in check/suppresses it/prevents over-expression..."

# Eya2 Gene: Eya2 is a member of the **eyes-absent family of genes** which, when mutated in drosophila, causes failure of eye development (https://www.nature.com/articles/srep23228). Eya homologs occur throughout metazoans.

36. Eya2 Information - GeneCards (http://www.genecards.org/): Type eya* into the search engine, and select Symbols instead of Keywords. GeneCards provides an array of information about human genes. How many human EYA paralogous genes do you find? What is the general function of the EYA2 protein?

37. Eya2 Information - MGI Mouse Genomics Informatics (http://www.informatics.jax.org/): Type eya* in the QuickSearch window. In the results, how many mouse Eya paralogous protein coding genes do you find? The others (there are more than 500) are all of the various known Eya mutants (you can expand by showing the first 100). The Eya2 knockout in this tutorial has not yet been entered into this system.

38. Eya2 Information - UniProt UniProt (http://www.uniprot.org/): Enter Eya2 mouse. Select the Eya2 Swiss-Prot entry (gold star) in the left Entry column. UniProt is a database that specializes in proteins, though you'll find links to genomic information as well. On the left, you can navigate to different information (or you can scroll down). Start with Function: What are the general functions of Eya2? With what protein is it known to be functionally redundant? Navigate to Pathology & Biotech: What is the result of a lack of (i.e. a mutation in) Eya2? In what condition is it lethal? Go to Expression: can you find information relevant to the study in this tutorial?

39. List of Genes for Downstream Pathway/Process Analysis: Return to your Excel spreadsheet with the DGE analysis. Into a new column, copy/paste just the names of the genes you identified as passing the threshold for expression level and fold-change. Add Eya2 to the list. Change gene name Adrbk2 to Grk3 (its more common name). In the next column, write up or down to keep track of whether the gene was up-regulated or down-regulated in response to Eya2 knockout. Group up- and down-regulated genes together.

# Downstream Pathway Analysis: A number of online tools are available for analyzing features such as: A) functional classifications, B) interactions among the genes in your dataset, C) interactions of the genes in your dataset with other genes, D) colocalization of expression, E) homology of genes/proteins. Keep in mind that these tools use known, human-curated information gathered from sources such as published research. Therefore, lack of findings with any given tool does not mean there is no relationship among genes; it could mean that none has been discovered *yet*. After all, this is one primary purpose of conducting DEG research...

40. GeneCards (http://www.genecards.org/): The list of genes in your dataset is small. Therefore, it is not onerous to acquire basic information on each gene for this tutorial. GeneCards is a good general-purpose lookup tool. Though it is a human database, we can generally assume analogous functions of mouse genes. You don't need to skim further than the Summaries panel for this, but you may need to click on diseases/conditions to see if there is relevance (the disease name is not always informative). What is the general function of each gene, and can you find a relationship to the Eya2 KO study in this tutorial? Are all genes protein-coding, and if not, what do they code? Are there diseases and conditions associated with each gene that may be relevant to this particular study? In your Excel spreadsheet, create a column for Function and another for Relevance (to the Eya2 KO study). Enter your previously-researched Eya2 information here also. Don't be concerned if you can't find anything particularly relevant for a given gene – we don't yet know everything, so you may discover something novel. Just enter ? if you don't find anything relevant. Many genes have little-to-no known function (yet).

41. GeneMania (http://genemania.org/): This tool provides a dense suite of information. Create columns in your Excel spreadsheet next to your gene list, one for each category: GeneMania shared protein domains, GeneMania physical interactions, GeneMania co-expression, GeneMania co-localization. Enter your gene list into the search window in the upper left. In the results, on the right, unclick all but shared protein domains. Do any genes in your dataset share protein domains with other proteins in your dataset? With genes not in your dataset? You can click on any of the circles and drag to move the components around; this helps to better see connections. Now unclick shared protein domains and click physical interactions. Do any of your genes share physical interactions with each other? With other genes not in your dataset? Do the same for co-expression and co-localization. If you click on any connecting lines, you will find a link to the literature reference for the connection. Does Eya2 have any known connections to any of the genes in your dataset? If not, what does that mean regarding your study?

42. Summarize Pathway Results: Write a paragraph that synthesizes your findings between GeneCards and GeneMania. This should be a descriptive assessment of known and potential interactions among your genes and in relation to your study. Eya2 is not a well-studied gene, so what conclusions/speculations can you draw that tie the interactions together among at least some of the genes? What hypotheses can you generate from your results?

# Alternative Splicing Analysis: Some of your genes may be alternatively spliced to produce protein isoforms with different properties and activity. Furthermore, the Eya2-KO experimental condition may induce a variation in splicing patterns of the other genes in your dataset.

43. Transcript Splice Variant Spreadsheet: The analysis of gene expression data is part of the story. Now you'll go back and look at transcript expression data to explore alternative splicing variants. In Galaxy, in your Mini-Mouse history, find Cuffdiff ...: transcript differential expression testing. Download and set up an Excel datasheet as you did in steps 24-29 for gene differential expression analysis.

44. Transcript Differential Expression: Highlight your data and sort on gene, alphabetically a → z. In red, highlight all genes for which you find more than one result. These are the splice variants. Apply the same 100 FPKM cutoff here that you did for gene expression. For which genes are alternative splice variants expressed above the threshold? *Italicize* those

rows for which the splice variant does not pass the threshold in both experimental and control. Which gene(s) has a substantively-expressed splice variant? Is the same splice variant expressed in both experimental and control? Would you estimate that the difference is significant for each splice variant between experimental and control (think carefully about the technicality of calculating statistics if a zero value occurs in one or the other FPKM column)?

45. Transcript Characterization: Go to Galaxy and find Cuffdiff …: transcript FPKM tracking. Click on the eye icon to open the file, and observe briefly the data. There is no need to download; you'll sort within Galaxy: In the tool panel, select Filter and Sort → Sort. Select your Transcript FPKM tracking, and sort on Column 10 (control_FPKM), descending order. Find your two different transcripts for the alternatively-spliced gene near the top of the spreadsheet. Copy and paste the associated accession number from Column 3 into the appropriate row for that gene in your Transcript Splice Variant spreadsheet. To be sure which is which, scroll over to the FPKM values in column 10 (control) and 14 (Eya2-KO) to reconcile these with your Excel spreadsheet.

46. NCBI Identification of Splice Variants: The accession numbers you retrieved for the splice variants are specific to the NCBI (National Center for Biotechnology Information) Database. Go to NCBI (https://www.ncbi.nlm.nih.gov/). Enter the gene name for your alternatively-spliced gene, and in the Databases dropdown menu, select Gene, and Search. In the results, click on the Mouse entry. Scroll down to NCBI Reference Sequences (RefSeq). Are your splice variants represented here? What is the nature of the variation? Does this variation alter the nature of the protein product? If not, what is a possible role of the variant? (If you are unsure, Wikipedia has a good first-pass summary of the roles of the gene region affected.)

47. Summarize Splicing Results: Write a statement explaining the influence of the Eya2-KO on splice variants, and the potential downstream consequences (in a general sense).

EXERCISE 2: Follow-up Big Data Analysis

There is a slight bait-and-switch here in the tools you will use – this is described below in red text.***

This analysis should be done on your own, outside of class/lab. Because that dataset will be larger, it may take a few days to run – especially the HISAT mapping step, and especially if run on public Galaxy instance. While this dataset is full-scale in that it encompasses the whole genome (versus a small segment of chr5), it is on the smaller side in that you will still work with one control and one experimental set. Therefore, it can be completed reasonably efficiently on the public Galaxy server. In a full-scale study, your sample size would be greater (N = three or more).

Goal: Analyze a full-scale set of RNA-seq data on the PUBLIC GALAXY SERVER (not Jax): Use the Mini-Mouse tutorial to analyze a full-sized set of data.

1. Create New History: In Galaxy, create and name a new history.

2. Get Data: In the Tool panel, go to Get Data → EBI SRA. This takes you to the European Nucleotide Archive. Enter PRJNA68307 into the search window. In the results, read the Description. Scroll down and click on the Navigation tab → Sample. You'll find 10 results. These are the same data with which the Mini-Mouse tutorial was prepared, using Mouse retina, Eya2 KO, ZT19 and Mouse Retina, control, ZT19. Briefly outline the entire study for yourself (ZT19 is midnight, ZT7 is midday). Select two datasets that make sense to compare. It is fine to re-use the sets used in the Mini-Mouse tutorial, since we only looked at ~14 genes in that activity. Write down the accession numbers of your selected files; you may need to access this information later.

3. Upload Data to Galaxy: Click on each dataset you've selected. To upload to Galaxy, click on the FASTQ files (Galaxy) links. These are paired-end data, so be sure to upload both for both datasets. These will upload into your open history; be patient – they are large files (~300-500 MB)

4. Decompress File: When your files are imported into Galaxy, they are compressed for efficiency of transfer and must now be uncompressed. Under NGS: QC and manipulation, select FASTQ Groomer. Select all of your files, and under Input FASTQ quality score type, select Sanger & Illumina 1.8+ (that refers to the type of sequencing – it is the most common/modern choice, if you don't know what to select). *Going forward through the analysis, use these FASTQ groomed files.*

5. Upload Mouse mm10 iGenomes Annotations: In Mini-Mouse, your data were isolated to chr5. You'll need the full genome annotation file for these datasets: https://usegalaxy.org/u/andrea-tilden/h/mouse-mm10-gene-annotations (these can also be found at Ensembl, UCSC Genome Browser, and various other sites)

*****6. Your pipeline in this analysis – *after HISAT* – will be htseq_count → DESeq2. While the pipeline in the tutorial works well for pedagogical purposes (ex. splice variants), the FPKM method of quantification has fallen out of favor and has been replaced by other normalization methods via pipelines such as htseq_count → DESeq2. (Note that you are likely to come across papers using either of these toolsets, in addition to others...)**

Count Methods: For an excellent discussion of the various count methodologies, see: https://github.com/hbc/knowledgebase/wiki/Count-normalization-methods. In sum, FPKM via the Cuff-tools should only be used for within-sample comparisons (which genes within your sample were expressed more/less than others in sample) – it normalizes well within a sample by correcting for transcript length. It does not correct so well between samples. DESeq2 employs a better between-samples normalization, important for the typical RNA-seq analysis (but it does not correct for gene length, so shouldn't be used for within-sample analysis).

6. Count Reads mapped to genes. htseq_count simply counts the number of reads that map to a gene. For example, if you were to go to IGV and manually count the reads mapped to a gene, this number should

match the htseq_count number. Under Aligned Sam/Bam file, select your HISAT files. Under GFF file, select the mouse genome annotations file. Keep default settings. In the results, you should see a 2-column file with a list of genes and counts for each gene.

8. Differential Expression with DESeq2: Here, under 1: Factor, give your analysis a name. In Factor level 1, select the dependent variable (ex. the knockout mouse), and in Factor level 2, select the control (this is the reverse of the order used by CuffDiff). The only default you need to change is Output normalized counts table: change this from no (default) to yes. This will give you a report of the normalized counts for each dataset which, unlike Cuffdiff, is not included in the differential analysis output.

9. More detailed DESeq2 and downstream instructions: ....Under construction!

Downstream Analysis - Tool List: (.............*a tutorial is in the works for this section; this is not an exhaustive tool list*)

GeneCards: Basic Gene-by-Gene information http://www.genecards.org/
GO – **Gene Ontology Enrichment Analysis**: http://www.geneontology.org/
GO PANTHER: http://pantherdb.org/
STRING: http://string-db.org/
ENRICHr: http://amp.pharm.mssm.edu/Enrichr/ (meta-database)
GORILLA: http://cbl-gorilla.cs.technion.ac.il/
MSigDB: http://software.broadinstitute.org/gsea/msigdb
DAVID: https://david.ncifcrf.gov/
Consensus Path DataBase – CPDB: http://cpdb.molgen.mpg.de (meta-database)
iPathwayGuide (free for a few days): https://www.advaitabio.com/ipathwayguide.html
CPDB: http://cpdb.molgen.mpg.de/
GeneMANIA: http://genemania.org/ (good for smaller subset of genes or a single gene of interest – high-density information)
GeneWeaver: https://geneweaver.org/ (good for cross-species comparisons)
Reactome: http://www.reactome.org/ (an especially rich/dense pathway visualization tool)
WikiPathways: http://www.wikipathways.org/index.php/WikiPathways (a good way to view particular pathways once you've identified a pathway you're interested in – not for analysis of list of genes)