Arabidopsis thaliana Multi-track RNA-Seq Browser.

Priyank K. Purohit, Nicholas Provart.
University of Toronto, CSB498H1Y, 2015-2016.

Table of Contents

Date: October 16, 2015

Agenda:

1. Check if having local BAM files can speed up data retrieval using samtools' mpileup () call.

Protocol:

1. Check if local BAM files can speed up data retrieval using samtools' mpileup () call.
    a. Downloaded the BAM file for experiment SRR547531 using wget ().
    b. Executed mpileup () through SSH.
        i. Used local BAM file and compare to iPlant BAM file
    c. Ran the output.cgi script with the two BAM files (local vs. iPlant).
        i. Used Chrome Dev Tools to analyze the TTFB (time to first byte).

Results:

1. Check if local BAM files can speed up data retrieval using samtools' mpileup () call.
    a. Done, file size = 662MB.
    b. Local BAM file mpileup call returns data very quickly (< 1 second), iPlant BAM file takes 10-20 seconds.
    c. Local BAM file returns data in ~600ms! The iPlant BAM file takes ~60 seconds!
        i. Only calling the generate_rnaseq_graph() function once to produce a single image.

Notes/Questions:

- Figure out why a URL to the BAM file doesn't work. Currently it works with a relative path..!
- See if BAM files hosted on Drop box result in data retrieval that is just as fast as local BAM files. This might implicate the iPlant server as the bottleneck and prove the HTTP request's innocence.
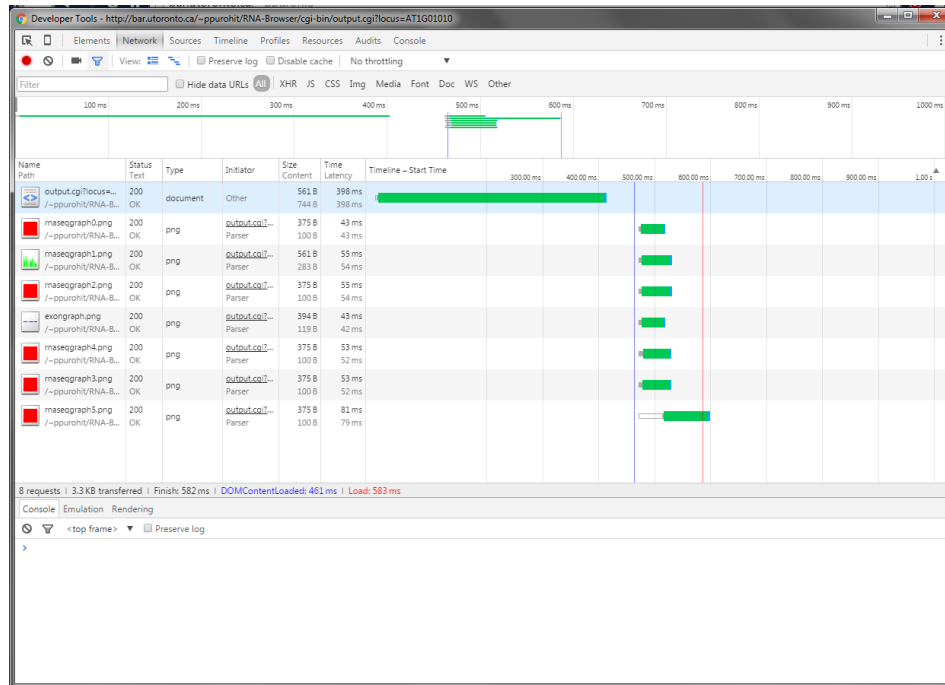
Figure 1: ~600ms for local BAM file.
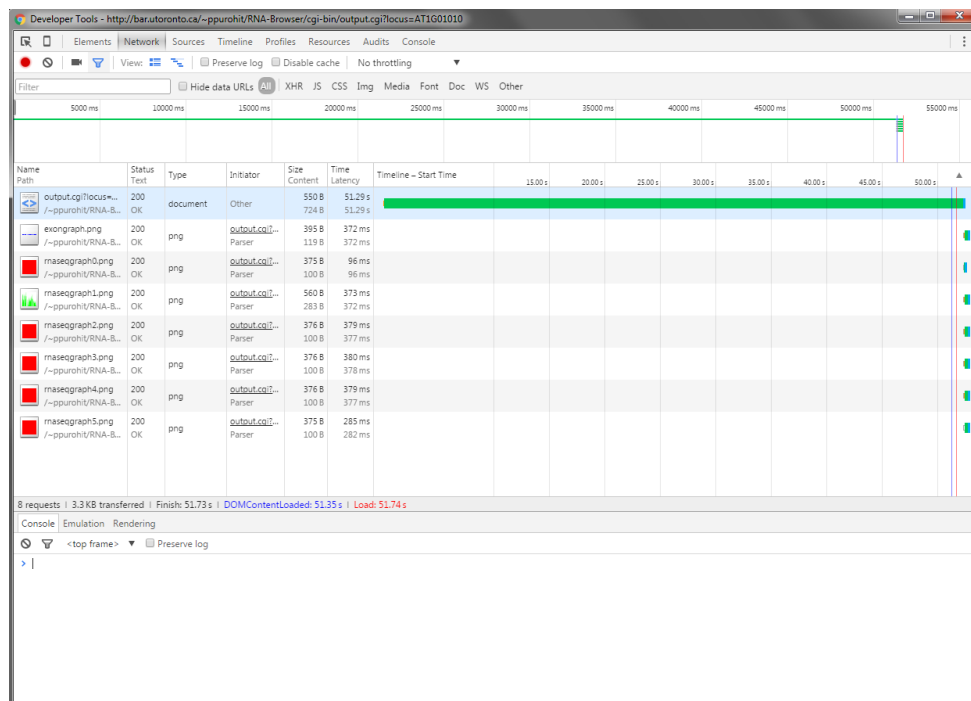
Figure 2: ~60 seconds for iPlant BAM file.

Date: October 21, 2015

Agenda:

1. Check if BAM files stored on other servers are just as fast as local BAM for the mpileup() call.

Protocol:

1. Executed 3 mpileup () commands through SSH.
    a. `time samtools mpileup -r chr2:8032000-10329941 http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeSydhRnaSeq/wgEncodeSydhRnaSeqK562Ifna6hPolyaAln.bam > ucsc.txt`
    b. `time samtools mpileup -r Chr2:10327050-10329941 http://vision.iplantcollaborative.org/iplant/home/araport/rnaseq_bam/aerial/SRR547531/accepted_hits.bam > iplant.txt`
    c. `time samtools mpileup -r Chr2:10327050-10329941 http://bar.utoronto.ca/~ppurohit/RNA-Browser/cgi-bin/data/iplant/home/araport/rnaseq_bam/aerial/SRR547531/accepted_hits.bam > bar.txt`

Results:

1. Done.
2. Executed mpileup () through SSH.
    a. UCSC.edu BAM file:
        i. 1.322 s for 6 260 175 bytes
    b. iPlant BAM file:
        i. 15.294 s for 1 248 931 bytes
    c. bar.utoronto.ca BAM file:
        i. 0.060 s for 1 248 931 bytes

Notes/Questions:

- The UCSC BAM file returned only ~1500 bytes for the `Chr2:10327050-10329941` region. This smaller size could be the reason why the command is fast. The query region was therefore increased to get 6x more data.
    a. … and it's still fast!

Date: October 23, 2015

To-do List:

1. Research proposal (Oct 29)
2. XML File Updates
    a. Change the Newland links to their new Vision links
    b. Add colours as discussed in Oct 22 meeting w/ NP
        i. Try to make them web safe colours without changing the shade too much
    c. Add the missing information in the XML file (some pictures missing)
3. Download the mpileup data for the default locus from all of the BAM files on iPlant
    a. Spread this out over 3-4 days
4. RNA Browser:
    a. Start working with locally stored mpileup data
    b. Change the image dimensions for RNA-Seq graph to ~250 x 50.
    c. Make the exon graph slim and add a horizontal line through the middle
    d. Start making dynamic requests for RNA-Seq graphs
        i. Sample flow:
            1. User comes on our app
            2. The RNA-Seq graphs of the default gene are pre-made and loaded on page load
            3. When the user enters a particular locus, the app will load 3 graphs, the rest are dynamically generated after page load …
    e. Image read map heights:
        i. Start with default height of 1000 reads
        ii. Have a button that allows the user to re-generate all images such that the max height is the maximum read for any base pair
    f. FPKM calculations:
        i. Genie is doing this, make sure to have the information she needs for these calculations

Date: October 25, 2015

Agenda:

1. Get Richard's new BAM Locator XML file and update the `newland` links to `vision` links.
2. Change the `displayxml.cgi` to account for changes made in the attribute names.

Protocol:

1. Got Richard's new BAM Locator XML file and updated the `newland` links to `vision` links.
   a. Wrote a [Java program](#) to go through an XML file, find the correct new link and replace it.
2. The attribute name changed to `svgname` from `subunitname` (correct name is there now).
   a. Changed the correct IF statement in displayxml.cgi to look for `svgname` as opposed to `subunitname`.

Results:

1. Success. The code replaced all files correctly.
2. Success. The `displayxml.cgi` script works correctly with the new BAM locator XML file.

Date: October 29, 2015

Agenda:

1. Add colours picked out by Dr. Provart to the BAM locator XML file's foreground column.
2. Update the BAM file links to the new Amazon S3 links.

Protocol:

1. Add colours picked out …
    a. Manually copy-pasted the new HEX colour codes.
    b. Fixed cases where the colour attribute was missing.
2. Update BAM file links to Amazon …
    a. Ran the Java code from Oct 25, 2015 w/ Amazon S3 prefix instead of the iPlant prefix

Results:

1. Add colours picked out…
    a. Successful.
    b. Live on BAR @ http://bar.utoronto.ca/~ppurohit/RNA-Browser/cgi-bin/displayxml.cgi
2. Update BAM file links to Amazon
    a. Success. It is live on BAR at the link shown above.

Notes:

1. The images still look incorrect for some of the experiments (i.e. the image should be root but is not)
    a. Look into this and fix it.

Date: November 1, 2015

Agenda:

1. Get the mpileup for a default gene of interest from all BAM files.
    a. Investigate the issue of remote BAMs not returning data but same local files would (reported by Vivek)

Protocol:

1. Get the mpileup for default gene of interest from all BAM files.
    a. Wrote a shell script to iterate over the BAM files in the `iplant_path_to_rnaseq_bam_files.txt` file.
    b. Each time, it executes the `samtools mpileup` call on the BAM file and outputs it to a smaller BAM file.
        i. Getting the mpileup for the first locus only (Chr1:3631-5899).
2. Based on the result for #1, the questions arise: are these files not returning data because there is no data? Or is it because there is some issue with the remote vs. local file?
    a. Executed mpileup through command line on a single BAM file that did not return data.
        i. `samtools mpileup -r Chr1:3631-5899` http://s3.amazonaws.com/iplant-cdn/iplant/home/araport/rnaseq_bam/leaf/SRR446034/accepted_hits.bam
    b. To see if the issue was resolved in the latest version of samtools, downloaded and installed the latest v1.2.1 of SAM Tools.
        i. Downloaded the source code
        ii. Executed the `makefile`
        iii. Installed by: `make -prefix=/path_to/install_folder/ install`.
        iv. Executed the call from 2(a)(i) with the new samtools exe (be sure to specify with a relative path to the new executable).
            1. `./samtools-1.2/exe/bin/samtools mpileup` `http://s3.amazonaws.com/iplant-`

```
cdn/iplant/home/araport/rnaseq_bam/leaf/SRR446034/accept
ed_hits.bam -r Chr1:3631-5899 -d 8000
```

    c. To see if BamView can show anything, the SRR446034 BAM file's Amazon link was used to see if there is any data in the file.

    d. Tried to redo 2(b)(iv) but when I'm in the directory of the newly installed BAM file

        i.
```
./samtools mpileup http://s3.amazonaws.com/iplant-
cdn/iplant/home/araport/rnaseq_bam/leaf/SRR446034/accepted_hit
s.bam -r Chr1:3631-5899 -d 8000
```

3. Rerun the shell script from the same directory as the latest SAM Tools ... (repeat #1)

Result:

1. Get the mpileup for default gene of interest from all BAMs.

    a. Got back data from the following BAM files (increase text size/zoom in to read):

        i. dark_SRR1019456_accepted_hits.bam
        ii. dark_SRR1019437_accepted_hits.bam
        iii. dark_SRR493238_accepted_hits.bam
        iv. dark_SRR493237_accepted_hits.bam
        v. root_SRR314814_accepted_hits.bam
        vi. receptacle_SRR401418_accepted_hits.bam
        vii. receptacle_SRR401416_accepted_hits.bam
        viii. aerial_SRR547531_accepted_hits.bam
        ix. receptacle_SRR401413_accepted_hits.bam
        x. receptacle_SRR401421_accepted_hits.bam
        xi. receptacle_SRR401419_accepted_hits.bam
        xii. receptacle_SRR401415_accepted_hits.bam
        xiii. aerial_ERR274310_accepted_hits.bam
        xiv. aerial_SRR847505_accepted_hits.bam
        xv. aerial_SRR847506_accepted_hits.bam
        xvi. aerial_SRR847504_accepted_hits.bam
        xvii. aerial_SRR548277_accepted_hits.bam
        xviii. receptacle_SRR401420_accepted_hits.bam
        xix. flower_SRR800753_accepted_hits.bam
        xx. receptacle_SRR401414_accepted_hits.bam
        xxi. flower_SRR800754_accepted_hits.bam
        xxii. leaf_SRR1159857_accepted_hits.bam

2. Try to get mpileup from a BAM that did not return data in 1(a)

    a. Command line mpileup call with SAM Tools v0.1.18

        i. open: No such file or directory

        ii. Segmentation fault

    b. Command line mpileup call with SAM Tools v1.2.1

        i. [knet_seek] SEEK_END is not supported for HTTP. Offset is unchanged.

        ii. [mpileup] 1 samples in 1 input files

    c. Checking the file with BamView program

<ol type="i">
Figure 3: SRR446034 BAM file's first locus in BamView program.
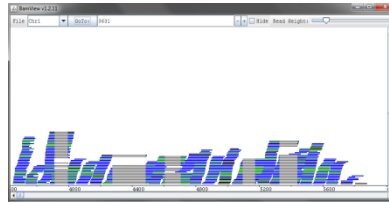
<ol type="a" start="4">
GOT BACK DATA! For some reason this works and returns data from that same BAM file.
<ol type="i">
But doesn't work for another experiment, SRR949989's BAM file…

<ol start="3">
Rerun the shell script from the same directory as the latest SAM Tools …
<ol type="a">
Got even less number of BAM files returning data

Date: November 5, 2015

Agenda:

1. Put together a rough version of the multi-track viewer.

Protocol:

1. Combined the relevant code from output.cgi and displayxml.cgi to read the mini-BAM files from the mpileups directory and output to files in the /img/ directory where all the image files are...
   a. The img files were generated using a shell script that generates blank images and `chmod`s them to 766.

Results:

1. Works well, but the mpileups are local. Doesn't tell is anything about what the end product will be like.

Date: November 12, 2015

Agenda:

1. Fix up the displayxml.cgi file and email a link to NP.
2. Add a horizontal line to the exon graph.

Protocol:

1. Cleaned up code, minor programming changes.
    a. Added missing parts as outlined by NP.
    b. Added colours as outlined by NP.
2. Added the horizontal line w/ another filledRectangle() call.

Results:

1. http://bar.utoronto.ca/~ppurohit/RNA-Browser/cgi-bin/displayxml.cgi
2.

Date: November 12, 2015

Agenda:

1. Produce the total number of mapped reads from a BAM file.

Protocol/Thought Process:

1. Information on total number of mapped reads should be available in the index file.
    a. Search the samtools manual for a convenient command to find this number
        i. Found and executed two potential commands to get this information on SRR547531 (the BAM file on BAR server):
            1. samtools idxstats <bam file>
                a. OUTPUT (tab delimited): seq name, seq length, # of mapped reads, # of unmapped reads
            2. samtools view −c −F 4 <bam file>
                a. ∑ of all mapped reads in the BAM file from all sequence names...
                b. This produces a single number that is the ∑ of all mapped reads from 1(a)(i)(1).

Result:

```
samtools idxstats accepted_hits.bam

        Chr1    30427671        1892640         0

        Chr2    19698289        2015247         0

        Chr3    23459830        2539031         0

        Chr4    18585056        1181593         0

        Chr5    26975502        1580658         0

        ChrC    154478          1792538         0

        ChrM    366924          68400           0

        *       0               0               0

samtools view -c -F 4 accepted_hits.bam
```

```
11070107
```

- Note that the ∑ of mapped reads = 11070107.

Date: November 18, 2015

Agenda:

1. Get the number of mapped reads for a given region rather than the whole file.

Protocol:

1. The number of lines returned by the `samtools view` command is correlated with the number of reads. Therefore, `samtools view Chr1:3631-5899 | wc –l` produces the number of reads mapped for that region.
   a. <mark>NEED TO VERIFY THIS ASSUMPTION</mark>.
2. Ran the following:
   a. `Samtools view accepted_hits.bam Chr1:3631-5899 | wc –l`
   b. `Samtools view accepted_hits.bam | wc –l`
   c. `Samtools view -c -F 4 accepted_hits.bam`

Results:

1. VERIFY, TO DO.
2.
   a. 120
   b. 11070107
   c. 11070107

Date: November 25, 2015

Agenda:

1. Create the front end of multitrack-rnaseq.html.
2. Display exon image by having the cgi script return a base64 image string. This image should be changing based on locus.
3. Display the RNA-Seq coverage images (not base64 for now, rather just the images generated by the CGI script). These images will have to be for the first locus since mpileups exist for that locus only.

Protocol:

1. Done, see November 25 and 26's commits to GitHub.
2. Done, see November 25 and 26's commits to GitHub.
3. Done, see November 25 and 26's commits to GitHub.

Results:
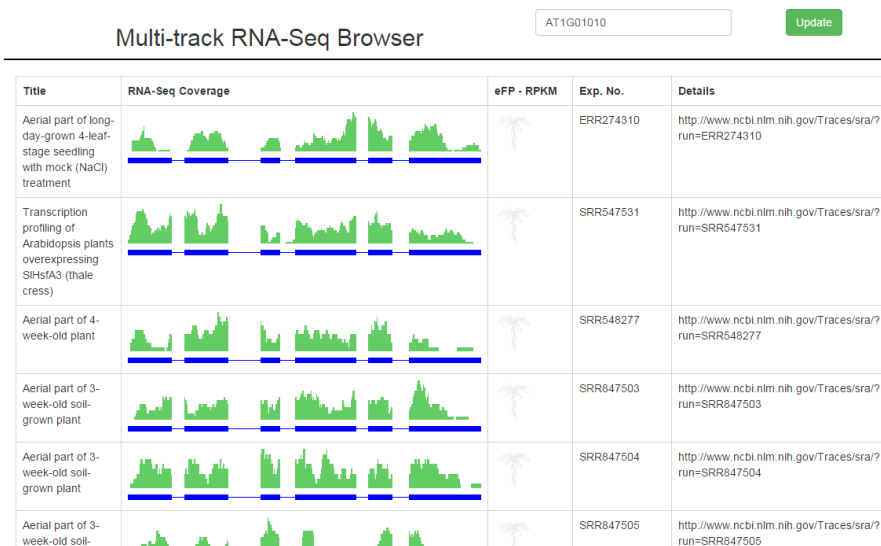
Date: November 26, 2015

Agenda:

1. Get `mpileups` for 4 more genes.

Protocol:

1. Get `mpileups` for 4 more genes
   a. Re-wrote the shell script (`mpileup_download_by_region.sh`) to download each BAM file and get `mpileups` for 4 more genes.
      i. Gene list:
         1. AT1G01010
         2. AT2G24270
         3. AT3G24650
         4. AT3G24660
         5. AT5G66460
   b. Added way to get mapped reads by counting samtools view <region>.

Results:

- Did not run the newly written script yet because to do next is: adding a way to get mapped reads by `bedtools` method; and to save both methods' mapped reads output.

Date: November 28, 2015

Agenda:

1. Get mapped reads counts for each of the 5 genes of interest from each of the 113 bam files by two methods.
2. Download mpileups and the read counts for 5 genes by running the shell script.

Protocol:

1. Two methods exist for getting read counts from a BAM file: word count method and using bedtools' multicov.
   a. Word count method:
      i. Code: samtools view <bam file> -r <region> -d 8000 | wc −l
   b. Bedtools method:
      i. Code: ./bedtools multicov −bams <bam file> -bed <bed file>
      ii. BED file generated by: echo −e "Chr1\t3631\t5899\tinterval1" > mybed.bed
2. Ran the shell script mpileup_download_by_region.sh after giving it 755 permission.

Results:

1. Running both commands one at a time on shell shows a general pattern that both methods give the same number. However it is not conclusive that they will always produce the same number.
   a. For each mpileup call, get the mapped reads by both methods and save them to compare later.
2. Worked. For 5 genes, mpileups and the mapped reads count was obtained.

Next Steps:

1. Check if both methods of counting mapped reads produce the same answer...

Date: November 28, 2015

Agenda:

1. Confirm that both methods of getting mapped reads give the same number of mapped reads.

Protocol:

1. A python script was written to read in the file contents and compare the two numbers.
    a. Script = mpileups/reads_mapped_methods_comparison.cgi

Results:

1. Both methods of getting mapped reads counts provided the same answer in each instance.

Date: January 19, 2016

Agenda:

1. Work on the UI to hide URLs
2. Set up a demo of sorting on images (work with the eFP image for now)
3. Read up about Y1H

Protocol:

1. UI changes:
    a. Detail column now has the experiment number, the publication link, and the total number of reads
2. Sort on image
    a. In front of an image, the number to sort on was made hidden using `<span>` tags.
    b. The table was then sorted using the `tablesorter.js` script.

Results:

1. UI changes were successfully implemented.
2. Sort on image by hiding a number in front of it works as expected!
    a. When Genie provides the R and P values, hide these values in front of their respective images and then let `tablesorter.js` sort the column.
    b. This may require an entire column to be hidden that will show the user the R and P values if the user wants to...

Notes:

Y1H = investigate DNA-protein interactions, allows us to find DNA-binding proteins. Falls under reverse genetics domain...

Date: January 24, 2016

Agenda:

1. Update the UI with columns and information NP suggested during the 20/01/ 2016 weekly meeting.
   a. "Column 1 from XML, groupings w/ grey shading, and controls"
2. Work on showing the exon/intron information (will have to modify the exongraph.png file).

Protocol:

1. Update the UI:
   a. Change the title of the column to "Description and Details" from "Details"
   b. Add NCBI SRA and PubMed links
   c. With a "More Details" button, allow user to expand a <div> with information about the number of reads and the controls information.
2. About exon-intro variants:
   a. Read more about the variants... from the biological point of view.
   b. Read today's latest email by Vivek and try out those services.

Results:

1. Update the UI

2. Exon-Intron variants
   a. Brushed up by reading the Wikipedia page and reading a paper...
   b. Tried out the service described by Vivek, it works perfectly as advertized

Date: February 1, 2016

Agenda:

1. Check if data is consistently returned from the new AWS hosting w/ mounted BAMs.
2. Play around with and understand webservice.cgi on AWS.

Protocol & Results:

1. Check if data is consistently returned from new AWS hosting:
    a. Download and compile the latest version of samtools on AWS server.
        i. Asher has already installed samtools v0.1.19.
    b. Write a script to find out whether data is consistently returned from each BAM file in mnt directory.
        i. Wrote data_return_consistency.sh script that will output the mpileups into a folder...
        ii. WORKED for two loci without any issues, safe to say that the original issue is solved because it would have returned data from ~15 BAMs as opposed to 113/113 (perfect result).
2. Understand webservice.cgi on AWS:
    a. Get rid of the pysam in favour of subprocess library to initiate a samtools call directly without going through pysam

Date: February 2, 2016

Agenda:

1. Figure out the components of the JSON object AWSS will return.
2. Integrate Gini's code …

Protocol:

1. JSON object design:
   a. The front end will make AJAX calls to the AWSS for RNA-Seq image and the related statistical (FPKM and PCC) and quantifying numbers (i.e. the number of mapped reads for that locus in the BAM file, and PERHAPS include SVG colouring details).
2. Gini's Code:
   a. Will she provide a web service for the FPKM and PCC values?

Results:

1. JSON object design:

```
{
status: "200/500/XXXX",
locus: "AT1G01010",
record: "ERR311526",
tissue: "leaf",
rnaseqbase64: "<a really long string of which denotes the RNA-seq image in base 64
format which can be rendered as an image in the browser…>",
fpkm: "456.45",
pcc: "0.9999",
svg_info: [
     colour: "#D3D3D3",
     intensity: "0.8756"
     ]
}
```

a. Error codes:

    i. Invalid locus = 1001

    ii. Invalid record = 1002

    iii. Invalid tissue = 1003

    iv. BAM file not found =1004

Date: February 3, 2016

Agenda:

1. Update the AWSS to return data in the format described in the February 2, 2016 entry.

   a. Edit the JSON object above if more information should be returned…

2. Do load testing to see how long does it take for the AWSS to return RNA-seq images for all 113 BAM files **IF the front end makes 113 AJAX requests**.

3. Integrate FPKM calculation steps in the AWSS.

4. Perform a sample test for SVG colouring.

Protocol:

1. Return JSON object from AWSS:

   a. Updated the output to match Feb 2 design by adding the missing attributes

   b. The code used by Asher is not the latest iteration, the Y-axis scaling is not present

      i. Fixed that code … but it's probably a good idea to get rid of all the extra code and make one file with the latest code

2. Load testing:

   a. Tried to do locust load test … but you need a Linux environment for that.

   b. Did two quick online tests

3. FPKM calculations integration:

   a. <pushed off until Gini's code is ready>

4. SVG Colouring Test:

   a. <PUSHED OFF, TO DO>

Results:

1. Return JSON object from AWSS:

   a. Successful, works fine but the code is old, so the output image back to the pre-multitrack stage (i.e. the colours are constant etc.).

   b. The scaling issue was fixed pretty easily but copying over the one line code for that

2. Load testing:

a. While the results are "positive," nothing can be inferred because they were very small scale trial version tests...

b. <TO DO> Repeat with Locust.

Date: February 4, 2016

Status update: Everything seems all over the place at the moment, taking half an hour to put in writing a project update entry…

Thoughts:

-   **Target**: multi track RNA-seq viewer "with exon-intron variants support"
-   MOST of the code is done, but there is a lot of testing that needs to be done
-   There is too much code, and it's all over the place. Although everything is *clearly labelled*, there are too many *clearly labelled pieces of code*!

Want:

1. UI with all the details (can get this from the XML file)
2. Web service that returns all the supported gene structure variants as base64 images
3. Web service that returns appropriately coloured, scaled, and axis-labelled RNA-seq image for a given locus, BAM file, and tissue type.

Specific features that are not the obvious features that one needs…

1. UI:
    a. Need a button that allows the user to rescale the RNA-seq image to the highest Y-axis scaling required.
    b. The button in (a) will also scale the eFP images' colours to highest being red and lowest being yellow…
2. Gene structure web service:
    a. Need a token from Araport people that does not expire
3. RNA-Seq web service:
    a. Should return all values required on the front end for SVG colouring and for FPKM calculations…

==**To Do List:**==

1. Clean up the BAR account, consolidate all the relevant code in one folder.

Date: February 5, 2016

Agenda:

1. Get an Araport token to make GET calls.

Protocol:

1. Get an Araport token to make GET calls.
   a. Followed the instructions here with AWS open in PuTTY.

Results:

1. Getting Araport Token:
   a. curl call to get API Keys: successful return output saved to a local file called 'public_keys_for_araport.json'
   b. curl call to get an authentication token: successful return output saved to a local file called 'authentication_token_for_araport.json'
   c. UPDATE:
      i. Asher is working on a BAR service to do the same; therefore, this has been put off until that service is ready.

Date: February 10, 2016

<div align="center">**Meeting topics**:</div>

What I did in the past week:

1. Designed the JSON object that will be returned by AWSS
2. The AWSS returns correct images now
   a. There is a small code issue that took up a lot of the time. Sometimes a blank image is returned... This happens when the mpileup call takes longer due to large amount of data returned.
      i. In C language I just put a wait statement on the pipes BUT I'm still learning how to do that in Python...
3. Did small scale load tests on the AWSS and the results were promising ... but I need a Linux environment to do a more extensive load test with Locust framework.
4. Cleaned up the BAR account.
5. Then spent ~2 days to make the Araport calls but now we're using the BAR service to get the data.

Next up:

1. Reading week + lots of midterms coming up. Will have to take a small hiatus from this project.
2. I plan to complete the webservice's RNA-Seq graph return portion by this weekend.
3. Then I will modify the front-end to make the AJAX calls & see how it works.
   a. This will be a prelim version that will not be user friendly (yet!).
4. I expect most of the kink to be worked out by mid-March to the point where one can ask someone to use the tool and give feedback.
   a. Leaves ~20 days to improve before the end of year!!

Date: February 17, 2016

Agenda:

1. Test out the new BAR-based gene structure web service written by Asher.
2. Work out a plan to avoid what happened between February 5 and today. Not much work was done at a level that would warrant a lab notebook entry (mostly admin stuff).
3. Look over Gini's work and respond to her email.

Protocol:

1. BAR Gene Structure Web Service Test:
    a. Understand what it outputs and the structure of the JSON object. How is it different from the previous version?
        i. Want a service that sends back all gene structure variants instead of just one
            1. Old (and current) service returns only one variant. New BAR service returns all different variants but they are all mixed together under one JSON group (refer to the email sent to AP and NP on Feb 17 for more details).
    b. Implement a stand-alone HTML page that calls the web service and shows the different gene structure variants in a single page.
2. Plan

Results:

1. BAR Gene Structure Web Service Test:
    a. The Araport service (araport11_gene_structure_by_locus v1.1) returns 6 unique splice variant structures for locus AT1G07350. Each variant's features are grouped under one JSON group.
    b. The BAR service returns all features of all 6 variants in one JSON group. This means that we can't figure out which feature belongs to which splice variant.
    c. Asked Asher to update the service to resolve this issue.
2. <See Trello>
3. Done.

Date: February 28, 29, and March 2, 2016

Agenda:

1. Develop the gene structure API using Asher's BAR web service that returns splice variants from Araport.
2. Develop RNA-Seq API on AWS.

Protocol:

1. Gene Structure API:
    a. See GitHub for code changes made.
    b. Get JSON from BAR web service with all splice variants.
    c. Figure out the start and end for the entire mRNA.
    d. Figure out start and end for each exon.
    e. Return all that information + base64 encoded image to show gene structure visually.
2. RNA-Seq API on AWS:
    a. As of Feb 28, the plan is to make 113 requests from front-end to the AWS for each RNA-Seq image…
    b. Figure out the number of mapped reads using `samtools view` and `wc -l` method.
    c. Return all information + RNA-Seq image encoded in base64.

Results:

1. Gene Structure API:
    a. Works beautifully as of March 2 @ 10:00 PM.
2. RNA-Seq API on AWS:
    a. Works too, but very slow as of March 2 @ 10:00 PM.

Date: March 3, 2016

Agenda:

1. Figure out why both API stopped working all of a sudden.

Result:

1. Araport was returning a 500 error and then 504 error because "were upgraded to close the DROWN vulnerability in SSL. Our Adama infrastructure, which hosts the web service adapters, did not come back up. I know that TACC folks were working on it last night but I don't see a fix in place yet" as per Jason Miller @ JCVI.

Date: March 4/5, 2016

Agenda:

1. Get SVG colouring code working at least partially by working out the FPKM code.

Protocol:

1. Partial SVG Colouring Code + FPKM Code:
   a. For now, the RNA-Seq API returns a random number instead of actual number of mapped reads for a gene…
   b. Using this mapped reads count, the absolute FPKM value is computed:

$$FPKM = \frac{\frac{(reads\ mapped\ to\ locus)}{(\frac{gene\ length}{1000})}}{(\frac{total\ reads\ in\ BAM\ file}{1000000})}$$

   c. Values are pushed into the `colours` array which stores [`target SVG ID, target SVG feature, absolute FPKM value, controls for the current BAM`].
   d. Once the entire DOM is populated by the data from BAM data XML, the average FPKM value for the controls is computed
      i. Go through all 113 elements of `colours`, and find the average controls FPKM value and store it in a separate attribute in `colours` array.
   e. For every element in colours, take log2 of the absolute FPKM to average controls FPKM ratio … this will be used to create the yellow-blue relative expression scale.
      i. When the user chooses to see the SVGs coloured by relative to each other, recolour all SVGs
      ii. As per NP: "Here the ratio will range from [0-1] for cases where the gene is expressed less in a sample after treatment, and [1-…] in cases where the expression level is increased. In log2, the range will be centred around 0 but will be negative if the expression is decreased, +ve if it's increased. Here the yellow-blue would denote negative values."
      iii. $Log_2\ FPKM\ Ratio = \log_2\left(\frac{sample\ absolute\ FPKM}{sample\ controls'average\ absolute\ FPKM}\right)$

    f.   Final `colours` array looks like `colours = [target SVG ID, target SVG feature, absolute FPKM, controls for the current BAM file, average controls FPKM, log2 ratio]`.

    g.   Now, go through each element of the `colours` array and colour SVGs appropriately

Results:

1.   Partial SVG Colouring + FPKM Code:

    a.   Good

    b.   Good

    c.   Good

    d.   Issue: need a signal that will call the colouring functions after ALL 113 rows have been populated properly

        i.   Tried to count the number of rows and then tried to call the colouring function BUT about half way down the table, Chrome would give a 'SVG not found' error.

        ii.   This error would not show up if the colouring function was manually called using the Update button. Suggests that the method described in (i) is still premature…

    e.   Problem: some BAM files in the BAM locator list controls but these control files do not have their own rows in the BAM locator leading to no average FPKM for the controls…

        i.   Other than this, the log2 ratio is calculated without issues

    f.   Good

    g.   Issue: there are some SVGs that are coloured gray … they look gray because of the blue-yellow hue. *{Is this okay? Should be…}*

Next Steps:

-   Consider shifting all calculations from JS to server side to make things cleaner

-   Remember to remove/add the factor of 10 on the temporary FPKM values that are generated

-   Consider shifting to 1 AJAX call to get all 113 RNA-Seq images from the current 113 AJAX calls design that gets 1 RNA-Seq image each…

Date: March 7, 2016

Agenda:

1. Come up with colours for the gene structure graph…
2. Sample data for gene structure API <no longer required, Araport is fully operational now>

Protocol:

1. Colours for gene structure graph: different parts of the data should be coloured differently to easily show what is where…
   a. *This is to see if we can solve the issue of not having enough exons in the gene graph after the switch to Araport GFF11 data by graphing additional parts of the gene variants.*
   b. *Previously, we may have 7-8 exons mapped for a gene; but now, some variants for the same gene are blank or have one huge exon … this does not make sense, so it is possible that according to the new standards, parts labelled CDS also have to be graphed to get the same/similar gene structure map.*
   c. Colours should be similar for similar gene parts; different for things that are radically different (i.e red for UTR but blue/green for CDS and exon).
   d. Gene parts & possible colours:
       i. 5' UTR: pink
       ii. Exon: blue
       iii. CDS: green (or light blue)
       iv. 3' UTR: orange
       v. Gene/mRNA: gets a solid 1px blue line vertically centered
           1. Assuming this is the entire mRNA…
2. Sample gene structure API
   a. No longer required since Araport is back up BUT saved a copy of the data returned for future outages

Results:

1.
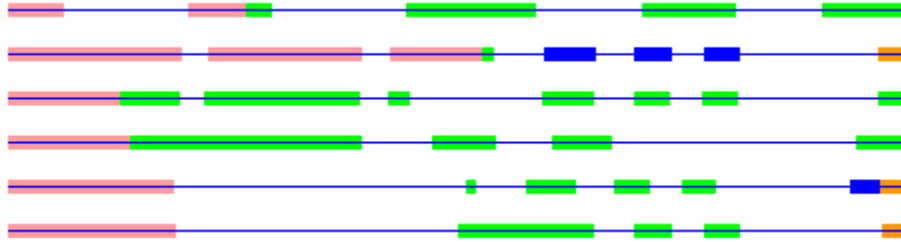
    a. Perhaps label too? Or should there be an index?

2. Done

Date: March 8, 2016

Agenda:

1. Check reliability of Araport and get the colouring demo ready for Dr. Provart.

Protocol & Result:

1. Reliability:
    a. NOT very reliable! Araport frequently responds with 504 error code. Extremely irritating now.
    b. Get rid of the reliance on Araport for now. Get sample data for AT2G24270 and update all API to return data for that locus.
    c. Get_gene_structures.cgi is now responding with constant data for that locus.
    d. AWSS is responding with dynamic BAM file data for that locus only!
        i. Actually, for colouring demo purposes, Araport is responding with static data… just to increase load speed.
    e. Both demo show colouring stuff fine right now.

Date: March 9, 2016

Agenda:

1. Clean up the get_gene_structures API
   a. Colour the gene structures images
   b. Fix the calculations for finding the starts and the ends of a locus.

Protocol:

1. get_gene_structures API:
   a. See March 8 for colour scheme.
   b. Problem space: currently the calculations assume that the start and end of the locus are the start and end of the first and last feature identified by Araport.
      i. However, this is wrong because it is possible that Araport does not explicitly identify all the features. For example, if the mRNA is (1000, 3000), and if Araport only tags two features: (2000, 2050) and (2500, 2600).
         1. In this case, the old calculations will "calculate" start and end as 2000 and 2600. However, the locus' true start and end SHOULD be 1000 and 3000.
         2. This issue is easily fixed by not calculating the start and end but instead by using what Araport provides!

Result:

1. get_gene_structures API:

    a. Old:
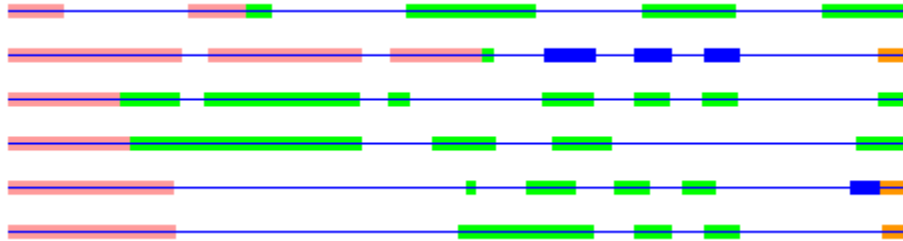
    b. New:
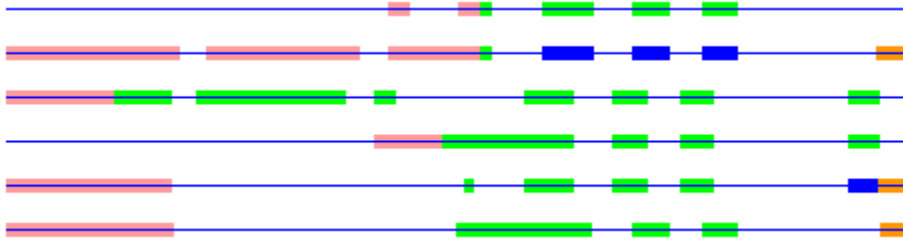
    c. The problem is clearly seen in the first variant. The old version would show the *zoomed in version…*

        i. *The image colours may change in future.*

Next steps:

- Colour the exons (blue) only above the midline; CDS (green), pink and orange (UTR) only below the midline to clearly visualize exons and other features.

    a. No other features overlap each other, exons overlap other features.

- May have to add the custom calculations for start/end again because it is possible that Araport start/end are WITHIN some of the identified features. Going back to the example above, a feature could be (900, 1050) which would fall outside of the Araport mRNA start/end of (1000, 3000).

Date: March 10, 2016

Agenda:

1. Clearly graph the different features like Araport does.
    a. Try to figure out the way Araport shows information.

Protocol:

1. Clearly graphing different features in get_gene_structures API.
    a. Increased height from 7 pixels to 8 pixels.
    b. Horizontal black line at y = height/2.
    c. All features labelled as "exon" will be graphed in blue from y = height/2 to 0 (i.e. above the horizontal black line).
    d. All features labelled as "CDS" will be graphed in green below the black line, "five_prime_UTR" will be graphed in orange, and "three_prime_UTR" in red.
    e. JBrowse shows exons and CDS features. Exons are the thick gray boxes, CDS features are the pink outlines. This is clear when you compare the API image and the JBrowse gene structures for locus = AT1G01010.

Results:

1. Clearly graphing get_gene_structures API:
    a. For locus = AT1G01010, API shows:

    b. JBrowse gene structure for comparison:
        i. Note that JBrowse does not show the UTR, also note how the green CDS features correspond to the pink outlines and the blue regions correspond to the thick gray boxes
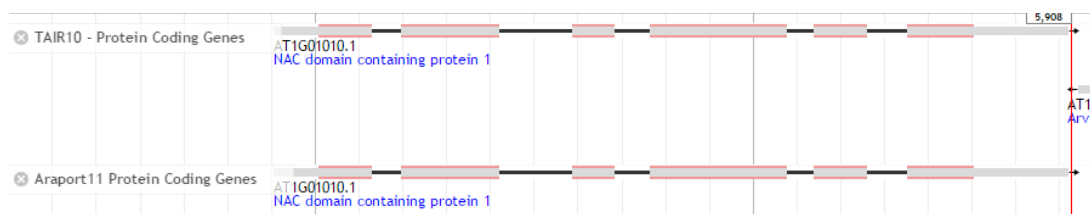
Date: March 10, 2016

Agenda:

1. Figure out why the Araport and the get_gene_structures API show different looking splice variants for genes like AT1G01020 and AT1G07350.

Protocol & Results:

1. Why do they look different?
    a. By clicking each variant shown on Araport, and manually looking at the features that are listed for that variant, it is seen that somehow, **JBrowse has more information** than what is retuned by the Araport Gene Structure web service.
        i. For example, JBrowse shows "exon" feature for all 6 variants of locus AT1G07350. However, Araport web service does not return "exon" feature for at least one of the 6 variants...
   b. There was another mistake in the old code ... the start and end of each variant were assumed to be the start/end of the "left/right most" feature retuned by Araport
        i. For example, if Araport returned 2 features (1000, 1200) and (1500, 1700), then start was 1000 and end was 1700. HOWEVER, if another variant returns two features (1100, 1300) and (1500, 1700) then graphing that with start = 1100 and end = 1700 would result in two images that won't show (visually) that the second (1500, 1700) feature was the same one. It would be slightly misaligned because of the different start positions. This was fixed by using the locus start and locus end which are the start/end of the left and right most features on ANY variant. In the scenario above, locus start would be 1000 and locus end would be 1700 - producing two image that would show that the second feature is the same one... This issue is clearly seen below in the top image and the fixed code result is shown in the bottom image.
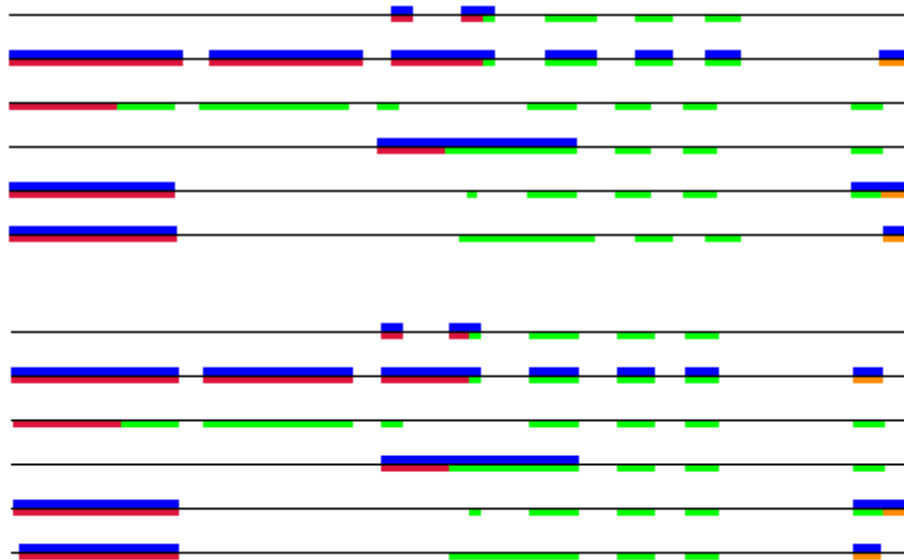
Top Image: misalignments seen clearly; bottom image is the fix applied.

ii. In short, the issue of misalignment is **because we are aligning to the variants and NOT the genome**. Consider the second and the third variants in the two images above. The second variant's exon and the third variant's CDS region (left side) are not aligned at the end. The CDS end is visually shown to be *earlier* than the exon's end. HOWEVER, the raw data show that they both end at nucleotide = 2257906. This is clearly and correctly shown in the bottom image.

Date: March 10, 2016

Agenda:

1. Improve RNA-Seq API
    a. Include Y-axis scale number which is the highest number of mapped reads for a particular nucleotide position

Procedure:

1. RNA-Seq API:
    a. Used python wrapper of GD library to insert a bit of text at the top-right corner, made it so that it can display a five digit number without cutting out.
    b. Added a line that shows a horizontal line that shows the highest mapped read …
    c. Better understood mpileup output format:
        i. Chr - Position - Base - Number of reads covering the base - Read bases - Base qualities
        ii. Heights at each nucleotide position = Number of reads covering the base – count of occurrences of "<|>"
            1. < and > denote the reference checks which are to be subtracted…
    d. Now using start and end of the locus to graph instead of the variant start/ends (reason same as the reason why the switch was made for gene structure graph).
    e. Now validating variant by first counting the number of variants returned by Araport and then making sure that the variant requested in the query does not exceed the count…

Results:

1. RNA-Seq API:
    a. For the most part, the visual did not change by much.
    b. Reason why 1(c)(ii) equation makes sense is seen if you do not subtract. The RNA-Seq images show stuff in the intronic region of the locus..!

Next Steps:

- RNA-Seq API should be one single call for all 113 images…
- RNA-Seq API calls the Gene Structures API but only to get the locus start/end. This information could be passed in as part of the query to the RNA-Seq API to speed up the API by ~5 seconds!

Date: March 11, 2016

Agenda:

1. Plan out the features for the UI.
2. Start making a better UI (create a new HTML page to access the browser).

Plan + Protocol:

1. UI Outline:
   a. Flow (what happens when the user clicks on a link to the tool?)
      i. Clicks link
      ii. Browser request multitrack-rnaseq.html which loads with data for default locus = AT2G24270.
      iii. Three AJAX requests are made for: BAM Locator XML, Gene Structures API, RNA-Seq images
      iv. The table is populated with the XML data, all rows are inserted with appropriate SVGs. Once populated, the table sorting is enabled.
      v. Once the gene structures API returns variants info, the first variant is inserted throughout the 113 rows…
      vi. When the RNA-Seq API returns all images, insert each in the appropriate position…
      vii. All SVGs are coloured by absolute FPKM
      viii. User can choose to sort, use a different variant for visual comparison, search a different locus, visualize expression in absolute or relative modes, choose to set a different maximum value for the y-axis scale
   b. Features (what capabilities will the user have?)
      i. <TO DO>
2. Better Version of Multi-track RNA-Seq Browser:
   a. Took parts of the old version and copy-pasted only the code that makes sense to the new version.

Date: March 18, 2016

To Do List:

1. Figure out where JBrowse is getting its gene structure information from <**DONE**>
2. Write an outline for the presentation + thesis paper
3. Colour RNA-Seq and Gene Structure images differently <**DONE**, IPR>
4. Implement SVG colouring on front-end
5. Implement PCC and FPKM calculations
6. Confirm the RNA-Seq API's output <**DONE**>
   a. Perhaps draw a vertical line at a specific nucleotide position and confirm that the two images are aligned correctly
   b. Ask NP what "reference skip" means and if it makes sense to subtract this from the number of mapped reads.
7. Give more options to user at the front-end (y-axis scale, etc.)
8. ALLOW THE USER TO RELOAD THE PAGE FOR ANY LOCUS! <**DONE**>

Date: March 22, 2016

Agenda:

1. Figure out where JBrowse is getting its gene structure information from
2. Colour RNA-Seq images based on the assigned colours for each record
3. Confirm RNA-Seq API's x-axis output
   a. To see if the RNA-Seq image aligns with gene structure image
4. Update front-end to allow user to update everything using the input box

Protocol:

1. JBrowse: link.
   a. This link shows details of each JBrowse track. Search for "protein coding" and find the 'Araport11 Pre-release 3' track for Araport11 Protein Coding Genes
   b. Araport web services page shows a particular service with description as "this service produces a JSON object in the format JBrowse requires"
2. Colour RNA-Seq images based on assigned colours for each record
   a. Used XPath query to extract out the `url` and the `foreground` attributes from each `<file>` elements from the BAM data XML file.
   b. Then used Sublime Text to format it correctly in the format required for creating a python dictionary variable (`dict = {'record' : 'colour HEX', …}`).
   c. In the webservices.cgi file, a function was copy pasted from stack overflow to covert a HEX value into RGB tuple(int, int, int).
   d. Whenever the code was drawing the rectangles, the colour was set to `dict[record]`.
3. Confirm RNA-Seq API output: want to see if the nucleotide positions in the RNA-Seq images correspond width-wise to the nucleotide positions in the gene structure API
   a. Code can be found by browsing GitHub on today's commits
   b. The code, which was deleted in a later commit after success, basically draws vertical red lines at certain known nucleotide positions (these positions are the starts or the ends of an exon/CDS/UTR) to see if the vertical lines line up to the gene structure image which is just below it...
4. Front-end: allow user to update the whole page for a different locus

a. Copy pasted a function from Stack Overflow to cancel existing XHR requests

b. Repopulated the table to destroy everything associated with old locus

c. Make brand new 114 requests (1 gene structure request + 113 RNA-Seq requests)

Results:

1. JBrowse uses the web service that Vivek sent but it MOST LIKELY uses it indirectly.

   a. JBrowse uses a second web service that takes the first's output and makes it compatible to JBrowse input.

   b. However, this still doesn't explain how they have more information about exons/CDS than the info returned by the first web service that Vivek told us to use.

   c. Manually checked to find that the JSON service from which JBrowse (apparently) gets its data returns the same data. But then, somehow, JBrowse shows exons that were not returned and maps them on its front-end along with CDS regions.

   d. It is possible that JBrowse assumes regions that are labelled just CDS are the same as exons...

2. Colour RNA-Seq images differently = GOOD

3. Confirm RNA-Seq API output = GOOD

   a. Everything is aligned correctly

4. Front-end: update page with info for a second locus

   a. Works but it seems choppy and there is no progress bar yet

   b. Almost certain there is a better method to do this than what is currently implemented

To Do List:

1. Upload our BAM and BAI files to JBrowse to confirm RNA-Seq image: https://www.araport.org/help/faq#t349n244

   a. To see if the image produced by the RNA-Seq API is the same as what would be produced by JBrowse.

Date: March 23, 2016

Agenda:

1. Modify RNA-Seq API to accept custom y-axis scaling

Protocol:

1. Custom y-axis scaling:
    a. Accepting an optional y-axis scale parameter in the RNA-Seq API
        i. If not specified, then the value is set to -1
        ii. If specified, the value is set to whatever is specified
    b. Highest_mapped_reads_count is set to the value specified

Result:

1. Custom y-axis scale:
    a. If no value is specified, the max read mapped count is used to set the y-axis scale.

    b. If a value is specified (specified 1500 in this case), it is used to set the y-axis scale.

Date: March 27, 2016

Agenda:

1. Move the absolute FPKM calculations server side.

2. Clean up the front-end.

Protocol:

1. Server side absolute FPKM calculations.

    a. Modified the XPath code on CDF server to extract record and number of reads
        mapped for that record.

    ```
    <files>
    {
            for $file in doc("bamdata_amazon_links.xml")//file
            return
            <file>
            {
                    <url>{data($file/@url)}</url>
            }
            {
                    <reads_count>{data($file/@numberofreads)}</reads_count>
            }
            </file>
    }
    </files>
    ```
    b. Created a python dictionary in webservice.cgi file.

    c. $FPKM = \dfrac{\dfrac{(reads\ mapped\ to\ locus)}{(\frac{gene\ length}{1000})}}{(\frac{total\ reads\ in\ BAM\ file}{1000000})}.$

    d. Returning the number of reads mapped AND the absolute FPKM value in the JSON.

2. Clean up front end.

    a. Moved the options in the same <div> as the variants. Variants occupy the left half…

    b. Flow

        i. Populate the input fields with correct default values

        ii. Populate table without gene structure and without RNA-Seq images

        iii. Load default gene's variant images

        iv. Load default gene's RNA-Seq images

Date: March 27, 2016

Agenda:

1. Get the multi-track RNA-Seq browser ready for presentation tomorrow.

Protocol:

1. Making the multi-track RNA-Seq browser presentable.
    a. Added non-functional Help and Cite buttons. At the end of the project, these will become functional with information.
    b. http://ec2-52-70-232-122.compute-1.amazonaws.com/~ppurohit/RNA-Browser/cgi-bin/webservice.cgi?tissue=light&record=SRR653575&locus=AT2G24270&variant=1&start=10326918&end=10330048 produces an error.
    c. Added inputs to let user change Y-axis scale, RPKM max value, SVG colouring mode.
    d. `colours` array index and values it holds:
        0. record + "_svg"
        1. 'Shapes'
        2. Controls
        3. absolute-fpkm
        4. average_controls_fpkm
        5. relative-fpkm
    e. Changed the colours array so that specific values are always stored at a specific index. Previously, it was just being appended to the end. This was causing problems when you load another locus…
    f. Implemented the option to let user change the eFP-RPKM scaling. Takes effect immediately, this could be a problem if the user changes it while the 113 requests are not complete because then SVGs will be scaled to different max values.
        ▪ Disable this option until all requests have been processed?
    g. Implemented a progress bar at the top (always shows up).
    h. Changed the gene structure colouring to match NCBI colour scheme. Needs more work.

i. Implemented basic table sorting code but needs more work. The update trigger doesn't seem to work. Code is thus commented out for now.

Result:
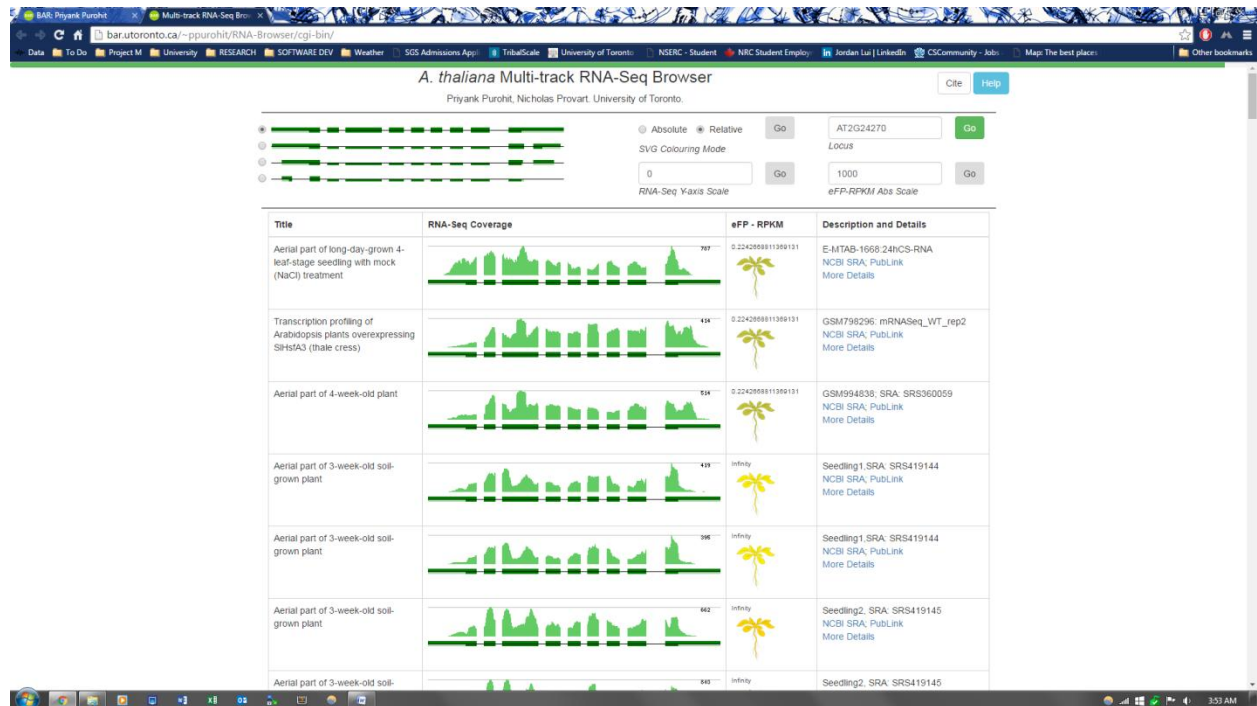
Date: April 3, 2016

To Do List:

1. Implement at least basic PCC calculations
2. Gene structure: make the structure 5' to 3' from left to right
3. Gene structure: fix the colour shades in the image, and the drawing shapes…
4. Show colour legends for absolute and relative SVG colouring scales
5. Implement custom y-axis scaling (default should be the max value, 0 means the max read mapped…).
6. Speed up default gene AT2G24270's display (can deliver the data faster AND somehow have a way for user to know this was the case OR regenerate data on server side every day?).
7. Figure out why the one BAM file doesn't get called
8. Sorting feature
9. Redesign "Descriptions and Details" column
10. Show the PCC and RPKM values on hover
11. When you press GO to change absolute RPKM scale, the radio button should also change to absolute mode if it is in relative mode…
12. Create a DTD for BAM locator XML file

Date: April 15, 2016

Agenda:

1. Create a DTD for the BAM locator XML file.

Protocol:

1. Create a DTD for the BAM Locator XML
    a. Working on CDF server because BAR servers don't have the xmllint program required for DTD-XML validation purposes
    b. Changed the element and attribute names to make things more clear
    c. Created a DTD that requires at least one bam_file, control, and groupwith element...
    d. Ran `xmllint --valid bamdata_amazon_links.xml > xmllint.txt`.
        i. Clearly outputs errors that occur throughout the file. Otherwise, the errors near the beginning of the file would get hidden because of the line limit of the shell

Results:

1. DTD for BAM locator XML file:
    a. XML was validated successfully with the DTD created.

Date: April 17, 2016

Agenda:

1. Code optimization
2. UI/UX improvements
3. PCC Implmentation

Protocol:

1. Code optimization
   a. Looking into modifying SAMtools source code to include the number of reads mapped to locus in the `mpileup` command.
   b.
2. UI/UX improvements
3. PCC Implmentation