A multi-track RNA-seq browser for visualization of Arabidopsis thaliana transcription patterns from different growth states and conditions.

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.

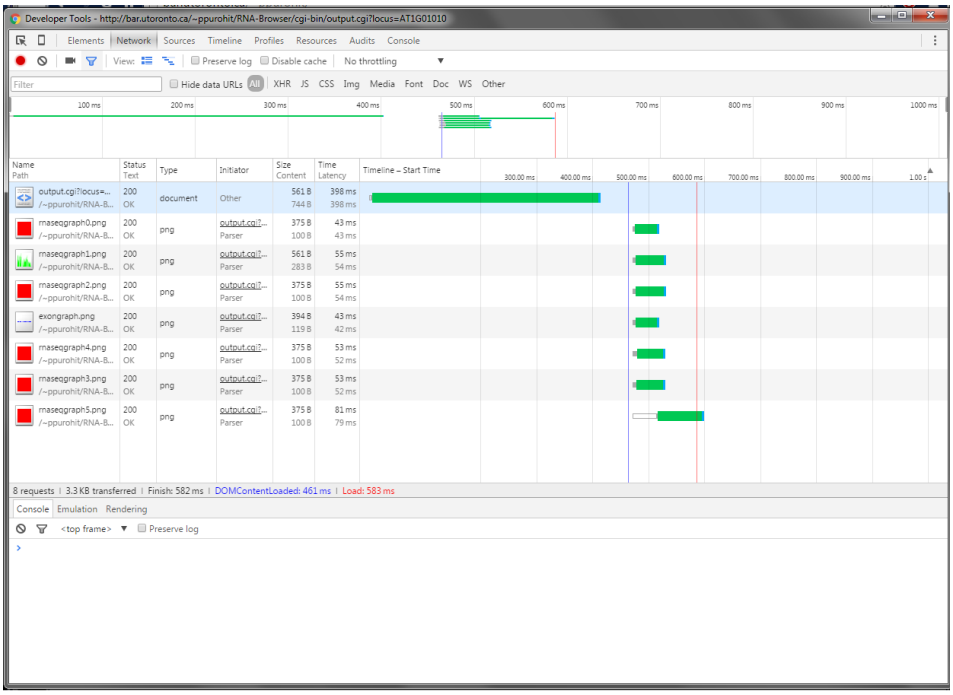


Figure 1: ~600ms for local BAM file.

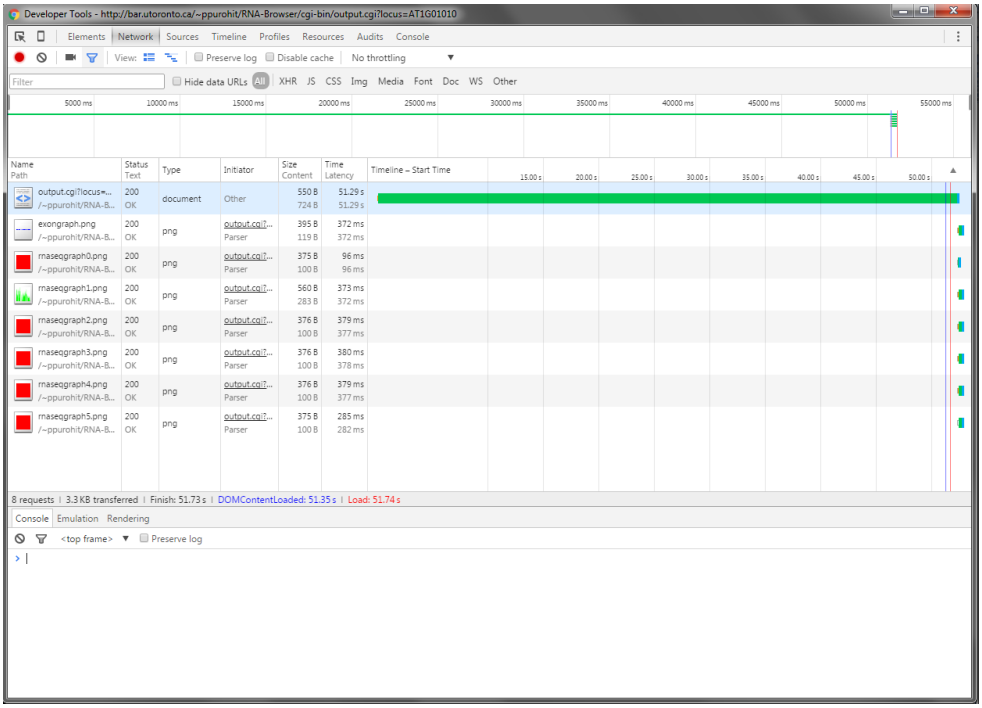


Figure 2: ~60 seconds for iPlant BAM file.

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.
    - … 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

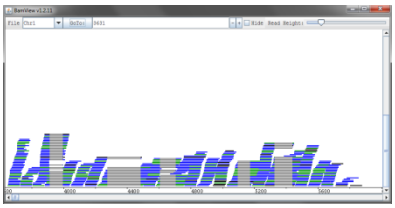
i. `./samtools mpileup http://s3.amazonaws.com/iplant-cdn/iplant/home/araport/rnaseq_bam/leaf/SRR446034/accepted_hits.bam -r Chr1:3631-5899 -d 8000`

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_SRR1019457_accepted_hits.bam
        iii. dark_SRR495258_accepted_hits.bam
        iv. dark_SRR495257_accepted_hits.bam
        v. root_SRR514814_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_ERR274510_accepted_hits.bam
        xiv. aerial_SRR847503_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

i. Figure 3: SRR446034 BAM file's first locus in BamView program.



d. GOT BACK DATA! For some reason this works and returns data from that same BAM file.

i. But doesn't work for another experiment, SRR949989's BAM file…

3. Rerun the shell script from the same directory as the latest SAM Tools …

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. **NEED TO VERIFY THIS ASSUMPTION**.
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

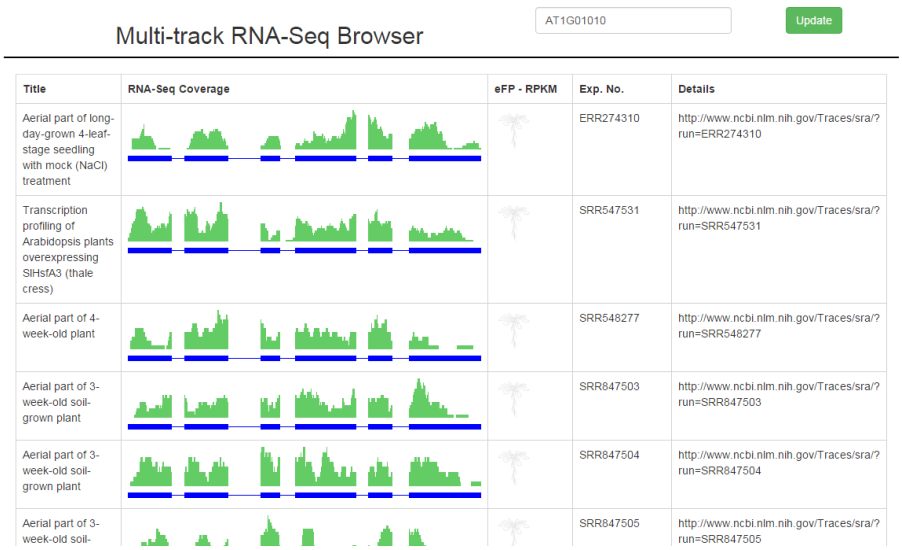
Agenda:

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.