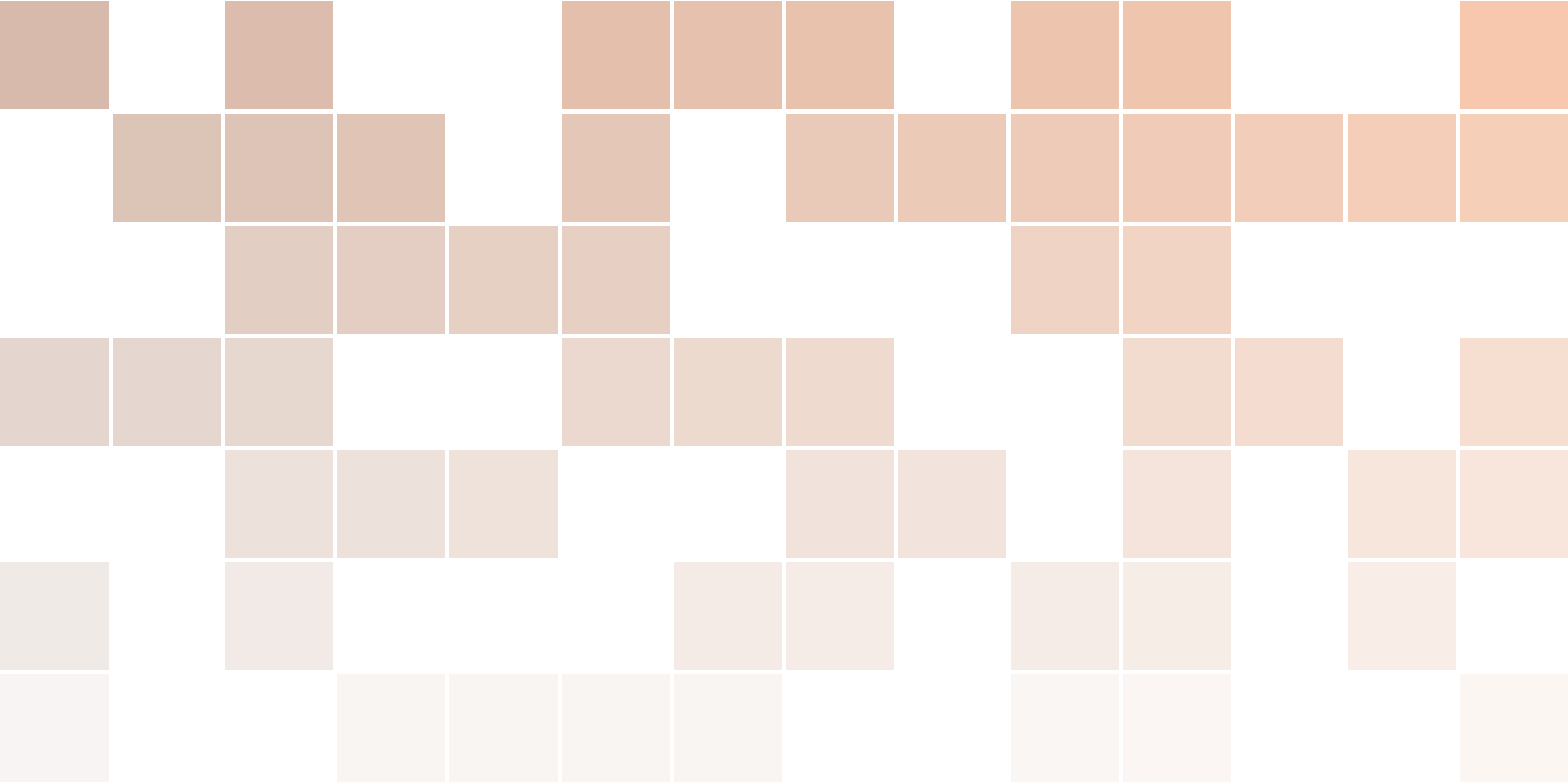
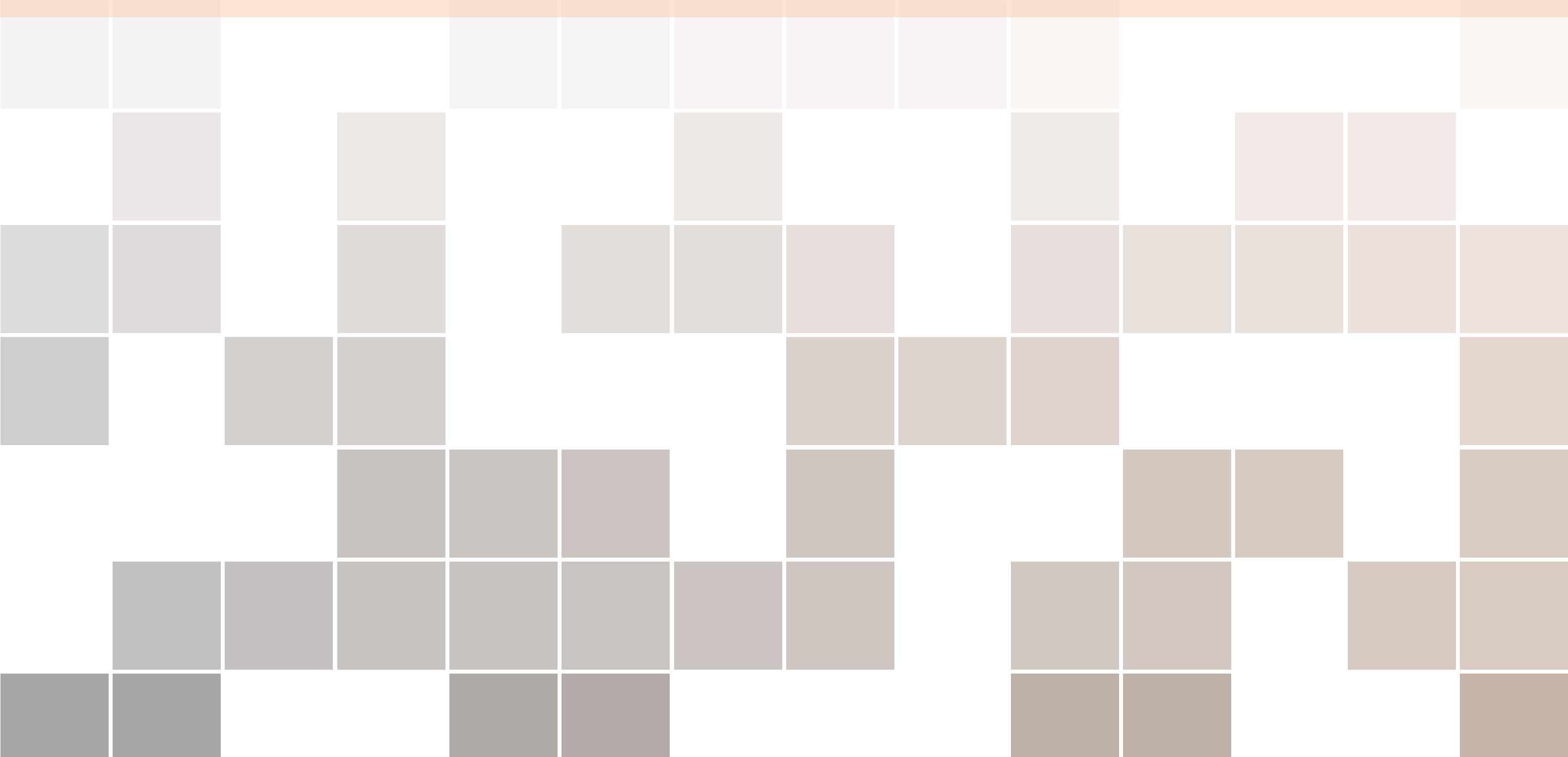


DEEPN 1.0

Illumina Data Processing

Dr. Venkatramanan Krishnamani
Dr. Robert C. Piper
Dr. Mark Stammnes



THIS SOFTWARE IS PROVIDED UNDER “THE MIT LICENSE” (MIT). *See Section 1.5*

DEPARTMENT OF MOLECULAR PHYSIOLOGY AND BIOPHYSICS, UNIVERSITY OF IOWA

HTTPS://GITHUB.COM/EMPTYEWER/DEEPN/RELEASES

This research was performed with the support of a grant awarded to Dr. Robert C. Piper by National Institute of Health (5R01GM058202).

First release, January 2016

Contents

I DEEPN Overview

1 DEEPN Overview . . . 7
1.1 About DEEPN 7
1.2 Contents within DEEPN 8
1.3 DEEPN WorkFlow Overview 9
1.4 Installation 9
1.4.1 Download Link . . . 9
1.4.2 Mac OS X Compatibility . . . 9
1.4.3 Windows Compatibility . . . 9
1.4.4 Linux Compatibility . . . 9
1.5 Open Source License 9

2 Initial Setup . . . 11
2.1 Preprocessing `.fastq files` 11
2.2 Initializing DEEPN 11
2.2.1 Launching . . . 11

II Processing Data

3 Gene Count . . . 17

4 Junction Make . . . 19

5 Running Gene Count & Junction Make 23

III Analyzing Data

6 Blast Query 27
6.1 Results Tab 28
6.2 Filtered Results Tab 29
6.3 Plot Tab 29
6.4 Export for Graphing 29
7 Read Depth 31
Index 33

I

DEEPN Overview

1 **DEEPN Overview** 7
1.1 About DEEPN
1.2 Contents within DEEPN
1.3 DEEPN WorkFlow Overview
1.4 Installation
1.5 Open Source License

2 **Initial Setup** 11
2.1 Preprocessing `.fastq files`
2.2 Initializing DEEPN

1. DEEPN Overview

1.1 About DEEPN

The **DEEPN** bioinformatics workflow is a collection of 4 programs

Gene Count counts the number of sequence reads found for every gene.

Junction Make finds and identifies all the sequences corresponding to the junction sequences that span the end of the bait plasmid with the cDNA insert in the library "prey" plasmid.

Blast Query allows the junction sequences to be analyzed.

Read Depth calculates the read depth for a particular cDNA, useful for predicting the 3' end of a cDNA insert.

DEEPN was developed to process and analyze sequence information from the Illumina platform that produces 110-140 bp reads. Both single and paried-end sequences are appropriate, **DEEPN** considers the different sides of a paried-end sequence as two separate sequences. **DEEPN** requires sequence files in `.sam` format, in which sequences have been mapped to the genome. Processing of .fastq sequence files with `Tophat2` will work, producing unmapped and mapped `.sam` files for each fastq read file. **DEEPN** requires BOTH mapped and unmapped `.sam` files to fully analyze a sequence dataset. See Section 1.4.1 for download link.

Later releases of DEEPN for Mac will also contain functions to automatically map `.fastq` files with Tophat2 to allow for seamless integration of processing sequence data.

`.fastq` → Tophat2/Bowtie → Gene Count → Junction Make → Blast Query → Read Depth.

The **DEEPN** application provides a graphic user interface to guide the launch and operation of Gene Count , Junction Make , Blast Query , or Read Depth modules within it. **DEEPN** comes in versions that run on Windows and Mac operating systems. See Section 1.4.1 for download link.

This user guide describes use of the standalone **DEEPN** application and how to operate the modules within it.

1.2 Contents within DEEPN

1. Program modules
    - Gene Count
    - Junction Make
    - Blast Query
    - Read Depth

   which are launched from within the main DEEPN.app or DEEPN.exe.
2. Databases for the Gene and mRNA coordinates
    - mouse mm10 genome and mouse RefSeq data
    - Gene and ORF coordinates for the SacCer3 genome

   These allow analysis of mouse cDNA Y2H libraries and yeast genomic Y2H libraries. The mouse RefSeq database that **DEEPN** uses contains just the known annotated mRNAs, basically the entries that have an `NM_*` nomenclature in genbank. It does not contain microRNAs, long non-coding RNAs, and theoretical splice variants. **DEEPN** contains a database of yeast genes, with a hybrid nomenclature of their systematic SGD name and their genebank `NM_*` nomenclature. For simplicity, a given yeast gene consists of the protein coding sequence flanked by 100 bp of untranslated region.
3. Database of "junction tags" for different libraries. Currently, analysis of the mouse mm10 data defaults to the use of the Clontech mate/plate pGADT7 plasmid. And analysis of the yeast libraries is tied to the Phil James libraries housed in pGAD-C1, C2, and C3. These default junction tags are:
    - cDNA insert (mouse)

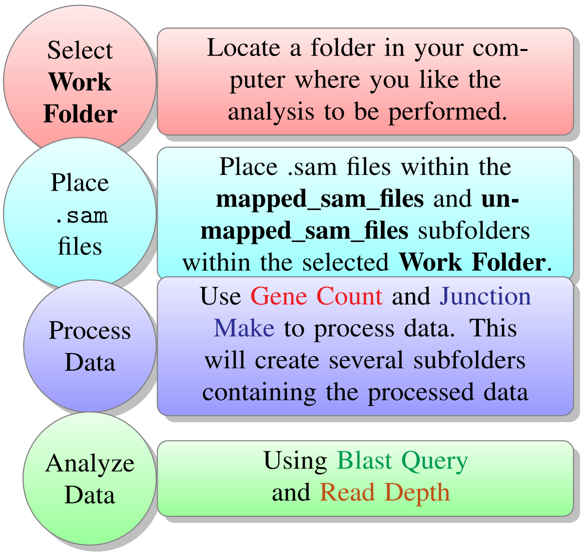
      ```
      AATTCCACCCAAGCAGTGGTATCAACGCAGAGTGGCCATTACGGCCGGGG
      ```

    - genomic fragment insert (*S. cerevisiae*)

      ```
      ATACCCCACCAAACCCAAAAAAAGAGATCGAATTCCCCGGGGGATCCATC
      ```

   Users can insert their own junction sequence into the **DEEPN** dialog box if using a different library. For a more permanent solution, users can modify the SQL database that houses these data (see below)
4. **DEEPN** operates the `blastn` program while its processing data. That is called upon by the Junction Make program. All of the relevant files required to blast search mouse mRNAs or yeast genes are included in internal resources. Stand-alone Blastn program and associated databases to perform `blastn` locally from within the **DEEPN** application.

1.3 DEEPN WorkFlow Overview



Locate a folder in your computer where you like the analysis to be performed.

Place .sam files within the **mapped_sam_files** and **unmapped_sam_files** subfolders within the selected **Work Folder**.

Use Gene Count and Junction Make to process data. This will create several subfolders containing the processed data

Using Blast Query and Read Depth

Step-by-step screen-shots and instructions are detailed in the following chapters.

1.4 Installation

1.4.1 Download Link

Platform-specific compiled binaries (*Mac OS X, Windows and Linux*) of **DEEPN** can be downloaded from the below URL.

`https://github.com/emptyewer/DEEPN/releases`

1.4.2 Mac OS X Compatibility

`Mac OS X (10.10+) Yosemite and above`

1.4.3 Windows Compatibility

`64-bit or 32 bit Windows 7 and above. Note that DEEPN itself is a 32-bit software.`

1.4.4 Linux Compatibility

`Scheduled for release in Version 2.0 of DEEPN.`

1.5 Open Source License

The MIT License (MIT)

Copyright (c) 2016 Venkatramanan Krishnamani, Robert C. Piper, Mark Stammnes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. Initial Setup

2.1 Preprocessing `.fastq files`

The current DEEPN application requires that `.sam` files have been generated from the `.fastq` illumine sequence files. This is done using the mapping program `Tophat2` that uses `Bowtie`. It is imperative that downstream processing by DEEPN uses the same databases that `Tophat2` uses to map the sequence files. These are...

1. Mouse: mm10GRCm38 2011 *Mus musculus* assembly (Genome Reference Consortium Mouse Build 38 (`GCA_000001635.2`)
   `https://goo.gl/T6OT2F`
2. Yeast: sacCer3 2011 *Saccharomyces cerevisiae* S288c assembly from Saccharomyces Genome Database (`GCA_000146055.2`)
   `https://goo.gl/wfPbvA`

`Tophat2` should produce sets of .sam files of Mapped Reads and Unmapped Reads for every input .fastq file. DEEPN will use both of these files.

2.2 Initializing DEEPN

2.2.1 Launching

Open the **DEEPN** application by double clicking. This opens a window (DEEPN) that can be used to run the other modules.

Step 1. Select Parameters from the list menu in the top. Figure 2.1
  - Selecting the M. musculus option selects the mm10 mouse databases
  - Selecting the S. cerevisiae option selects the sacCer3 databases
  - Once this is selected, the “Select Work Folder” will be activated for use

Step 2. Create a work folder Figure 2.2
  - **DEEPN** needs a folder to write its files into and to read sequence files from. This is done using the “Select Work Folder” button Select Work Folder... . Here create a new folder or select an existing one.

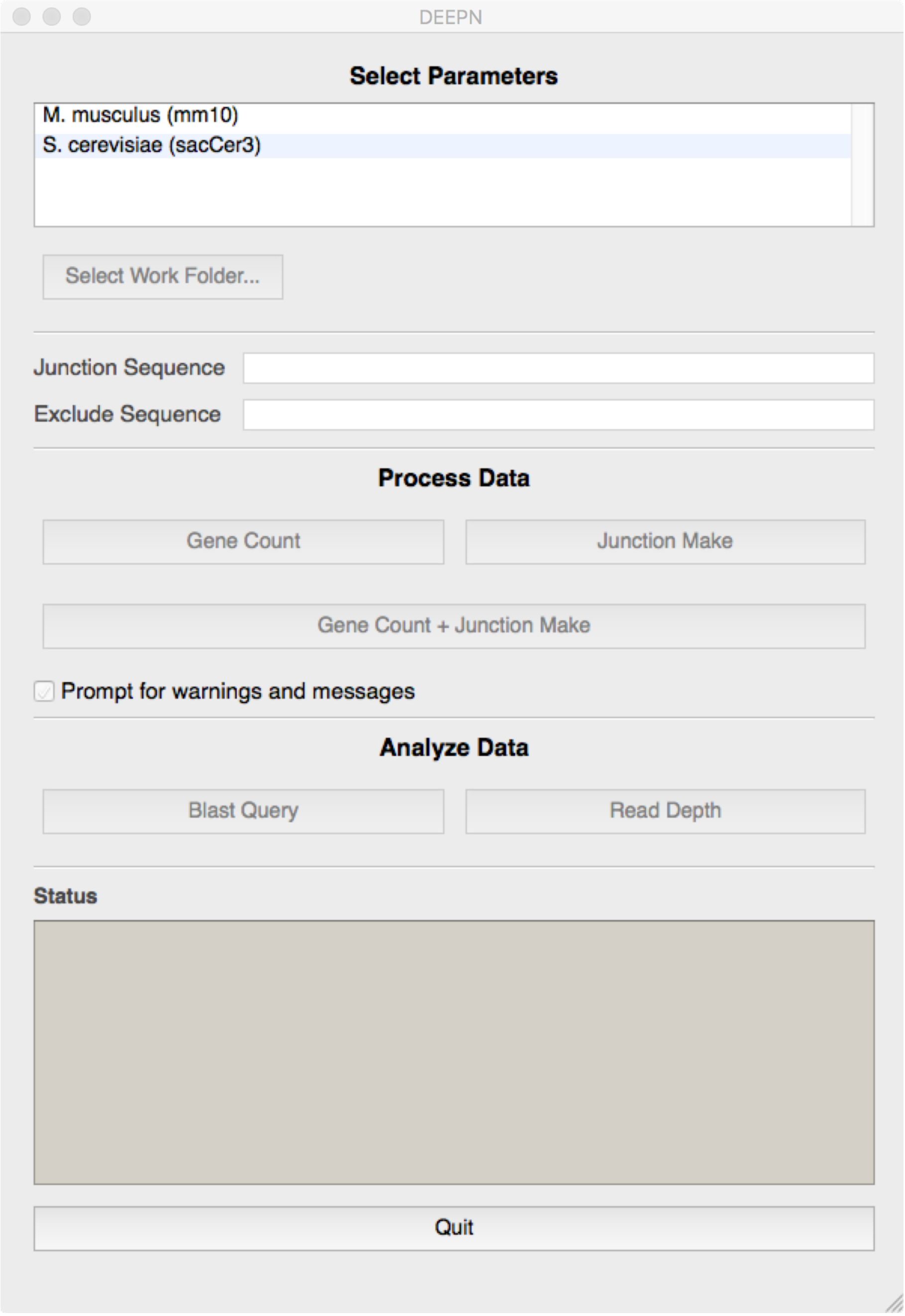
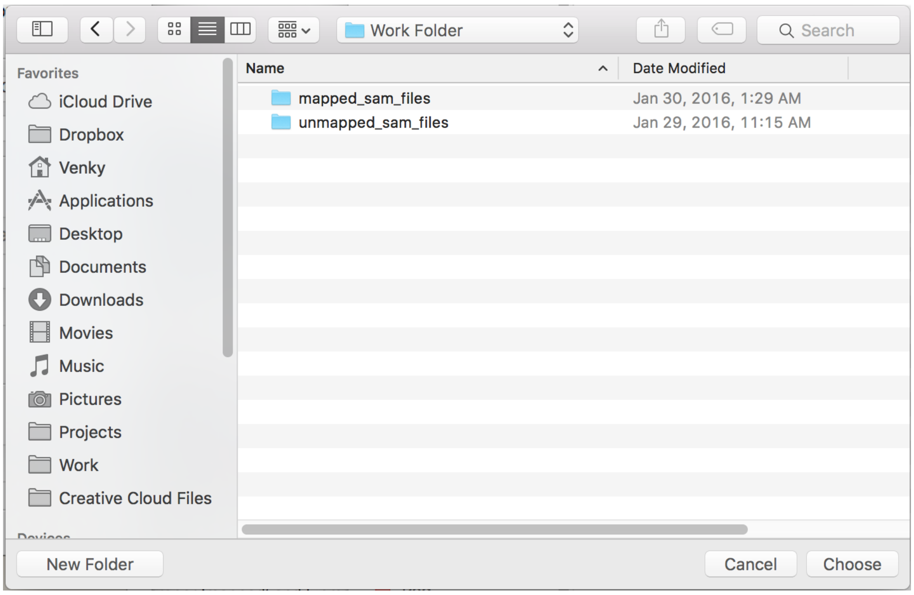

DEEPN
Select Parameters
M. musculus (mm10)
S. cerevisiae (sacCer3)
Select Work Folder...
Junction Sequence
Exclude Sequence
Process Data
Gene Count
Junction Make
Gene Count + Junction Make
☑ Prompt for warnings and messages
Analyze Data
Blast Query
Read Depth
Status
Quit

Figure 2.1: **DEEPN** main interface.

- Once your Work Folder is designated, **DEEPN** will need to operate from two subfolders within it (See Figure 2.2). These folders are called...
  - `mapped_sam_files`
  - `unmapped_sam_files`
- If these folders already exist within the "Work Folder" because of previous processing, then **DEEPN** will use them.
- If the "Work Folder" is new and those folders do not exist, **DEEPN** will create them.

Step 3. To start things off, move your `.sam` files generated by Tophat2 into the `mapped_sam_files` and `unmapped_sam_files` folders.


Work Folder
Search
Favorites
iCloud Drive
Dropbox
Venky
Applications
Desktop
Documents
Downloads
Movies
Music
Pictures
Projects
Work
Creative Cloud Files
Devices
New Folder
Cancel
Choose

Name	Date Modified
mapped_sam_files	Jan 30, 2016, 1:29 AM
unmapped_sam_files	Jan 29, 2016, 11:15 AM

Figure 2.2: Folder with two subfolders, named `mapped_sam_files` and `unmapped_sam_files` is the starting state of **DEEPN** work folder before processing.

Gene Count module will process the `.sam` files placed in the `mapped_sam_files` folder. These files should contain the mapped read files outputted from Tophat2

Junction Make module will process the `.sam` files placed in the `unmapped_sam_files` folder. These files should contain the UNmapped read files outputted from Tophat2. These are the reads that were unable to to mapped adequately to the **SacCer3** or the **Mm10** genomes and that contains the bulk of junction reads. With Illumina 110-120 bp reads, the stretch of cDNA or gene DNA in these "Junction sequences" is too short to be mapped to a chromosome by Tophat2. This workflow assumes these types of short reads. Were one to have longer reads, the Junction Sequences might be able to be mapped, which would oblige the search for them to included the Mapped reads as well.

- Once `.sam` files are placed within `mapped_sam_files`, the Gene Count button is activated and the Gene Count processing can begin by clicking the button.
- Once `.sam` files are placed within `unmapped_sam_files`, the Junction Make button is activated and the Junction Make processing can begin by clicking the button.

A warning message may appear if **DEEPN** detects folders created by previous processing runs. **DEEPN** will add to these folders, but users run the risk that if file names are the same, the old files will be written over by the new files. To avoid any problems, one can move the processed data folders to a new location.

II

Processing Data

3 Gene Count 17

4 Junction Make 19

5 Running Gene Count & Junction Make 23

3. Gene Count

Gene Count will process all the `.sam` files that are in the folder `mapped_sam_files`

Once the `.sam` files are moved to this folder, click the Gene Count button.

Clicking the Gene Count button will only be possible if there are files in `mapped_sam_files` folder.

After starting, Gene Count will report to you the following:

```
>>>GENEcountY2H
Gene Count will process the mapped .sam files present in the folder
    mapped_sam_files
Gene Count will generate two folders for its output data:
gene_count_summary contains a summary files of genes and their count frequency.
chromosome_files contains more granular data for each gene.

Be patient....This program is slow but will keep you posted.
>>>END
```

Gene Count will populate the `gene_count_summary` and `chromosome_files` folders with files that have names corresponding to the input files.

For an input file named `Dataset1.sam`

- The `gene_count_summary` folder will contain `Dataset1_summary.csv`
- The `chromosome_files` folder will contain `Dataset1_ChrGene.csv`

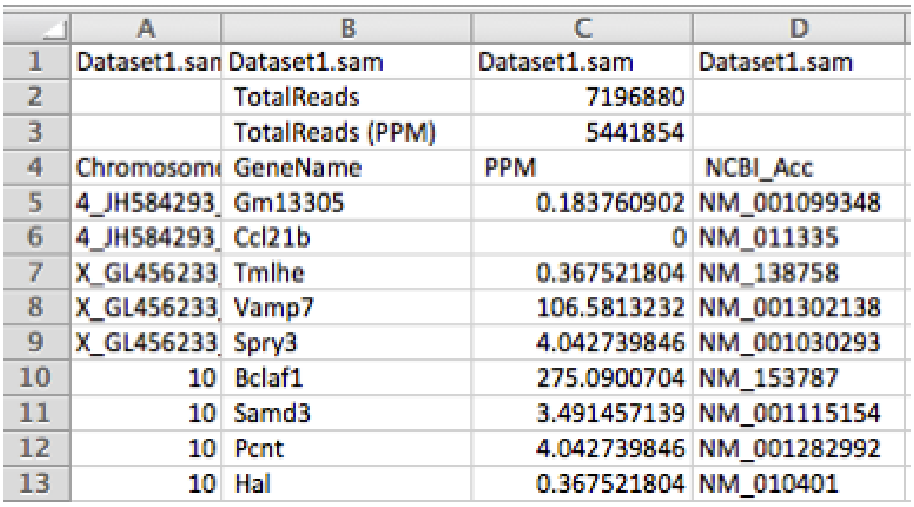
The `_summary.csv` files generated by Gene Count have the following format when opened in `Microsoft Excel`. See Figure 3.1.

- The name of the `.sam` file processed is found along the top.
- **Column A** shows Chromosome on which each gene is located

- **Column B** shows gene name
- **Column C** shows the frequency of reads for that gene in parts per million (PPM)
- **Columns D and greater** show corresponding NCBI genbank accession numbers that describe annotated mRNA sequences for that gene

Shown are the “Total Mapped Reads” that were found in the starting `.sam` file (TotalReads). Also shown are the “Total Mapped Reads” that were used in the PPM calculation (TotalReads(PPM))

TotalReads(PPM)) is the number of reads that were found to have a position corresponding to a known exon as annotated in the corresponding database used. Since some exons have yet to be annotated, some of the reads may not be able to be assigned to a particular gene, which accounts for the discrepancy between TotalReads(PPM) and TotalReads.



	A	B	C	D
1	Dataset1.san	Dataset1.sam	Dataset1.sam	Dataset1.sam
2		TotalReads	7196880	
3		TotalReads (PPM)	5441854	
4	Chromosom	GeneName	PPM	NCBI_Acc
5	4_JH584293_	Gm13305	0.183760902	NM_001099348
6	4_JH584293_	Ccl21b	0	NM_011335
7	X_GL456233	Tmlhe	0.367521804	NM_138758
8	X_GL456233	Vamp7	106.5813232	NM_001302138
9	X_GL456233	Spry3	4.042739846	NM_001030293
10	10	Bclaf1	275.0900704	NM_153787
11	10	Samd3	3.491457139	NM_001115154
12	10	Pcnt	4.042739846	NM_001282992
13	10	Hal	0.367521804	NM_010401

Figure 3.1: Screen-shot of `_summary.csv` file generated by Gene Count .

4. Junction Make

Junction Make will process all the `.sam` files that are in the folder `unmapped_sam_files`.

Once the `.sam` files are moved to this folder, click the Junction Make button.

R Clicking the Junction Make button will only be possible if there are files in `unmapped_sam_files` folder.

Junction Make will report to you the following:

```
>>>Comment1
- Make sure all ".SAM" files from your UNmapped reads are in the folder:
unmapped_sam_files
- This program will scan for junction sequences that span the Gal4 activation
    domain and the prey.
- The junction tag sequence used will be the one entered in the Junction
    Sequence textbox
- Output files will be placed in the junction_files folder as .junctions.txt
    files.
- Blast identified reads will be placed in the blast_results folder as
    .blast.txt files
- A database of identified junctions will be placed in the blast_results_query
    folder as .p files
>>>END
```

Junction Make will make the `junction_files`, `blast_results`, and `blast_results_query` folders to accept the new files it will produce. If these folders already exist, a warning will be issued to alert the user that files might be overwritten if Junction Make is run again. To avoid this, move the files out of the `junction_files`, `blast_results`, and `blast_results_query` folders to a new place or Abort, rename the `junction_files`, `blast_results`, and `blast_results_query` folders and start Junction Make again.

Junction Make will look for different junction “tag” sequence. When different `Gal4AD-` libraries are used, here is where some user input may be necessary.

For the **clontech mate and plate library** the junction “tag” sequence looked for is:

```
AATTCCACCCAAGCAGTGGTATCAACGCAGAGTGGCCATTACGGCCGGGG
```

So Junction Sequences look like this for the mouse cDNA Mate/Plate pGADT7 library (Figure 4.1):

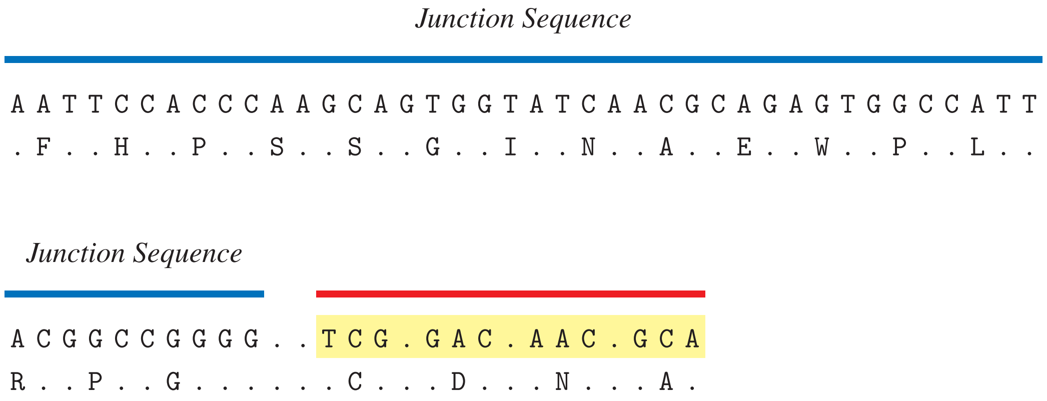

Junction Sequence
A A T T C C A C C C A A G C A G T G G T A T C A A C G C A G A G T G G C C A T T
. F . . H . . P . . S . . S . . G . . I . . N . . A . . E . . W . . P . . L . .
Junction Sequence
A C G G C C G G G G . . T C G . G A C . A A C . G C A
R . . P . . G . . . . C . . . D . . . N . . . A .

Figure 4.1: Junction Sequences look like this for the mouse cDNA Mate/Plate pGADT7 library

For the **yeast genomic Phil James (pGAD-C1,2,3) library** the junction “tag” sequence looked for is:

```
ATACCCCACCAAACCCAAAAAAAGAGATCGAATTCCCCGGGGGATCCATC
```

So junction sequences look like this for the pGAD-C yeast genomic library (Figure 4.2):

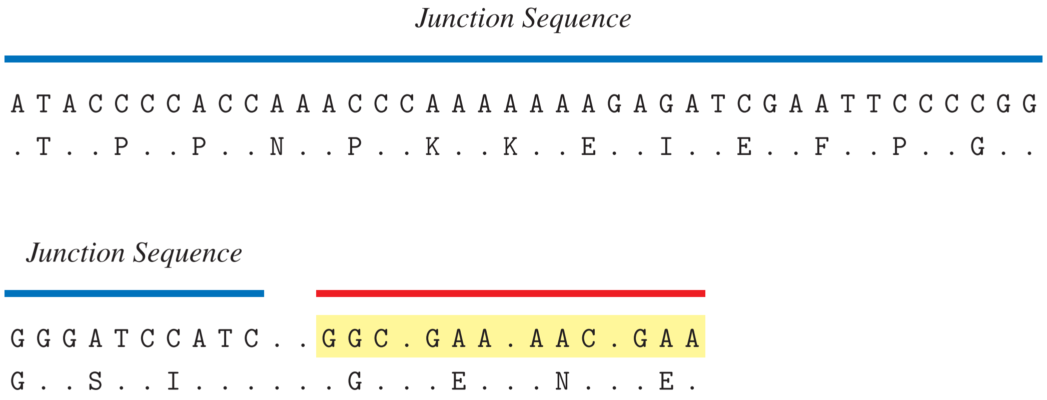

Junction Sequence
A T A C C C C A C C A A A C C C A A A A A A A G A G A T C G A A T T C C C C G G
. T . . P . . P . . N . . P . . K . . K . . E . . I . . E . . F . . P . . G . .
Junction Sequence
G G G A T C C A T C . . G G C . G A A . A A C . G A A
G . . S . . I . . . . G . . . E . . . N . . . E .

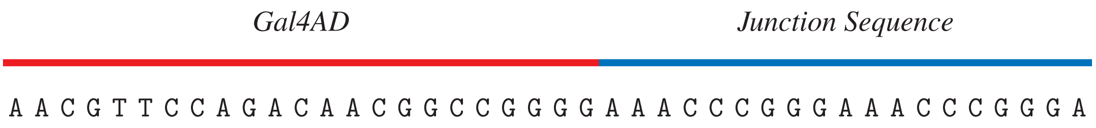
Figure 4.2: Junction sequence for the pGAD-C yeast genomic library.

If you need to use another **Junction Sequence** you can do so by pasting it directly into the textbox labeled “junction sequence”

A new junction sequence should be

- UPPER case and be 50 nt long.
- Be immediately upstream of the cDNA/fragment fusion site
- Have the last 3 nt define a complete codon for the preceding reading frame. In the examples above the last 3 nucleotides define the operative frame (GGG → Glycine or ATC → Isoleucine)

Junction Make takes the 50 nt sequence and creates 3 sequences “tag” from it. It will then make a new file of all the reads that contain one or more of these sequence tags. These lists can be found in the folder `junction_files`. Consider the following sequence:

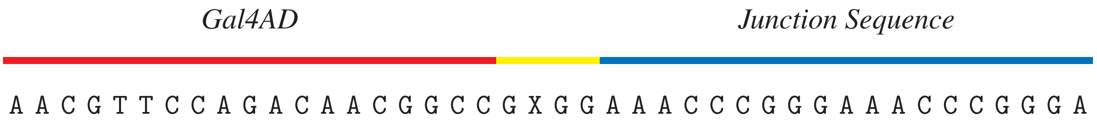

Gal4AD
Junction Sequence
A A C G T T C C A G A C A A C G G C C G G G G A A A C C C G G G A A A C C C G G G A

Junction Make will first search for the tag `AGACAACGGCCGGGG`.

Once found it will determine the Downstream Reading Frame (the fusion point of the cDNA), which would be: `AAACCCGGGAAACCCGGGA`.

Sometimes that there is a cloning mismatch in where a cDNA is inserted into the library. This could be a base substitution or a missing nucleotide. To compensate, Junction Make will also look for a 15 bp upstream sequence 4 nt back from what it first looked for. It only does this if the read in question does not have a perfect match to the primary junction sequence query above.

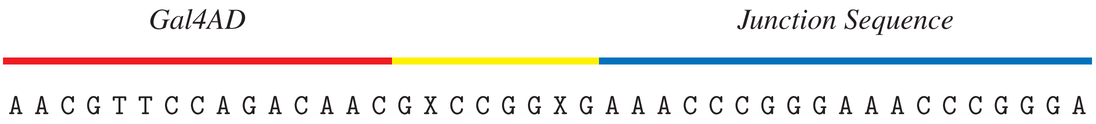
This looks like the following:


Gal4AD
Junction Sequence
A A C G T T C C A G A C A A C G G C C G X G G A A A C C C G G G A A A C C C G G G A

The sequence to be searched for is: `TTCCAGACAACGGCC`

The Downstream Reading Frame returned remains: `AAACCCGGGAAACCCGGGA`

We have even found this does not fully capture all the junction sequences there are so Junction Make will do the same thing again, going back another another 4 nt to yield.


Gal4AD
Junction Sequence
A A C G T T C C A G A C A A C G X C C G G X G A A A C C C G G G A A A C C C G G G A

The sequence to be searched for is: `AACGTTCCAGACAAC`
The Downstream Reading Frame returned remains: `AAACCCGGGAAACCCGGGA`

Junction Make will generate these 3 junction tags that it will look for an apprise you of its status by stating

```
>>>Comment2
The primary, secondary, and tertiary sequences that will be searched for are:
1st sequence
2nd sequence
3rd sequence
>>>END
```

Junction Make will then notify you that it has started to search the `.sam` files using

```
>>>Comment3
Junction Make is searching .sam files for the junctions that span the GAL4-AD
    and library insert
The next step converts files to a FASTA file format used for blastn search
The FASTA files are temporary \_TEMP.fa files are located in the blastResults
    folder

\_TEMP.fa files are being converted into blast.txt files that contain the blastn
    results for each junction.
This is done by searching each sequence against the reference cDNA database
    using blastn.
This takes a while...
>>>END
```

During this time, Junction Make runs a `blastn` search of each of the junction reads against a database of annotated RefSeq mRNAs or yeast genes. Results from this blast search are found in the “blast_results” folder. Files from this folder are then used to create a searchable format that can be used by the analysis program Blast Query . Junction Make creates a Python dictionary “.p” file for each dataset are stores this in the `blast_results_query` folder.

5. Running Gene Count & Junction Make

The **DEEPN** application also provides a Gene Count + Junction Make button. Clicking this sequentially runs Gene Count and Junction Make on the contents of “mapped_sam_files” and the “unmapped_sam_files” automatically. Thus, one can queue in the files and let processing complete overnight. To run this option be sure that the “Work Folder” contains only:

- `mapped_sam_files` folder with mapped `.sam` files
- `unmapped_sam_files` folder with corresponding unmapped `.sam` files

If there are no other folders, you can be sure to avoid any **WARNINGS** that might interrupt the processing workflow. Alternatively, you can unclick the box on the **DEEPN** window to skip such prompts.

III

Analyzing Data

6 **Blast Query** 27

6.1 Results Tab

6.2 Filtered Results Tab

6.3 Plot Tab

6.4 Export for Graphing

7 **Read Depth** 31

Index 33

6. Blast Query

This program allows you to assess the fusion point between the Gal4AD and each gene/cDNA in question.

- This program queries the blast searches done previously in `blast_results_query` folder
- Once loaded, you simply type in the NCBI reference number (**NM_*****)
- The fusion points and their frequency in ppm are displayed

Once Junction Make has loaded the `blast_results_query` folder with processed ".p" files, Blast Query can be used to analyze the junctions. Clicking the Blast Query button launches the module and a new graphic user interface.

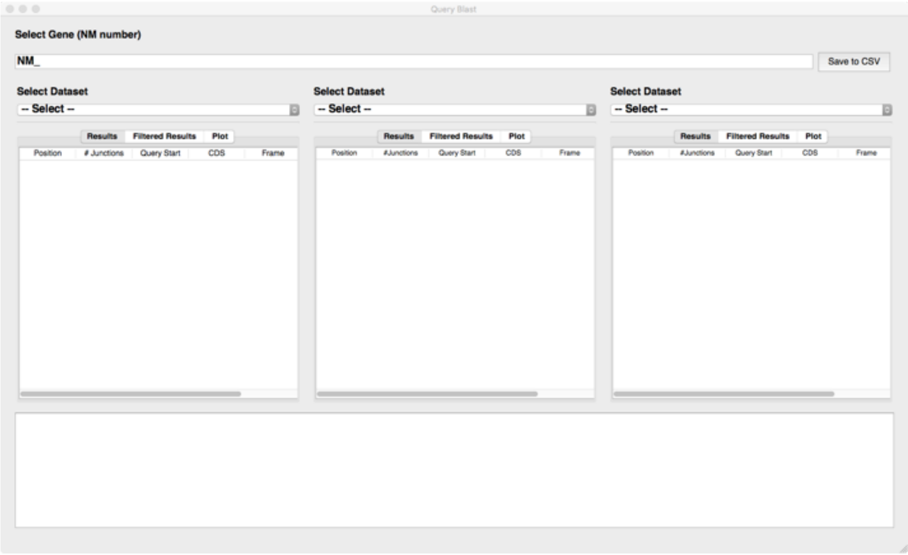
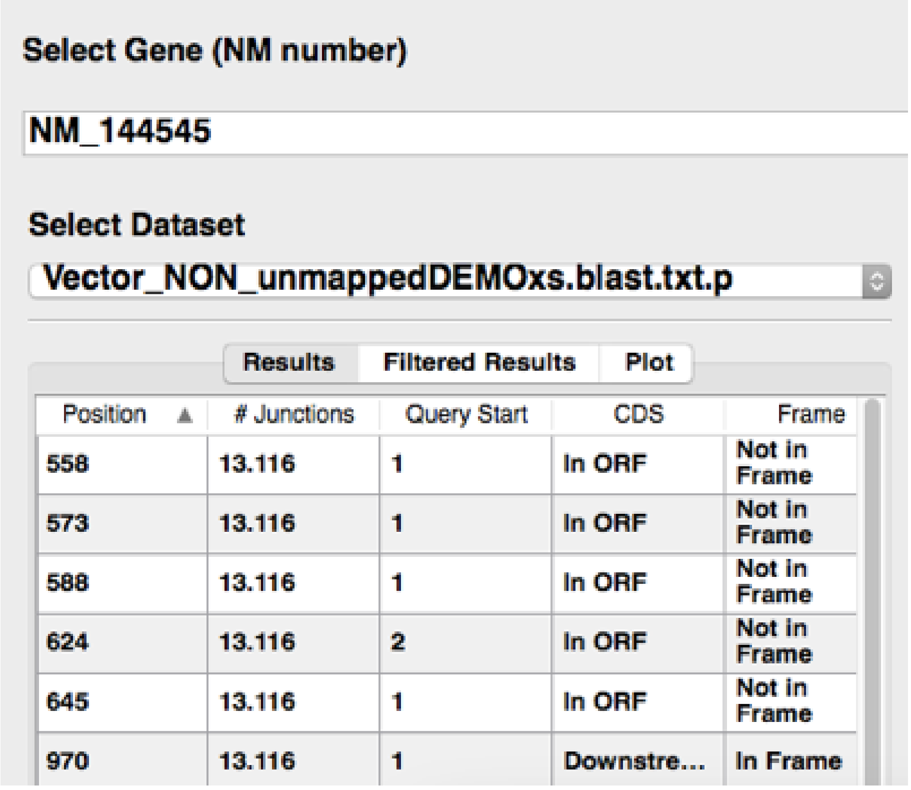

Query Blast
Select Gene (NM number)
NM_
Save to CSV
Select Dataset
-- Select --
Results
Filtered Results
Plot
Position
# Junctions
Query Start
CDS
Frame
Select Dataset
-- Select --
Results
Filtered Results
Plot
Position
#Junctions
Query Start
CDS
Frame
Select Dataset
-- Select --
Results
Filtered Results
Plot
Position
#Junctions
Query Start
CDS
Frame

Figure 6.1: Screen shot of Blast Query user interface.

Enter in GeneID in the text box above in Figure 6.1. These are **NM_*** identifiers that can be found in relation to gene names in the `summary.csv` files generated by Gene Count . An example is **NM_146001**.

Then use the pull-down menu to select which datasets to compare. The list of datasets is generated by reading what ".p" files are in the `blast_results_query` folder that is within your "Work Folder". If the right .p file is not in that folder, you can simply move it in from somewhere else.

The data window can be selected to display **Results** , **Filtered Results** or **Plot** as shown in Figure 7.1.



Select Gene (NM number)
NM_144545

Select Dataset
Vector_NON_unmappedDEMOxs.blast.txt.p

Position ▲	# Junctions	Query Start	CDS	Frame
558	13.116	1	In ORF	Not in Frame
573	13.116	1	In ORF	Not in Frame
588	13.116	1	In ORF	Not in Frame
624	13.116	2	In ORF	Not in Frame
645	13.116	1	In ORF	Not in Frame
970	13.116	1	Downstre...	In Frame

Figure 6.2: Screen shot of Blast Query user interface.

6.1 Results Tab

The "Results Tab" shows

- **Position** (which is the position of first nucleotide of the given insert that found for a particular junction read).
- **#Junctions** is a count, in PPM, for how abundant that particularjunction is
- **QueryStart** comes from the q.start value of the `blastn` search used to identify the downstream gene fused to the Gal4AD. It refers to how many nucleotides are between the junction tag and the matched cDNA position. Often, this number is 1, but sometimes there are cloning artifacts that generate extra sequence between the end of the Gal4-AD and where the match is to the cDNA. These extra nucleotides will have an impact on the translation reading frame. So the Position and the Query Start are used to calculate whether a particular junction is within the Coding Region (CDS) and whether it is in-frame.
- **CDS** shows whether the junction site in the mRNA of interest is upstream of the coding region, downstream of the coding region, or within (In ORF) of the coding region.

- **Frame** calculates whether the downstream library insert that encodes the candidate protein of interest is in the same frame as the upstream Gal4AD.

6.2 Filtered Results Tab

Sometimes, sequencing the same junction multiple times leads to sequence errors which can produce reads that have the same Position but different Q.Starts, or that have Positions that are 1-2 bp different from the main Position. To simplify comparisons, one can use the “Filtered Results” tab, which collapses all similar Positions that may have different Q.starts into a single value. Filtered Results also displays only the Positions that are found within the **Left** most dataset. Thus, by placing, say, a total unselected library dataset on the left, the **Right**-hand datasets will only display position sites that are in common with the unselected library dataset.

6.3 Plot Tab

The plot tab creates a quick graphic displaying where each junction site is along the given mRNA or gene. The abundance of each junction sequence within the dataset are plotted on the Y axis in ppm. If a junction site is downstream of the CDS or out of translational frame, the bar is grey. If it is within the CDS and in-frame it is dark blue, and if it is upstream of the CDS start but within the correct translational reading frame, it is cyan. The start and stop sites for translation are shown with red bars. The mRNA/gene sequence itself is given in the text box below, where the protein coding sequence is shown in black text and the upstream and downstream untranslated regions are in grey.

6.4 Export for Graphing

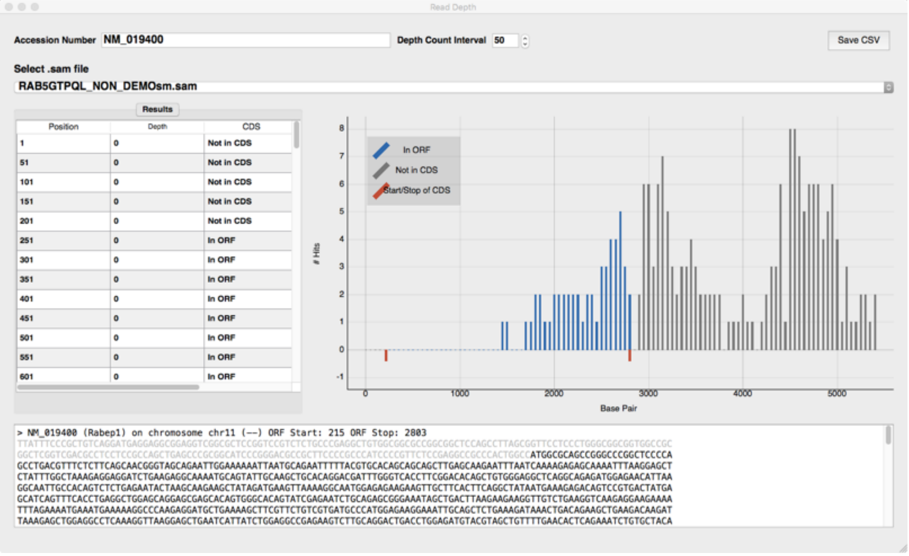
A Save to CSV button is found at the top right hand corner. This will save the current **Results** and **Filtered Results** Tables in a `.csv` file that can be opened in `Microsoft Excel`. Results from each selected dataset are saved in a different sheet.

This output pasted program for further analysis or graphing such as GraphPad Prism

http://www.graphpad.com/scientific-software/prism/

7. Read Depth

Read Depth is used as an aid to determine the 3’ end of the interacting cDNA insert. Most genes that enrich upon selection are represented by a single plasmid, which can be surmised by having a dominant junction site. Thus, much of a given cDNA insert extends downstream can be determined by read-depth along the cDNA sequence. Read Depth determines this by measuring how many sequences within the dataset contain a 25 bp interval of the cDNA sequence in question. Read Depth first makes a subset of all the mapped reads that correspond to the gene of interest and then counts how many of these reads contain sequence that match along the length of the cDNA. The up/down arrows can be used to adjust the interval distance in increments of 50 nt. Users can also directly enter a desired interval, with a minimum interval of 50 nt.



Accession Number NM_019400

Depth Count Interval 50

Select .sam file

RAB5GTPQL_NON_DEMOsm.sam

Position	Depth	CDS
1	0	Not in CDS
51	0	Not in CDS
101	0	Not in CDS
151	0	Not in CDS
201	0	Not in CDS
251	0	In ORF
301	0	In ORF
351	0	In ORF
401	0	In ORF
451	0	In ORF
501	0	In ORF
551	0	In ORF
601	0	In ORF

```
> NM_019400 (Rabep1) on chromosome chr11 (--) ORF Start: 215 ORF Stop: 2803
TTATTTCCCGCTGTCAGGATGAGGAGGCGGAGGTCGGCGCTCCGGTCCGTCTCTGCCCGAGGCTGTGGCGGCGCCGGCGGCTCCAGCCTTAGCGGTTCCTCCCTGGGCGGCGGTGGCCGC
GGCTCGGTCGACGCCTCCTCCGCCAGCTGAGCCCGCGGCATCCCGGGACGCCGCTTCCCCGCCCATCCCCGTTCTCCGAGGCCGCCCACTGGCCATGGCGCAGCCGGGCCCGGCTCCCCA
GCCTGACGTTTCTCTTCAGCAACGGGTAGCAGAATTGGAAAAAATTAATGCAGAATTTTTACGTGCACAGCAGCAGCTTGAGCAAGAATTTAATCAAAAGAGAGCAAAATTTAAGGAGCT
CTATTTGGCTAAAGAGGAGGATCTGAAGAGGCAAAATGCAGTATTGCAAGCTGCACAGGACGATTTGGGTCACCTTCGGACACAGCTGTGGGAGGCTCAGGCAGAGATGGAGAACATTAA
GGCAATTGCCACAGTCTCTGAGAATACTAAGCAAGAAGCTATAGATGAAGTTAAAAGGCAATGGAGAGAAGAAGTTGCTTCACTTCAGGCTATAATGAAAGAGACAGTCCGTGACTATGA
GCATCAGTTTCACCTGAGGCTGGAGCAGGAGCGAGCACAGTGGGCACAGTATCGAGAATCTGCAGAGCGGGAAATAGCTGACTTAAGAAGAAGGTTGTCTGAAGGTCAAGAGGAAGAAAA
TTTAGAAAATGAAATGAAAAAGGCCCAAGAGGATGCTGAAAAGCTTCGTTCTGTCGTGATGCCCATGGAGAAGGAAATTGCAGCTCTGAAAGATAAACTGACAGAAGCTGAAGACAAGAT
TAAAGAGCTGGAGGCCTCAAAGGTTAAGGAGCTGAATCATTATCTGGAGGCCGAGAAGTCTTGCAGGACTGACCTGGAGATGTACGTAGCTGTTTTGAACACTCAGAAATCTGTGCTACA
```

Figure 7.1: Screen shot of Read Depth user interface.

Index

About DEEPN, 7

Blast Query, 27
Blast Query Filtered Results, 29
Blast Query Plot Tab, 29
Blast Query Results, 28

Contents within DEEPN, 8

DEEPN Workflow Overview, 9
Download Link, 9

Export for Graphing, 29

Gene Count, 17

Initializing DEEPN, 11
Installation, 9

Junction Make, 19

Launching DEEPN, 11
Linux Compatibility, 9

Mac OS X Compatibility, 9

Open Source License, 9

Preprocessing `.fastq files`, 11

Read Depth, 31
Running Gene Count & Junction Make, 23

Windows Compatibility, 9