

# **Differential Gene Expression Analysis Using Grotto, an IGS Web-based Tool, from Fastq Files**

## **Introduction**

The following protocol takes fastq files as input to perform differential gene expression analysis. The fastq sequences are aligned against a reference fasta file. The read counts for each gene are determined using HTseq and the differential gene expression is determined using DEseq.

To practice launching this pipeline, example files are available on the server at /local/projects/RNASEQ/SOPs/grottosOP/. In the illumina\_reads directory are the 12 fastq files. There are two groups of samples; Human Brain Reference (HBR) which is total RNA isolated from the brains of 23 Caucasians, male and female, of varying age and Universal Human Reference (UHR) with is total RNA isolated from a diverse set of 10 cancer cell lines. There are three samples of each group with two files (a file for each of the ends of the paired end reads) for each sample. The reference directory has the reference sequence, the reference annotations files, and a gene annotation map file. The grotto\_input\_files directory has examples of a sample file and a config file. To use these input files in grotto, you will need to download these to the local laptop or desktop computer you are using to access grotto.

## **Requirements**

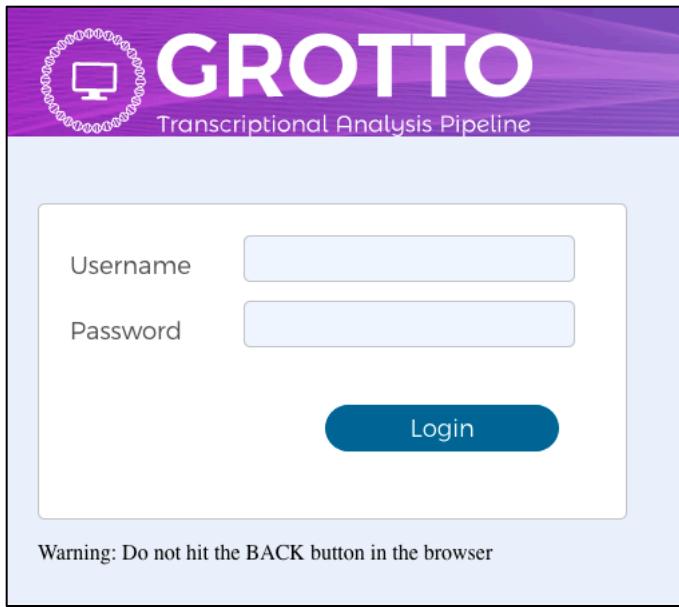
There must be a certain set of directories on the server and a project area in ergatis. See the SOP for “Creating work areas required by grotto.”

## **Procedure**

1. Download genome reference file (a fasta file) and the genome annotation file (a gtf or gff3 file) for your study organism.  
Put a copy of each in /local/projects/XYABC/rnaseq/reference, where XYABC is your project area.

For this example or for testing, we can use files from  
/local/projects/RNASEQ/SOPs/grottoSOP/reference

2. Log onto grotto-staging.igs.umaryland.edu with your IGS credentials.



Next you have the choice of enter your samples manually or by upload. Instructions for using the upload follow the instructions for manual entry.

3. Enter sample information. This can be done either manually on the screen (4a) or automatically by file upload (4b).

3a. Enter the information for your samples manually.

For each sample, assign it to a group (such as experiment and control or, in our example, HBR and UHR), and enter the path in the IGS servers for each of the fastq files for the sample.

**Please note that sample names and group names should not start with a number.**

To test grotto for yourself using example data, there are fastq files in  
/local/projects/RNASEQ/SOPs/grottoSOP/illumina\_reads

Sample Info File

Sample Info File Location ?  No file selected.

	Sample ID	Group ID	File 1	File 2	
1	HBR1	HBR	/local/projects/RNASEQ/SO	/local/projects/RNASEQ/SO	-
2	HBR2	HBR	/local/projects/RNASEQ/SO	/local/projects/RNASEQ/SO	-
3	HBR3	HBR	/local/projects/RNASEQ/SO	/local/projects/RNASEQ/SO	-
4	UHR1	UHR	/local/projects/RNASEQ/SO	/local/projects/RNASEQ/SO	-
5	UHR2	UHR	/local/projects/RNASEQ/SO	/local/projects/RNASEQ/SO	-
6	UHR3	UHR	/local/projects/RNASEQ/SO	/local/projects/RNASEQ/SO	-

Warning: Do not hit the BACK button in the browser

### 3b. Enter the information for your samples by upload.

Create a tab-separated text file with a line for each sample that includes the sample name, group designation, read 1 and read2.

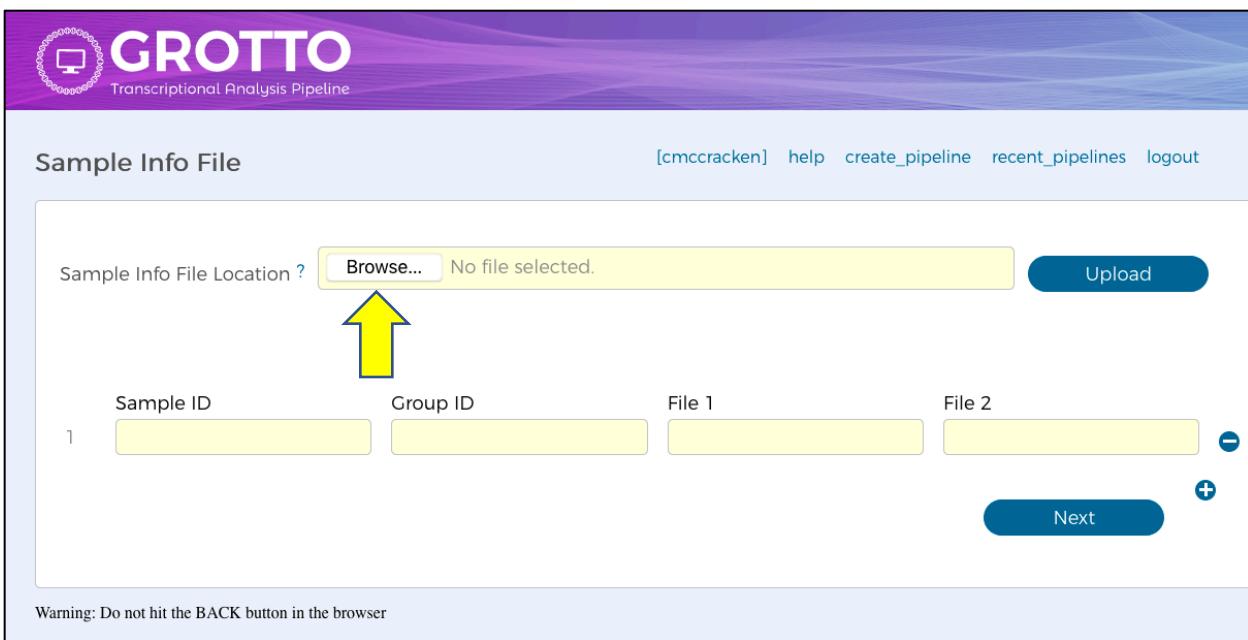
```
Sample1      Group1 /path_to_file/Sample1_R1.fa.gz    /path_to_file/Sample1_R2.fa.gz
Sample2      Group1 /path_to_file/Sample2_R1.fa.gz    /path_to_file/Sample2_R2.fa.gz
Sample3      Group2 /path_to_file /Sample3_R1.fa.gz   /path_to_file/Sample3_R2.fa.gz
Sample4      Group2 /path_to_file /Sample4_R1.fa.gz   /path_to_file/Sample4_R2.fa.gz
```

**Please note that sample names and group names should not start with a number.**

For testing, you can download

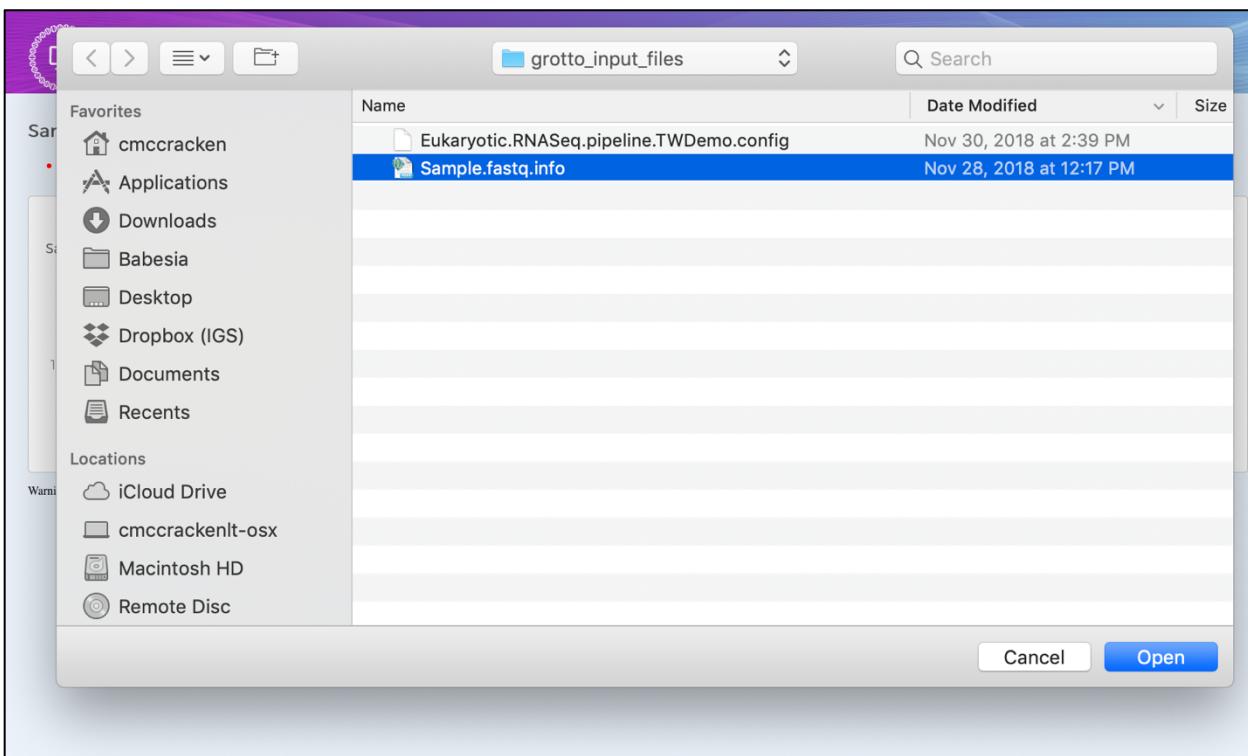
/local/projects/RNASEQ/SOPs/grottoSOP/grotto\_input\_files/Sample.fastq.info to your local computer.

In the grotto window, click the browse button.



The screenshot shows the Grotto Transcriptional Analysis Pipeline interface. At the top, there is a purple header bar with the Grotto logo and the text "Transcriptional Analysis Pipeline". Below the header, the page title is "Sample Info File". On the right side of the header, there are links for "[cmccracken]", "help", "create\_pipeline", "recent\_pipelines", and "logout". The main content area has a light blue background. It contains a form for uploading a sample info file. The first field is "Sample Info File Location ?" with a "Browse..." button and a yellow arrow pointing up to it. A message "No file selected." is displayed next to the button. To the right of the button is a blue "Upload" button. Below this field are four input fields: "Sample ID" (containing "1"), "Group ID", "File 1", and "File 2". To the right of "File 2" is a minus sign (-) button. Below these fields are two buttons: a blue plus sign (+) button and a blue "Next" button. At the bottom of the form, there is a warning message: "Warning: Do not hit the BACK button in the browser".

Select file from your local computer.



Click on Upload button.

The screenshot shows the GROTTO Transcriptional Analysis Pipeline interface. At the top, there's a purple header bar with the GROTTO logo and the text "Transcriptional Analysis Pipeline". Below the header, the page title is "Sample Info File". In the top right corner, there are links for "[cmccracken]", "help", "create\_pipeline", "recent\_pipelines", and "logout". The main content area has a light blue background. It features a "Sample Info File Location" input field containing "Eukaryotic.RNASeq.pipeline.TWDemo.config", a "Browse..." button, and an "Upload" button. Below this, there are four rows for sample entries. Each row has "Sample ID" (with value 1), "Group ID", "File 1", and "File 2" fields. To the right of each row are minus and plus buttons. A "Next" button is located at the bottom right. A yellow arrow points upwards from the "File 2" field towards the "Upload" button. At the very bottom, a warning message says "Warning: Do not hit the BACK button in the browser".

Click the Next button.

This screenshot shows the same GROTTO interface after the file has been uploaded. The "File 2" fields now contain local file paths: "/local/projects/RNASEQ/SC". The "Next" button is highlighted with a yellow arrow pointing to it. The rest of the interface is identical to the previous screenshot, including the sample entries and the warning message at the bottom.

On Grotto, you may see “?” next to an expected input, such as next to the Sample Info File Location in the above window. If you hover your cursor over the “?”, an explanation of what is expected will pop up.

#### 4. Set up your pipeline options

Enter the paths on the servers to the reference genome, the reference annotation, and the ergatis repository root.

Add your project ID and select the annotation format (GTF or GFF3).

Select the boxes for Build indexes, alignment, visualization, RPKM analysis, and Differential Gene Expression. Building an index is necessary for creating alignments and the alignments are necessary for differential gene expression analysis. The visualization tool will create files that can be loaded into IGV to view the alignments graphically. The RPKM analysis is useful for profiling the gene expression in the samples.

Add your comparison.

To test the example data, you can use the paths to reference files. These are the files that you copied from /local/projects/RNASEQ/SOPs/grottoSOP/reference/. Now they will be in /local/projects/XYABC/rnaseq/reference, where XYABC is your project area.

Click the Next button.



Pipeline Options

download\_SOP help create\_pipeline recent\_pipelines

Organism Type	Eukaryotic <input checked="" type="radio"/>	Prokaryotic <input type="radio"/>
Reference	/local/projects/RNASEQ/SOPs/grottoSOP/reference/chr22_with_ERCC92.fa <a href="#">?</a>	
GFF3/GTF	/local/projects/RNASEQ/SOPs/grottoSOP/reference/chr22_with_ERCC92.gtf <a href="#">?</a>	
Repository Root	/local/projects/RNASEQ/XYABC/rnaseq/ergatis <a href="#">?</a>	
Project ID	XYABC <a href="#">?</a>	
Annotation Format	GTF <a href="#">?</a>	
<input type="checkbox"/> Use Tophat in place of HiSat2 <a href="#">?</a> <input checked="" type="checkbox"/> Build Indexes <a href="#">?</a> <input type="checkbox"/> Quality Stats <a href="#">?</a> <input type="checkbox"/> Quality Trimming <a href="#">?</a> <input checked="" type="checkbox"/> Alignment <a href="#">?</a> <input checked="" type="checkbox"/> Visualization <a href="#">?</a> <input checked="" type="checkbox"/> RPKM Analysis <a href="#">?</a> <input checked="" type="checkbox"/> Differential Gene Expression <a href="#">?</a> Comparison Groups <input type="text" value="HBRvsUHR"/> <a href="#">?</a> <input type="checkbox"/> Isoform Analysis <a href="#">?</a> <input type="checkbox"/> Differential Isoform Analysis <a href="#">?</a>		

**Next →**

You can either configure your pipelines manually or by upload. Instructions for using the upload follow the instructions for manual entry.

5. Configure your pipeline. This can be done either manually on the screen (6a) or automatically by file upload (6b).

#### 5a. Configure your pipeline manually

There are default settings for many of the pipeline parameters. The pipeline components with the green dot are have sufficient settings with the default parameters, but you will need to open the ones with the yellow or red dots to add parameters where there are not default. If you are interested in further exploring the options, you will need to read the documentation for that component. Start by clicking on the arrow to the right for hisat2.



## Config File Form

[cmccracken] help create\_pipeline recent\_pipelines logout

(Optional) Upload a pre-made config file.

Config File Location

[Browse...](#)

No file selected.

[Upload](#)

● Sufficient default values    ● Default specific requirements    ● Multiple required fields

● <b>Hisat2 Build</b>	Build HiSat2 reference indexes	▼
● <b>Hisat2</b>	Run HiSat2 aligner	 ▼
● <b>Percent Mapped Stats</b>	Calculate mapping statistics of aligned reads	▼
● <b>RPKM Coverage Stats</b>	Calculate coverage statistics using BEDtools	▼
● <b>Bam2BigWig</b>	Convert BAM to BigWig format	▼
● <b>DESeq</b>	Analyze differential gene expression from HTSeq read counts	▼
● <b>Filter DESeq</b>	Filter DESeq output on various metrics	▼
● <b>EdgeR</b>	Analyze differential gene expression from HTSeq read counts	▼
● <b>Filter EdgeR</b>	Filter EdgeR output on various metrics	▼
● <b>Expression Plots</b>	Creates plots for DESeq/Cuffdiff or RPKM analysis	▼

[Next](#)

Warning: Do not hit the BACK button in the browser

For most of these parameters, the default is usually OK. It is important to be aware of the strandness of your reads. The default is unstranded. The example sample data are unstranded.

Next click on the arrow to the right of Percent Mapped Stats.

● **Hisat2** Run Hisat2 aligner ^

Fields with \* are required.

HISAT2_BIN_DIR	* \$.HISAT2_BIN\$.	?
SAMTOOLS_BIN_DIR	* \$.SAMTOOLS_BIN\$.	?
MISMATCH_PENALTIES		?
SOFTCLIP_PENALTIES		?
READ_GAP_PENALTIES		?
REF_GAP_PENALTIES		?
MIN_ALIGNMENT_SCORE		?
PEN_CANSPLICE		?
PEN_NONCANSPLICE		?
PEN_CANINTRONLEN		?
PEN_NONCANINTRONLEN		?
MIN_INTRON_LENGTH		?
MAX_INTRON_LENGTH		?
RNA_STRANDNESS		? --rna-strandness <string> = Specify strand-specific information: the default is unstranded. F or R for single-end reads. FR or RF for paired-end reads.
DTA_CUFFLINKS		?
NUM_ALIGNMENTS		?
MIN_FRAGMENT_LENGTH		?
MAX_FRAGMENT_LENGTH		?
SUPPRESS_UNALIGNMENTS		?
NUMBER_THREADS		?
OTHER_PARAMETERS		?
OTHER_ARGS	--V	?

● **Percent Mapped Stats** Calculate mapping statistics of aligned reads  ▼

The feature\_type is the feature from column 3 of your gtf file. The attribute\_ID is a unique identifier from column 9 of your gtf. Groupby\_ID is the same ID as attribute\_ID. Next click on the arrow to the right of RPKM Coverage Stats.

<b>Percent Mapped Stats</b>	Calculate mapping statistics of aligned reads	^																		
<p>Fields with * are required.</p> <table border="0"> <tr> <td>FEATURE_TYPE</td> <td>* exon</td> <td>?</td> </tr> <tr> <td>ATTRIBUTE_ID</td> <td>* gene_id</td> <td>?</td> </tr> <tr> <td>GROUPBY_ID</td> <td>* gene_id</td> <td>?</td> </tr> <tr> <td>BEDTOOLS_BIN_DIR</td> <td>* \$:BEDTOOLS_BIN\$:</td> <td>?</td> </tr> <tr> <td>SAMTOOLS_BIN_DIR</td> <td>* \$:SAMTOOLS_BIN\$:</td> <td>?</td> </tr> <tr> <td>OTHER_PARAMETERS</td> <td></td> <td>?</td> </tr> </table>			FEATURE_TYPE	* exon	?	ATTRIBUTE_ID	* gene_id	?	GROUPBY_ID	* gene_id	?	BEDTOOLS_BIN_DIR	* \$:BEDTOOLS_BIN\$:	?	SAMTOOLS_BIN_DIR	* \$:SAMTOOLS_BIN\$:	?	OTHER_PARAMETERS		?
FEATURE_TYPE	* exon	?																		
ATTRIBUTE_ID	* gene_id	?																		
GROUPBY_ID	* gene_id	?																		
BEDTOOLS_BIN_DIR	* \$:BEDTOOLS_BIN\$:	?																		
SAMTOOLS_BIN_DIR	* \$:SAMTOOLS_BIN\$:	?																		
OTHER_PARAMETERS		?																		
<b>RPKM Coverage Stats</b>	Calculate coverage statistics using BEDtools	 ▾																		

The parameters for RPKM Coverage Stats are the same as for Percent Mapped Stats.  
Click on the arrow to the right of Filter DESeq.

<b>RPKM Coverage Stats</b>	Calculate coverage statistics using BEDtools	^																		
<p>Fields with * are required.</p> <table border="0"> <tr> <td>FEATURE_ID</td> <td>* exon</td> <td>?</td> </tr> <tr> <td>ATTRIBUTE_ID</td> <td>* gene_id</td> <td>?</td> </tr> <tr> <td>GROUPBY_ID</td> <td>* gene_id</td> <td>?</td> </tr> <tr> <td>BEDTOOLS_BIN_DIR</td> <td>* \$:BEDTOOLS_BIN\$:</td> <td>?</td> </tr> <tr> <td>SAMTOOLS_BIN_DIR</td> <td>* \$:SAMTOOLS_BIN\$:</td> <td>?</td> </tr> <tr> <td>OTHER_ARGS</td> <td>--V</td> <td>?</td> </tr> </table>			FEATURE_ID	* exon	?	ATTRIBUTE_ID	* gene_id	?	GROUPBY_ID	* gene_id	?	BEDTOOLS_BIN_DIR	* \$:BEDTOOLS_BIN\$:	?	SAMTOOLS_BIN_DIR	* \$:SAMTOOLS_BIN\$:	?	OTHER_ARGS	--V	?
FEATURE_ID	* exon	?																		
ATTRIBUTE_ID	* gene_id	?																		
GROUPBY_ID	* gene_id	?																		
BEDTOOLS_BIN_DIR	* \$:BEDTOOLS_BIN\$:	?																		
SAMTOOLS_BIN_DIR	* \$:SAMTOOLS_BIN\$:	?																		
OTHER_ARGS	--V	?																		
<b>Bam2BigWig</b>	Convert BAM to BigWig format	▼																		
<b>DESeq</b>	Analyze differential gene expression from HTSeq read counts	▼																		
<b>Filter DESeq</b>	Filter DESeq output on various metrics	 ▼																		

Enter the path for the mapping file in the MAPFILE\_PATH. Click on the arrow to the right of FilterEdgeR.

● **Filter DESeq** Filter DESeq output on various metrics ^

FILTERS\_VAL • FDR=0.05,RCP=0.1,UFC=1,D ? Fields with \* are required.

PROJECT\_NAME • XRSEQ ?

MAPFILE\_PATH ith\_ERCC92.gene.mapping ?

● **EdgeR** Analyze differential gene expression from HTSeq read counts ▼

● **Filter EdgeR** Filter EdgeR output on various metrics ▼ 

Enter the path for the mapping file in the MAPFILE\_PATH. Click on the Next button.

● **Filter DESeq** Filter DESeq output on various metrics ^

FILTERS\_VAL • FDR=0.05,RCP=0.1,UFC=1,D ? Fields with \* are required.

PROJECT\_NAME • XRSEQ ?

MAPFILE\_PATH ith\_ERCC92.gene.mapping ?

● **EdgeR** Analyze differential gene expression from HTSeq read counts ▼

● **Filter EdgeR** Filter EdgeR output on various metrics ▼

● **Expression Plots** Creates plots for DESeq/Cuffdiff or RPKM analysis ▼  

## 5b. Configure your pipeline by upload.

Click on the Browse button

**GROTTO**  
Transcriptional Analysis Pipeline

Config File Form

(Optional) Upload a pre-made config file.

Config File Location  No file selected.



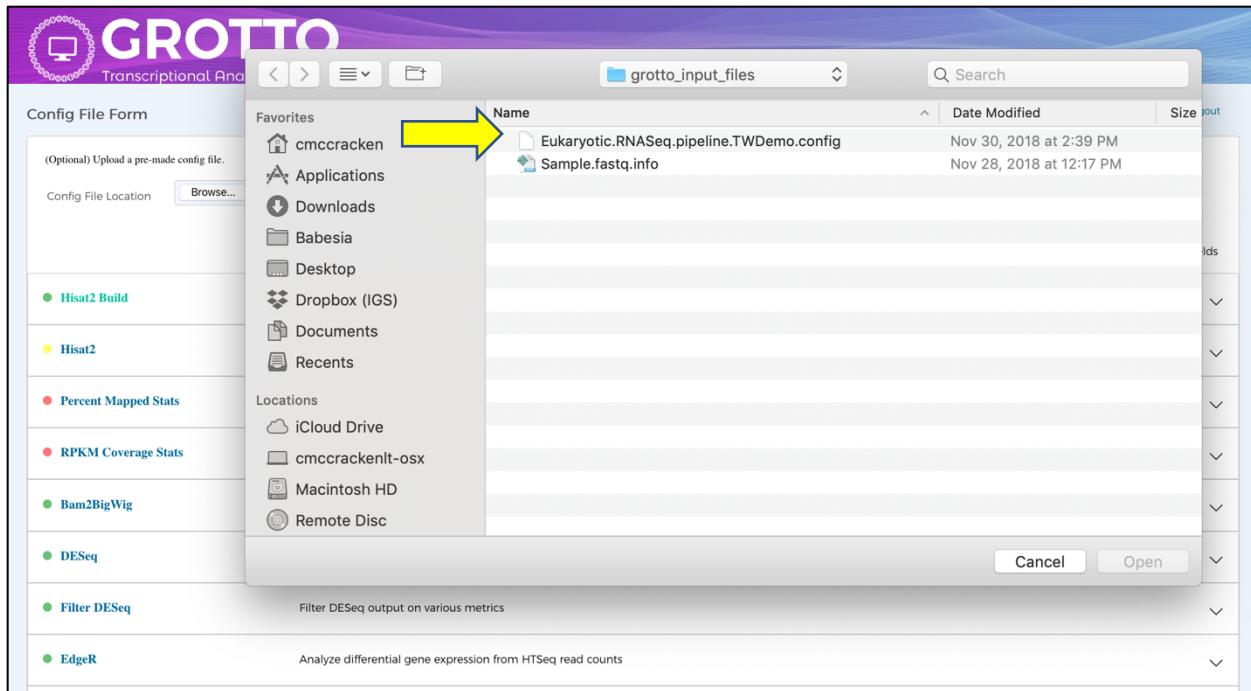
● Sufficient default values   ● Default specific requirements   ● Multiple required fields

<span style="color: green;">●</span> Hisat2 Build	Build HiSat2 reference indexes	▼
<span style="color: yellow;">●</span> Hisat2	Run HiSat2 aligner	▼
<span style="color: red;">●</span> Percent Mapped Stats	Calculate mapping statistics of aligned reads	▼
<span style="color: red;">●</span> RPKM Coverage Stats	Calculate coverage statistics using BEDtools	▼
<span style="color: green;">●</span> Bam2BigWig	Convert BAM to BigWig format	▼
<span style="color: green;">●</span> DESeq	Analyze differential gene expression from HTSeq read counts	▼
<span style="color: green;">●</span> Filter DESeq	Filter DESeq output on various metrics	▼
<span style="color: green;">●</span> EdgeR	Analyze differential gene expression from HTSeq read counts	▼
<span style="color: green;">●</span> Filter EdgeR	Filter EdgeR output on various metrics	▼
<span style="color: green;">●</span> Expression Plots	Creates plots for DESeq/Cuffdiff or RPKM analysis	▼

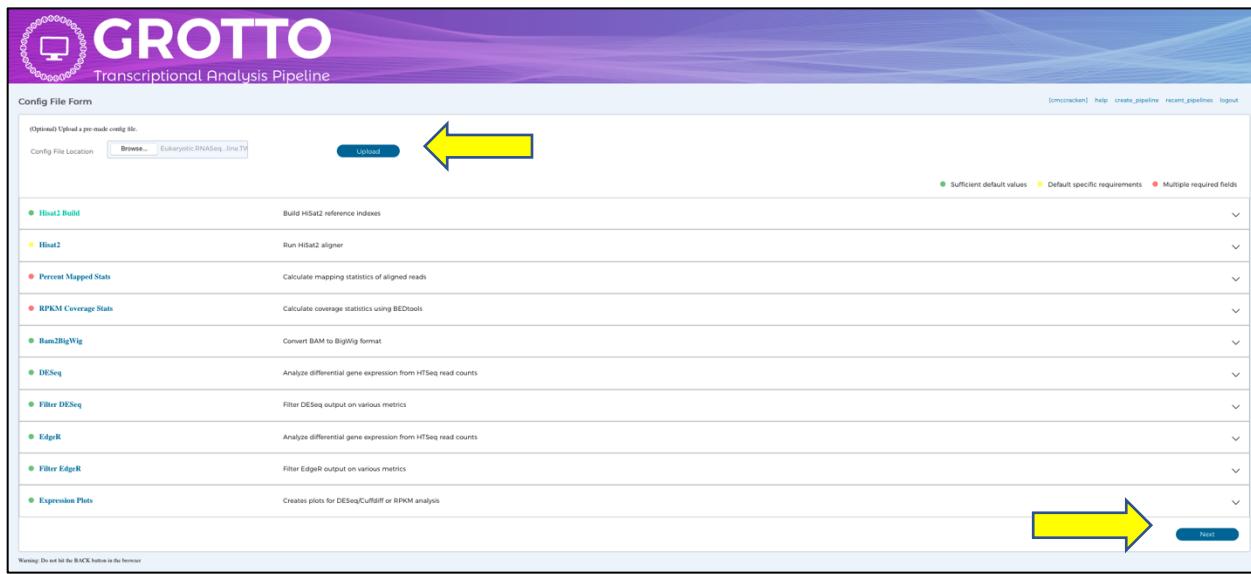
### Select your config file

For this example, you can download

/local/projects/RNASEQ/SOPs/grottoSOP/grotto\_input\_files/Eukaryotic.RNASeq.pipeline.TWD  
emo.config to your computer locally



Click on the Upload button and then the Next button



## 6. Launch your pipeline

You can click on the Sample Info File, Pipeline Options, or the Configuration File Form to review. If you need to make changes, each expanded view has an Edit button in the lower right corner. Then click the Submit button.

 **GROTTO**  
Transcriptional Analysis Pipeline

Summary [cmccracken] help create\_pipeline recent\_pipelines logout

**Sample Info File**

**Pipeline Options**

**Configuration File Form**

 **Submit**

Warning: Do not hit the BACK button in the browser

7. Watch the progress of your pipeline in ergatis. Grotto has a progress bar, but you can also click on the View in Ergatis button for a detailed view

 **GROTTO**  
Transcriptional Analysis Pipeline

Pipeline 13307617699 [cmccracken] help create\_pipeline recent\_pipelines logout

Component	Status	Time Elapsed
Progress	View in Ergatis	

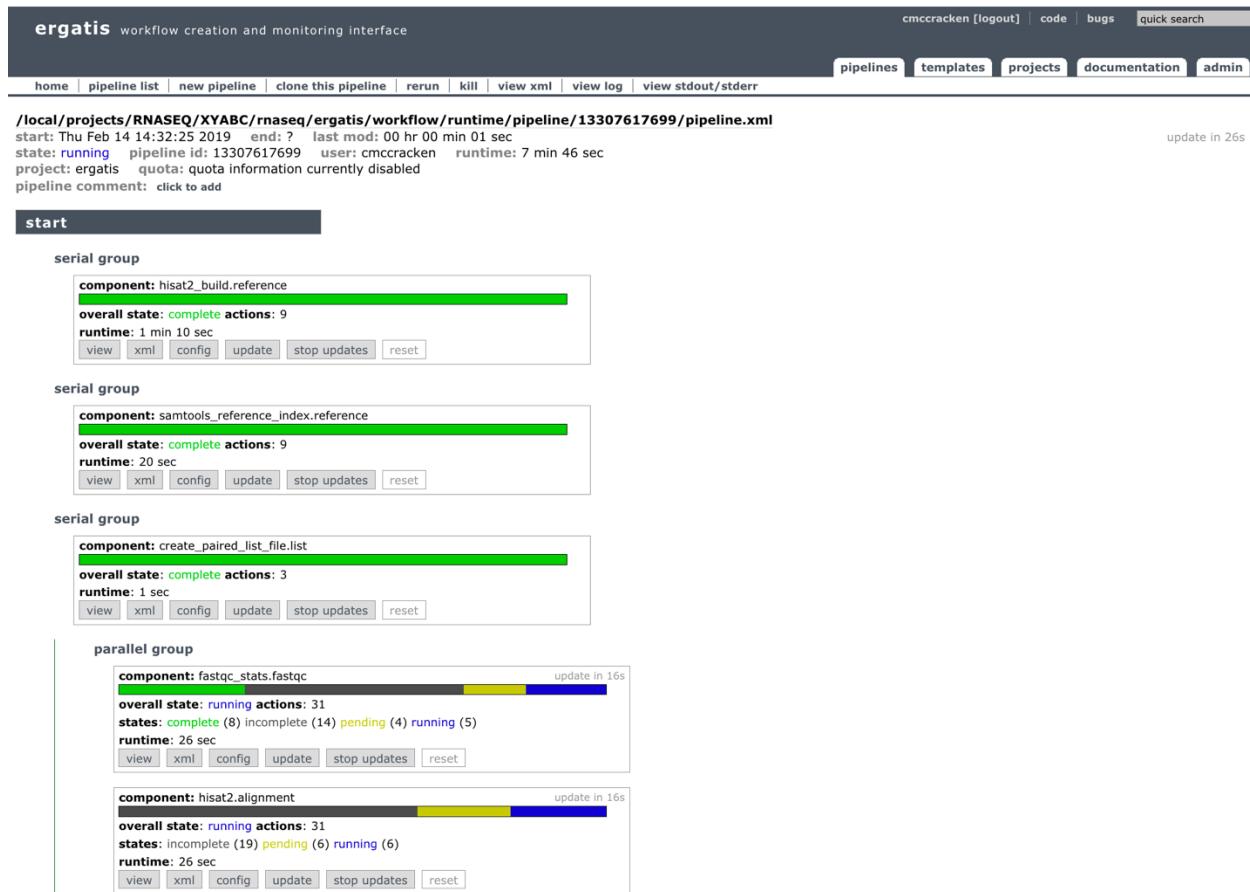
Status	Launch Time	Time Elapsed
running	02/14/19 - 14:32	0hr 4min

**Refresh Page** **Create BD Bag**

**Generate Report** **Download Bag**

Warning: Do not hit the BACK button in the browser

The Ergatis window is another view for watching progress of your pipeline



When the pipeline is complete, the entire progress bar will be green. You can now hit the “Create BDBag” button.

**GROTTO**  
Transcriptional Analysis Pipeline

Pipeline 13307617699

[cmccracken] help create\_pipeline recent\_pipelines logout

Progress	View in Ergatis	

Status	Launch Time	Time Elapsed
complete	02/14/19 - 14:32	0hr 12min

[Refresh Page](#) [Create BDBag](#)

[Generate Report](#) [Download Bag](#)

Warning: Do not hit the BACK button in the browser

You have created a BDBag. Now you can hit the Generate Report button.

**GROTTO**  
Transcriptional Analysis Pipeline

Pipeline 13307617699

[cmccracken] help create\_pipeline recent\_pipelines logout

- BDBag object can be found at /local/projects/RNASEQ/bdbag\_output
- BDBag object has been successfully created!

Component	Status	Time Elapsed

Status	Launch Time	Time Elapsed
complete	02/14/19 - 14:32	0hr 12min

[Refresh Page](#) [Create BDBag](#)

[Generate Report](#) [Download Bag](#)

Warning: Do not hit the BACK button in the browser

This process can take a few minutes. When you get the message on your screen that your report has been successfully generated, you can download the bag that contains your reports as a zip archive.

The screenshot shows the Grotto Transcriptional Analysis Pipeline interface. At the top, there's a purple header with the Grotto logo and the text "Transcriptional Analysis Pipeline". Below the header, the pipeline ID "Pipeline 13307617699" is displayed. To the right of the pipeline ID are navigation links: [cmccracken], help, create\_pipeline, recent\_pipelines, and logout. A green progress bar indicates the pipeline is complete. Next to it is a link "View in Ergatis". On the right side, a message box displays two bullet points: "Generated reports can be found at /local/projects/RNASEQ /bdbag\_output/reports/XYABC\_13307617699" and "Report has been successfully generated!". Below this message is a table with three columns: Component, Status, and Time Elapsed. The table shows one row with the status "complete". At the bottom of the page are several buttons: Refresh Page, Create BDBag, Generate Report, and Download Bag. A yellow arrow points to the "Download Bag" button. A warning message at the bottom left says: "Warning: Do not hit the BACK button in the browser".

## Troubleshooting pipelines that fail to progress

In grotto, if the arrows in the progress bar turn green without another turning blue before reaching the conclusion of the pipeline, your pipeline may be stalled.

There have been problems with workflow. First, wait a few minutes. It can take a minute or two for the window to update. If you still have a pipeline that is running but all component are either complete or incomplete and no components are running or pending, you will need to fix this.

Log onto the grid and enter qstat

```
[cmccracken@kano ~]$ qstat
job-ID  prior  name      user      state submit/start at      queue      slots ja-task-ID
-----  -----
8138966 0.75500 pipeline_1 cmccracken  r      01/10/2019 10:57:03 submit.q@grid-1-3-2.igs.umaryl  1
```

See how there is a pipeline running but no workflow. We need to restart our pipeline. We start by deleting the stalled job. We do this with command “qdel” and the job ID for the stalled pipeline.

```
[cmccracken@kano ~]$ qdel 8138966  
cmccracken has registered the job 8138966 for deletion
```

Now return to the ergatis window and hit the “rerun” button.

The screenshot shows the ergatis interface with a running pipeline. The top navigation bar includes links for pipelines, templates, projects, documentation, and admin. A yellow arrow points to the 'rerun' button in the top right of the header. The main content area displays pipeline details: /local/projects/RNASEQ/XYABC/rnaseq/ergatis/workflow/pipeline/13307617699/pipeline.xml. It shows the start time (Thu Feb 14 14:32:25 2019), end time (indicated as '?'), last modification (00 hours), runtime (1 min 01 sec), state (running), pipeline ID (13307617699), user (cmccracken), and project (ergatis). Quota information is noted as currently disabled. Below this, there are sections for 'start', 'serial group', and 'parallel group', each containing component status boxes with 'view', 'xml', 'config', 'update', 'stop updates', and 'reset' buttons. An 'overall state' box indicates 'complete' for the first serial group and 'incomplete' for the parallel group. A 'serial group' section also shows an 'overall state' of 'incomplete'. The interface includes a 'quick search' bar at the top right and a 'update in 13s' timer.