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.

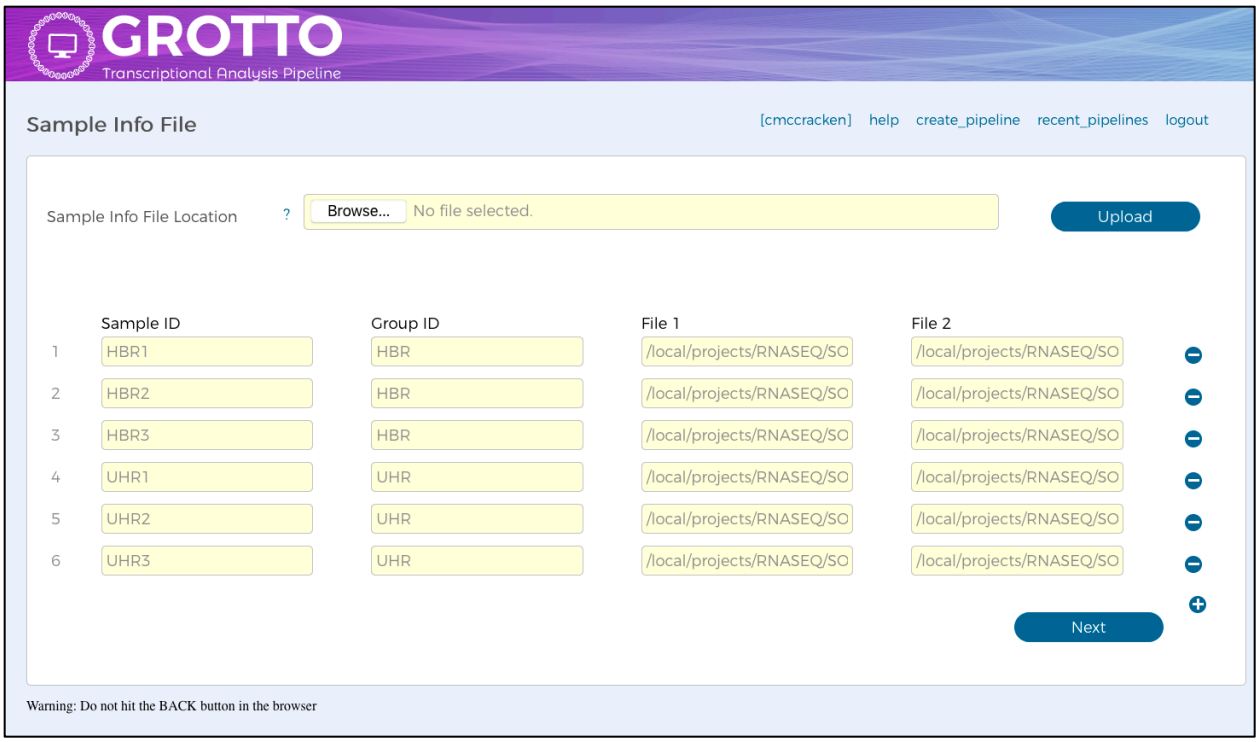
3. Enter sample information. This can be done either manually on the screen (4a) or automatically by file upload (4b).

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



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

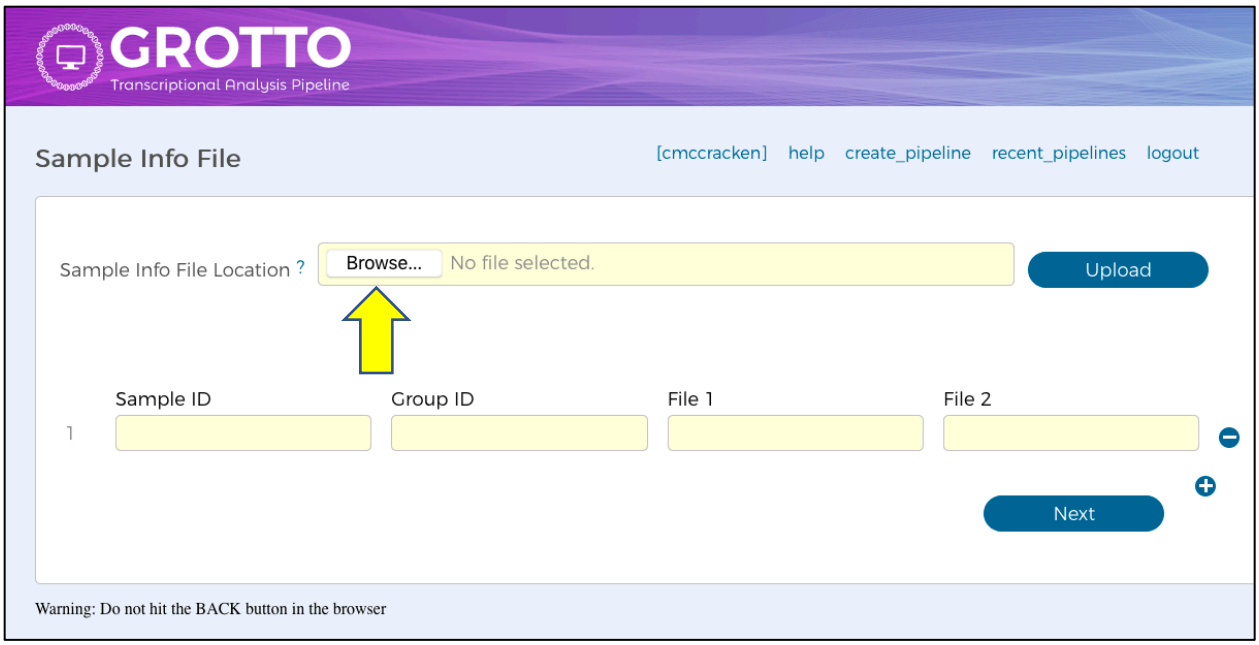
Create a tab-separated text file with a line for each sample that includes the sample name, group designation, read 1 and read2.

```
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
```

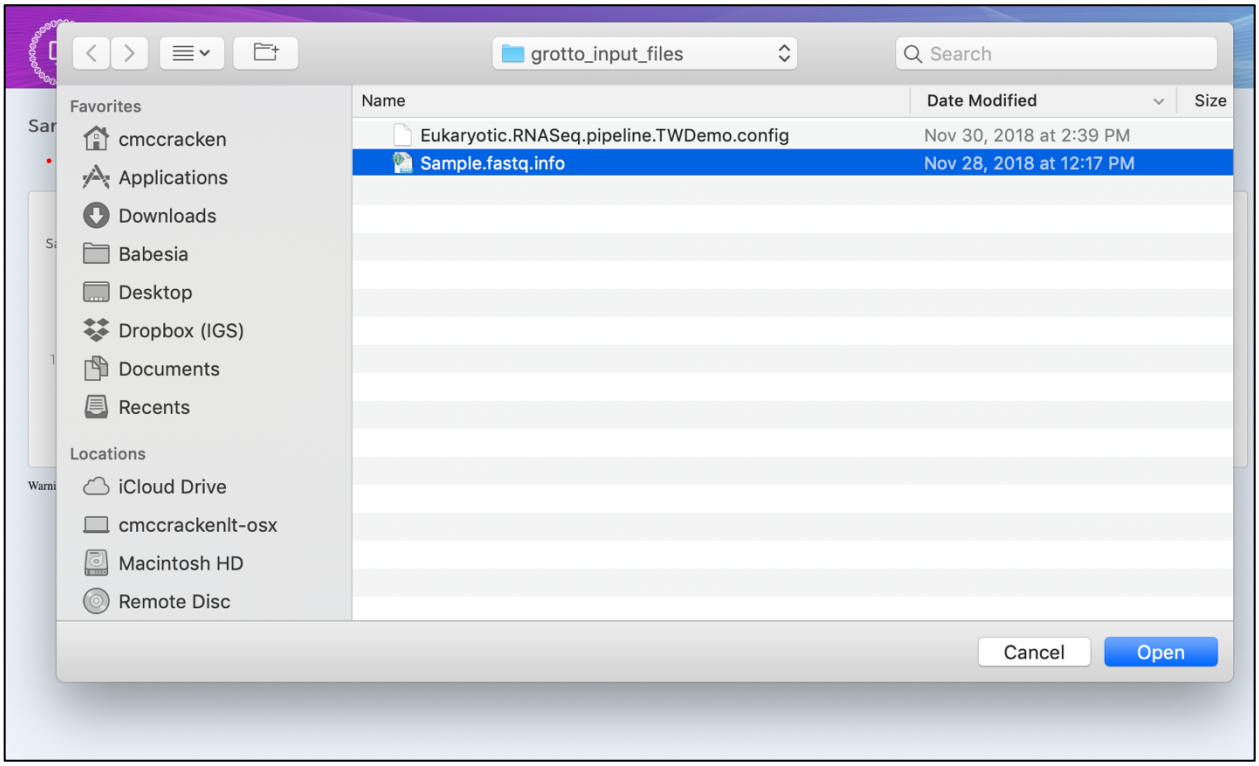
Please note that sample names and group names should not start with a number.

For testing, you can download /local/projects/RNASEQ/SOPs/grottoSOP/grotto_input_files/Sample.fastq.info to your local computer.

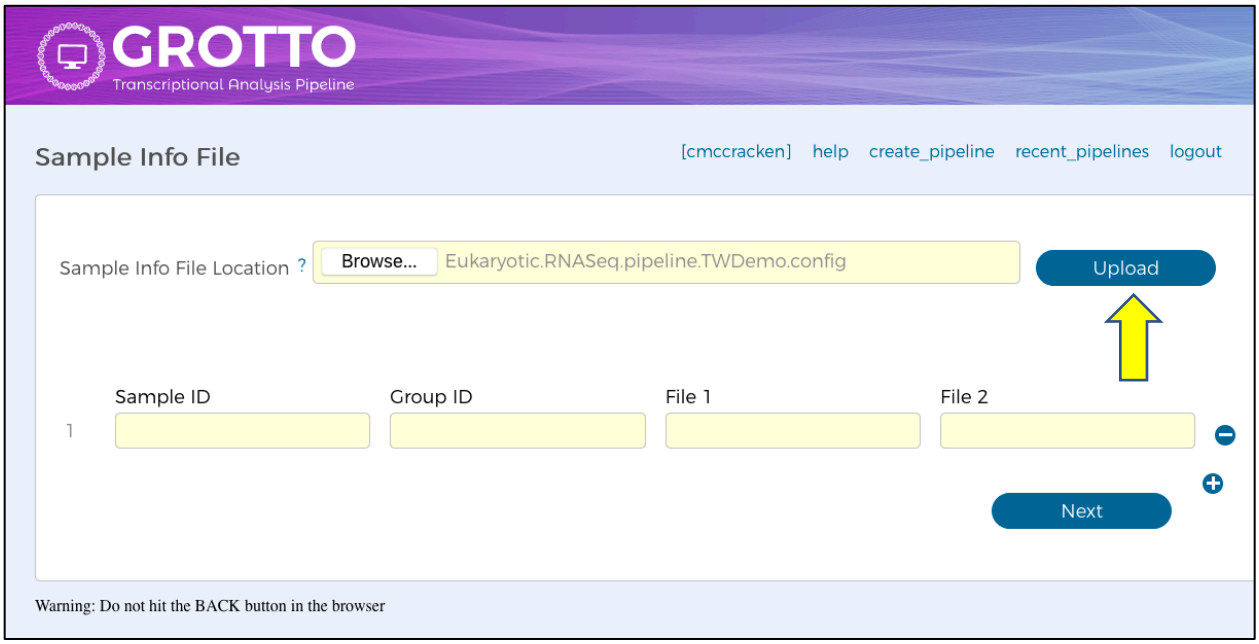
In the grotto window, click the browse button.



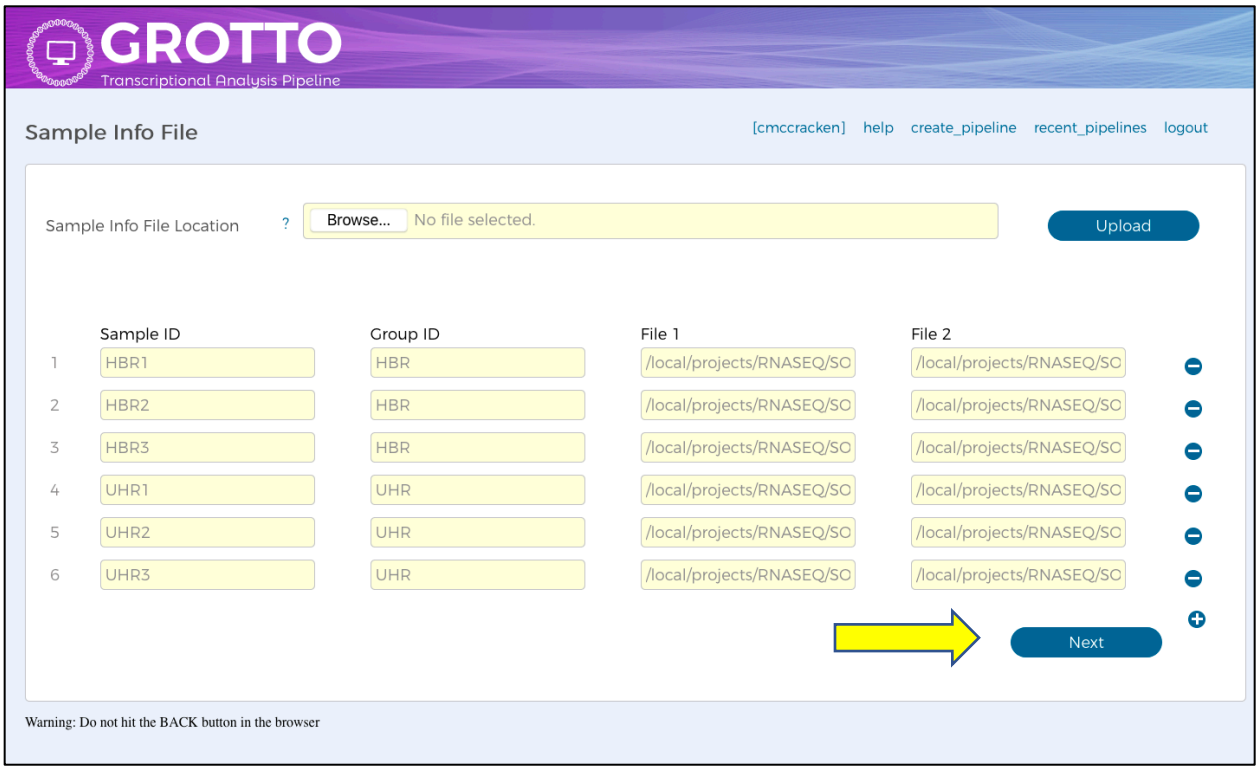
Select file from your local computer.



Click on Upload button.



Click the Next button.



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.

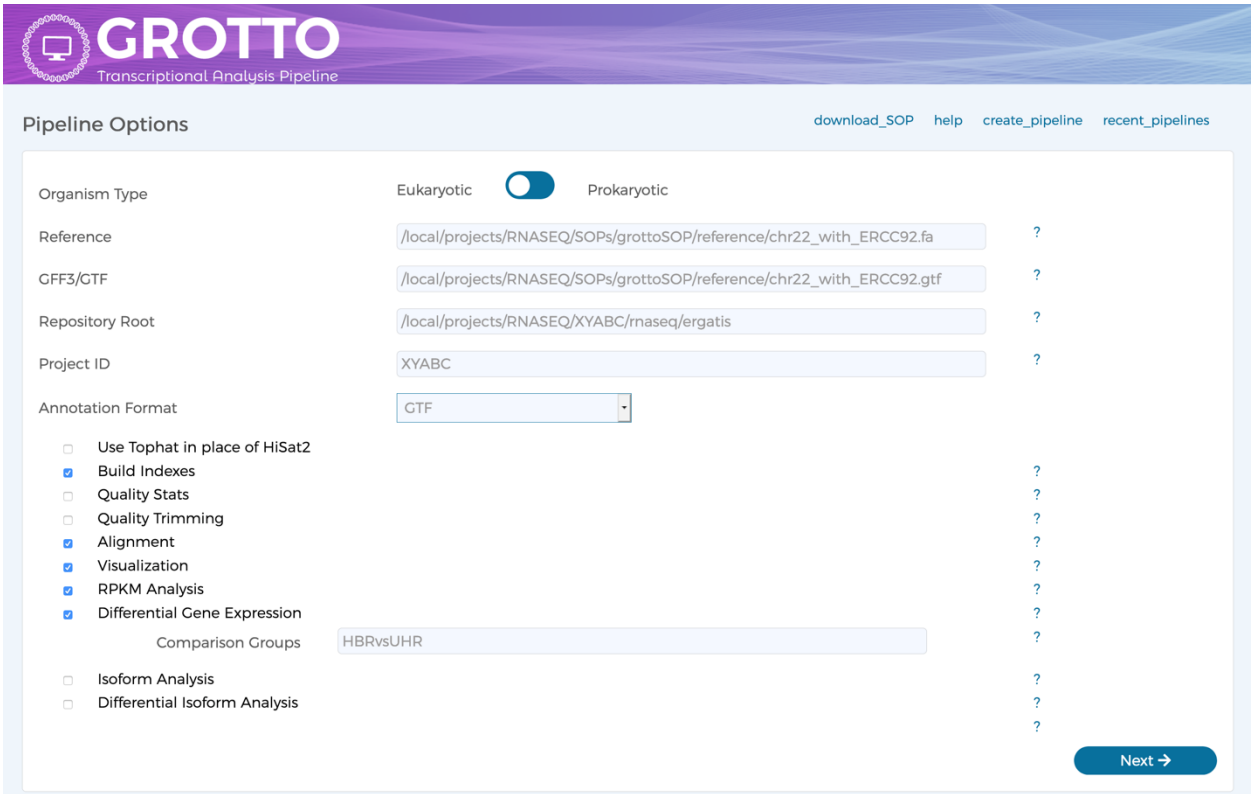
Add your project ID and select the annotation format (GTF or GFF3).

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.

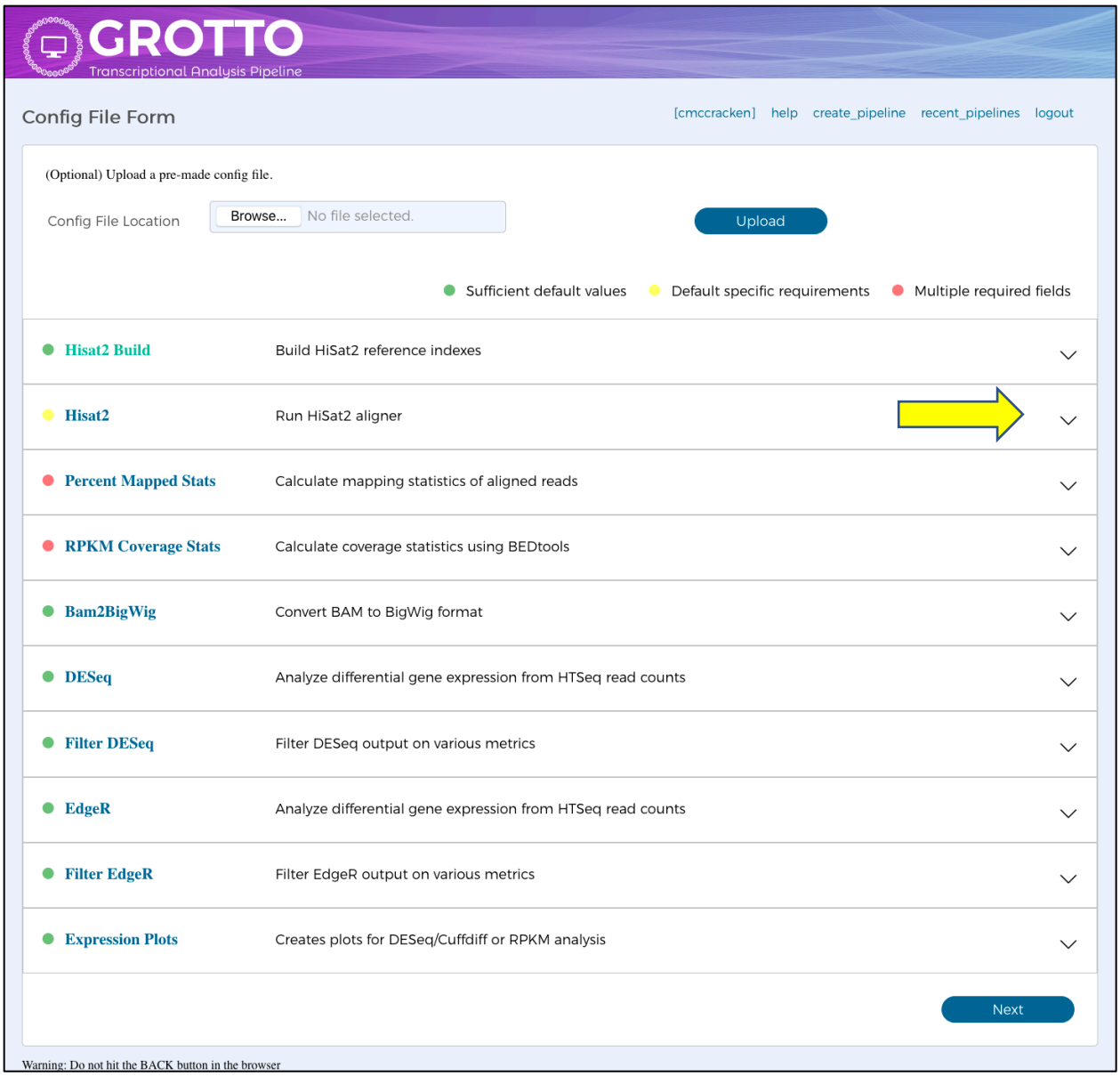


You can either configure your pipelines manually or by upload. Instructions for using the upload follow the instructions for manual entry.

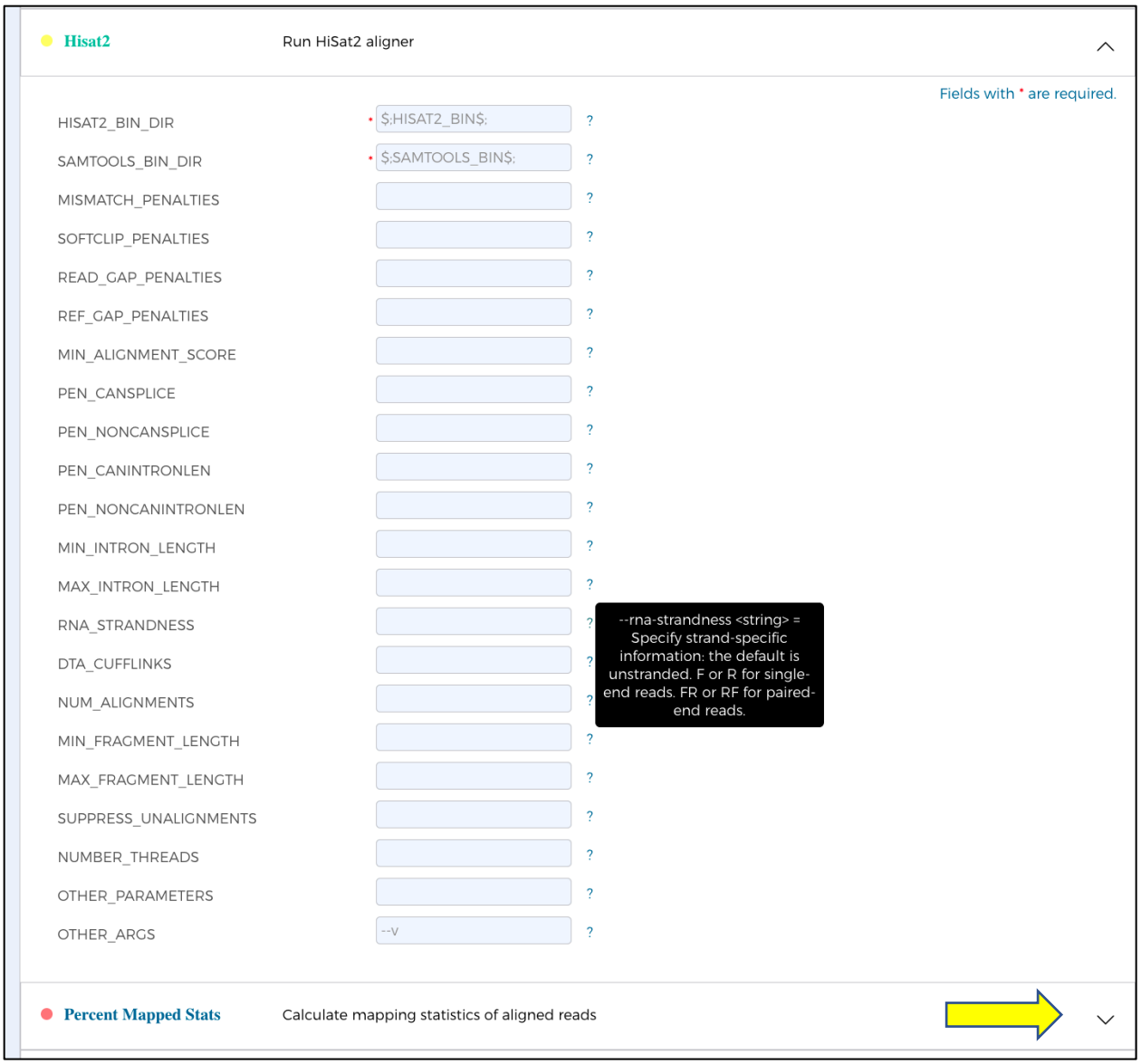
5. Configure your pipeline. This can be done either manually on the screen (6a) or automatically by file upload (6b).

5a. Configure your pipeline manually

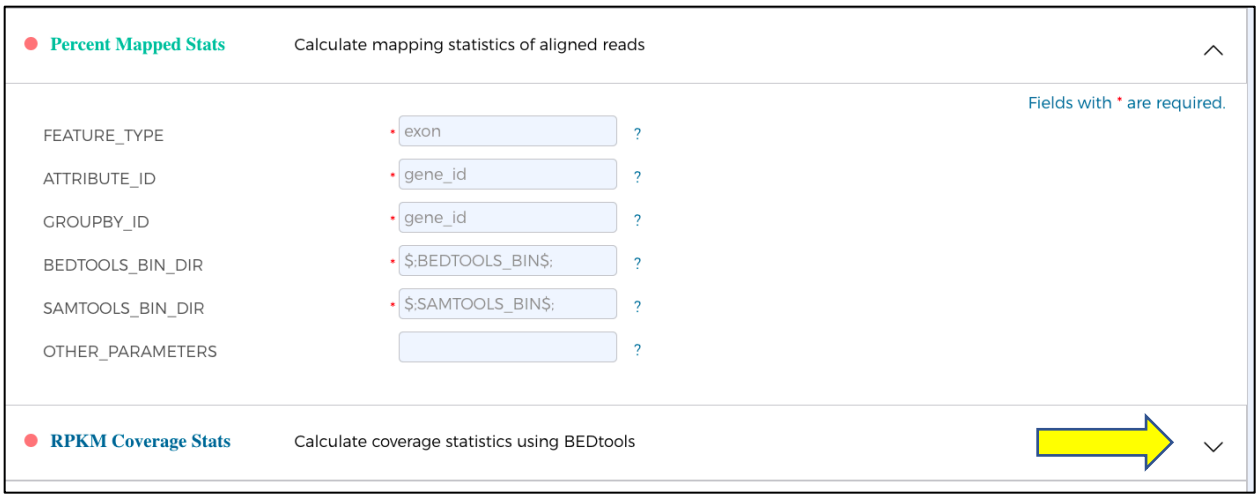
There are default settings for many of the pipeline parameters. The pipeline components with the green dot are have sufficient settings with the default parameters, but you will need to open the ones with the yellow or red dots to add parameters where there are not default. If you are interested in further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



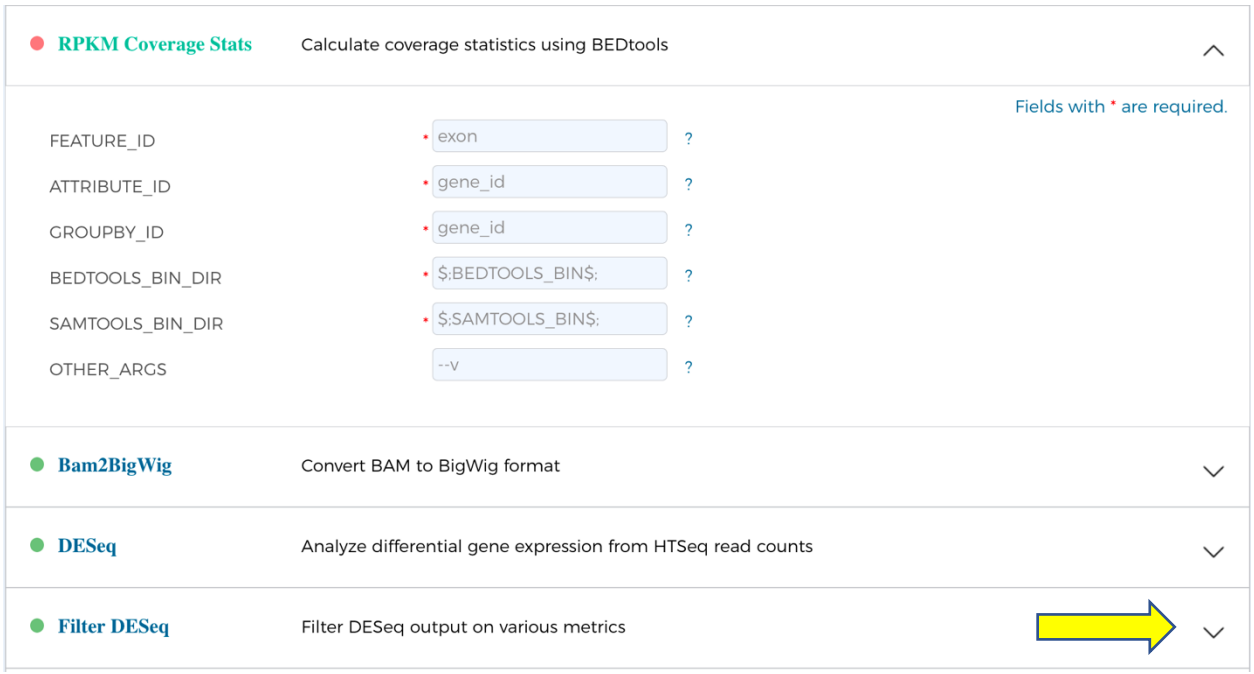
For most of these parameters, the default is usually OK. It is important to be aware of the strandness of your reads. The default is unstranded. The example sample data are unstranded.
Next click on the arrow to the right of Percent Mapped Stats.



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.

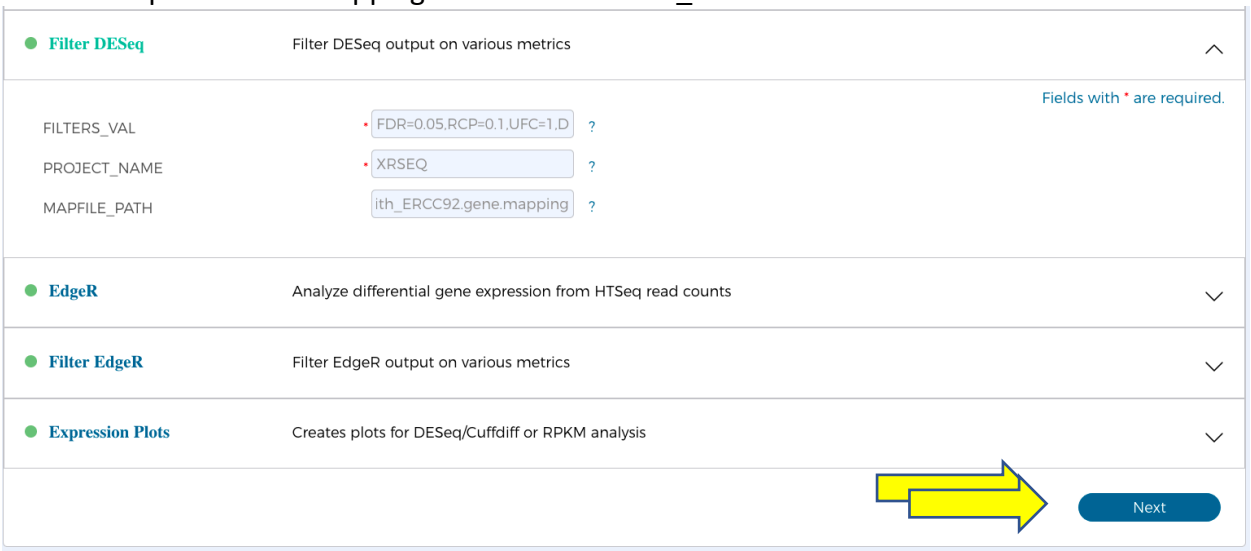


The parameters for RPKM Coverage Stats are the same as for Percent Mapped Stats. Click on the arrow to the right of Filter DESeq.



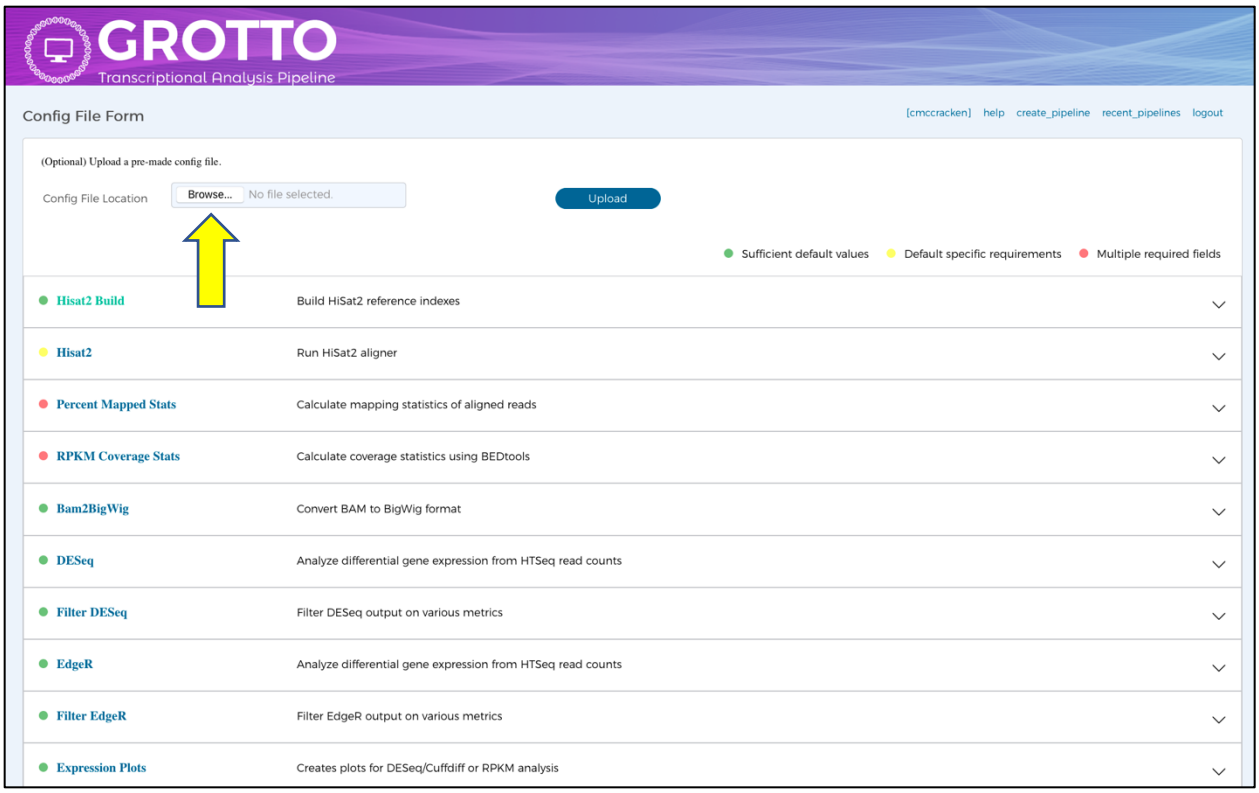
Enter the path for the mapping file in the MAPFILE_PATH. Click on the arrow to the right of FilterEdgeR.

Enter the path for the mapping file in the MAPFILE_PATH. Click on the Next button.

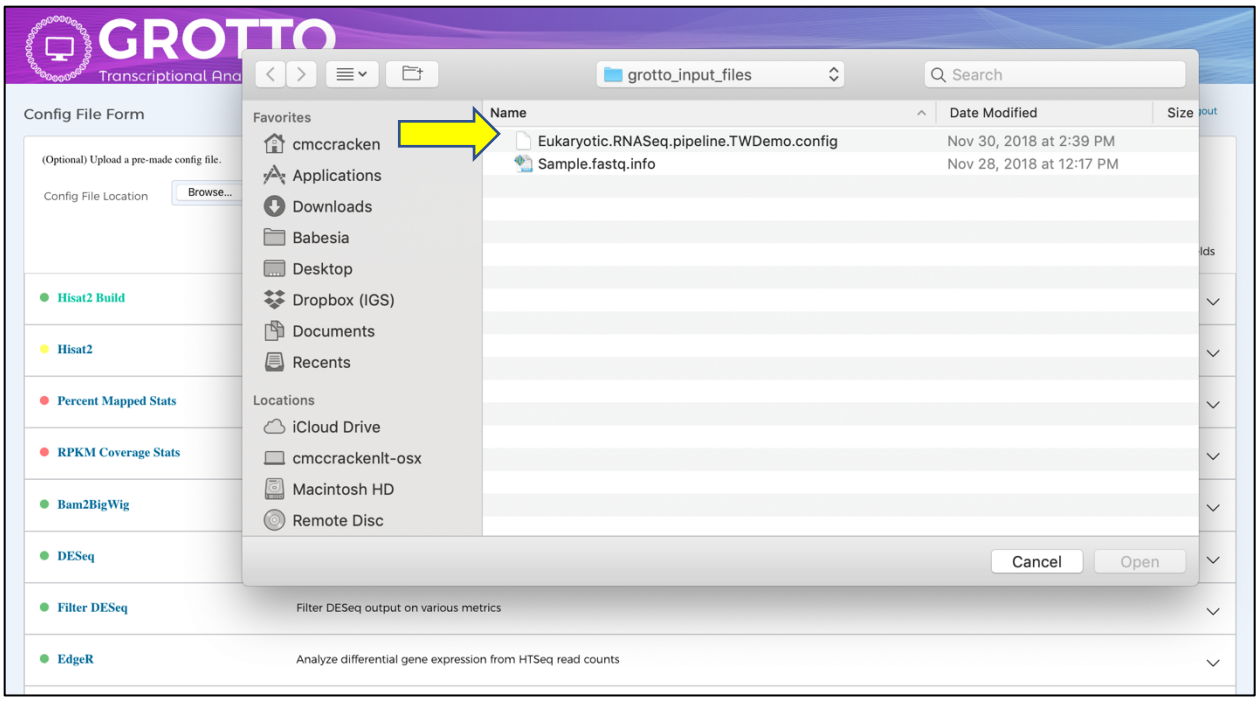


5b. Configure your pipeline by upload.

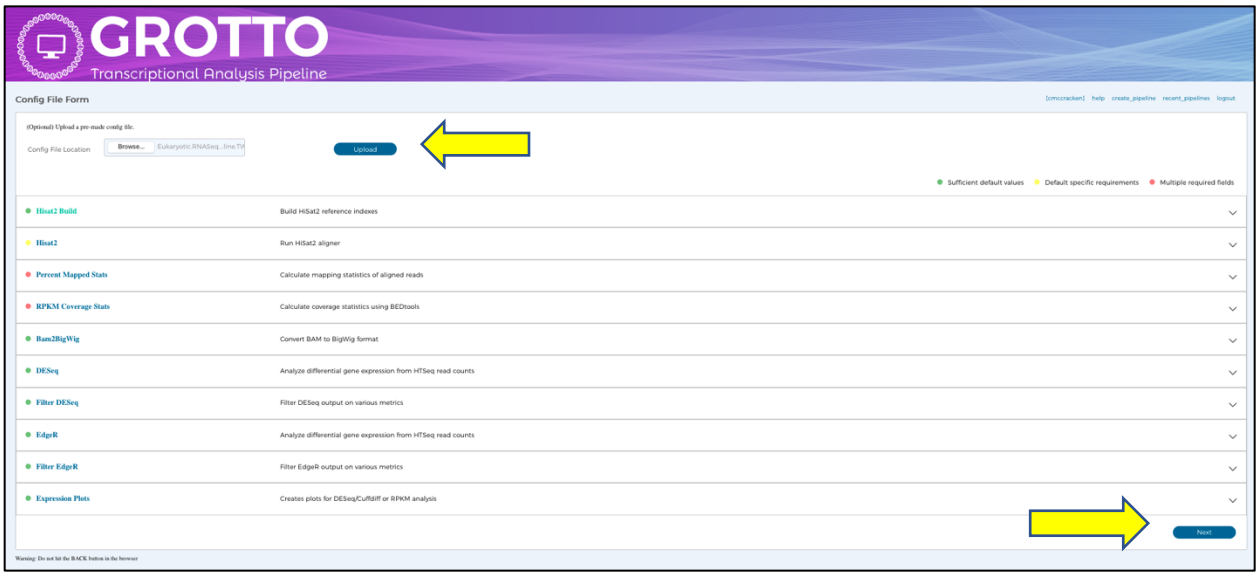
Click on the Browse button



Select your config file
For this example, you can download /local/projects/RNASEQ/SOPs/grottoSOP/grotto_input_files/Eukaryotic.RNASeq.pipeline.TWDemo.config to your computer locally

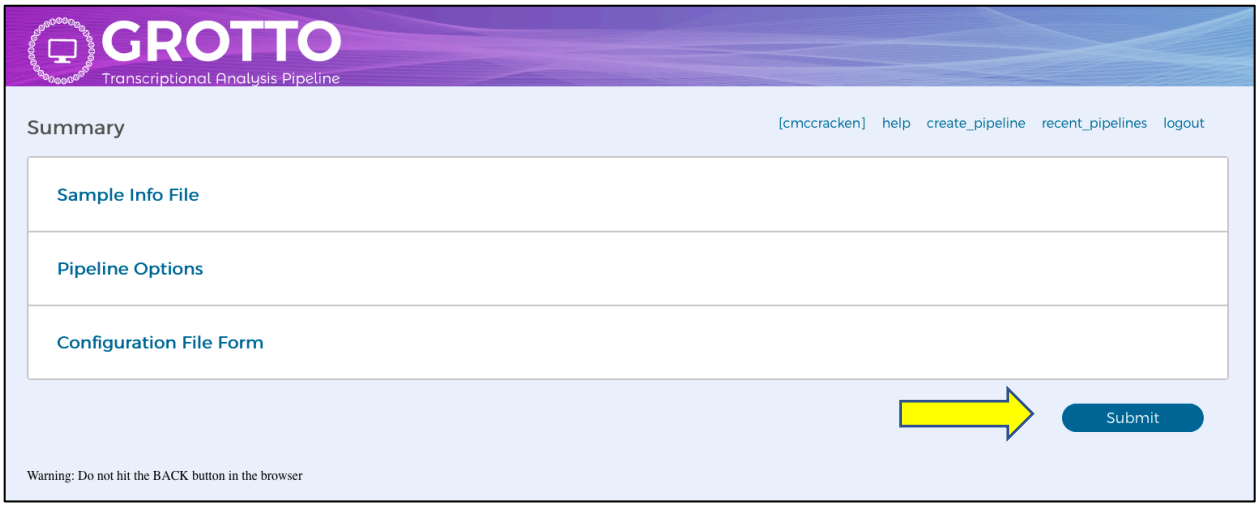


Click on the Upload button and then the Next button



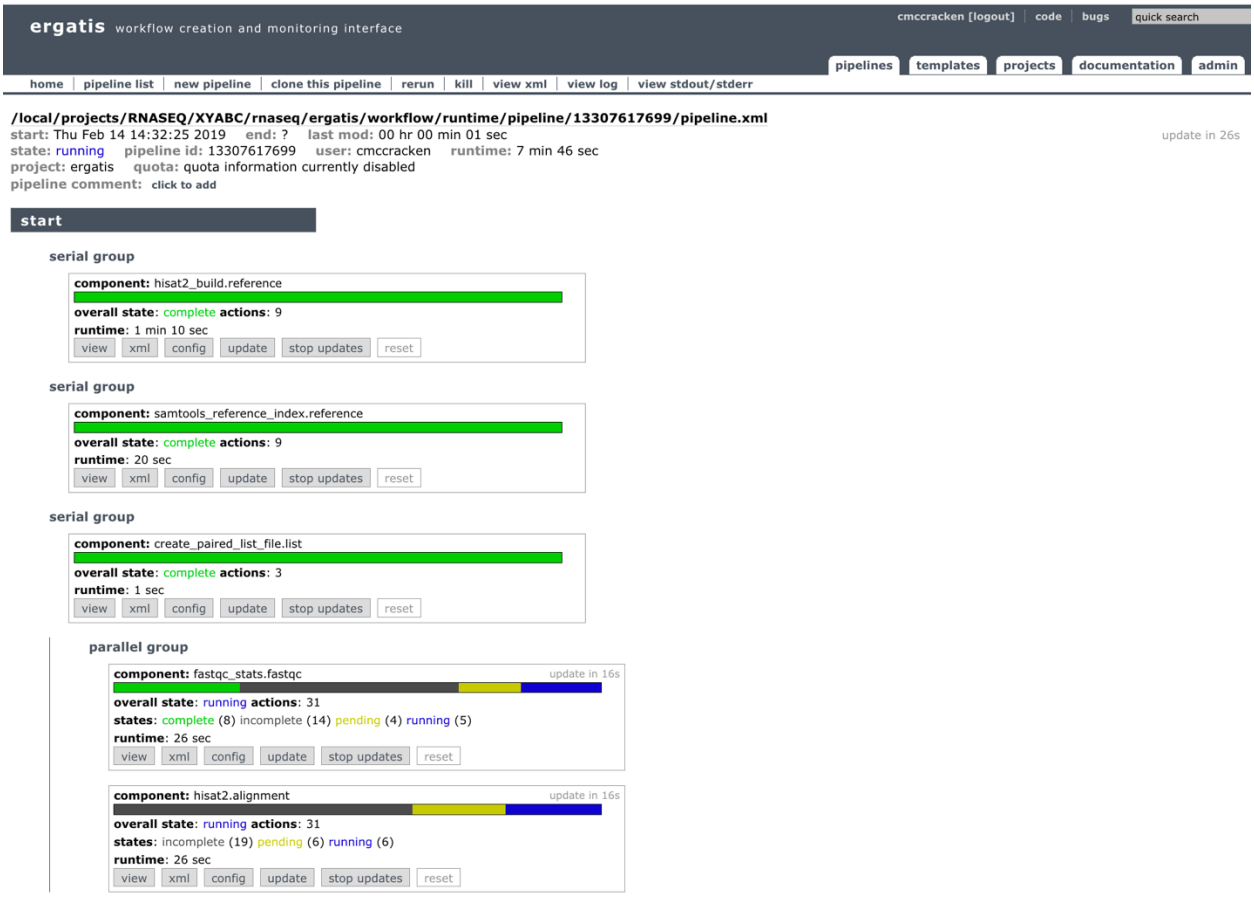
6. Launch your pipeline

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.

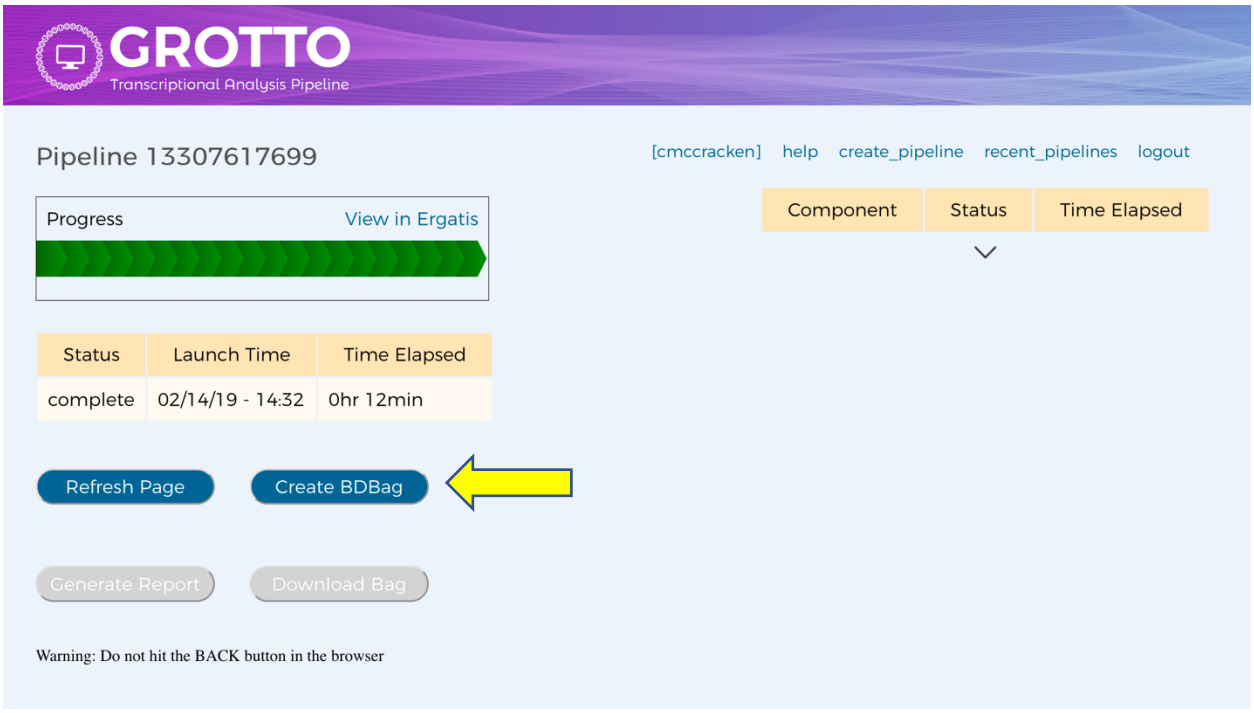


7. Watch the progress of your pipeline in ergatis. Grotto has a progress bar, but you can also click on the View in Ergatis button for a detailed view

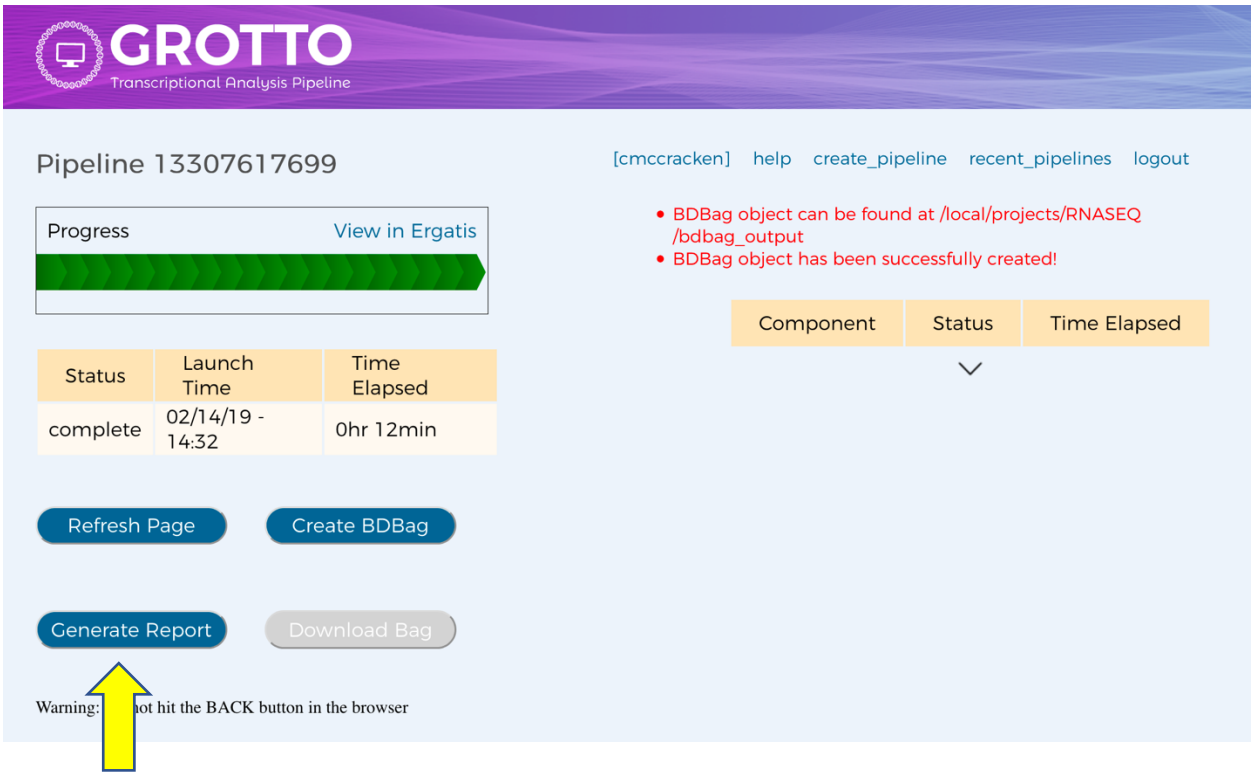
The Ergatis window is another view for watching progress of your pipeline



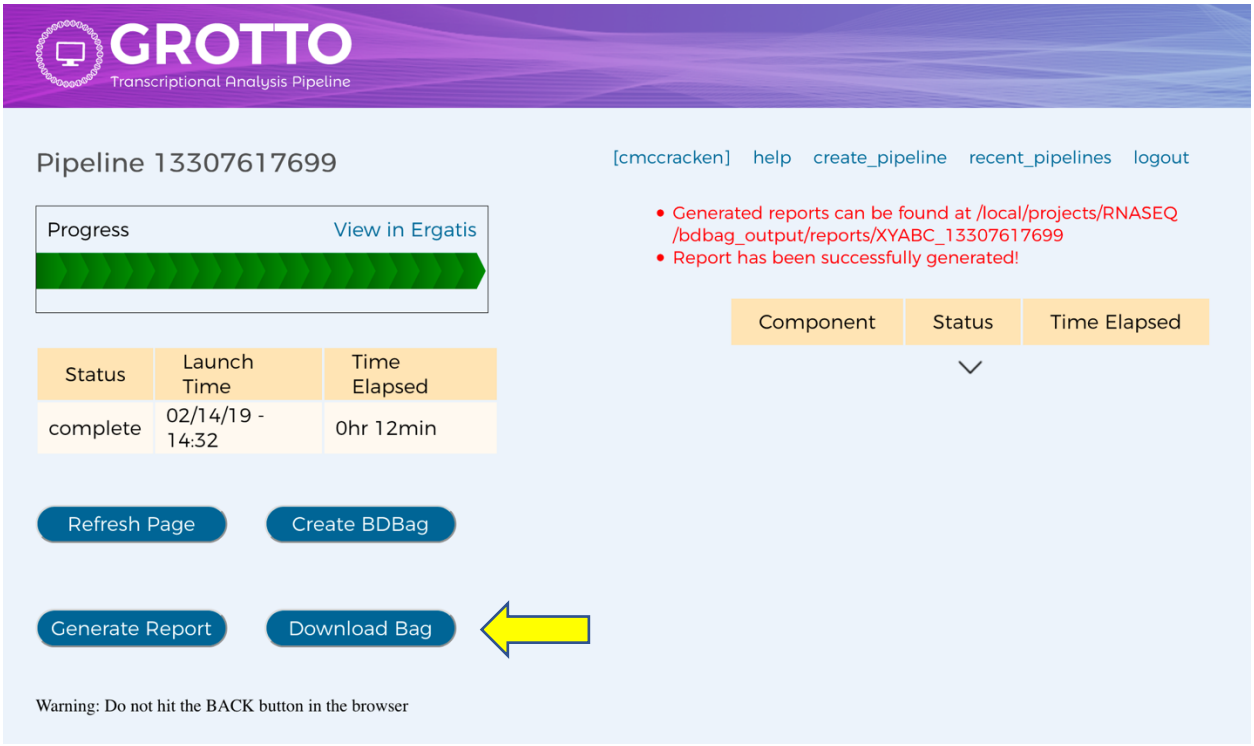
When the pipeline is complete, the entire progress bar will be green. You can now hit the “Create BDBag” button.



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



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.



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.