GEMmaker Documentation

Release 1.0

Connot Wytko, Shawna Spoor, Brian Soto, Stephen Ficklin

Jun 22, 2021

Contents:

GEMmaker is a Nextflow workflow for large-scale gene expression sample processing, expression-level quantification and Gene Expression Matrix (GEM) construction. Results from GEMmaker are useful for differential gene expression (DGE) and gene co-expression network (GCN) analyses. The GEMmaker workflow currently supports Illumina RNA-seq datasets.

CHAPTER 1

nf-core Compatibility

GEMmaker is an nf-core compatible workflow, however, GEMmaker is not an official nf-core workflow. This is because nf-core offers the nf-core/rnaseq workflow which is an excellent workflow for RNA-seq analysis that provides similar functionality to GEMmaker. However, GEMmaker is different in that it can scale to thousands of samples without exceeding local storage resources by running samples in batches and removing intermediate files. It can do the same for smaller sample sets on machines with less computational resources. This ability to scale is a unique adaption that is currently not provided by Nextflow. When Nextflow does provide support for batching and scaling, the nf-core/rnaseq will be updated and GEMmaker will probably be retired in favor of the nf-core workflow. Until then, if you are limited by storage GEMmaker can help!

CHAPTER 2

Acknowledgments

Development of GEMmaker was funded by the U.S. National Science Foundation Award #1659300.

2.1 Prerequisites

GEMmaker uses a variety of bioinformatics software packages, which will be downloaded to your machine automatically. **The GEMmaker workflow is downloaded automatically to your machine when you run it via Nextflow for the first time**. To do this, there are two dependencies you will need: Nextflow and your choice of container executor (Singularity or Docker).

2.1.1 Required Dependencies

At a minimum, GEMmaker requires the following:

- Nextflow : Executes the workflow. Nextflow also requires an installation of Java.
- Singularity or Docker. **(Singularity Recommended)**

2.1.2 Container Support

All of the software tools needed to run GEMmaker have been pre-installed into a Docker container. Therefore, you do not need to install them! Using the container can ensure that results from GEMmaker are always reproducible because the environment in which the software is executed will not change even if the host computational computer is updated. Please ensure one of these containerization services are installed.

- Singularity Community Edition (CE) **(recommended)**.
- Docker.

Choice of Computing Environment

2.1.3 Local Machine

GEMmaker can be run on a local computer; in particular, it has been tested on Ubuntu. GEMmaker has been optimized to allow for execution of large sample sizes without consuming large amounts of storage space. However, the time required to execute is dependent on the amount of computing power the machine has. You should consider using a High-Performance Computing (HPC) system if you have a large number of samples and would like GEMmaker to execute faster.

2.1.4 High-Performance Computing (HPC) Setup

On an HPC system it is recommended to use containerized dependencies as most HPC users do not have access to install software and HPC systems can change yielding results that may be hard to reproduce after time has passed and system setups have changed. Additionally, most HPC setups do not allow users to run Docker, but rather support Singularity instead. Using singularity is recommended on an HPC system rather than installing software dependencies manually. You will need to make sure that nextflow and Singularity are installed on your cluster (you may need the help of your HPC administrator).

2.2 Test GEMmaker

GEMmaker provides example data to quickly show how it works. This data consists of a small set of local files (contained with GEMmaker) and a remote sample from the NCBI's SRA repository. These samples are small to demonstrate usage for a mixed set of local and remote files. This example assumes you have Singularity installed.

Note: For the examples on this page, Singularity will be used. Singularity will automatically retrieve the GEMmaker Docker images and by default will store them in the `work` folder that Nextflow creates. However, Nextflow may warn that a cache directory is not set. If you intend to run GEMmaker multipe times, you may wish to designate a permanent cache directory by seting the `NXF_SINGULARITY_CACHEDIR` prior to running GEMmaker. You can learn more at the nf-core tools page

---

You can run the example by executing the following command within the GEMmaker directory:

```
nextflow run systemsgenetics/gemmaker -profile test,<docker/singularity/podman/
↪shifter/charliecloud/conda/institute>
```

Replace the text *<docker/singularity/podman/shifter/charliecloud/conda/institute>* with the execution profile of your choice. For example to test GEMmaker in a Singularity image the command would be.

```
nextflow run systemsgenetics/gemmaker -profile test,singularity
```

Results are stored in the `results` directory. You can find more information about the results in the **Use GEMmaker section**

2.2.1 About the Demo Test Data

The demo data provided by GEMmaker belongs to the imaginary CORG organism. For the local example, we use a set of 3 artificially made RNA-seq runs. The fictitious CORG organism has a very small "genome" of only 2,336 nucleotides, 3 "chromosomes" and 6 "genes". The 6 genes are named `gene_Alpha`, `gene_Beta`, `gene_Zeta`, `gene_Gamma`, `gene_Delta`, `gene_Epsilon`.

For the remote data file, GEMmaker automatically downloads a very small RNA-seq file from NCBI. This dataset is from an uncharacterized bacteria, but luckily, CORG shares 3 of the genes with this bacteria so we can use CORG's reference file. This remote sample was selected becasue it is an unusually small file, making it ideal for the example dataset.

2.3 Step 1: Prepare Genome Data

GEMmaker supports use of Hisat2, Kallisto and Salmon, and allows you to select one of these tools to use for quantification of gene expression. Each tool requires that transcript sequences of the genome are indexed prior to usage.

First, you must obtain the appropriate genome reference files and have them available on your local machine for indexing. Once you have obtained the files and placed them in a directory, you can index the genome by following the instructions in the sections below.

You only need to index the files for the tool (i.e. kallisto, salmon or hisat2) that you would like GEMmaker to use.

Note: Sometimes the genome assembly for a species may have successive releases, with each improving either on the genome assembly or the genome structural annotations (i.e. identified genes) or both. However, sometimes the functional annotation of genes may be lacking in most recent version as research communities developing the genome may release the genome for use prior to fully annotating it. When you select a genome reference for use with GEMmaker, choose the assembly with the best functional annotations for genes, or ensure that functional annotations for the release you choose are available. This will not affect the performance of GEMmaker but will affect downstream analyses such as identifying function of deferentially expressed genes or modules of genes from a co-expression network.

The instructions below provide examples for indexing the *Arabidopsis thaliana* genome as if it were obtained from Ensemble Plants.

2.3.1 Kallisto

Kallisto is the default tool that GEMmaker uses for gene transcript quantification. For Kallisto, you need the nucleotide sequences of all transcripts for your species in FASTA format. Kallisto recommends, for example, using the cDNA FASTA files similar to what you find on Ensembl genomes. After obtaining the file, you must use the `kallisto index` command to index the file. For more usage information please, see the Kallisto Online Manual.

For example, to retrieve the Arabidopsis cDNA file:

```
wget ftp://ftp.ensemblgenomes.org/pub/plants/release-50/fasta/arabidopsis_thaliana/
↪cdna/Arabidopsis_thaliana.TAIR10.cdna.all.fa.gz
```

Index Kallisto using Singularity

If you do not have Kallisto indexes already prepared for your reference genome, you can use the GEMmaker docker image to perform the indexing. For example, you can use Singularity in the following way:

```
singularity exec -B ${PWD} docker://systemsgenetics/gemmaker kallisto index -i␣
↪Arabidopsis_thaliana.TAIR10.kallisto.indexed Arabidopsis_thaliana.TAIR10.cdna.all.
↪fa.gz
```

The command above uses the `gemmaker/gemmaker` image that was built for GEMmaker. The image will be downloaded if it does not already exist on your machine. The command above uses the `-B ${PWD}` argument to automatically mount the current directory onto the same directory in the image. From there the Kallisto index command can be executed.

Index Kallisto using Docker

If you do not have Kallisto indexes already prepared for your reference genome, you can use the GEMmaker docker image to perform the indexing. For example, you can use Docker in the following way:

```
docker run -v ${PWD}:/reference -u $(id -u ${USER}):$(id -g ${USER}) systemsgenetics/
↪gemmaker /bin/bash -c "cd reference; kallisto index -i Arabidopsis_thaliana.TAIR10.
↪kallisto.indexed Arabidopsis_thaliana.TAIR10.cdna.all.fa.gz"
```

The command above uses the `gemmaker/gemmaker` image that was built for GEMmaker. The image will be downloaded if it does not already exist on your machine. The `-v ${PWD}:/references` argument instructs Docker to mount the current directory (i.e.: `${PWD}`) onto a new directory in the image named `/reference` and gives the image access to the transcript file for indexing. From there the Kallisto index command can be executed. The `-c` argument provides the Kallisto index command needed to index the files. The `-u $(id -u ${USER}):$(id -g ${USER})` argument instructs Docker to run Kallisto as you rather than the system root user.

2.3.2 Salmon

Salmon is similar in terms of performance and results as Kallisto. For Salmon, you need the nucleotide sequences of all transcripts for your species in FASTA format. Be sure to find a FASTA file containing cDNA sequences. After obtaining the file, you must use the `salmon index` command to index the file. For more usage information please, see the Salmon Online Manual.

As an example, to retrieve the Arabidopsis cDNA file:

```
wget ftp://ftp.ensemblgenomes.org/pub/plants/release-50/fasta/arabidopsis_thaliana/
↪cdna/Arabidopsis_thaliana.TAIR10.cdna.all.fa.gz
```

Index Salmon using Singularity

If you do not have Salmon indexes already prepared for your reference genome, you can use the GEMmaker docker image to perform the indexing. For example, you can use Singularity in the following way:

```
singularity exec -B ${PWD} docker://systemsgenetics/gemmaker salmon index index -t␣
↪Arabidopsis_thaliana.TAIR10.cdna.all.fa.gz -i Arabidopsis_thaliana.TAIR10.salmon.
↪indexed
```

The command above uses the `systemsgenetics/gemmaker` image to index the transcripts. The image will be downloaded if it does not already exist on your machine. The command above uses the `-B ${PWD}` argument to automatically mount the current directory onto the same directory in the image. From there the Salmon index command can be executed.

Index Salmon using Docker

If you do not have Salmon indexes already prepared for your reference genome, you can use the GEMmaker docker image to perform the indexing. For example, you can use Docker in the following way:

```
docker run -v ${PWD}:/reference -u $(id -u ${USER}):$(id -g ${USER}) systemsgenetics/
↪gemmaker /bin/bash -c "cd /reference; salmon index index -t Arabidopsis_thaliana.
↪TAIR10.cdna.all.fa.gz -i Arabidopsis_thaliana.TAIR10.salmon.indexed"
```

The command above uses the `systemsgenetics/gemmaker` image that was built by the GEMmaker development team to index the transcripts. The image will be downloaded if it does not already exist on your machine. The `-v ${PWD}:/references` argument instructs Docker to mount the current directory (i.e.: `${PWD}`) onto a new directory in the image named `/reference` and gives the image access to the transcript file for indexing. The `-c` argument provides the Salmon index command needed to index the files. The `-u $(id -u ${USER}):$(id -g ${USER})` argument instructs Docker to run Salmon as you rather than the system root user.

2.3.3 Hisat2

Hisat2 is different from Kallisto and Salmon in that it requires multiple steps that include alignment of RNA-seq reads to a genomic reference sequence followed by quantification of expression using the tool StringTie. You must therefore obtain the following files:

- A FASTA file containing the full genomic sequence in FASTA format (either pseudomolecules or scaffolds).
- A GTF file containing the gene models.

As an example, to retreive the Arabidopsis files:

```
wget ftp://ftp.ensemblgenomes.org/pub/plants/release-50/fasta/arabidopsis_thaliana/
↪dna/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa.gz
gunzip Arabidopsis_thaliana.TAIR10.dna.toplevel.fa.gz

wget ftp://ftp.ensemblgenomes.org/pub/plants/release-50/gff3/arabidopsis_thaliana/
↪Arabidopsis_thaliana.TAIR10.50.gff3.gz
gunzip Arabidopsis_thaliana.TAIR10.50.gff3.gz
```

Note: If your genome file is extremely large with hundreds of thousands of contigs/scaffolds, you may want to reduce the size of the FASTA file to contain only those contigs/scaffolds with predicted annotated genes.

Sometimes a genome assembly does not provide a GTF file, but rather provides a GFF3 file. This is the case for the Arabidopsis genome provided by Ensemble You can convert the GFF file to a GTF file using the gffread. Examples for using gffread are provdied below.

Index Hisat2 using Singularity

If you do not have a GTF or Hisat2 indexes already prepared for your reference genome, you can use the GEMmaker docker image to create the GTF and perform the indexing. For example, you can use Singularity in the following way:

To create the GTF file:

```
singularity exec -B ${PWD} docker://systemsgenetics/gemmaker  gffread Arabidopsis_
↪thaliana.TAIR10.50.gff3.gz -T -o Arabidopsis_thaliana.TAIR10.gtf
```

To index the reference:

```
singularity exec -B ${PWD} docker://systemsgenetics/gemmaker hisat2-build -f␣
↪Arabidopsis_thaliana.TAIR10.dna.toplevel.fa Arabidopsis_thaliana.TAIR10
```

The following describes the meaning of the arguments in the command-line above:

The command above uses the `systemsgenetics/gemmaker` image. The image will be downloaded if it does not already exist on your machine. The command above uses the `-B ${PWD}` argument to automatically mount the current directory onto the same directory in the image. From there the Hisat2 index command can be executed.

Index Hisat2 using Docker

If you do not have a GTF or Hisat2 indexes already prepared for your reference genome, you can use the GEMmaker docker image to create the GTF and perform the indexing. For example, you can use Docker in the following way:

To create the GTF file:

```
docker run -v ${PWD}:/reference -u $(id -u ${USER}):$(id -g ${USER}) systemsgenetics/
↪gemmaker /bin/bash -c "cd /reference; gffread Arabidopsis_thaliana.TAIR10.50.gff3 -
↪T -o Arabidopsis_thaliana.TAIR10.gtf"
```

To index the reference:

```
docker run -v ${PWD}:/reference -u $(id -u ${USER}):$(id -g ${USER}) systemsgenetics/
↪gemmaker  /bin/bash -c "cd /reference; hisat2-build -f Arabidopsis_thaliana.TAIR10.
↪dna.toplevel.fa Arabidopsis_thaliana.TAIR10"
```

The command above uses the `systemsgenetics/gemmaker` image. The image will be downloaded if it does not already exist on your machine. The `-v ${PWD}:/reference` argument instructs Docker to mount the current directory (i.e.: `${PWD}`) onto a new directory in the image named `/references` and gives the image access to the transcript file for indexing. The `-c` argument provides the Salmon index command needed to index the files. The `-u $(id -u ${USER}):$(id -g ${USER})` argument instructs Docker to run `hisat2-build` as you rather than the system root user.

2.4 Step 2: Prepare Sample Data

GEMmaker is capable of processing both locally stored RNA-seq files and automatically downloading samples stored in the NCBI SRA database. You can provide both types of files to be included in a single run of GEMmaker, or use only local or only remote files.

2.4.1 Using Samples From NCBI SRA

GEMmaker supports automatic download and processing of samples from the NCBI SRA repository. To use samples from the SRA, you must first find the list of NCBI SRA Run IDs of the samples you want to process. The run IDs typically start with an **SRR**, **ERR**, or **DRR** prefix. Do not confuse these with the Experiment IDs which typically start with SRX, ERX or DRX. The run IDs must be placed, one per line, in a file.

Example of a remote ID File:

```
SRR1058270
SRR1058271
SRR1058272
SRR1058273
SRR1058274
SRR1058275
SRR1058276
SRR1058277
```

2.4.2 Using Samples Stored Locally

By default, GEMmaker expects that FASTQ files are uncompressed (not GZ compressed). They can be stored in any directory on the local filesystem.

Paired FASTQ files

By default, paired files must have a `_1.fastq` and a `_2.fastq` suffix at the end of the filename. GEMmaker uses the `_1` and `_2` designation to differentiate and match paired files.

Non-Paired FASTQ files

By default, if your data is non-paired, GEMmaker expects all files to have a `_1.fastq` suffix at the end of the filename.

2.5 Step 3: Run GEMmaker

2.5.1 How to Launch GEMmaker

The demonstrate how to to use GEMmaker the *Arabidopsis thaliana* reference genome available from Ensembl Plants was prepared in Step 2. As an example, we will indicate 3 SRA files for automatic retrieval and processing by listing them in a file named `SRAs.txt`:

```
SRR1058270
SRR1058271
SRR1058272
```

If you followed the example in the previous step you should have the reference genome already indexed.

Note: For the examples on this page, Singularity will be used. Singularity will automatically retrieve the GEMmaker Docker images and by default will store them in the `work` folder that Nextflow creates. However, Nextflow may warn that a cache directory is not set. If you intend to run GEMmaker multipe times, you may wish to designate a permanent cache directory by seting the `NXF_SINGULARITY_CACHEDIR` prior to running GEMmaker. You can learn more at the nf-core tools page

---

Use Kallisto

To run Kallisto you need to specify:

- The path to the genome reference indexed file
- A file containing a set of SRA run IDs you want to download or the path were FASTQ files are stored on the local system.

For example:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --sras SRAs.txt
```

Use Salmon

To run Salmon you need to specify:

- The path to the directory containing the genome reference index files.
- A file containing a set of SRA run IDs you want to download or the path were FASTQ files are stored on the local system.

For example:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline salmon \
  --salmon_index_path Arabidopsis_thaliana.TAIR10.salmon.indexed \
  --sras SRAs.txt
```

Use Hisat2

To run Hiast2 you need to specify:

- The path to directory containing the Hisat2 genome reference indexed files
- The base name of the whole genome. All Hisat2 index files use this base name. For this example, the base name used is `Arabidopsis_thaliana.TAIR10`.
- The GTF file containing the gene annotations.
- A file containing a set of SRA run IDs you want to download or the path were FASTQ files are stored on the local system.

For example:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline hisat2 \
  --sras SRAs.txt \
  --hisat2_base_name Arabidopsis_thaliana.TAIR10 \
  --hisat2_index_dir hisat2_indexes \
  --hisat2_gtf_file Arabidopsis_thaliana.TAIR10.gtf
```

Additionally, you can control the Trimmomatic trimming step by adding any of the following parameters:

- `--trimmomatic_clip_file`: the location for a custom file of sequences to clip. GEMmaker provides a default version so you only need to set this if you have custom sequences.
- `--trimmomatic_MINLEN`: corresponds to the `MINLEN` argument of Trimmomatic. Defaults to 0.7.
- `--trimmomatic_SLIDINGWINDOW`: corresponds to the `SLIDINGWINDOW` argument of Trimmomatic. Defaults to "4:15"
- `--trimmomatic_LEADING`: corresponds to the `LEADING` argument of Trimmomatic. Defults to 3.
- `--trimmomatic_TRAILING`: correponds to teh `TRAILING` argument of Trimmomatic. Defaults to 6.

Use Local FASTQ Files

If your FASTQ files are local to your computer you must provide the `--input` argument when launching Nextflow and indicate the GLOB pattern than is needed to find the files:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --input "../../01-input_data/RNA-seq/fastq/*{1,2}.fastq"
```

In the example above the `--input` argument indicates that FASTQ files are found in the `../../01-input_data/RNA-seq/fastq/` directory and GEMmaker should use all files that match the GLOB pattern `*{1,2}.fastq`.

Note: GEMmaker currently expects that all fASTQ files have a *1* or *2* suffix. For paired files two files with the same name but each suffix respectively.

Use Both Local and SRA Files

You can combine data from the NCBI SRA with local files in a single run of GEMmaker by providing both the `--sras` and `--input` arguments.

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --input "../../01-input_data/RNA-seq/fastq/*{1,2}.fastq" \
  --sras SRAs.txt
```

Using Paired-End Local Data

If your data is paired-end you must provide a GLOB pattern for the `--input` argument that can distinguish between the sample name and the suffix that indicates the pair. Usually, paired-files have a `1.fastq` or `2.fastq` suffix on all file names. Therefore, the GLOB given example given above is appropriate: `*{1,2}.fastq`. The `{1,2}` indicates where the `1` and `2` are at in file name. However, if your files are named differently, be sure to use a GLOB pattern that can differentiate the pairs.

Warning: If the GLOB you provide cannot distinguish between pairs then GEMmaker will treat them as non-paired.

Using Non Paired-End Local Data

If your data is not paired-end then the GLOB pattern for the `--input` argument simply needs to find all of the FASTQ files. For example, if your FASTQ files have a `.fastq` suffix the following GLOB would be appropriate: `*.fastq"`.

Using Both Paired-End and Non Paired Local Data

GEMmaker can work with both paired and non-paired data in the same data set. The only stipulation is that the non-paired data must follow the same naming convention as the paired data. See the section *Using Paired-End Local data*. For example, if your paired files have a `1.fastq` and `2.fastq` extension, then the non-paired files should have a `1.fastq` suffix as well.

2.5.2 Resuming After Failure

If for some reason GEMmaker fails to fully complete and Nextflow reports some form of error. You can resume execution of the workflow, afer correcting any problems, by passing the `-resume` flag to GEMmaker. For example to resume a failed Kallisto run:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  -resume \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --sras SRAs.txt
```

GEMmaker should resume processing of samples without starting over.

Skipping Samples

You may find that a sample is problematic. It may be corrupt, does not align or has other problems that may cause GEMaker to fail. For such samples that cause GEMmaker to fail, you have two options. You can either remove the bad samples and restart GEMmaker or you can resume, as just described in the previous section, but first add the sample names to a new file, one per line, then, use the `--skip_samples` argument to tell GEMmaker about this file. For example:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --sras SRAs.txt \
  --skip_samples samples2skip.txt
```

In the example above any samples that should be skipped should be added to the `samples2skip.txt` file.

Warning: Note, when you provide SRA IDs to GEMmaker you provide the RUN IDs, but multiple run IDs can be contained in a single sample. To skip a sample, you must provide the sample ID. For SRA, these begin with the prefix SRX, DRX or ERX, where as run IDs begin with SRR, DRR or ERR.

2.5.3 Running on a Cluster

If you want to run GEMmaker on a local High Performance Computing Cluster (HPC) that uses a scheduler such as SLURM or PBS, you must first create a configuration file to help GEMmaker know how to submit jobs. The file should be named `nextflow.config` and be placed in the same directory where you are running GEMmaker. Below is an example `nextflow.config` file for executing GEMmaker on a cluster that uses the SLURM scheduler.

```
profiles {
   my_cluster {
      process {
         executor = "slurm"
         queue = "<queue name>"
         clusterOptions = ""
      }
      executor {
         queueSize = 120
      }
   }
}
```

In the example above we created a new profile named `my_cluster`. Within the stanza, the placeholder text `<queue name>` should be replaced with the name of the queue on which you are allowed to submit jobs. If you need to provide specific options that you would normally provide in a SLURM submission script (such as an account or other node targetting settings) you can use the `clusterOptions` setting.

Next, is an example SLURM submission script for submitting a job to run GEMmaker. Please note, this is just an example and your specific cluster may require slightly different configuration/usage. The script assumes your cluster uses the lmod system for specifying software.

```
#!/bin/sh
#SBATCH --partition=<queue_name>
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=10:00:00
#SBATCH --job-name=GEMmaker
#SBATCH --output=%x-%j.out
#SBATCH --error=%x-%j.err

module add java nextflow singularity

nextflow run systemsgenetics/gemmaker \
  -profile my_cluster,singularity \
  -resume \
  --pipeline kallisto \
  --kallisto_index_path Araport11_genes.201606.cdna.indexed \
  --sras  SRA_IDs.txt \
  --max_cpus 120
```

Notice in the call to nextflow, the profile `my_cluster` has been added along with `singularity`, also, the `--max_cpus` argument has been set to the same size as the `queueSize` value in the config file. The default value of `--max_cpus` is 4 and won't allow the workflow to expand beyond 4 CPUs if it is not increased to match the config file.

2.5.4 Intermediate Files

GEMmaker was designed to limit the storage requirements in order to allow for processing of large numbers of FASTQ files without overrunning storage requirement. By default it will remove all large intermediate files to keep space usage to a minimum. However, you can indicate what intermediate files you would like to keep by providing any of the following arguments and setting them to `true`. For example, to keep the downloaded SRA files the `keep_sra` argument would be provided and set to true:

```
nextflow run systemsgenetics/gemmaker -profile singularity \
  --pipeline salmon \
  --salmon_index_path Arabidopsis_thaliana.TAIR10.salmon.indexed \
  --sras SRAs.txt \
  --keep_sra true
```

The following is a listing of all arguments that can control which intermediate files are kept.

SRA Files

The following arguments can be used if the `--sras` option is used.

- `--keep_sra`: Set to true to keep all downloaded SRA files .

- `--keep_retrieved_fastq`: Set to true to keep the FASTQ files that are derived from downloaded SRA files.

Kallisto Files

The following arguments can be used if the `--pipeline kallisto` option is used.

- `--kallisto_keep_data`: Set to true to keep the intermediate files created by Kallisto.

Salmon Files

The following arguments can be used if the `--pipeline salmon` option is used.

- `--kallisto_keep_data`: Set to true to keep the intermediate files created by Salmon.

Hisat2 Files

The following arguments can be used if the `--pipeline hisat2` option is used.

- `--hisat2_keep_data`: Set to true to keep the stringtie output.
- `--hisat2_keep_sam`: Set to true to keep the SAM files created by Hisat2.
- `--hisat2_keep_bam`: Set to true to keep the BAM files created by Hisat2.
- `--trimmomatic_keep_trimmed_fastq`: Set to true to keep the trimmed FASTQ files after trimmomatic is run.

2.5.5 Configuration

The instructions above provide details for running GEMmaker using Singularity. For most instances you probably won't need to make customizations to the workflow configuration. However, should you need to, GEMmaker is a nf-core compatible workflow. Therefore, it follows the general approach for workflow configuration which is described at the nf-core Pipeline Configuration page. Please see those instructions for the various platforms and settings you can configure. However, below are some quick tips for tweaking GEMmaker.

In all cases, if you need to set some customizations you must first create a configuration file. The file should be named `nextflow.config` and be placed in the same directory where you are running GEMmaker.

Configuration for a Cluster

To run GEMmaker on a computational cluster you will need to to create a custom configuration. Instructions and examples are provided in the *Running on a Cluster* section.

Increasing Resources

You may find that default resources are not adequate for the size of your data set. You can alter resources requested for each step of the GEMmaker workflow by using the `withLabel` scope selector in a custom `nextflow.config` file.

For example, if you have thousands of SRA data sets to process, you may need more memory allocated to the `retrieve_sra_metadata` step of the workflow. All steps in the workflow have a "label" that you can use to indicate which step resources should be changed. Below is an example `nextflow.config` file where a new profile named `custom` is provided where the memory has been increased for the `retrieve_sra_metadata`.

```
profiles {
    custom {
        process {
            withLabel:retrieve_sra_metadata {
                memory = "10.GB"
                }
        }
    }
}
```

This new `custom` profile can be used when calling GEMmaker. The following is an example Kallisto run of GEMmaker using the custom and singularity profiles:

```
nextflow run systemsgenetics/gemmaker -profile custom,singularity \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --sras SRAs.txt
```

Nextflow provides many "directives", such as `memory` that you can use to alter or customize the resources of any step (or process) in the workflow. You can find more about these in the Nextflow documentation. Some useful directives are:

- memory: change the amount of memory allocated to the step.
- time: change the amount of time allocated to the step.
- disk: defines how much local storage is required.
- cpus: defines how many threads (or CPUs) the task can use.

The "labels" that GEMmaker provides and which you can set custom directives include:

- `retrieve_sra_metadata`: For the step that retrieves metadata from the NCBI web services for the SRR run IDs that were provided. This step can require more memory than the defaults if there are huge numbers of samples.
- `download_runs`: For the step is used for downloading SRA files from NCBI.
- `fastq_dump`: For the step that is used after downloading SRA files and converting them to FASTQ files.
- `fastqc`: For the step where the FastQC program is used which generates quality reports on FASTQ files.
- `kallisto`: For the step the runs the Kallisto tool.
- `salmon`: For the step that runs the Salmon tool.
- `trimmomatic`: For the step that runs the Trimmomatic step which only runs when hisat2 is the desired pipeline.
- `hisat2`: For the step that runs the hisat2 tool.
- `samtools`: For the step that runs when the samtools tool is used after Hisat2 runs. This step only runs when the hisat2 pipeline is used.
- `stringtie`: For the step that runs the stringtie tool and which only runs when the hisat2 pipeline is used.
- `multiqc`: For the step that runs the MultiQC results summary report.
- `create_gem`: For the step that creates the final GEM files.
- `multithreaded`: For all of the tools that support multithreading you can use this label to set a default number of CPUs using the `cpus` directive. These tools include Salmon, Kallisto, Trimmomatic, Hisat2 and Stringtie. By using this label you set set the same number of `cpus` for all multithreaded steps at once.

2.5.6 Using the Development Version

New updates to GEMmaker, prior to issuing a formal release, are held in the `dev` branch of the GEMmaker github repository. It is recommended to always use a formal release of GEMmaker, however, you can test the most recent improvements prior to release. To do so, use the `-r dev` argument when running GEMmaker. For example:

```
nextflow run systemsgenetics/gemmaker -r dev -profile singularity \
  --pipeline kallisto \
  --kallisto_index_path Arabidopsis_thaliana.TAIR10.kallisto.indexed \
  --sras SRAs.txt
```

The `-r dev` argument forces Nextflow to use the development version of GEMmaker rather than the most recent stable version.

Note: You can find the most recent documentation for the `dev` branch at https://gemmaker.readthedocs.io/en/dev/

2.6 Step 4: View Output and Results

2.6.1 The Gene Expression Matrix (GEM)

After GEMmaker completes, it will have created a Gene Expression Matrix (GEM) that can be found in the `results/GEMs/` directory by default. This directory contains the final gene-expression matrices in raw, TPM and FPKM form, depending on the tool used.

2.6.2 Sample files

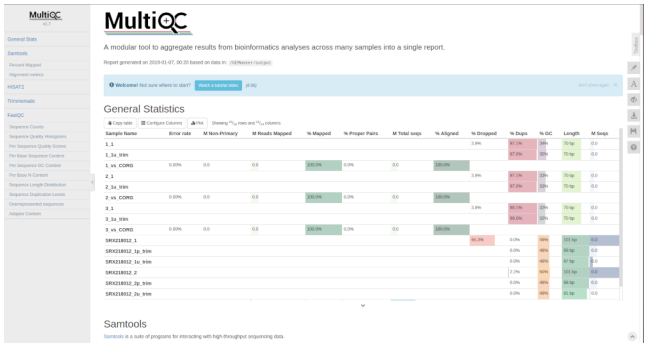
As GEMmaker processes each RNA-seq sample, it creates a directory with all intermediate files and report files that you indicated should be kept. These can include the downloaded SRA or FASTQ files (if remote files are used), trimmed FASTQ files, SAM or BAM files (if Hisat2 is used), FASTQC reports, and raw, FPKM or TPM output files (depending on the tool selected). These files are found withing the `results/Samples` folder of GEMmaker with each sample having its own directory.

2.6.3 The MultiQC Report

The MultiQC tool is used as the last step in the GEMmaker workflow. It will examine the output files from each tool and generate a report that allows you to examine the quality of the results. You can find this report in the `results/reports` folder of GEMmaker. Please refer to the MultiQC website for how to use this report. An example screenshot of a MultiQC report generated by GEMmaker is provided below.

2.6.4 The Failed Runs Report

When SRA runs are requested for download by GEMmaker, sometimes those runs fail. For example, some causes may included that the NCBI SRA servers providing the files could have connectivity issues, samples can be missing metadata, or downloaded samples could be corrupted. GEMmaker will not terminate running in the case of SRA failures, and instead will continue on with samples it can retrieve. Once completed it will create an HTML report listing any failed runs and their cause. You can find this report in the `results/reports` folder of GEMmaker. The report is named `failed_SRA_run_report.html`. An example screen shot shows two examples where runs failed.



GEMmaker Failed Runs Report

If SRA runs files are listed in any of the tables below then these runs were excluded from the GEMmaker results.

SRA Runs with Metadata Failures

SRA Run ID	Reason for Failure
SRR2927686	Metadata was not returned by NCBI for this run.

SRA Runs with Failed Downloads

SRA Run ID	SRA Sample ID	Reason for Failure
No Runs Failed		

SRA Runs with Failed Dumps

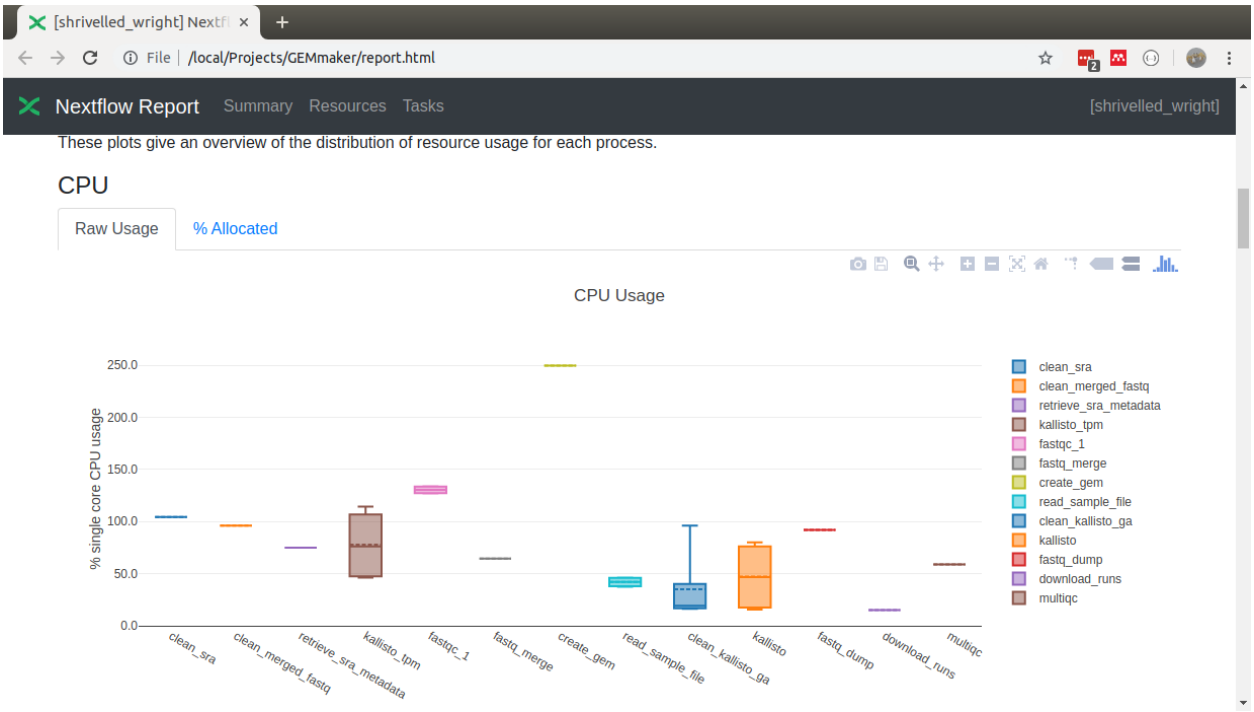
SRA Run ID	SRA Sample ID	Reason for Failure
SRR3461660	SRX1734704	2021-03-26T23:31:52 fastq-dump.2.10.0 err: data corrupt while executing funct == An error occurred during processing. A report was generated into the file '/home/ficklin/ncbi_error_report.txt'. If the problem persists, you may consider sending the file to 'sra-tools@ncbi.nlm.nih.gov' for assistance. ==

2.6.5 Nextflow Reports

GEMmaker will automatically request that Nextflow generate three reports: a summary report, a timeline report and a trace report.

Summary Report

The summary report is found in the `results/reports` folder of GEMmaker and is named `report.html`. You can open this file with a web browser to view it. It contains summary information such as graphs showing the amount of CPU and Memory (RAM) usage, Input/Output (I/O) , job duration and a description of each task. The following screenshot shows the CPU usage section of the report from a run of the GEMmaker example data:

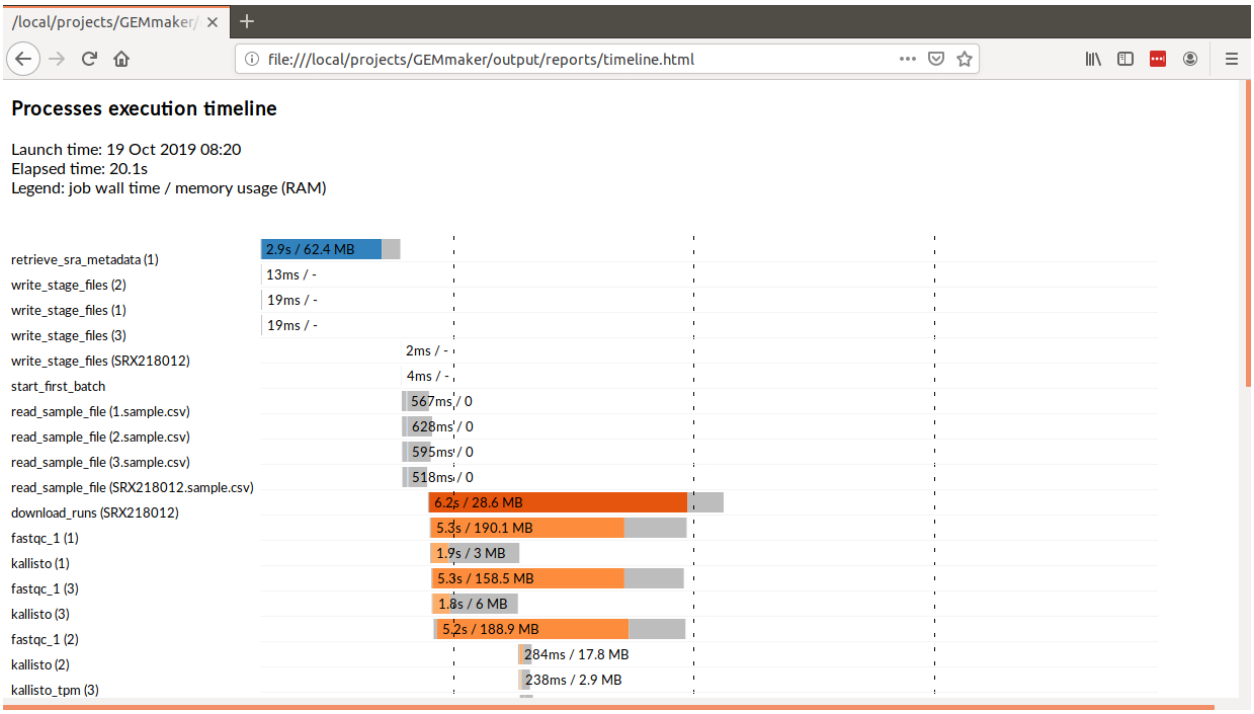


Timeline Report

The timeline report shows the order and time required to execute each of the jobs launched by GEMmaker. It is found in the `results/reports` folder of GEMmaker and is named `timeline.html`. You can open this file with a web browser to view it. The following screenshot shows a section of the report from a run of the GEMmaker example data:

Trace Report

The trace report is named `trace.txt` and is also found in the `results/reports` folder of GEMmaker. It contains the raw data used to create the Summary and Timeline reports.



2.7 What to do with the GEM?

The Gene Expression Matrix (GEM) created by GEMmaker can be used for either Differential Gene Expression (DGE) analyssi or Gene Co-expression Network (GCN) analysis.

2.7.1 DGE Analysis

The raw GEM can be used for differential gene expression (DGE) analysis in edgeR and DESeq2.

2.7.2 Network Analysis

The GEM can be used to construct a gene co-expression network (GCN). The most common tool for construction of GCNs is WGCNA. However, the developers of GEMmaker have also developed a new tool for constructing condition-specific GCNs called KINC. It is a high-performance application that can construct networks using Pearson or Spearman for pairwise correlation, as well as Gaussian mixture models (GMMs) for pairwise clustering. KINC is a Qt/ACE application that is capable of running on CPUs and GPUs, which means that it can scale to larger workloads.

Note: Prior to network construction it is recommended to normalized (such as with quantile normalization) and log-transform the GEM. GEMmaker does not provide this functionality.

2.8 Troubleshooting

2.8.1 WARNING about "sticked on revision"

If you encounter a similar warning message when running GEMmaker:

```
Project `systemsgenetics/gemmaker` currently is sticked on revision: dev -- you need␣
↪to specify explicitly a revision with the option `-r` to use it
```

This means you have multiple versions of GEMmaker cached by Nextflow and it isn't sure which one to use. You must specify the `-r <version>` parameter when running Nextflow. For example, to use the `dev` version of GEMmaker you would provdie `-r dev`.

2.8.2 ERROR : Unknown image format/type

If you encounter the following error:

```
ERROR  : Unknown image format/type: <directory to a singularity image file>
ABORT  : Retval = 255
```

Most likely, Singularity encountered some problem when retrieving and building the software images that GEMmaker uses. The solution is to just resume the GEMmaker workflow and the problem will most likely resolve itself.

2.8.3 ERROR : No valid /bin/sh in container

If you encounter the following error message:

```
ERROR  : No valid /bin/sh in container
ABORT  : Retval = 255
```

Then this can be caused by any of the following problems:

- Nextflow will automatically download and build the singularity images for you. If your *umask* is not set to create files that are readable and executable then you can get this error. Setting a umask such as, `umask u=rwx,g=rx,o=rx`, prior to running Nextflow will ensure the images are readable and executable.

2.8.4 ERROR : Failed to set loop flags on loop device: Resource temporarily unavailable

On local machines, you may encounter the following Singularity error:

```
ERROR  : Failed to set loop flags on loop device: Resource temporarily unavailable
ABORT  : Retval = 255
```

This is caused by Singularity attempting to access the same image with more than one thread. The first process to access the image will lock it until it is read into memory. This can be safely ignored, as GEMmaker will automatically retry the process.

2.8.5 Exception in thread "main" java.lang.OutOfMemoryError: Java heap space

This error can occur when GEMmaker runs the FastQC process and there is insufficient RAM available. If running on a local machine you can adjust the `--max_cpus` argument to decrease the number of concurrent jobs that run at a given time. If running on an HPC system, you can increase the amount of memory requested for the job by altering the setting in the `--max_memory` argument. See the nf-core Pipeline Configuration page for more in-depth details for providing more custom configuration files.

2.8.6 Why is it taking so long to pull a docker/singularity image?

This is dependent directly on your internet speed. The first time GEMmaker is run, it must download the Docker image it needs to run. This means it may take a little while longer to run the first time it is run on your machine.

2.8.7 GEMmaker seems hung and does not complete

If GEMmaker seems to have processed all the samples provided, but does not move on to the `create_gem` step it may be hung. Sometimes this can occur if you have tried to run GEMmaker multiple times but changed the list of samples between runs. The best solution is to only run GEMmaker with one set of samples in a single directory, and to use a different directory for other samples.

2.8.8 SLURM: exceeded memory limit

If you are launching GEMmaker on an HPC system with the SLURM scheduler you can sometimes get an *exceeded memory limit* similar to the following:

```
slurmstepd: error: Job 12254566 exceeded memory limit (7871840 > 6553600), being␣
↪killed
```

If you have a lot of samples, Nextflow may need more memory. Increasing the amount of memory in your SLURM submission script will correct this problem. Remember to restart GEMmaker with the `-resume` flag to have it continue where it left off.

2.9 Get Help or Suggest Improvements

If you have questions, comments, suggestions for improvement or require help with setup and execution of GEMmaker please consider posting to the GEMmaker issue board on Github.