13-Apul-sRNAseq-ShortStack

Sam White

2023-11-03

Contents

1 Set R variables **2**

2 Create a Bash variables file **2**

3 Load ShortStack conda environment **3**

4 Download miRBase mature miRNA FastA **4**

5 Run ShortStack **4**

5.1 Modify genome filename for ShortStack compatability 4

5.2 Excecute ShortStack command 5

5.3 Check runtime 5

6 Results **6**

6.1 ShortStack synopsis 6

6.2 Inspect `Results.txt` 6

Citations **9**

Use ShortStack (Axtell 2013; Shahid and Axtell 2014; Johnson et al. 2016)to perform alignment of sRNAseq data and annotation of sRNA-producing genes.

The *A.millepora* genome will be used as the reference genome for *A.pulchra*, as *A.pulchra* does not currently have a sequenced genome and *A.millepora* had highest alignment rates for standard RNAseq data compared to other published genomes tested.

Inputs:

- Requires trimmed sRNAseq files generated by 08-Apul-sRNAseq-trimming.Rmd
  - Filenames formatted: `*flexbar_trim.25bp*.gz`
- *A.millepora* genome FastA. See 12-Apul-sRNAseq-MirMachine.Rmd for download info if needed.

Outputs:

- See ShortStack outputs documentation for full list and detailed descriptions.

Software requirements:

- Utilizes a ShortStack Conda/Mamba environment, per the installation instructions.

Replace with name of your ShortStack environment and the path to the corresponding conda installation (find this *after* you've activated the environment).

E.g.

```
# Activate environment
conda activate ShortStack4_env

# Find conda path
which conda
```

---

1 Set R variables

```
shortstack_conda_env_name <- c("ShortStack4_env")
shortstack_cond_path <- c("/home/sam/programs/mambaforge/condabin/conda")
```

2 Create a Bash variables file

This allows usage of Bash variables across R Markdown chunks.

```
{
echo "#### Assign Variables ####"
echo ""

echo "# Trimmed FastQ naming pattern"
echo "export trimmed_fastqs_pattern='*flexbar_trim.25bp*.fastq.gz'"

echo "# Data directories"
echo 'export deep_dive_dir=/home/shared/8TB_HDD_01/sam/gitrepos/deep-dive'
echo 'export deep_dive_data_dir="${deep_dive_dir}/data"'
echo 'export output_dir_top=${deep_dive_dir}/D-Apul/output/13-Apul-sRNAseq-ShortStack'
echo 'export trimmed_fastqs_dir="${deep_dive_dir}/D-Apul/output/08-Apul-sRNAseq-trimming/trimmed-reads"
echo ""

echo "# Input/Output files"
echo 'export genome_fasta_dir=${deep_dive_dir}/D-Apul/data/Amil/ncbi_dataset/data/GCF_013753865.1'
echo 'export genome_fasta_name="GCF_013753865.1_Amil_v2.1_genomic.fna"'
echo 'export shortstack_genome_fasta_name="GCF_013753865.1_Amil_v2.1_genomic.fa"'
```

```
echo 'export mirbase_mature_fasta=mature.fa'
echo 'export mirbase_mature_fasta_version=mirbase-mature-v22.1.fa'
echo 'export genome_fasta="${genome_fasta_dir}/${shortstack_genome_fasta_name}"'
echo ""

echo "# External data URLs"
echo 'export mirbase_fasta_url="https://mirbase.org/download_version_files/22.1/"'
echo ""

echo "# Set number of CPUs to use"
echo 'export threads=46'
echo ""

echo "# Initialize arrays"
echo 'export trimmed_fastqs_array=()'


} > .bashvars

cat .bashvars
```

```
#### Assign Variables ####

# Trimmed FastQ naming pattern
export trimmed_fastqs_pattern='*flexbar_trim.25bp*.fastq.gz'
# Data directories
export deep_dive_dir=/home/shared/8TB_HDD_01/sam/gitrepos/deep-dive
export deep_dive_data_dir="${deep_dive_dir}/data"
export output_dir_top=${deep_dive_dir}/D-Apul/output/13-Apul-sRNAseq-ShortStack
export trimmed_fastqs_dir="${deep_dive_dir}/D-Apul/output/08-Apul-sRNAseq-trimming/trimmed-reads"

# Input/Output files
export genome_fasta_dir=${deep_dive_dir}/D-Apul/data/Amil/ncbi_dataset/data/GCF_013753865.1
export genome_fasta_name="GCF_013753865.1_Amil_v2.1_genomic.fna"
export shortstack_genome_fasta_name="GCF_013753865.1_Amil_v2.1_genomic.fa"
export mirbase_mature_fasta=mature.fa
export mirbase_mature_fasta_version=mirbase-mature-v22.1.fa
export genome_fasta="${genome_fasta_dir}/${shortstack_genome_fasta_name}"

# External data URLs
export mirbase_fasta_url="https://mirbase.org/download_version_files/22.1/"

# Set number of CPUs to use
export threads=46

# Initialize arrays
export trimmed_fastqs_array=()
```

3 Load ShortStack conda environment

If this is successful, the first line of output should show that the Python being used is the one in your [ShortStack](https://github.com/MikeAxtell/ShortStack conda environment path.

E.g.

```
python:         /home/sam/programs/mambaforge/envs/mirmachine_env/bin/python
```

```
use_condaenv(condaenv = shortstack_conda_env_name, conda = shortstack_cond_path)

# Check successful env loading
py_config()
```

```
python:         /home/sam/programs/mambaforge/envs/ShortStack4_env/bin/python
libpython:      /home/sam/programs/mambaforge/envs/ShortStack4_env/lib/libpython3.10.so
pythonhome:     /home/sam/programs/mambaforge/envs/ShortStack4_env:/home/sam/programs/mambaforge/envs/Sh
version:        3.10.13 | packaged by conda-forge | (main, Oct 26 2023, 18:07:37) [GCC 12.3.0]
numpy:          /home/sam/programs/mambaforge/envs/ShortStack4_env/lib/python3.10/site-packages/numpy
numpy_version:  1.26.0

NOTE: Python version was forced by use_python() function
```

4 Download miRBase mature miRNA FastA

```
# Load bash variables into memory
source .bashvars

wget \
--directory-prefix ${deep_dive_data_dir} \
--recursive \
--no-check-certificate \
--continue \
--no-host-directories \
--no-directories \
--no-parent \
--quiet \
--execute robots=off \
 ${mirbase_fasta_url}/${mirbase_mature_fasta}

# Rename to indicate miRBase FastA version
mv ${deep_dive_data_dir}/${mirbase_mature_fasta} ${deep_dive_data_dir}/${mirbase_mature_fasta_version}

ls -lh "${deep_dive_data_dir}"
```

```
total 3.7M
-rw-r--r-- 1 sam sam 3.7M Nov  6 12:40 mirbase-mature-v22.1.fa
```

5 Run ShortStack

5.1 Modify genome filename for ShortStack compatability

```
# Load bash variables into memory
source .bashvars

# Copy genome FastA to ShortStack-compatible filename (ending with .fa)
cp ${genome_fasta_dir}/${genome_fasta_name} ${genome_fasta_dir}/${shortstack_genome_fasta_name}

# Confirm
ls -lh ${genome_fasta_dir}/${shortstack_genome_fasta_name}
```

```
-rw-r--r-- 1 sam sam 460M Nov  6 12:40 /home/shared/8TB_HDD_01/sam/gitrepos/deep-dive/D-Apul/data/Amil/r
```

5.2 Excecute ShortStack command

Uses the `--dn_mirna` option to identify miRNAs in the genome, without relying on the `--known_miRNAs`.

This part of the code redirects the output of `time` to the end of `shortstack.log` file.

- `; } \ 2>> ${output_dir_top}/shortstack.log`

```
# Load bash variables into memory
source .bashvars

# Create array of trimmed FastQs
trimmed_fastqs_array=(${trimmed_fastqs_dir}/${trimmed_fastqs_pattern})


# Pass array contents to new variable as space-delimited list
trimmed_fastqs_list=$(echo "${trimmed_fastqs_array[*]}")


###### Run ShortStack ######
{ time \
ShortStack \
--genomefile "${genome_fasta}" \
--readfile ${trimmed_fastqs_list} \
--known_miRNAs ${deep_dive_data_dir}/${mirbase_mature_fasta_version} \
--dn_mirna \
--threads ${threads} \
--outdir ${output_dir_top}/ShortStack_out \
&> ${output_dir_top}/shortstack.log ; } \
2>> ${output_dir_top}/shortstack.log
```

5.3 Check runtime

```
# Load bash variables into memory
source .bashvars

tail -n 3 ${output_dir_top}/shortstack.log \
| grep "real" \
| awk '{print "ShortStack runtime:" "\t" $2}'
```

```
ShortStack runtime: 142m36.973s
```

6 Results

6.1 ShortStack synopsis

```
# Load bash variables into memory
source .bashvars

tail -n 20 ${output_dir_top}/shortstack.log
```

```
Screening of possible de novo microRNAs

No microRNA loci were found!

Writing final files

Non-MIRNA loci by DicerCall:
N 18676
22 45
23 36
21 10
24 5

Mon 06 Nov 2023 11:50:36 -0800 PST
Run Completed!

real    142m36.973s
user    2955m32.601s
sys 1100m59.754s
```

ShortStack didn't identify *any* miRNAs.

6.2 Inspect `Results.txt`

```
# Load bash variables into memory
source .bashvars

head ${output_dir_top}/ShortStack_out/Results.txt

echo ""
echo "------------------------------------------------------------"
echo ""

echo "Nummber of potential loci:"
awk '(NR>1)' ${output_dir_top}/ShortStack_out/Results.txt | wc -l
```

```
Locus   Name    Chrom   Start   End Length  Reads   UniqueReads FracTop Strand  MajorRNA    MajorRNARea
NC_058066.1:161118-161784   Cluster_1   NC_058066.1 161118  161784  667 1363    392 0.6573734409391049
NC_058066.1:171557-171958   Cluster_2   NC_058066.1 171557  171958  402 366 108 0.5683060109289617  .
NC_058066.1:204734-205143   Cluster_3   NC_058066.1 204734  205143  410 525 180 0.6342857142857142  .
NC_058066.1:205754-206966   Cluster_4   NC_058066.1 205754  206966  1213    3040    509 0.37697368421051
NC_058066.1:210858-211343   Cluster_5   NC_058066.1 210858  211343  486 1422    317 0.2883263009845288
NC_058066.1:243461-243885   Cluster_6   NC_058066.1 243461  243885  425 446 46  0.8497757847533632  +
NC_058066.1:349656-351296   Cluster_7   NC_058066.1 349656  351296  1641    5821    1435    0.515718948
NC_058066.1:351494-353435   Cluster_8   NC_058066.1 351494  353435  1942    17924   2140    0.571356839
NC_058066.1:776275-776775   Cluster_9   NC_058066.1 776275  776775  501 2260    216 0.8433628318584071

------------------------------------------------------------

Nummber of potential loci:
18772
```

Column 20 of the `Results.txt` file identifies if a cluster is a miRNA or not (`Y` or `N`).

```
# Load bash variables into memory
source .bashvars

echo "Number of loci characterized as miRNA:"
awk '$20=="Y" {print $0}' ${output_dir_top}/ShortStack_out/Results.txt \
| wc -l
echo ""

echo "------------------------------------------------------------"

echo ""
echo "Number of loci _not_ characterized as miRNA:"
awk '$20=="N" {print $0}' ${output_dir_top}/ShortStack_out/Results.txt \
| wc -l
```

```
Number of loci characterized as miRNA:
0

------------------------------------------------------------

Number of loci _not_ characterized as miRNA:
18772
```

Column 21 of the `Results.txt` file identifies if a cluster aligned to a known miRNA (miRBase) or not (`Y` or `NA`).

Since there are no miRNAs, the following code will *not* print any output.

The `echo` command after the `awk` command is simply there to prove that the chunk executed.

```
# Load bash variables into memory
source .bashvars

echo "Number of loci matching miRBase miRNAs:"
awk '$21!="NA" {print $0}' ${output_dir_top}/ShortStack_out/Results.txt \
| wc -l
```

```
echo ""

echo "------------------------------------------------------------"

echo ""
echo "Number of loci _not_ matching miRBase miRNAs:"
awk '$21=="NA" {print $0}' ${output_dir_top}/ShortStack_out/Results.txt \
| wc -l
```

```
Number of loci matching miRBase miRNAs:
46

------------------------------------------------------------

Number of loci _not_ matching miRBase miRNAs:
18727
```

Although there are loci with matches to miRBase miRNAs, ShortStack did *not* annotated these clusters as miRNAs likely because they do not *also* match secondary structure criteria.

6.2.1 Directory tree of all ShortStack outputs

Many of these are large (by GitHub standards) BAM files, so will not be added to the repo.

Additionally, it's unlikely we'll utilize most of the other files (bigwig) generated by ShortStack.

```
# Load bash variables into memory
source .bashvars

tree -h ${output_dir_top}/
```

```
/home/shared/8TB_HDD_01/sam/gitrepos/deep-dive/D-Apul/output/13-Apul-sRNAseq-ShortStack/
   [ 28K]  shortstack.log
   [ 36K]  ShortStack_out
       [ 47K]  alignment_details.tsv
       [1.4M]  Counts.txt
       [ 87K]  known_miRNAs.gff3
       [1.8M]  known_miRNAs_unaligned.fasta
       [9.8M]  merged_alignments_21_m.bw
       [ 10M]  merged_alignments_21_p.bw
       [9.5M]  merged_alignments_22_m.bw
       [9.9M]  merged_alignments_22_p.bw
       [ 19M]  merged_alignments_23-24_m.bw
       [ 20M]  merged_alignments_23-24_p.bw
       [2.7G]  merged_alignments.bam
       [227K]  merged_alignments.bam.csi
       [123M]  merged_alignments_other_m.bw
       [126M]  merged_alignments_other_p.bw
       [ 48M]  merged_alignments_sRNA-ACR-140-S1-TP2.flexbar_trim.25bp_1.bw
       [ 48M]  merged_alignments_sRNA-ACR-140-S1-TP2.flexbar_trim.25bp_2.bw
       [ 52M]  merged_alignments_sRNA-ACR-145-S1-TP2.flexbar_trim.25bp_1.bw
       [ 52M]  merged_alignments_sRNA-ACR-145-S1-TP2.flexbar_trim.25bp_2.bw
```

```
[ 50M]  merged_alignments_sRNA-ACR-150-S1-TP2.flexbar_trim.25bp_1.bw
[ 49M]  merged_alignments_sRNA-ACR-150-S1-TP2.flexbar_trim.25bp_2.bw
[ 43M]  merged_alignments_sRNA-ACR-173-S1-TP2.flexbar_trim.25bp_1.bw
[ 43M]  merged_alignments_sRNA-ACR-173-S1-TP2.flexbar_trim.25bp_2.bw
[ 44M]  merged_alignments_sRNA-ACR-178-S1-TP2.flexbar_trim.25bp_1.bw
[ 43M]  merged_alignments_sRNA-ACR-178-S1-TP2.flexbar_trim.25bp_2.bw
[1.9M]  Results.gff3
[2.8M]  Results.txt
[246M]  sRNA-ACR-140-S1-TP2.flexbar_trim.25bp_1.bam
[224K]  sRNA-ACR-140-S1-TP2.flexbar_trim.25bp_1.bam.csi
[266M]  sRNA-ACR-140-S1-TP2.flexbar_trim.25bp_2.bam
[229K]  sRNA-ACR-140-S1-TP2.flexbar_trim.25bp_2.bam.csi
[279M]  sRNA-ACR-145-S1-TP2.flexbar_trim.25bp_1.bam
[228K]  sRNA-ACR-145-S1-TP2.flexbar_trim.25bp_1.bam.csi
[298M]  sRNA-ACR-145-S1-TP2.flexbar_trim.25bp_2.bam
[230K]  sRNA-ACR-145-S1-TP2.flexbar_trim.25bp_2.bam.csi
[297M]  sRNA-ACR-150-S1-TP2.flexbar_trim.25bp_1.bam
[228K]  sRNA-ACR-150-S1-TP2.flexbar_trim.25bp_1.bam.csi
[316M]  sRNA-ACR-150-S1-TP2.flexbar_trim.25bp_2.bam
[229K]  sRNA-ACR-150-S1-TP2.flexbar_trim.25bp_2.bam.csi
[255M]  sRNA-ACR-173-S1-TP2.flexbar_trim.25bp_1.bam
[229K]  sRNA-ACR-173-S1-TP2.flexbar_trim.25bp_1.bam.csi
[275M]  sRNA-ACR-173-S1-TP2.flexbar_trim.25bp_2.bam
[230K]  sRNA-ACR-173-S1-TP2.flexbar_trim.25bp_2.bam.csi
[234M]  sRNA-ACR-178-S1-TP2.flexbar_trim.25bp_1.bam
[229K]  sRNA-ACR-178-S1-TP2.flexbar_trim.25bp_1.bam.csi
[248M]  sRNA-ACR-178-S1-TP2.flexbar_trim.25bp_2.bam
[230K]  sRNA-ACR-178-S1-TP2.flexbar_trim.25bp_2.bam.csi

1 directory, 47 files
```

Citations

Axtell, Michael J. 2013. "ShortStack: Comprehensive Annotation and Quantification of Small RNA Genes." *RNA* 19 (6): 740–51. https://doi.org/10.1261/rna.035279.112.

Johnson, Nathan R, Jonathan M Yeoh, Ceyda Coruh, and Michael J Axtell. 2016. "Improved Placement of Multi-Mapping Small RNAs." *G3 Genes|Genomes|Genetics* 6 (7): 2103–11. https://doi.org/10.1534/g3.116.030452.

Shahid, Saima, and Michael J. Axtell. 2014. "Identification and Annotation of Small RNA Genes Using ShortStack." *Methods* 67 (1): 20–27. https://doi.org/10.1016/j.ymeth.2013.10.004.