Smartseq3 Template Workflow

Project: Benjamin's Lab Book

Author: Benjamin Clark

Entry Created On: 01 Nov 2022 15:33:21 UTC
Entry Last Modified: 29 Jul 2024 16:40:49 UTC
Export Generated On: 29 Jul 2024 16:41:07 UTC

MARDI 01/11/2022

Index mm10 genome

~/scratch/burst/ss3_2/mm10/scripts/genome_index.sh

```
#!/bin/bash
#SBATCH --time=02:00:00
#SBATCH --cpus-per-task=4
#SBATCH --mem=40G
#SBATCH --job-name=STAR_index


module load star/2.7.9a

STAR --runThreadN 4 \
 --runMode genomeGenerate \
 --genomeDir /home/clarkb/scratch/burst/ss3_2/mm10  \
 --genomeFastaFiles /home/clarkb/scratch/burst/ss3_2/mm10/mm10.fa \
 --sjdbGTFfile /home/clarkb/scratch/burst/ss3_2/mm10/mm10.ensGene.gtf  \
 --sjdbOverhang 100
```

Notes:
-sjdbOverhang might need to be 150
-Judging by the SS3 github, they might have generated genome indices without a gtf file.

Create N-masked BL6/CAST genome

~/scratch/burst/ss3_2/CAST/make_masked_genome.sh

```
#!/bin/bash
#SBATCH --job-name=n_mask
#SBATCH --ntasks=1
#SBATCH --mem=40G
#SBATCH --time=00:30:00

perl /home/clarkb/projects/def-robertf/clarkb/SNPsplit/SNPsplit_genome_preparation \
 --vcf_file ~/projects/def-robertf/clarkb/Smart-seq3/allele_level_expression/CAST.SNPs.validated.vcf.gz \
 --reference_genome /home/clarkb/scratch/burst/ss3_2/mm10 --strain CAST_EiJ
```

Merge chromosome files into one:

~/scratch/burst/ss3_2/CAST/CAST_EiJ_N-masked

```
cat *.fa >> CAST_N-masked.fasta
```

Index N-masked genome

~/scratch/burst/ss3_2/CAST/index_masked.sh

```
#!/bin/bash
#SBATCH --time=02:00:00
#SBATCH --cpus-per-task=4
#SBATCH --mem=40G
#SBATCH --job-name=STAR_index



module load star/2.7.9a

STAR --runThreadN 4 \
 --runMode genomeGenerate \
 --genomeDir /home/clarkb/scratch/burst/ss3_2/CAST/index  \
 --genomeFastaFiles /home/clarkb/scratch/burst/ss3_2/CAST/CAST_EiJ_N-masked/CAST_N-masked.fasta \
 --sjdbGTFfile /home/clarkb/scratch/burst/ss3_2/mm10/mm10.ensGene.gtf  \
 --sjdbOverhang 100
```

zUMIs

~/scratch/burst/ss3_2/scripts/zUMIs_master_2.sh

```
#!/bin/bash
#SBATCH --job-name=zUMI_run_2021_2h1_%j
#SBATCH --mem=200G
#SBATCH --time=48:00:00
#SBATCH --cpus-per-task=30
#SBATCH --mail-user=benjamin_r.clark@live.com
#SBATCH --mail-type=END,FAIL

module load StdEnv/2020
module load star
module load gcc/9.3.0
module load hdf5/1.12.1
module load r/4.0.2
module load python
source ~/projects/def-robertf/clarkb/pysam-env/env/bin/activate

/home/clarkb/projects/def-robertf/clarkb/zUMIs/zUMIs.sh -y
/home/clarkb/scratch/ss3_2/scripts/zUMIs_2021/zUMIs_master2.yaml
```

yaml file

~/scratch/burst/ss3_2/scripts/ zUMIs_master_2.yaml

```
project: zUMIs_2021_star2.5.4b_FULL_h1
sequence_files:
  file1:
    name: /home/clarkb/scratch/ss3_2/reads/plate1.read1.fq.gz
    base_definition:
    - cDNA(23-100)
    - UMI(12-19)
    find_pattern: ATTGCGCAATG
  file2:
    name: /home/clarkb/scratch/ss3_2/reads/plate1.read2.fq.gz
    base_definition:
    - cDNA(1-100)
    - BC(103-108,111-118)
reference:
  STAR_index: /home/clarkb/scratch/ss3_2/CAST/N_mask_index_no_gtf
  GTF_file: /home/clarkb/scratch/ss3_2/mm10/mm10.ensGene.gtf
  additional_STAR_params: --limitSjdbInsertNsj 2000000  --clip3pAdapterSeq
    CTGTCTCTTATACACATCT
  additional_files: ~
out_dir: /home/clarkb/scratch/ss3_2/zUMIs/fullh1
num_threads: 30
mem_limit: 50
filter_cutoffs:
  BC_filter:
    num_bases: 1
    phred: 20
  UMI_filter:
    num_bases: 1
    phred: 20
barcodes:
  barcode_num: null
  barcode_file: /home/clarkb/scratch/ss3_2/reads/barcodes.txt
  automatic: no
  BarcodeBinning: 0
  nReadsperCell: 100
counting_opts:
  introns: yes
  downsampling: '0'
  strand: 0
  Ham_Dist: 1
  velocyto: no
  primaryHit: yes
  twoPass: no
make_stats: yes
```

```
which_Stage: Filtering
Rscript_exec: Rscript
STAR_exec: /home/clarkb/projects/def-robertf/clarkb/STAR-2.7.3a/source/STAR
pigz_exec: pigz
samtools_exec: /home/clarkb/projects/def-robertf/clarkb/samtools/bin/samtools
zUMIs_directory: /lustre06/project/6002165/clarkb/zUMIs
read_layout: PE
```

Notes:

- The main takeaway here is that the memory limit parameter in the yaml doesn't seem to do much. When testing mem consumnption it actually takes double the amount (e.i here I gave it 50, which means it takes over 100G). Make sure the slurm job has more than double the memory requirement. This job took about 16hrs so 50G is fine.

Allele Assignment

~/scratch/burst/ss3_2/scripts/allele_assign.sh

```
#!/bin/bash
#SBATCH --job-name=allele_assign      # Job name
#SBATCH --mail-type=END,FAIL          # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=benjamin_r.clark@live.com           # Where to send mail
#SBATCH --ntasks=1                    # Run on a single CPU
#SBATCH --mem=24gb                    # Job memory request
#SBATCH --time=03:00:00               # Time limit hrs:min:sec
#SBATCH --output=allele_assign_j.log # Standard output and error log
#SBATCH --cpus-per-task=1             # Multi-threading

module load r/4.1.2

Rscript ~/projects/def-robertf/clarkb/Smart-seq3/allele_level_expression/get_variant_overlap_CAST.R --yaml
~/scratch/burst/ss3_2/scripts/zUMIs_master_2.run.yaml \
       --vcf  ~/projects/def-robertf/clarkb/Smart-seq3/allele_level_expression/CAST.SNPs.validated.vcf.gz
```

Notes:

- Requires to use the .run.yaml file for the extra line denoting read type.

Parameter Estimation

Notes:

- Currently I'm using nextflow which just chains two scripts together: txburstML.py and txburstPL.py. I have a bunch of other processes here but I'm only using the three listed in the workflow block. Essentially all thats happening is that I'm converting outputs from the above script into csv format and piping sequencially into ML and PL scripts and outputing them into a directory named output.
- The resume option in the bash script doesn't work yet for some reason.

- It may be that nextflow generates a new instance of the variable whenever I call it however this seems to say I'm doing it properly https://nextflow-io.github.io/patterns/channel-duplication/ . I might need to test it further and run the scripts by themselves again.

~/scratch/burst/ss3_2/nextflow/tx_pipe.sh

```
#SBATCH --mail-user=benjamin_r.clark@live.com           # Where to send mail
#SBATCH --ntasks=1                    # Run on a single CPU
#SBATCH --mem=24gb                     # Job memory request
#SBATCH --time=07:00:00               # Time limit hrs:min:sec
#SBATCH --output=txburst_pipe_%j.log   # Standard output and error log
#SBATCH --cpus-per-task=30             # Multi-threading

module load scipy-stack
module load nextflow
source /home/clarkb/projects/def-robertf/clarkb/txburst/env/bin/activate
resume=false

while getopts 'r' flag
do
    case "${flag}" in
        r) resume=true ;;
        *) echo 'Unknown flag' >&2
           exit 1
    esac
done

if "$resume"; then
        echo 'resuming...'
        nextflow run main.nf -resume
else
        nextflow run main.nf
fi
```

~/scratch/burst/ss3_2/nextflow/main.nf

```
#!/usr/bin/env nextflow

//Parameters
params.outdir = "output"
params.counts = "/home/clarkb/scratch/burst/ss3_2/zUMIs/gelcut/zUMIs_output/allelic/*.txt"
//params.bam = "/home/clarkb/scratch/burst/ss3_2/stitcher-out/stitched_transcripts.bam"
//params.yaml = "/home/clarkb/scratch/burst/ss3_2/scripts/zUMIs_master_sam.run.yaml"


process stitch {
        cpus = 10
        memory = 24.GB
        time = "1 hour"
        input:
                path ss3_bam
                path gtf
                path isoform_json

        output:
                path "stitched_transcripts.bam"
        """
        stitcher.py -i $ss3_bam -o "stitched_transcripts.bam" -g $gtf --isoform $isoform_json -t 10
        """
}

process getDir {
        input:
                path yaml
        output:
                stdout
        """
        #!/usr/bin/env python
        import yaml
        with open("$yaml") as file:
                input = yaml.safe_load(file)
        print(input["out_dir"] + "/zUMIs_output/allelic/*.txt")
        """

}

process tab2CSV {
        input:
                path tab
        output:
```

```
                path "${tab.baseName}.csv"
        """
        cat $tab | tr -s "\\t" "," > "${tab.baseName}.csv"
        """
}

process alleleLevelExpression {
        cpus = 5
        memory = 24.GB
        input:
                path yaml
                val outdir
        output:
                path "${outdir}*.txt"

        """
        module load r/4.1.2;
        get_variant_overlap_CAST.R --yaml $yaml --vcf $baseDir/CAST.SNPs.validated.vcf.gz
        """
}

process txburstML {
        publishDir "$params.outdir/ML"
        input:
                path csv
        output:
                path "*ML.pkl"

        """
        txburstML.py --njobs 20  $csv
        """
}

process txburstPL {
        publishDir "$params.outdir/PL"
        input:
                path csv
                path ML
        output:
                path "*PL.pkl"
        """
        txburstPL.py --njobs 20 --file $csv --MLFile $ML
        """
}
```

```
workflow {

        counts_in = Channel.fromPath(params.counts)

        tab2CSV(counts_in)

        csv = tab2CSV.out

        txburstML(csv)

        txburstPL(csv, txburstML.out)

}
```

DIMANCHE 18/06/2023