# Align RNAseq data using STAR

## Generate STAR genome index

TBD

For now we will just use genome generated by mathemagician Tracy. It is located in:

/srv/gs1/projects/montgomery/tnance/genomes/STAR

More information:

* sjdbOverhang clarification: [link](https://groups.google.com/forum/#!topic/rna-star/h9oh10UlvhI)

## STAR 1-pass

* Use **run\_star.sh**

Currently run\_star.sh only supports paired-end alignment. I will add single-end alignment in the future.

# Shell Scripting

## Find

The find command in one of the most versatile tools on Linux.

To find files larger than 1G under home directory.

*find ~ -type f –size +5G –exec ls –lh {} \;*

## Variables $

substitution

${var/<string\_to\_be\_replaced>/<replacement>}

To know more about variable manipulation, click [here](http://stackoverflow.com/questions/2188199/how-to-use-double-or-single-bracket-parentheses-curly-braces).

## Disk Quota

Use this command to check disk quota:

*ssh scg3-0-1 /srv/gs1/software/scg\_quota -p montgomery*

## Disk Usage

To check the size of a directory:

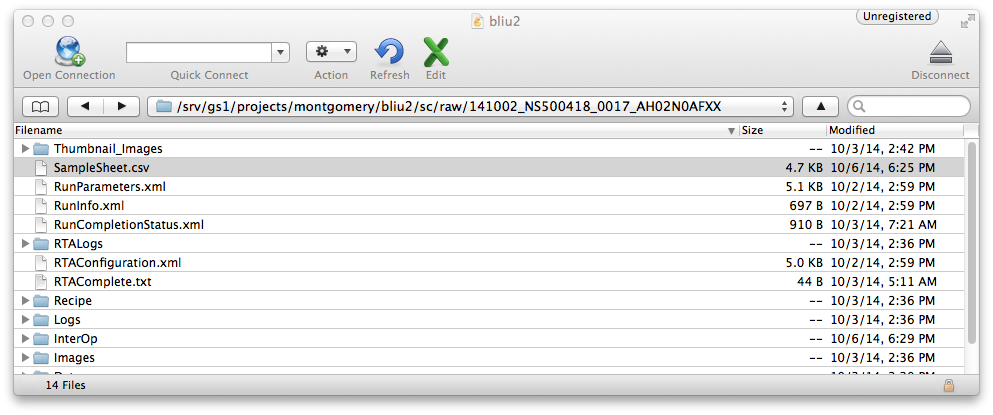
du –sh <directory>

To check the size of a directory and its sub-directories:

du –h <directory>

# Convert NextSeq basecall to fastq

First, put a “SampleSheet.csv” in the top directory. E.g.



Sample sheet templates:

* **NextSeq500\_SampleSheet\_DualIndex.csv:** template for dual index
* **NextSeq\_SampleSheet\_SingleIndex.csv:** template for single index

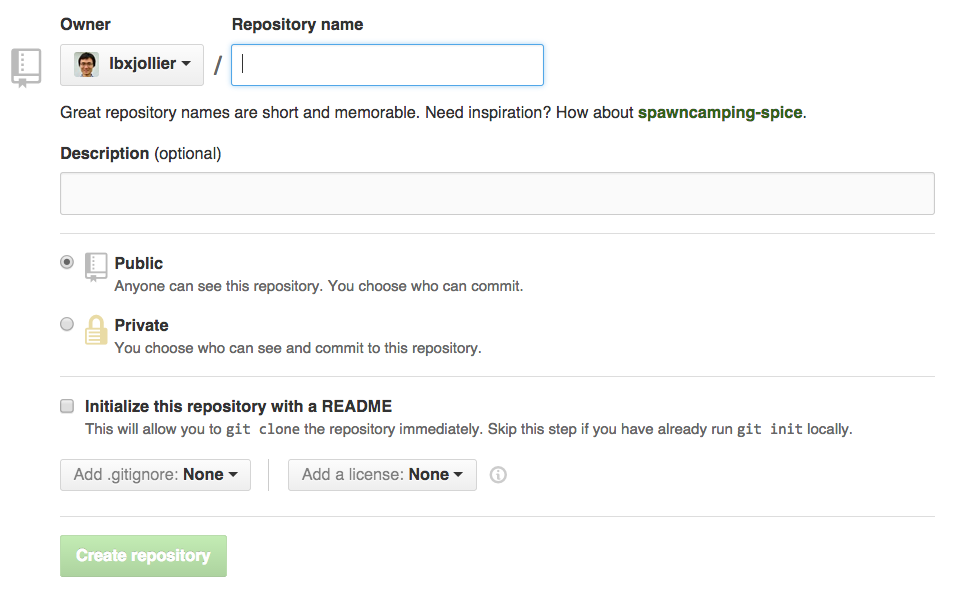
To convert NextSeq bcl files to fastq format, use **run\_bcl2fastq2.sh**

# Git

## Basic operations

To be a successful bioinformatician, the first thing you need to do is to create a git repository to store your scripts.

First, let us create a git remote repo on github. If you don’t have a github account, you should really create one on <https://www.github.com>. After you have a personal account, clikc on the ‘plus’ button on the topright corner to create a new repository. You will see the following page.



Enter repository name, and click on ‘Create repository’.

Now clone this repository to your local machine:

*$ git clone git@github.com:lbxjollier/bioinformatics\_toolbox.git*

# If you encounter the following error, follow this [link](https://help.github.com/articles/generating-ssh-keys/) to generate a public ssh key.

*Permission denied (publickey).*

*fatal: The remote end hung up unexpectedly*

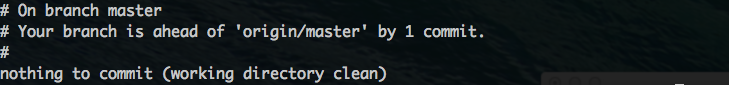
This command will create a directory named “bioinformatics\_toolbox”. Add some files into this directory, and type:

Let usdo our first commit:

*$ git commit –m ‘initial commit’*

This command tells git to keep a permanent copy of the current files. To check whether a desired files have been committed, use:

*$ git status*

You will see: 

This reflects that we have committed the newly added files locally, but have not ‘pushed’ the local file to our remove repo on github. To push, use:

*$ git push origin master*

To exclude a file, first create a .gitignore file:

$ touch .gitignore

Then add files you don’t want to include. For example:

$ echo '.nfs0000000006ee455100000edb' >> .gitignore

# FastQ Quality Control:

First, let us check the quality of the sequencing result using fastQC.

Let’s say your file is located in <input\_directory>. To make our life easier, let’s create an atlas.

*in=<input\_directory>*

*out=<output\_directory>*

To run fastQC, first let us load the module (I’m on a cluster, but you can also download fastQC from [here](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/).

*module load fastqc/0.11.2*

To run, type:

*fastqc $in/<input\_file> --outdir=$out*

Now we will have our results in the output directory. The document from [here](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/) will help you understand the result.

## Trim adapters

If you see adaptor contamination especially towards the end. This is because if you template is not long enough, the adapter sequences will be sequences right after the template ends. If this is the case, we will need to cut the adatpters with cutadapt.

If you don’t have cutadapt, install using:

*pip install cutadapt*

To trim illumina single-ended indexed libraries:

*cutadapt -a GATCGGAAGAGCACACGTCTGAACTCCAGTCACNNNNNNATCTCGTATGCCGTCTTCTGCTTG -o trimmed.1.fastq reads.1.fastq > -o trimmed.1.log*

# note to self: Rui’s mmPCR adapters has an additional ‘A’ at the beginning. So the read will be *AGATCGGAAGAGCACACGTCTGAACTCCAGTCACNNNNNNATCTCGTATGCCGTCTTCTGCTTG*

To run cutadapt on all files in a folder, use the script **run\_cutadapt.sh.**

If you don’t understand why adapters need to be trimmed, look [here](http://www.med.unc.edu/pharm/calabreselab/files/tufts-sequencing-primer).

# Calling Variants using Bowtie2 + Samtools

## Alignment

In order to call variants, the first step is to align the short reads to the reference genome. In this case, we will be using Bowtie2 as our aligner. For a description of Bowtie2, click [here](http://bowtie-bio.sourceforge.net/bowtie2/index.shtml).

I am using Bowtie2 on the Stanford scg cluster, so I first login to a large memory node and load bowtie using:

$ ssh <user\_id>@greenie

$ module load bowtie/2.2.1

The basic command of bowtie2 is:

$ bowtie2 -x <bowtie2\_index> -U <input\_fastq\_files> -S <output\_sam\_files>

The –x, -U and –S are called flags. There meanings are:

* -x indicates this command line input is the bowtie2 inde[x].
* -U Comma-separated list of files containing [U]npaired reads to be aligned, e.g. lane1.fq,lane2.fq,lane3.fq,lane4.fq. Reads may be a mix of different lengths. If - is specified, bowtie2 gets the reads from the "standard in" or "stdin" filehandle.
* -S output [S]am files

To make our lives easier, let’s create a few atlases first:

fastq\_dir=<directory\_to\_fastq>

sam\_dir=<directory\_to\_sam>

index\_dir=<directory\_to\_bowtie\_index>

log\_dir=<directory\_to\_logs>

Now let us run bowtie2:

bowtie2 -x $index\_dir/<index\_basename> -U $fastq\_dir/<fastq\_file> -S $sam\_dir/<sam\_file> 2><log\_file>

The last part of the command “2> <log\_file>” redirects stderr to a log file for later reference.

A moment later, bowtie2 spits out:

137294 reads; of these:

137294 (100.00%) were unpaired; of these:

2189 (1.59%) aligned 0 times

128799 (93.81%) aligned exactly 1 time

6306 (4.59%) aligned >1 times

98.41% overall alignment rate

That’s pretty much all we need to do about alignment. Simple, right? A few additional resources.

* Bowtie2 manual ([link](http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml))
* Run bowtie2 on all files in a directory: **run\_bowtie2.sh**

## Compress and Sort

After alignment has finished, the resultant sam files need to sorted and compressed using:

samtools view -bS <sam\_file> | samtools sort - <bam\_file\_prefix>

The dash (-) tells samtools to take stdin as input.

For instance:

samtools view -bS 020311-1\_S40\_L001\_R1\_001.sam | samtools sort - 020311-1\_S40\_L001\_R1\_001.sorted

Ignore the following warning, it is a known bug.

[bam\_header\_read] EOF marker is absent. The input is probably truncated.

To compress and sort all files in a directory, use **compress\_and\_sort.sh.**

## Call Genotype Likelihoods using Samtools

In this pipeline we will use the Mpileup command in Samtools to call genotype likelihoods or to call variants.

Type the following to call genotype likelihood:

samtools mpileup -g -f <genome> <sorted\_bam\_file> > <bcf\_file>

* -g Compute genotype likelihoods and output them in the binary call format (BCF).
* -f indicate <genome> is indexed by samtools faidx (a module in samtools).

For example:

hg19=/srv/gs1/projects/montgomery/shared/genome/hg19/hg19.fa

samtools mpileup -g -f $hg19 020311-1\_S40\_L001\_R1\_001.sorted.bam > 020311-1\_S40\_L001\_R1\_001.bcf

To inspect bcf files, use bcftools, which comes with samtools.

bcftools view 020311-1\_S40\_L001\_R1\_001.bcf

Additional information:

* bcf file format specification: [link](http://www.1000genomes.org/wiki/analysis/variant-call-format/bcf-binary-vcf-version-2)
* sam file format specification: [pdf](http://samtools.github.io/hts-specs/SAMv1.pdf)
* call genotype over all files in a directory: **call\_genotype\_likelihood.sh**

## Call Variants With Samtools

### One diploid individual:

samtools mpileup -uf ref.fa aln1.bam

bcftools view -bvcg <bcf\_file> > <variants\_bcf\_file>

bcftools view <variants\_bcf\_file> | vcfutils.pl varFilter –D 100 > <filtered\_ variants\_bcf\_file>

samtools mpileup flags

* -u compute genotype likelihood and output in [u]ncompressed format (vcf).
* -f samtools [f]aidx indexed file

bcftools flags

* -b output in bcf format
* -v output variant sites only
* -c call variants using Bayesian inference
* -g call per-sample genotype at variant site

vcfutils flags

* -D 100 controls the maximum read depth, which should be adjusted to about twice the average read depth

### Multiple diploid individuals

TBD

### More information:

* Powerpoint from EBI explaining this pipeline: [pdf](https://www.ebi.ac.uk/training/sites/ebi.ac.uk.training/files/materials/2014/140217_AgriOmics/dan_bolser_snp_calling.pdf)
* batch implementation: **call\_variants\_samtools.sh**

# Call Variants using GATK

## Preprocessing

### Alignment

Align with BWA using:

bwa mem -M -R ’<read group info>’ reference.fa raw\_reads.fq > aligned\_reads.sam

The ‘<read group info>’ can be:

@RG\tID:group1\tSM:sample1\tPL:illumina\tLB:lib1\tPU:unit1

Batch implementation: **run\_bwa\_single\_end.sh**

### Dedupping

Use Picard. This step is only necessary for whole-genome sequencing; you don’t need to dedup for mmPCR-seq data.

### Indel Realignment

TBD

### Base Recalibration

TBD

## Variant Discovery

TBD

## Troubleshoot

* SAM file doesn't have any read groups defined in the header. The GATK no longer supports SAM files without read groups

Many GATK algorithms needs information that certain reads are sequenced together as these algorithms try to compensate the variation from one sequencing to the next. A read group, or RG, provides this information. Reads in the same read group are assumed to be sequenced on one lane. ([ref](https://www.biostars.org/p/43897/))

Implementation: **add\_read\_groups.sh** to add read group.

Click [here](https://www.broadinstitute.org/gatk/guide/article?id=2909) to know more about how to fix a badly formatted BAM.

TODO: add batch functionality to **add\_read\_groups.sh**

* Input files reads and reference have incompatible contigs: Relative ordering of overlapping contigs differs, which is unsafe.

Solution: reorder the sam/bam file with picard/ReorderSam. ([ref](https://www.broadinstitute.org/gatk/guide/article?id=58), [help](http://broadinstitute.github.io/picard/command-line-overview.html#ReorderSam))

Implementation: **reorder\_sam.sh**

TODO: add batch functionality to reoder\_sam.sh