

▼ Install Kallisto

▼ Install Kallisto On a Mac

- Open Terminal
- ▼ Install HomeBrew
 - `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`

▼ Install Kallisto

- `brew tap homebrew/science`
- `brew install kallisto`

▼ Test whether it is properly installed by typing “kallisto”

- should see this output:

• `kallisto 0.42.4`

```
Usage: kallisto <CMD> [arguments] ..
```

```
Where <CMD> can be one of:
```

<code>index</code>	Builds a kallisto index
<code>quant</code>	Runs the quantification algorithm
<code>h5dump</code>	Converts HDF5-formatted results to plaintext
<code>version</code>	Prints version information

```
Running kallisto <CMD> without arguments prints usage information for <CMD>
```

▼ Build Index from reference transcriptome fasta file

- Get reference transcriptome files from here: <http://useast.ensembl.org/info/data/ftp/index.html>
- Choose organism and cDNA, then download the file that has “cDNA.all.fa.gz”

▼ Build your index. Here’s the command for human:

- `kallisto index -i myHumanIndex Hs.trans.shortnames.fasta`

▼ Output will look like this, and will create a single file called “myHumanIndex” in your directory

```
[build] loading fasta file Homo_sapiens.GRCh38.cdna.all.fa.gz
[build] k-mer length: 31
[build] warning: clipped off poly-A tail (longer than 10)
           from 1400 target sequences
[build] warning: replaced 83 non-ACGUT characters in the input sequence
           with pseudorandom nucleotides
[build] counting k-mers ... done.
[build] building target de Bruijn graph ... done
[build] creating equivalence classes ... done
[build] target de Bruijn graph has 1057821 contigs and contains 104600239 k-mers
```

▼ Quantify transcript expression

- ▼ Run the following command for pseudo alignment of **single-end** reads to index. NOTE: bootstrapping (-b command in the line below) adds significant time to the mapping, but is essential for accurate quantification. With a ‘typical’ fastq file for the mouse genome, I find it takes about 15sec per bootstrap. So expect this to add ~30 min to the mapping time for each sample.

- Note: avoid putting hyphens in the name of the kallisto output...causes a problem down the line

- `kallisto quant -i myHumanIndex -o Sample1.mapped -b 100 --single -l 275 -s 20 read1.fastq.gz`

- ▼ Run the following command for pseudo alignment of **paired-end** reads to index

- `kallisto quant -i myMouseIndex -o Sample1.mapped -b 100 read1.fastq.gz read2.fastq.gz`

- ▼ When the read mapping is complete, you should see some summary statistics on the screen

- ```
[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 31
[index] number of targets: 96,626
[index] number of k-mers: 87,312,671
[index] number of equivalence classes: 333,129
[quant] running in paired-end mode
[quant] will process pair 1: B6-WT-untreat-rep1_S1_mergedLanes_read1.fastq.gz
 B6-WT-untreat-rep1_S1_mergedLanes_read2.fastq.gz
[quant] finding pseudoalignments for the reads ... done
[quant] processed 28,405,354 reads, 21,336,303 reads pseudoaligned
[quant] estimated average fragment length: 136.291
[em] quantifying the abundances ... done
[em] the Expectation-Maximization algorithm ran for 1,116 rounds
```

#### ▼ Install Sleuth

- Open RStudio and install the 'rhdf5' package from the BioC website
- install devtools package (if you don't already have it)
- ▼ install sleuth from lior's github page
  - `devtools::install_github("pachterlab/sleuth")`