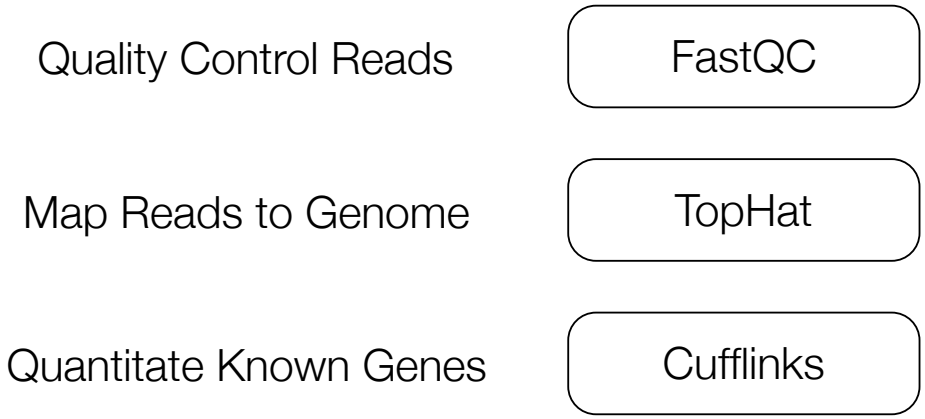
Quantitative Biology Bootcamp

Intro to RNA-seq

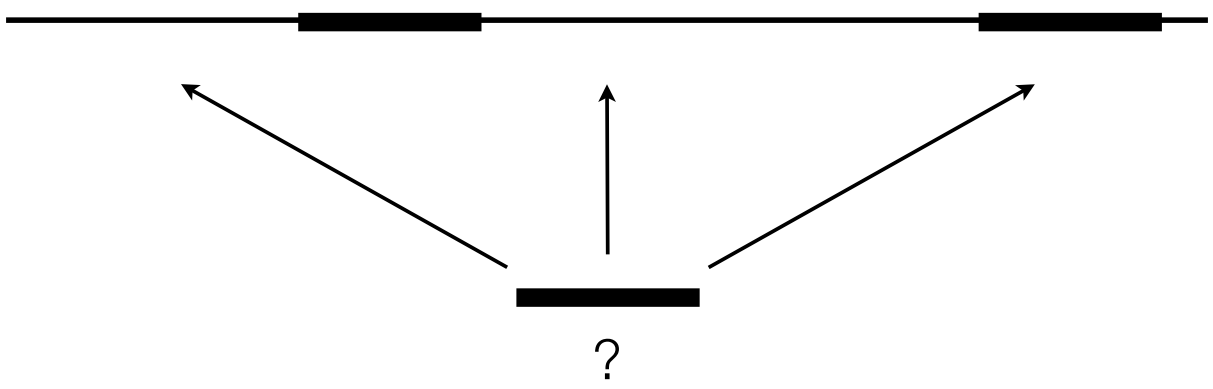
Frederick J Tan
Bioinformatics Research Faculty
Carnegie Institution of Washington, Department of Embryology

2 September 2014

RNA-seq Analysis Pipeline



The Mapping Problem



GOALS

Find match quickly

Find true match

Avoid missing a match

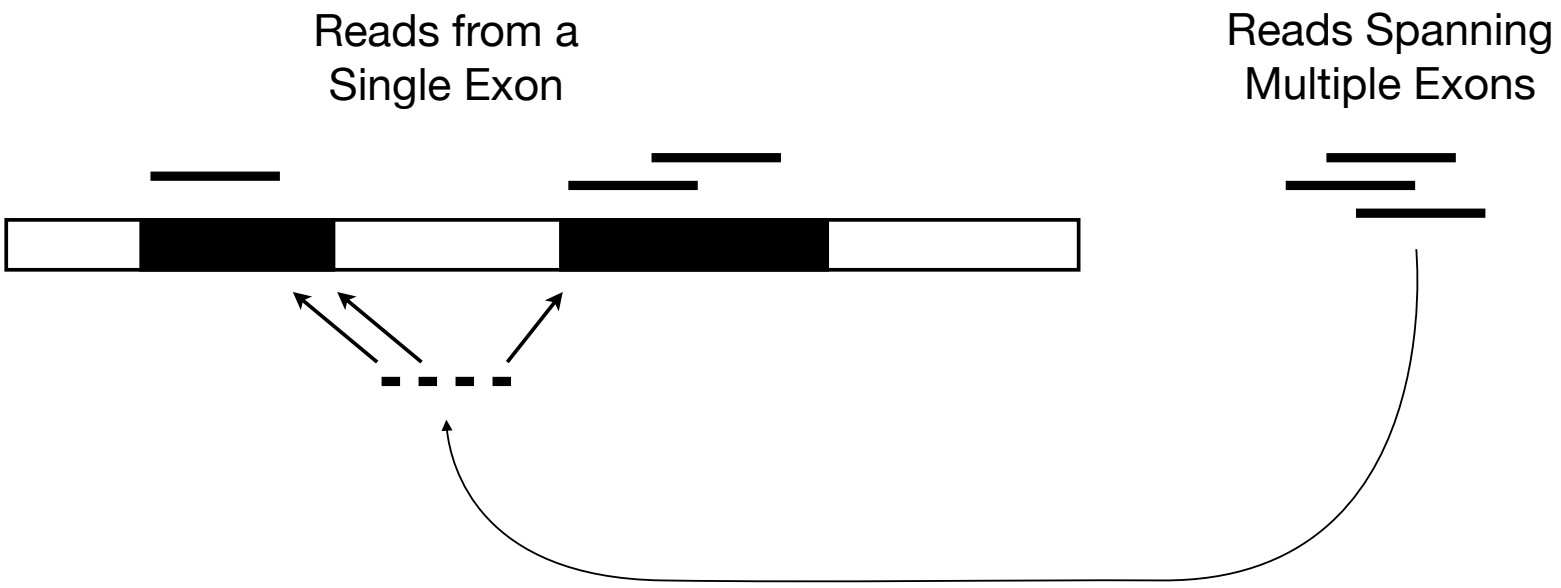
COMPLICATIONS

Large genome, many reads

Paralogs, pseudo-genes, repeats

Sequencing errors, polymorphisms

The Junctions Read Problem



Mapping Solutions

<CTRL>-F **BLAST** **BLAT** **Bowtie2** **TopHat**

how many sequences do you need to search?

hundreds or billions

how many genomes are you searching against?

one or many

how many mismatches will you consider?

few or many

An "ultrafast and memory-efficient tool" for aligning reads to a genome

Table of Contents

Version **2.2.2**

billions of reads
(50 - 1000+ bases)

gapped and local alignments

reports a single alignment (by default)

bowtie-bio.sourceforge.net/bowtie2/manual.shtml

Cheatsheet for `bowtie2` Syntax

```
$ man bowtie2
```

```
No manual entry for bowtie2
See 'man 7 undocumented' for help when manual pages are not available.
```

```
$ bowtie2
```

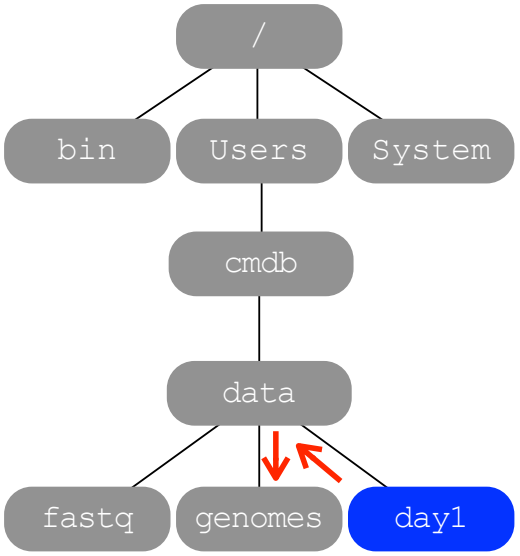
```
No index, query, or output file specified!
Bowtie 2 version 2.1.0 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
Usage:
  bowtie2 [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S <sam>]

  <bt2-idx>  Index filename prefix (minus trailing .X.bt2).
             NOTE: Bowtie 1 and Bowtie 2 indexes are not compatible.
  <m1>       Files with #1 mates, paired with files in <m2>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <m2>       Files with #2 mates, paired with files in <m1>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <r>        Files with unpaired reads.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <sam>      File for SAM output (default: stdout)
```

Task 1: Change working directory to `/Users/cmdb/data/genomes`

```
/Users/cmdb/data/day1 $ cd ..
```

```
/Users/cmdb/data $ cd genomes
```




```
/Users/cmdb/data/day1 $ cd ../genomes
```

```
/Users/cmdb/data/day1 $ cd /Users/cmdb/data/genomes
```

```
/Users/cmdb/data/day1 $ cd ~/data/genomes
```

↑ ~ is a nickname for `/Users/cmdb`

Task 2: Examine contents of dmel-all-chromosome-r5.57.fasta

```
$ ls
```

```
$ head dmel-all-chromosome-r5.57.fasta
```

```
$ less dmel-all-chromosome-r5.57.fasta
```

```
>YHet type=chromosome_arm; loc=YHet:1..347038; ID=YHet; dbxref=REFSEQ:NW_00184
AGGGTCACGTAATGCTGATCCAGTCTTGTTTTTATTTTCATTCATGTTCC
CGCTCTTGCTTTGATTCCGACTTCTAACGTTTAACCTGTGATCAGACGTT
TCACTGCTCCATATTTTACGTGTGCCTGCCGGTCATCTTGGGTAGAGTTA
GCATATCCGTTAATTAATATAAGCGGGGTCTTTCCCATTTCCTATTAAGA
GTAATTTCTTACTTATTTTTTGTATGGGTATATCTGCCTCGTAATCACAG
ATTGTGCTTTTCTAGTTTTTGTATATTAATGGTAACAACTTTCTCTTTTT
```

(**f**)orward
(**b**)ack
(**q**)uit
(**h**)elp

```
$ grep "^>" dmel-all-chromosome-r5.57.fasta
```

Task 3: Build a Bowtie2 genome index

```
$ bowtie2-build
```

```
No input sequence or sequence file specified!
Bowtie 2 version 2.1.0 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
Usage: bowtie2-build [options]* <reference_in> <bt2_index_base>
    reference_in            comma-separated list of files with ref sequences
    bt2_index_base          write .bt2 data to files with this dir/basename
```

```
$ bowtie2-build dmel-all-chromosome-r5.57.fasta dmel5
```

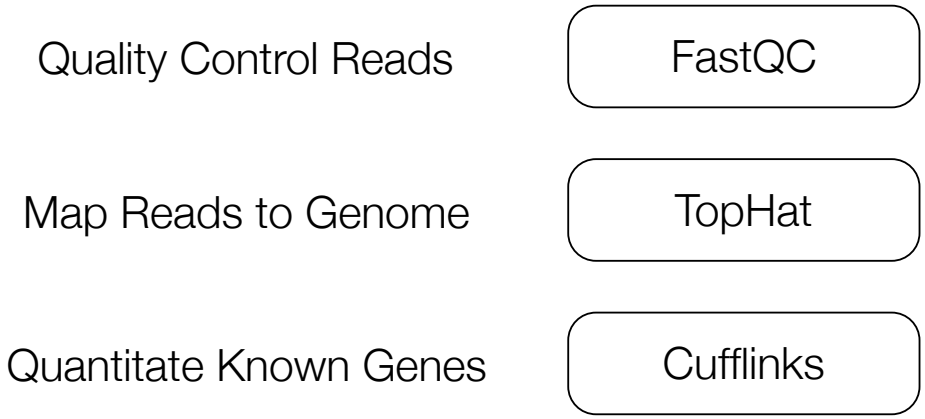
Task 4: Monitor activity after using `top`

```
$ top
```

```
Processes: 152 total, 4 running, 2 stuck, 146 sleeping, 698 threads    07:26:48
Load Avg: 1.10, 1.02, 1.03  CPU usage: 2.11% user, 2.81% sys, 95.7% idle
SharedLibs: 10M resident, 11M data, 0B linkedit.
MemRegions: 27389 total, 794M resident, 50M private, 242M shared.
PhysMem: 3571M used (551M wired), 72M unused.
VM: 372G vsize, 1065M framework vsize, 5583147(0) swapins, 6168331(0) swapouts.
Networks: packets: 15256292/19G in, 5058847/900M out.
Disks: 2724645/300G read, 2184694/158G written.

PID    COMMAND      %CPU      TIME     #TH   #WQ  #PORT #MREG MEM    RPRVT
93282  AppleSpell   0.0       00:15.52 3    1    69    102   2252K  1004K
72277  AirPlayUIAge 0.0       00:10.02 4    1    191   85    1452K  1016K
72274  storeagent   0.0       00:07.28 5    1    252   111   5028K  4528K
72270  pbs          0.0       00:00.95 2    1    53    61    2272K  1732K
72260  com.apple.Sh 0.0       00:08.67 4    1    142   164   1604K  1248K
72254  Quicksilver  0.0       02:41.68 5    1    202   485   19M    13M
72248  CalendarAgen 0.0       00:08.00 3    1    94    103   3200K  2788K
72244  identityserv 0.0       00:08.17 5    1    103   85    2672K  2132K
72243  imagent      0.0       00:01.69 3    1    97    90    1708K  1276K
72241  lsboxd       0.0       00:01.22 3    2    71    102   1448K  812K
72240  Notification 0.0       00:16.80 4    1    177   265   7232K  6468K
72236  SocialPushAg 0.0       00:00.15 2    0    49    57    608K   432K
72235  usernoted    0.0       00:02.09 2    0    83    48    1620K  1296K
```

RNA-seq Analysis Pipeline



Quantitative Biology Bootcamp

Installing BEDTools

Frederick J Tan
Bioinformatics Research Faculty
Carnegie Institution of Washington, Department of Embryology

2 September 2014

Follow the Yellow Brick `$PATH`

```
$ echo $PATH
```

```
$ cd
```

```
$ mkdir bin src
```

```
$ tm .profile export PATH=/Users/cmdb:$PATH
```

```
$ source .profile
```

Task 5: Download BEDTools source code and compile into binary form

```
$ cd src
```

```
$ wget https://github.com/arq5x/bedtools2/releases/download/v2.20.1/
bedtools-2.20.1.tar.gz
```

```
$ tar xzvf bedtools-2.20.1.tar.gz
```

```
$ cd b<TAB>2<TAB>
```

```
$ ls
```

```
$ ls bin
```

```
$ make
```

`Makefile`s Help Automate Software Compilation (and Analysis!)

`$ less Makefile`

```
# ============================
# BEDTools Makefile
# (c) 2009 Aaron Quinlan
# ============================

SHELL := /bin/bash -e

VERSION_FILE=./src/utils/version/version_git.h
RELEASED_VERSION_FILE=./src/utils/version/version_release.txt


# define our object and binary directories
export OBJ_DIR  = obj
export BIN_DIR  = bin
export SRC_DIR  = src
export UTIL_DIR = src/utils
export CXX              = g++
#ifeq ($(DEBUG),1)
#export CXXFLAGS = -Wall -O0 -g -fno-inline -fkeep-inline-functions -
D_FILE_OFFSET_BITS=64 -fPIC -DDEBUG -D_DEBUG
#else
export CXXFLAGS = -Wall -O2 -D_FILE_OFFSET_BITS=64 -fPIC $(INCLUDE)
:
```

Task 6: Install `bedtools` to your personal `bin` folder

```
$ ls bin
```

```
$ cd bin
```

```
$ ls -l
```

```
$ which bedtools
```

```
$ cp bedtools ~/bin
```

```
$ which bedtools
```

NOTE: current directory not searched

Quantitative Biology Bootcamp

Bash scripting

Frederick J Tan
Bioinformatics Research Faculty
Carnegie Institution of Washington, Department of Embryology

2 September 2014

Hello, World!

Open TextMate

Write script:

```
echo "Hello, world"
```

Save as hello_world.sh in /Users/cmdb

```
/Users/cmdb $ ls -l
```

```
/Users/cmdb $ chmod +x hello_world.sh
```

```
/Users/cmdb $ ls -l
```

```
/Users/cmdb $ hello_world.sh
```

```
-bash: hello_world.sh: command not found
```

```
/Users/cmdb $ ./hello_world.sh
```

Useful Concepts

#! hash-bang sha-bang

Global Variables

Comments

For loop

A Very Buggy Bash Script

```
#!/bin/bash

#
# Day 1 - Homework: Part 2 - debug this bash script
#

echo "There are around 6 mistakes"

FASTQ_DIR=/Users/cmdb/data/fastq
OUTPUT_DIR=/Users/cmdb/data/day1

GENOME_DIR=/Users/cmdb/data/genomes
=dmel5
ANNOTATION=dmel-all-r5.57.gff

CORES=4

for i in
  echo fastqc $FASTQ_DIR/$SAMPLE_PREFIX$i\.fastq.gz -o $OUTPUT_DIR
  echo tophat
  echo cufflinks
```