DNA mappers

Preliminaries

Work directory

IMPORTANT please do not write your output files to the /share/data drive. If everyone tries to do that it is likely to create I/O problems for everyone. Please do output writing to your local disk. You should have created a folder for this by now

```
~/Day45/results
```

If not go ahead and do so and make sure to direct all output there or sub folders there. You can do this in many ways. Perhaps one easy way would be to add a variable to your `config.sh` file:

```
RES=~/Day45/results
```

and then always prepend this to any output file

```
$ bwa mem $GENOME R1.f.gz R2.f.gz >$RES/sample.sam
```

or cd to that directory and do all your work from there.

Check path to mapping programs

Most of the modern mappers are pretty easy to use; at least in default mode. First check that

- `bwa`
- `STAR`

are on your path. Just time the name of the command and you should get a nice help screen. For example:

```
$ bwa

Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.12-r1039
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
         mem           BWA-MEM algorithm
...
```

Another way to tell if a command is on your path is to type:

```
which CMD
```

and if it is on your path you will see the full directory path to the CMD.

Get test genome path

All of the data for this exercise used a reduced version of the human genome (NCBI build b37) which contained Chromosomes chr4, chr7, chr12, and chr17. This was to the reduce the size of the files, the amount of memory needed and to reduce run time. The path to the root folder with all the necesarry genome files is:

```
$ROOT45/genomes/H.Sapiens/b37_hl
```

Remember that `$ROOT45` is the folder we previous set in our `config.sh` file in `~/Day45/code`. Now is a good time to add some more useful variables to your `config.sh` file. There is not hard and fast convention but I recommend:

```
# Compgen2016 Day 4,5 configuration file

# Path to root of lab data directories

ROOT45=/share/data/compgen2016/day45_Intro2Seq_VarCalling

# Genome files

GENOME_BUILD=human_b37
GENOME_ROOT=$ROOT45/genomes/H.Sapiens/b37_hl
GENOME_FASTA=$GENOME_ROOT/b37_hl.fa
GENOME_DICT=$GENOME_ROOT/b37_hl.dict
GENOME_BWA=$GENOME_ROOT/index/bwa/b37_hl.fa

# Note for STAR you need to pick
#    NoGTF
#    Gencode_75
GENOME_STAR=$GENOME_ROOT/index/star
```

Once you have your `config.sh` file setup with the new variables remember to `source` it to get those variables set in your environment. Remember you need to source this file for every new terminal window. If you like you can add this source command to your `.profile` or `.bashrc` file so it is done automatically.

To check that everything is working ok do the following:

```
$ cat ${GENOME_FASTA}.fai
```

if all is good you should see:

```
4       191154276       3       60      61
```

```
7       159138663       194340187       60      61
12      133851895       356131166       60      61
17      81195210        492213930       60      61
```

which is the `samtools` fasta index for this genome.

Mapping synthetic DNA data.

To generate these datasets I used the program `wgsim` from the author of BWA. It is not installed but is prehaps the easiest program of all the ones you are looking at to install. At the end of this exercise I have the location for it for those you want to play with generating their own data.

Building indexes

Normally the first step when using a mapper for the first time with a new genome is to build the genome index. This is a time consuming process but it greatly speeds up the subsequent mapping runs.

To save time I have pre-build the indexes for three aligners on the test genome and we already have set the paths to them in the `config.sh` file. However if you are comfortable with UNIX and the command line and would like to get some experience with index building please do so. If you re-build the indexes make sure you keep track of where they are and fix the paths in `config.sh` accordingly

Data set 1: default wgsim generated

The first test is a single end run and consists of one file:

```
$ROOT45/Labs/2_Mapping/data/b37Test2_DEF_1_SE.fastq.gz
```

This one is a single end set so just one read file. Map it with BWA and even though it is DNA data try and map it with STAR.

For BWA use BWA MEM first. If you have time you can experiment with the other modes of BWA. For Bowtie and STAR (and BWA-MEM) us default mapping options. If you are finished early try bowtie2. You will need to build the index for that one so only try that if you have plenty of time or come back to it at the end.

The main point of this exercise is to learn how to use the mappers; so I am not going to give explicit command lines for each. Look at the help screens (remember STAR does not have one) look at the manuals. You can go online for BWA and Bowtie; STAR's manual is a PDF so I put it in the repository at:

```
$ROOT45/Labs/man/STAR/STARmanual.pdf
```

Note, all the mappers can output uncompressed plain text SAM files so you can look at them with `more` or `less` or maybe open them with an editor. In fact if you take the first 10-20 lines (with head) you can maybe even open them in excel or some other spreadsheet program.

Some things to do.

- Record the time it takes to run each.
  - To time a command you can use the unix `time` function. If you do

    ```
    time COMMAND
    ```

    That will run the command and print out the time it took.

    Timing benchmarks are notoriously hard to get right. We do not want to get bogged down here on that but I suggests running each program twice and taking the smaller of the two numbers. But if you have time and ideas try other testing strategies
    - Do you understand why STAR is not used for everything
- Check the % of mapped reads. Think about how to do this; but do not spend too much time. There is a quick script in:
  `bash $ROOT45/Labs/2_Mapping/code/calcPercentMapped.sh`

which will give the %-mapped and unmapped. In the next module we will use a much more powerfull program to compute mapping statistics.

- Look at the different SAM files in details. In particular look at the TAGS, especially the custom tags for each.
  - BWA: XA, XS
  - Bowtie: XG, XM, XN, XO, XS
    You should learn what those they mean and while there is no point memorizing them you should be able to know how to find the meanings of them when needed
- If you have time: **wrap** the commands with `Bash` scripts. STAR is a prime candiate for this as it has a complex command line and writes lots of files as output so you really one to create a new directory for each STAR run.

Data set 2: paired end data set

The second set is paired end data. There are two files this time

```
b37Test2_DEF_2_R1.fastq.gz
b37Test2_DEF_2_R2.fastq.gz
```

This follows the Illumina convention: R1 is the 'left' read, R2 the 'right' (in sequencing space not necessary genome space)

Map this data; all the mappers have modes for paired end mapping. You can drop STAR now unless you are convinced it is a good idea for DNA sequencing.

Do the same things you did for the single end case:

- Time them
- Measure % mapped/unmapped
- Look into the files

Then if there is time; play with the various options in the mappers and see how they effect things. Maybe try BWA ALN (which is not really used anymore but had some nice properties). If you are really adventurous the extra credit is to find other mappers to download and try them; SHRiMP is a personal favorite but it is no longer maintained. But if you need to deal with color data from the SOLiD sequencer is is one of the easiest to use.

Data set 3: Larger set of target capture data

This next set is a larger set of sequencing data from a target capture array. It is not a real sample as real data from humans is usually protected and requires special permission for access. So it is synthetically generated but tries to mimic real cature data (in the case captures from the IMPACT panel).

The folder with the data in it is

```
$ROOT45/suppData/DNA
```

and the files are:

```
normalP_L016__R1_001.fastq.gz
normalP_L016__R2_001.fastq.gz
```

Map this data with the mapper of your choice. We will use it in the next two labs. Note, it is single end data with two (2) different samples.

These datasets are large so they might take awhile to map. I suggest running in the background and moving on to the next step mapping RNAseq data

MultiCore tests

Your computers have multiple CPU's. To find out how many do

```
cat /proc/cpuinfo | fgrep processor | wc -l
```

Read the manuals and try remapping the data specifying multiple threads. See how much of a speed up you get.