February 2, 2012

LAB2, BB2490 Analysis of data from high-throughput molecular biology experiments

LAB2:

A. Using the Uppmax computers
B. Mapping of MPS sequence data to the human reference genome
C. DD2399 only: Advanced assignments

Welcome to the second computer exercise of BB2490/DD2399 spring 2012. In this exercise, you will practice how to access and use the Uppmax high-performance computing cluster Kalkyl (that some of you will use for the project as well), and specifically how to map sequence data. There is also an advanced assignment (C.) for students taking the DD2399 course.

Uppmax/Uppnex/Kalkyl

Uppmax stands for Uppsala Multidisciplinary center for advanced computational science, and is part of the Swedish National Infrastructure for Computing (SNIC). Uppmax is Uppsala University's resource for high-performance computing and related know-how, but as it is a "national infrastructure", any researcher affiliated with a Swedish university is welcome to use the resource. More information is available at http://www.uppmax.uu.se.
Uppnex is part of Uppmax and specializes in the handling and analysis of data from massively parallel sequencing (MPS; a.k.a. next-generation sequencing). More information is available at http://www.uppnex.uu.se.
Kalkyl is the login node of Uppnex, i.e., the actual computer to which you log in to access the computing power.

Report

You should write the answers to all the below questions in a lab report (one single file). Use the program OpenOffice to write the report ([1] Applications -> Accessories -> OpenOffice; [2] Archive->New->Text Document) . Start by saving the report as a file named lab2_<first_name>_<first_name>.odp (example: lab2_osqulda_osquar.odp). The file type .odp is the default file format of OpenOffice text documents, but you can also choose to save it as a .doc file if you wish. Check where the file is saved. Remember to save the file now and then as you go along. In the report, start all answers with the question number (Q1:, Q2:…).

Contents required: The report should contain answers to questions and the required figures. You should also write the names of the two students that co-authored the submitted report (i.e., yourself and your lab partner).
Instructions for file format: Submit the report as **one single file**, use .doc, .pdf, .odt or .odp format. Please include all your figures (if any) in the .doc/.pdf/.odt/.odp file you submit. Do not submit figures as individual files, make sure to include them in the main report (in OpenOffice,

use Infoga->Bildobjekt->Från fil). To save a document as .pdf, use (in OpenOffice) Arkiv->Exportera som PDF.
Instructions for submission: To submit the lab report, email it to kristoffer.sahlin@scilifelab.se. The lab report should be submitted **before Thursday 9 february, 2012, at 17:00**. Again, note that it should all be in **one single file**. Don't forget to write your names on the first page of the report.

PART A.
Accessing Uppmax computer cluster through Kalkyl, and how to run jobs on the cluster

You will find the documents related to this exercise at KTH Social (for DD2399) or on the Kalkyl file system.

Linux commands and descriptions of commands can be found here:
http://www.computerhope.com/unix/overview.htm
Or you can look at a Linux introduction at the Uppnex web site:
https://www.uppnex.uu.se/uppnex-book/getting-started/screencast

A.1. LOG IN TO KALKYL

This computer exercise will be performed on the computer cluster **Kalkyl**, hosted at the Uppmax computing center in Uppsala.

You should use your Uppmax account user name and password to log in to Kalkyl.

Follow the instructions on this web page to log in: https://www.uppnex.uu.se/uppnex-book/getting-started/login-ssh
(This web page also contains information about how to copy files to/from your Uppmax account, and how to log out).

Once logged in, you should be greeted with text that looks like this:

```
Last login: Thu Jan  6 14:59:36 2011 from 130-157-217-38.scilifelab.ki.se
------------------------------------------------------------------------
                        Welcome to Kalkyl
     Latest & Greatest cluster at Uppmax, Uppsala Universitet
                     access node kalkyl1
------------------------------------------------------------------------
```

and a lot more text. **Note**: if this is the first time you log in to your Uppmax account, you will first be asked to change your temporary password (which you should have received in an email shortly after you applied for your account).

When you logged in, you were entered into your home directory on the file system. The path to your home directory is /home/YOUR_USER_NAME.
In your home directory you can create files and subdirectories. For this lab, create a Lab2 directory:

```
mkdir Lab2
```

Then go to that directory:

```
cd Lab2
```

No further instructions will be provided regarding how to organize your files during this lab. You may want to follow the directory structure of the Noble paper.

More information about your home directory at Uppnex can be found here:
https://www.uppnex.uu.se/uppnex-book/using-resources/filedata-storage/home-file-system

More information about Uppnex, Kalkyl and how to use these resources can be found here:
https://www.uppnex.uu.se/content/support

A.2. FILES YOU NEED FOR THIS LAB

The files you need for this lab are present in /proj/g2012009/INBOX/BB2490_Lab2. Whenever a sequence file is needed in this lab you should copy it to your own /home/YOUR_USER_NAME/Lab2 directory (or suitable subdirectory). Do not copy the index files (see below for information about index files).

A.3. HOW TO RUN COMPUTING JOBS ON KALKYL

We will use something called SLURM to start computing jobs. It is a scheduling system that enables many users to share computing resources in a fair way. You "submit" your computing jobs, then your jobs enter a "queue", and once it's your turn you will be "allocated resources" (a resource is a computing node) and your job is run.

Specifically, to run a job on Kalkyl you need to put the job in a "queue" with the `sbatch` command. Even more specifically for this computer exercise we have reserved 9 nodes (between 08:00 and 12:00 today) so your job should be started more or less immediately. But there might occasionally be some waiting time. It's normal.

A.4. HOW TO USE SBATCH AND THE MODULE SYSTEM

Using `sbatch` means that you will have to write down the commands you would like to execute in a bash script (file name ends with .sh)

For instance, if you'd like to run the job
`echo "hello world"`
you'd simply put this line in a script, and submit the script (see *SUBMITTING JOBS* below).

There is an example sbatch script here:
/proj/g2012009/INBOX/BB2490_Lab2/sbatch_template.sh

Copy it to your own /home/YOUR_USER_NAME/Lab2 directory. You are supposed to modify and use the sbatch_template.sh script during this lab. You can change the name if you want. Look at the file using `less, more` or `cat.` Modify it using `emacs, vi` or any other text editor you prefer.

This line in the template file is important. It specifies what options you should use when running the sbatch script:
`–A g2012009 –t 2:00:00 –p node –n 8 --reservation=g2012009 –o my_job_name.out`

Here's what these options mean:

`-A g2012009` specifies that you are working with project g2012009 (which is the project of this course)
`-t 2:00:00` specifies that you're requesting 2 hrs of computation time (the stuff you'll do in this computer lab will easily be run within this time limit).
`-p node` specifies that you're requesting a node partition.
`-n 8` (this is rather technical, see `man sbatch` if you're interested)
`--reservation=g2012009` specifies that you would like to use one of the nodes that have been reserved for this computer exercise
`-o my_job_name.out` specifies the output file where the STDOUT from the job will end up. You may change this.

THE MODULE SYSTEM
Kalkyl uses the "module system". The module system is a way to get access to certain softwares that have been installed on the computer. In order to use, e.g., the module that enables the read mapping program bwa, you have to *add* the bwa module. At **Kalkyl** you should first type the command:
`module add bioinfo-tools`

which will give you access to a lot of different bioinformatics related modules (including the bwa module). Then you specify that you want the bwa module:
`module add bwa`

Once you've typed "module add bwa", you can use the bwa program. There are many different modules available on Kalkyl, you can see a list by typing
`module avail`

And to find out what modules are already added, type:
`module list`

Once you've added a module, it will be available until you log out.

SUBMITTING JOBS
To submit the job(s) specified in the sbatch_template.sh, run this command:
`sbatch <OPTIONS> sbatch_template.sh`
(For the options, see above).

Then, the jobs specified inside the script will enter a queue. You can check the status of the queue using
`squeue`

You should modify the sbatch_template.sh using `emacs –nw` or any other text editor you like in order to suit the assignments in this lab.

For each task in PART B, you should enter the command into the sbatch_template.sh script file, save the script file, and submit the script file as above.
You may also in the script file specify several commands to be run sequentially. This is very useful. More information and an example (`bwa aln`) are in the sbatch_template.sh script file!

PART B.
Mapping (aligning) a set of sequence reads to a reference genome

B.1. THE DATA

Here is a data set from the 1000 genomes (a.k.a. 1KG) project. It contains genomic DNA reads from one Homo sapiens individual. The reads are available as a .fastq file here:

`/proj/g2012009/INBOX/BB2490_Lab2/ERR001014.filt.fastq`

Copy the file to your Lab2 directory (or a suitable subdirectory therein).

The INBOX subdirectory of a project directory is the standard place where the sequencing facilities deliver the sequence data.

Q1: Use the Unix commands `wc` and `expr` to determine how many reads there are in this file. (Use `man wc` *or* `man expr` if you'd like to know more about the commands. Exit the man page with q).
The data set is ready to be aligned. It has been filtered (by the 1KG crew) according to this file: ftp://ftp.ncbi.nih.gov/1000genomes/ftp/README.sequence_data (see section 2, paragraph Sequence checks). (Furthermore, to reduce the waiting times during this lab, we have also removed ~75% of the reads).

B.2. ALIGNING THE SEQUENCE READS

Use BWA to align (map) the reads of the file onto the reference genome. To do this you need an indexed version of the genome to map against (the human reference genome GRCh37 in this case). Normally you would do this by performing these two actions:
1. download the reference genome (in fasta format), from http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/human/index.shtml
2. index the fasta file using `bwa index -a bwtsw all_chr.hg19.fa`
But since the indexing takes about an hour, it has been prepared for you. The index is available here:
`/proj/g2012009/INBOX/BB2490_Lab2/all_chr.hg19.fa`

(Don't copy this file). Now start the alignment. From the BWA web page you can get the instructions: http://bio-bwa.sourceforge.net/bwa.shtml. But we provide suitable command lines below (to be entered into your sbatch_template.sh script file).

Start the alignment (and note, this line is already in the sbatch_template.sh script):

```
bwa aln /proj/g2012009/INBOX/BB2490_Lab2/all_chr.hg19.fa
ERR001014.filt.fastq >  ERR001014.filt.bwa_aln.default.sai
```

The resulting .sai file is a binary file that contains the suffix array, it is not human readable. Note that this will take a while to run. Proceed with the section about SRA below while it is running.

Then convert our alignment output file (the .sai file) to the .sam format:

```
bwa samse /proj/g2012009/INBOX/BB2490_Lab2/all_chr.hg19.fa
ERR001014.filt.bwa_aln.default.sai ERR001014.filt.fastq >
ERR001014.filt.bwa_aln.default.sam
```

This command will also take a while. The .sam file is human readable.

B.3. SRA- SEQUENCE READ ARCHIVE

While the above commands are running you are asked to find some information about this data set and the aligner bwa on the internet.

First, find out more about the data set we're using. It is available in the Sequence Read Archive, http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi, where data sets from MPS (massively parallel

sequencing) experiments are deposited. Go to the web site, choose the Search tab, choose "SRA Objects" and enter the name of this data set, ERR001014. In the results page, click on the SRA Experiments line.

Q2. What sequencing technology was used to sequence these reads?

Q3. The data set ERR001014 is from a human individual. Can you find the sex of this individual?

Go back to the Sequence Read Archive web page mentioned above. Pick to Main tab, then About, to learn a little about SRA.

Q4. What organization created SRA?

Q5. What is the difference between the Sequence Read Archive and the Trace Archive?

Hopefully your alignment is soon ready. The following tasks in this section should be done after the BWA mapping has finished.

B.4. INVESTIGATING THE ALIGNMENT

We will now use SAMtools to get this file into a format in which we can actually look at the alignment. Apply it on your .sam format outputfile `ERR001014.filt.bwa_aln.default.sam`

But also here we first need to index the genome before using SAMtools (this index is then used by SAMtools. It is not the same index as was created for BWA). This is the command for the indexing:

```
samtools faidx all_chr.hg19.fa
```

This creates a file all_chr.hg19.fa.fai. **Note**: This indexing has already been done for you and the index file is available here:
`/proj/g2012009/INBOX/BB2490_Lab2/all_chr.hg19.fa.fai`

Then we need to do a series of operations (use sbatch where appropriate):

1. Add the module samtools:

```
module add samtools
```

2. Convert SAM to BAM (BAM is the binary format that SAMtools works on)

```
samtools import /proj/g2012009/INBOX/BB2490_Lab2/all_chr.hg19.fa.fai
ERR001014.filt.bwa_aln.default.sam ERR001014.filt.bwa_aln.default.bam
```

3. Sort the BAM file, in order to browse the alignment

```
samtools  sort ERR001014.filt.bwa_aln.default.bam
ERR001014.filt.bwa_aln.default.sorted
```

4. Index the BAM file to enable fast random access:

```
samtools index ERR001014.filt.bwa_aln.default.sorted.bam
```

5. View using samtools view

```
samtools view ERR001014.filt.bwa_aln.default.sorted.bam
```

(Hit CTRL-C to stop the output).
You can direct your output towards certain regions. E.g. the following will give you the .sam formated output of all reads matching region 18,000,000 – 18,080,000 on chr22 into a file.

```
samtools view ERR001014.filt.bwa_aln.default.sorted.bam 'chr22_hg19:18000000-
18080000' > chr22_part.ERR001014.filt.bwa_aln.default.sam
```

Q6. How many reads do you have within this region of chr 22?

Q7. Look at the SAM format definition (from the lecture notes or the SAMtools paper or on the Internet). How many of the reads within this region of chr 22 have a mapping quality above 30? How many have a mapping quality of 0 ? (A bwa mapping quality of 0 means the mapping is ambiguous, ie. the read matches to more than one place on the genome).

B.5. VIEW THE ALIGNMENT WITH TVIEW

SAMtools also has an alignment viewer, tview. Use information on the web page http://samtools.sourceforge.net/samtools.shtml to launch `samtools tview`.

Q8. Try `tview` for a few minutes. (Use '?' to get help inside `tview`). What do you think about this alignment viewer?

Look at your reads at UCSC Genome Browser. The UCSC Genome Browser is a web based genome browser that contains a lot of information. Go to the web site http://genome.ucsc.edu/cgi-bin/hgGateway, which is the start page for the human genome (the UCSC Genome Browser can display other genomes as well).

A random region has been entered into the fields at the top of the page. Press the 'submit' button to the right, and take a look at the page you come to.

You will see a lot of information, and you can use the lists below the main picture to display many more. You can change region of the genome, and zoom in and out using the controls at the top of the page. You can also use the computer mouse to select a region to zoom in on. Play around for a while.

One great thing with the UCSC Genome Browser is that you can upload your own data to it.

A simple file format used in the UCSC Genome Browser is the .bed format. It's a human readable format that is well suited to define regions on a chromosome: http://genome.ucsc.edu/FAQ/FAQformat.html#format1

Q9. What are the 3 required fields (i.e., columns) of a .bed file?

We will use `bamToBed`, which comes with the bedtools package, to convert our alignment to a .bed file. **Note**! You need to add the `BEDTools` module first – do it now. (Type BEDTools with the upper/lower case pattern given here).
To save disk space we will use the `awk` command to restrict the output to only a small part of chr22:

```
bamToBed -i ERR001014.filt.bwa_aln.default.sorted.bam | awk '/^chr22_hg19/
{if ($2>=18000000 && $2<=18080000) print "chr22", substr($0,11)}' >
chr22_part.ERR001014.filt.bwa_aln.default.sorted.bed
```

You may choose another chromosome or region – change the `awk` command accordingly (more information about awk can be found here: http://www.vectorsite.net/tsawk.html).

Now you should upload the .bed file to the UCSC Genome Browser.
Hit the 'add custom tracks' button. Upload your file: browse for it, then hit 'submit'.
Click the 'go to genome browser' when youre file has been uploaded.
You can now see your reads on the Genome Browser! It will appear under the name "User Track". (Use the position/search field to zoom in on the interesting region, chr22:18,000,000-18,080,000).

Q10. Are your reads mostly within exons or within introns of the refseq gene CECR2 in this region? (Hint: exons are thicker lines than introns).

Q11. Take a screenshot of the UCSC Genome Browser window with your submitted track visible. Include the screenshot in the report.

End of lab 2 for BB2490 students.

PART C: Advanced assignments for DD2399

For those of you taking DD2399, the grade is set based on the number of "advanced assignments" you solve. For the genomics part of the course, there are two assignments:

- Understanding SOLiD sequencing, and
- SMAQ: the simplified read mapper.

You find both of these assignments on the web page

February 2, 2012

LAB2, BB2490 Analysis of data from high-throughput molecular biology experiments

http://www.csc.kth.se/dd2399/omsys12/labs

These implementation exercises are due on Feb 9.