2019

Designing Computational Biology Workflows with Perl - Part 2

Esma Yildirim
CUNY Queensborough Community College

Title: Designing Computational Biology Workflows with Perl – Part2

Author/Affiliation: Esma Yildirim / Queensborough Community College

Date: 05/15/2019

Material Type: Lab

CS + Computational Biology

Software/Equipment Dependencies:
An Amazon Web Services (AWS) account, a web browser and a command line interpreter program (e.g. Putty on Windows, Terminal on Linux/MacOSX)

Prior Knowledge Needed (if any): The material covered in “Designing Computational Biology Workflows with Perl – Part1”

Keywords: Gene-sequencing file formats, SNPs, INDELs, Alignment, Variant Calling, Perl

Approximate time needed: 1 hour

Description: This material introduces the AWS console interface, describes how to create an instance on AWS with the VMI provided and connect to that machine instance using the SSH protocol. Once connected, it requires the students to write a script to automate the tasks to create VCF files from two different sample genomes belonging to E.coli microorganisms by using the FASTA and FASTQ files in the input folder of the virtual machine. The same exercise can be applied if the VMI is installed on a local machine using virtualization software (e.g. Oracle VirtualBox). In this case, the Terminal program of the VMI can be used to do the exercise.

Designing Computational Biology Workflows with Perl – Part 2

In this lab, you will learn how to create a virtual machine instance on AWS cloud and write a Perl script to automate the tasks to create a VCF file from FASTA and FASTQ files.

1. What is a Virtual Machine Image(VMI)?

A virtual machine image is the software representation of a machine with its operating system, hardware settings, installed programs and all the data in its file system. It can be configured on any matching hardware settings that might be running on any type of operating system through the use of *virtualization software* (e.g. Oracle VirtualBox).

Example: An Ubuntu Linux Virtual Machine with 10GB hard disk, 4 GB memory settings can be launched on top of a computer that runs Windows operating system and has 250GB of disk, 8GB of memory and a 4-core CPU.

The Virtual Machine Image provided as part of this course is a Ubuntu Linux machine with 4GM of memory and 20GB of disk space, prepared via Oracle VirtualBox Version 5.2.20 r125813 (Qt5.6.2) virtualization software that ran on a Windows operating system.

2. Why do we need a VMI?

The open source software packages used in gene sequencing analysis usually run on top of a UNIX/Linux based system which is not a common operating system type used in a Lab setting. However, they can easily be installed on a UNIX/Linux machine and exported as a VMI to be launched in a Cloud Computing environment through the use of a web browser and a command line interpreter program.

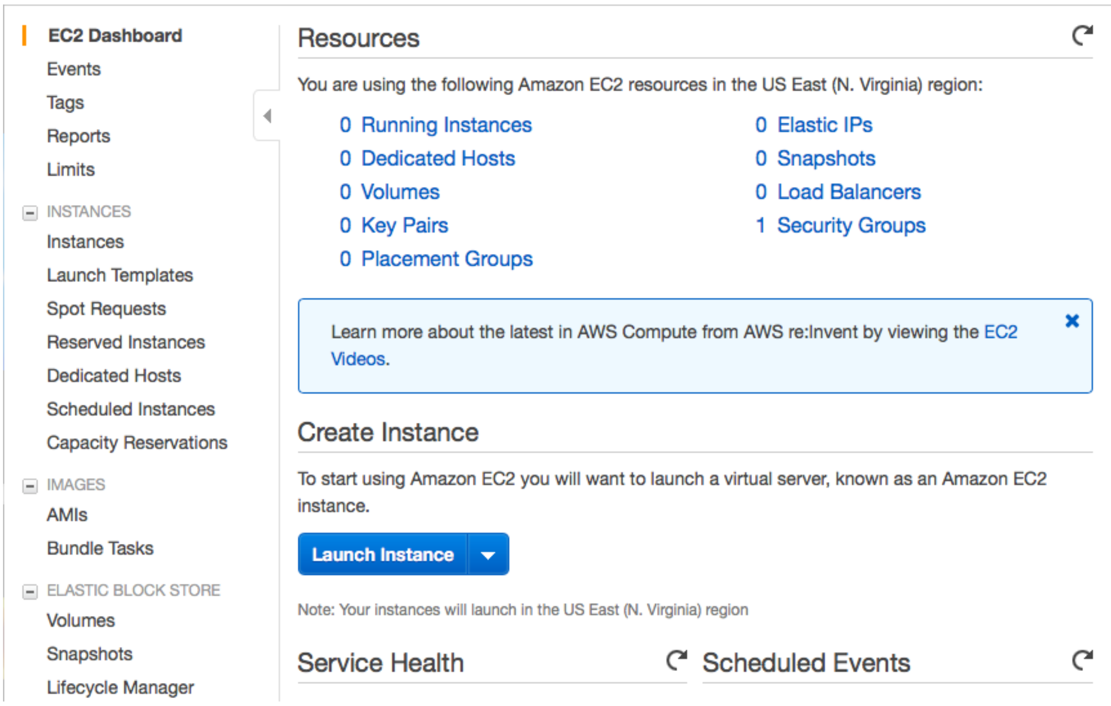
Also, the installation process of gene sequencing software packages is a cumbersome process and they have a lot of dependencies. The VMI comes with all the necessary software pre-installed along with the input data needed to do the lab exercises. After the instructor converts the VMI to an AWS compatible AMI (Amazon Machine Image), shares that AMI through his/her AWS account. You can then launch an instance on AWS cloud using the converted AMI in a few seconds and connect to the machine to do the lab exercises.

Alternatively, use virtualization software like Oracle VirtualBox to import the VMI provided by your instructor and start it on your local desktop machine.

3. Create a machine instance on AWS [Optional]

This section can be used only after the students register and activate their AWS accounts or the instructor creates IAM user accounts for the students through his/her account. The services provided by AWS cloud are many. But the one service that allows us to create machine instances is called the “EC2” service.

Step 1. Go to AWS console web page using this link and sign in. From the services menu at the top, select EC2.



You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

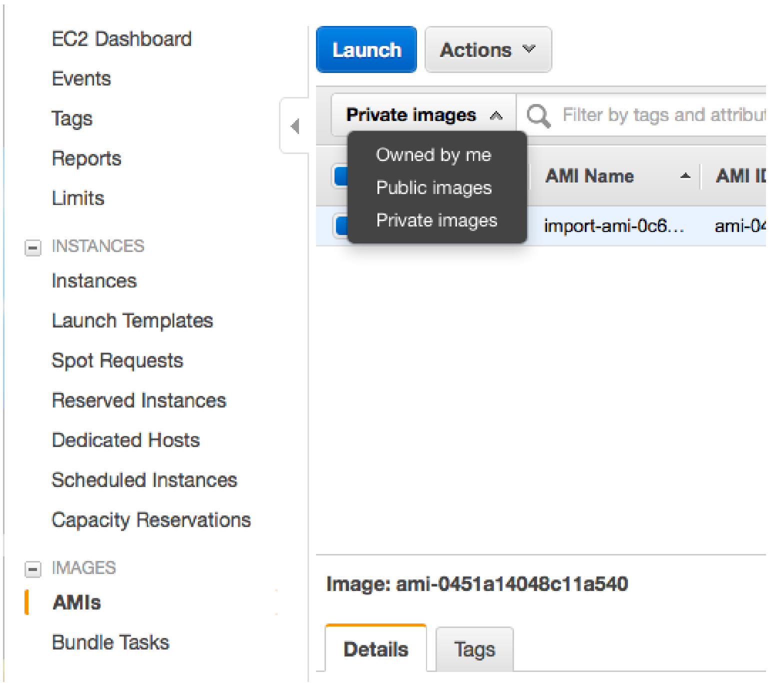
Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Figure 1. EC2 Dashboard

Step2. From the left menu, select “AMIs” under “IMAGES”. The list of AMIs will appear on the right hand side frame. From the drop down menu, select “Private images” and you will see the AMI added by your instructor.



AMI Name	AMI ID
import-ami-0c6…	ami-04

Image: ami-0451a14048c11a540

Figure 2. AMI list.

Step 3. Select the AMI from the list and select “Launch” from the “Actions” menu.

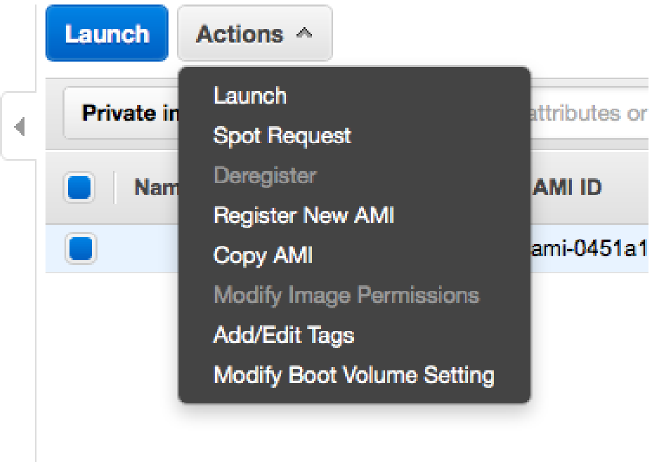
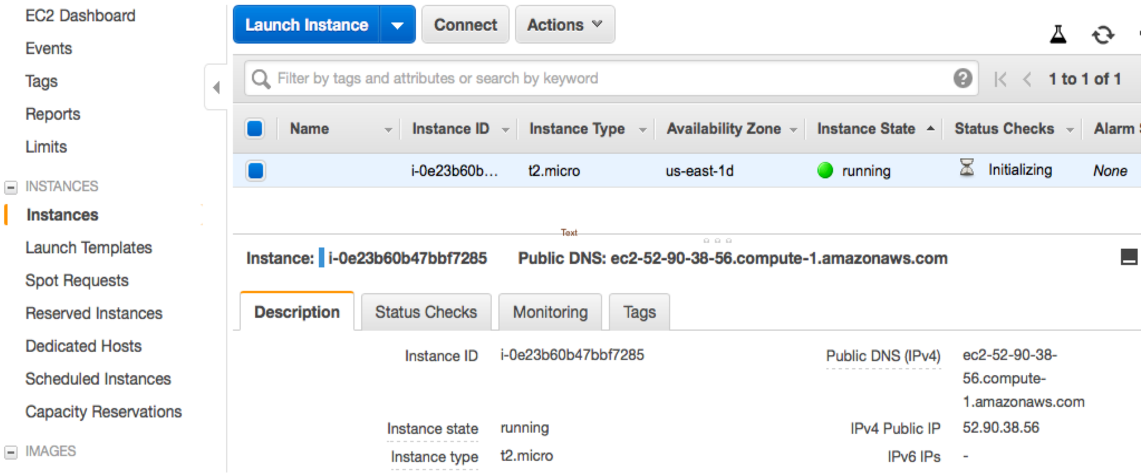


Figure 3. Launch your AMI

Step 4. For the following pages that come after “Launch”, do not make any changes and follow the instructions on the pages. The settings will automatically be selected by the specifications of the AMI. If it is not, select an instance type with **4GB of memory** and **20GB of harddrive**. Once you come to the page that asks for keypairs, select the option to “continue without a keypair” from the dropdown menu. When you are done, go back to the EC2 dashboard and select instances on the left menu. Your launched instance along with its description will appear on the right frame. Make a note of the public IP address.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
	i-0e23b60b...	t2.micro	us-east-1d	● running	Initializing	None

Instance: i-0e23b60b47bbf7285 Public DNS: ec2-52-90-38-56.compute-1.amazonaws.com

Instance ID	i-0e23b60b47bbf7285	Public DNS (IPv4)	ec2-52-90-38-56.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	52.90.38.56
Instance type	t2.micro	IPv6 IPs	-

Figure 4. Instance specification

Once the status of your instance indicates “running”, you are ready to connect to the instance remotely.

Step5. Launch your Terminal program on your local machine. Make sure that SSH is installed on it. In Windows machines, a Putty client can be used as well. Use the following command to connect to your remote machine:

```
$ ssh ubuntu@52.90.38.56
```

The numbers 52.90.38.56 indicate the IP address of the remote machine and it can be different for each instance. The username “ubuntu” is fixed, because that is the only username configured with this AMI.

Type ‘Yes’ to the question that appears and type ‘ubuntu2967’ for password. You will connect to the instance. After that, any command that you type will be executed on the remote machine instance.

Step6. Once you’re finished with the lab exercises, from the Actions menu click ‘Instance state’ and select ‘Terminate’. If the instance is not terminated, it will charge for the hours that the instance is alive. Therefore, it is very important that, once you are done, terminate the instance. Before that, save your work. Once the instance is terminated, all the changes you make in the file system will disappear.

4. Automation of Gene Sequencing Preprocessing Workflow

The gene-sequencing preprocessing workflow consists of index creation, alignment, sorting, marking duplicates and finally variant calling tasks. In the following subsections you will write a Perl script to automate these tasks.

4.1. Index creation

This step allows tools to access certain sections of the reference genome faster. The gene-sequencing input data resides under the directory "/home/ubuntu/input/". This directory includes a FASTA file for the reference genome and 4 FASTQ files(a pair for each sample) for two sample strands of the E.coli microorganism.

Step1. Create another directory called "scripts" under /home/ubuntu/ by using **mkdir** command.

Then, use **ls –l** command to see all files and directories under /home/ubuntu/ directory. If "scripts" directory is in this list, you are ready to move to the next step.

Step2. Go inside "scripts" directory by using the **cd** command. Then use **pwd** command to print the current working directory. If the output of the command is "/home/ubuntu/scripts", then, move on to the next step.

Step3. Create a script called *lab2.pl* by using **vim** command. Go into *insert mode* and then do the following tasks one by one:

- Create a variable named $path and assign the value of $ARGV[0] to it. This path will come from the command line as an argument, when you run the script. You will give the path of the input directory as an argument, which is /home/ubuntu/input. Print the path.

- Use backquotes `` to run **ls** command to list all the files that end with the extension "*.fasta"*. Use the wildcard character * to do that. Assign the returned output to a variable named $ref. Use print function to print the value of this variable to make sure that it includes the entire path of the FASTA file. Use **chomp()** function to remove any new line character from the $ref variable.

- Use the **system()** function with "samtools" to create an index file that ends with the extension .fai. samtools should take "faidx" option and $ref variable as arguments.

- Use the system() function with "bwa" to create several index files. bwa should take "index" option and $ref variable as arguments.

Step 4. Escape insert mode and exit vim text editor by saving your changes to the file.

Step 5. Run your script as follows:

```
$ perl lab2.pl /home/ubuntu/input
```

Step 6. Use "ls -l" command to list all the files in "/home/ubuntu/input". Your output should include the following index files in addition to the fasta and fastq files:

```
-rw-r--r-- 1 ubuntu ubuntu 12 Apr 30 11:34 E.coli-str.K-
12substr.MG1655.fasta.amb
-rw-r--r-- 1 ubuntu ubuntu 98 Apr 30 11:34 E.coli-str.K-
12substr.MG1655.fasta.ann
-rw-r--r-- 1 ubuntu ubuntu 4641732 Apr 30 11:34 E.coli-str.K-
12substr.MG1655.fasta.bwt
-rw-r--r-- 1 ubuntu ubuntu 29 Apr 30 11:34 E.coli-str.K-
12substr.MG1655.fasta.fai
-rw-r--r-- 1 ubuntu ubuntu 1160415 Apr 30 11:34 E.coli-str.K-
12substr.MG1655.fasta.pac
-rw-r--r-- 1 ubuntu ubuntu 2320880 Apr 30 11:34 E.coli-str.K-
12substr.MG1655.fasta.sa
```

4.2. Alignment

There are two sample strands of E.coli DNA in the input directory. We are going to assume that we do not know their names and use Perl to gather this information before starting the alignment process on the first sample.

Step 1. Open *lab2.pl* with **vim** editor and go into **Insert** Mode. Comment the lines with the system function(put # before system() call) so that the next time you run the script, the index files are not created again.

Step 2. Use “ls” command with backquotes ` ` to list all the files that end with the extension “.fastq”. Then, assign the result to an array named @samples. Use the print function to display the contents of the @samples array. You should see the paths for 4 FASTQ files. The first pair belongs to the first sample, while the second pair belongs to the second sample. We will be working on the first sample first. So the paths of the FASTQ files for it will be stored in $samples[0] and $samples[1].

Step 3. Create a variable $output1 just to include the path of the samples without its extension using the following line:

```
$output1 = substr $samples[0], 0, -9
```

This will throw away 9 characters from the end of the path and assign the new value to $output1 variable. We are going to use this variable to create output file names with different extensions. Then use chomp() function on both $sample[0] and $samples[1] to remove new line characters.

Step 4. Use the system() function with “bwa” to start the alignment process.

- The ID and SM should be assigned the name “K12”.
- -t option should be used with the number of processors the VMI is running on. If it has two cores it should be –t 2.
- $ref variable should be used for the path of the reference genome
- $samples[0] and $samples[1] should be used for the paths of the FASTQ files and $output1.raw.bam should be used as the output BAM file to be generated.

Step 5. Exit insert mode and save your file. Then run your script.

```
$perl lab2.pl /home/ubuntu/input
```

Step 6. Use ls –l command to see all the files that ends with the extension .raw.bam. Your aligned file should be listed.

```
$ls –l *.raw.bam
```

4.3. Sorting and Marking Duplicates

Now that the alignment process is over and the raw BAM file is created, it is time to sort and mark duplicates over that file. We are going to use **sambamba** tool to do both tasks.

Step1. Open *lab2.pl* with **vim** and comment all the lines with system() function call, so that the previous alignment task is not run again.

Step2. Call the system() function to run sambamba with “sort” option and give $output1.raw.bam as the second argument. A file with the extension “.raw.sorted.bam” will be created.

Step3. Call the system() function to run sambamba with “markdup” option, set the hash-table-size to 4194304, give $output1.raw.sorted.bam as the third argument and finally $output1.bam as the fourth argument. A file with the extension .bam will be created (without .raw and .sorted extensions).

Step4. Exit insert mode and save your file. Then run your perl script. You should see the BAM output files with the ls command.

```
$ ls -l /home/ubuntu/input/*.bam
total 5665320
-rw-r--r-- 1 ubuntu ubuntu  204859553 May  9 13:46 SRR1770413.bam
-rw-r--r-- 1 ubuntu ubuntu  282074721 May  9 13:31 SRR1770413.raw.bam
-rw-r--r-- 1 ubuntu ubuntu  204793669 May  9 13:44
SRR1770413.raw.sorted.bam
```

4.5. Variant Calling

The last step in the workflow is the creation of the VCF file to do variant calling analysis. We are going to use the **freebayes** tool to do that.

Step 1. Open *lab2.pl* with **vim** and comment all the lines with system() function call, so that the previous tasks are not run again.

Step 2. Run system() function with the command “freebayes”. Give $ref variable to –f option as an argument, set “--ploidy 1” and “$output1.bam > $output1.vcf” as the third and fourth argument.

Step 4. Exit insert mode and save your file. Then run your script. You should see the created VCF file in the output of the ls command.

```
$ ls -l /home/ubuntu/input/*.vcf
total 5665320
-rw-r--r-- 1 ubuntu ubuntu   17752445 May  9 14:10 SRR1770413.vcf
```

4.6. Repeat

Repeat all the steps in 4.2-4.5 for the second sample with little alterations:

- The ID and SM options should be given a different name: O104_H4.
- $samples[0] and $samples[1] should be replaced with $samples[2] and $samples[3].
- $output1 should be replaced with $output2.

Once the VCF files are created, submit your properly commented Perl script and the output of the ls command to list all the files in /home/ubuntu/input directory.

4.7. Visualize[Optional]

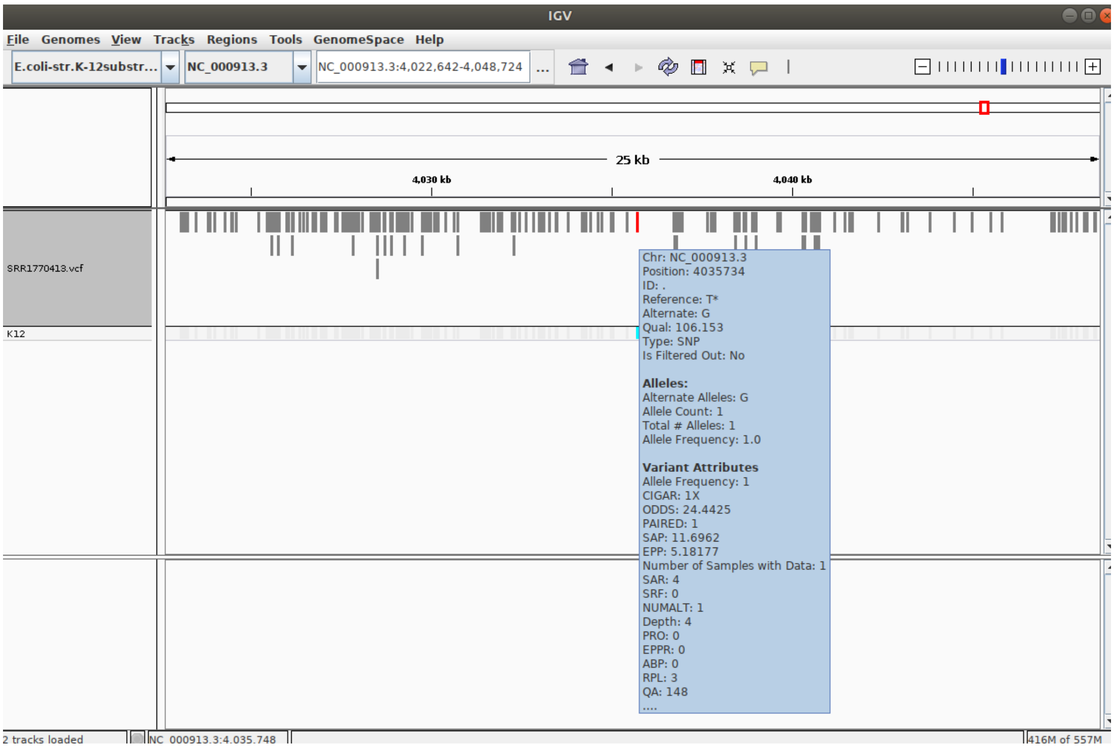
If you are running the VMI on a local machine and have access to the Desktop view, you may run the IGV program from the Downloads folder.

From the Terminal window, go to /home/ubuntu/Downloads/igv with cd command.

Run igv.sh with the following command:

```
$./igv.sh
```

A window will appear. Select “Genomes” from the Menu and load Reference FASTA file. Then select “File” from the Menu and load one of the VCF files you created from /home/ubuntu/input. Zoom in from the + sign on the top right corner and hover over one of the bars to see the type of variation(Figure 5).



```
Chr: NC_000913.3
Position: 4035734
ID: .
Reference: T*
Alternate: G
Qual: 106.153
Type: SNP
Is Filtered Out: No

Alleles:
Alternate Alleles: G
Allele Count: 1
Total # Alleles: 1
Allele Frequency: 1.0

Variant Attributes
Allele Frequency: 1
CIGAR: 1X
ODDS: 24.4425
PAIRED: 1
SAP: 11.6962
EPP: 5.18177
Number of Samples with Data: 1
SAR: 4
SRF: 0
NUMALT: 1
Depth: 4
PRO: 0
EPPR: 0
ABP: 0
RPL: 3
QA: 148
....
```

Figure 5. IGV view of VCF file

This OER material was produced as a result of the CS04ALL CUNY OER project.