Module 3 Exam

Quiz, 24 questions

24/24 points (100%)

✓ Congratulations! You passed!

Next Item

✓ 1 / 1 points

1.
How many sequences were in the genome?

7

Correct Response
This can be determined by searching for all fasta header lines in the genome file (% below denotes the terminal prompt):

```
% grep -c “^>” wu_0.v7.fas
```

✓ 1 / 1 points

2.
What was the name of the third sequence in the genome file? Give the name only, without the “>” sign.

Chr3

Correct Response
Extract all fasta headers from the genome file as in Q1, then determine the third sequence name either by visual inspection or with the command below:

```
% grep “^>” wu_0.v7.fas | head -3 | tail -1
```

✓ 1 / 1 points

3.

What was the name of the last sequence in the genome file? Give the name only, without the “>” sign.

mitochondria

Correct Response

Similarly, extract all fasta headers from the genome file and then select only the last line:

```
% grep “^>” wu_0.v7.fas | tail -1
```

✓ 1 / 1 points

4.

How many index files did the operation create?

6

Correct Response

Step 1: Suppose the current directory is ‘/media/sf_gencommand_proj3_data/’, which is where supporting data for Exam 3 are stored on the virtual machine for this course. First, we generate the bowtie2 index for the genome file provided. We create a sub-directory ‘wu_0’ to store the index, then invoke ‘bowtie2-build’:

```
% mkdir wu_0
% bowtie2-build wu_0.v7.fas /media/sf_gencommand_proj3_data/wu_0/wu_0
```

Notes: If your work directory is different, substitute it in the command above. Also, the 'bt2_index_base' in the bowtie2-build command line usage should consist of both the path to the index directory and the prefix used for the index files.As a reminder, you can use:

```
% bowtie2-build >& bowtie2-build.log
```

to save and then review the command line usage for the program and decide on the relevant parameters.Step 2: To answer the question, then simply inspect the content of the index directory and directly observe the number of index files:

```
% ls wu_0/
```

✓ 1 / 1 points

5.
What is the 3-character extension for the index files created?

bt2

Correct Response
List the files in the index directory and observe their extension (n.b., all index files have the extension 'bt2'):

```
% ls wu_0/
```

✓ 1 / 1 points

6.
How many reads were in the original fastq file?

147354

Correct Response
Each fastq record is represented on 4 lines in the 'wu_0_A_wgs.fastq' reads file. Simply count the number of lines in the file and divide by 4:

```
% wc -l wu_0_A_wgs.fastq
```

✓ 1 / 1 points

7.

How many matches (alignments) were reported for the original (full-match) setting? Exclude lines in the file containing unmapped reads.

137719

Correct Response

Step 1: We first run bowtie2 with two sets of parameters: i) the default parameters, to generate end-to-end read alignments; and ii) the '--local' option, to produce potential partial alignments of a read. For both runs, display the output as SAM alignments (option '-S'):

```
% bowtie2 -x wu_0/wu_0 -U wu_0_A_wgs.fastq -S out.full.sam
% bowtie2 -x wu_0/wu_0 -U wu_0_A_wgs.fastq -S out.local.sam --local
```

These will create the SAM files 'out.full.sam' and 'out.local.sam'. Upon completion, each run also prints a set of summary statistics on the number of reads unmapped/mapped exactly once/mapped multiple times. Save these to local files, say 'mapped.full.stats' and 'mapped.local.stats'.Step 2: To answer question Q7, we inspect the SAM files created and determine the number of alignment lines, excluding lines that refer to unmapped reads. A SAM line indicating an unmapped read can be recognized by a '*' in column 3 (chrom). Additionally, we need to exclude the SAM header:

```
% cat out.full.sam | grep -v “^@” | cut -f3 | grep -v “*” | wc -l
% cat out.local.sam | grep -v “^@” | cut -f3 | grep -v “*” | wc -l
```

✓ 1 / 1 points

8.

How many matches (alignments) were reported with the local-match setting? Exclude lines in the file containing unmapped reads.

141044

Correct Response

Step 1: We first run bowtie2 with two sets of parameters: i) the default parameters, to generate end-to-end read alignments; and ii) the '--local' option, to produce potential partial alignments of a read. For both runs, display the output as SAM alignments (option '-S'):

```
% bowtie2 -x wu_0/wu_0 -U wu_0_A_wgs.fastq -S out.full.sam
% bowtie2 -x wu_0/wu_0 -U wu_0_A_wgs.fastq -S out.local.sam --local
```

These will create the SAM files ‘out.full.sam’ and ‘out.local.sam’. Upon completion, each run also prints a set of summary statistics on the number of reads unmapped/mapped exactly once/mapped multiple times. Save these to local files, say ‘mapped.full.stats’ and ‘mapped.local.stats’.Step 2: To answer question Q7, we inspect the SAM files created and determine the number of alignment lines, excluding lines that refer to unmapped reads. A SAM line indicating an unmapped read can be recognized by a ‘*’ in column 3 (chrom). Additionally, we need to exclude the SAM header:

```
% cat out.full.sam | grep -v “^@” | cut -f3 | grep -v “*” | wc -l
% cat out.local.sam | grep -v “^@” | cut -f3 | grep -v “*” | wc -l
```

✓ 1 / 1 points

9.

How many reads were mapped in the scenario in Question 7?

137719

Correct Response

This question explores the difference between reads and alignments. As a reminder, a read may have 0 (unmapped), 1 (unique), or multiple alignments in a SAM file. The information on the number of reads in each category is contained in the ‘mapped.full.stats’ and ‘mapped.local.stats’ summary files above: simply add the numbers of reads reported to map exactly once and those reported to match multiple times.

Note that by default bowtie2 reports only one match per read, so in this case the number(s) of mapped reads at Q8 and the number(s) of alignments at Q7 will be the same.

✓ 1 / 1 points

10.

How many reads were mapped in the scenario in Question 8?

141044

Correct Response

This question explores the difference between reads and alignments. As a reminder, a read may have 0 (unmapped), 1 (unique), or multiple alignments in a SAM file. The information on the number of reads in each category is contained in the ‘mapped.full.stats’ and ‘mapped.local.stats’ summary files above: simply add the numbers of reads reported to map exactly once and those reported to match multiple times.

Note that by default bowtie2 reports only one match per read, so in this case the number(s) of mapped reads at Q8 and the number(s) of alignments at Q7 will be the same.

✓ 1 / 1 points

11.

How many reads had multiple matches in the scenario in Question 7? You can find this in the bowtie2 summary; note that by default bowtie2 only reports the best match for each read.

43939

Correct Response

Retrieve this information from the ‘mapped.{full,local}.stats’ files.

✓ 1 / 1 points

12.

How many reads had multiple matches in the scenario in Question 8? Use the format above. You can find this in the bowtie2 summary; note that by default bowtie2 only reports the best match for each read.

56105

Correct Response

Retrieve this information from the ‘mapped.{full,local}.stats’ files.

✓ 1 / 1 points

13.

How many alignments contained insertions and/or deletions, in the scenario in Question 7?

2782

Correct Response

This information is captured in the CIGAR field, marked with ‘D’ and ‘I’, respectively:

```
% cut -f6 out.full.sam | grep -c “[I,D]”
% cut -f6 out.local.sam | grep -c “[I,D]”
```

✓ 1 / 1 points

14.

How many alignments contained insertions and/or deletions, in the scenario in Question 8?

2614

Correct Response

This information is captured in the CIGAR field, marked with ‘D’ and ‘I’, respectively:

```
% cut -f6 out.full.sam | grep -c “[I,D]”
% cut -f6 out.local.sam | grep -c “[I,D]”
```

✓ 1 / 1 points

15.

How many entries were reported for Chr3?

360295

Correct Response

Step 1: Start by converting the SAM file to BAM format as indicated:

```
% samtools view -bT wu_0.v7.fas out.full.sam > out.full.bam
then sorting it:
% samtools sort out.full.bam out.full.sorted
```

This will create the BAM file ‘out.full.sorted.bam’, which will be used to determine sites of variation.Step 2: Determine candidate sites using ‘samtools mpileup’, providing the reference fasta genome (option ‘-f’) and using the option ‘-uv’ to report the output in uncompressed VCF format:

```
% samtools mpileup -f wu_0.v7.fas -uv out.full.sorted.bam > out.full.mpileup.vcf
```

Step 3: Count the number of entries in the VCF file located on Chr3. The chromosome information is listed in column 1, once we filter out the header lines (marked with “#”):

```
% cat out.full.mpileup.vcf | grep -v “^#” | cut -f1 | grep -c “^Chr3”
```

Potential pitfalls: It is critical to sort the BAM file before analyzing it with samtools.

✓ 1 / 1 points

16.

How many entries have ‘A’ as the corresponding genome letter?

1150985

Correct Response

This information is contained in column 4:

```
% cat out.full.mpileup.vcf | grep -v “^#” | cut -f4 | grep -P “^A$”
```

where “^A$” tells ‘grep’ to look for patterns that consist exclusively of ‘A’ (i.e., between the start ‘^’ and end ‘$’ of the line).

✓ 1 / 1 points

17.

How many entries have exactly 20 supporting reads (read depth)?

1816

Correct Response

Read depth is indicated by the ‘DP=’ field in column 8:

```
% cat out.full.mpileup.vcf | grep -v “^#” | grep -c “DP=20;”
```

✓ 1 / 1 points

18.

How many entries represent indels?

1972

Correct Response

This information is marked with the keyword ‘INDEL’ in the variant line:

```
% cat out.full.mpileup.vcf | grep -v “^#” | grep -c INDEL
```

✓ 1 / 1 points

19.

How many entries are reported for position 175672 on Chr1?

2

Correct Response

This information is stored in columns 1 and 2 of the VCF file:

```
% cat out.full.mpileup.vcf | grep -v “^#” | cut -f1,2 | grep Chr1 | grep 175672
```

then select only the entries corresponding to position 175672.

Module 3 Exam

Quiz, 24 questions

24/24 points (100%)

✓ 1 / 1 points

20.

How many variants are called on Chr3?

398

Correct Response

Step 1: First re-run ‘SAMtools mpileup’ with the BCF output option ‘-g’:

```
% samtools mpileup -f wu_0.v7.fas -g out.full.sorted.bam > out.full.mpileup.bcf
```

then call variants using ‘BCFtools call’ with the multi-allelic caller (option ‘-m’), showing only variant sites (‘-v’) and presenting the output in uncompressed VCF format (‘-O v’), as instructed:

```
% bcftools call -m -v -O v out.full.mpileup.bcf > out.final.vcf
```

Step 2: To answer the question, we count all reported variants that show ‘Chr3’ in column 1:

```
% cat out.final.vcf | grep -v “^#” |  cut -f1 | sort | uniq -c | grep “Chr3”
```

✓ 1 / 1 points

21.

How many variants represent an A->T SNP? If useful, you can use ‘grep –P’ to allow tabular spaces in the search term.

392

Correct Response

This information is stored across columns 2 and 3. An A->T SNP would be represented as an ‘A’ in column 2 and a ‘T’ in column 3:

```
% cat out.final.vcf | grep -v “^#” | cut -f4,5 | grep -P “^A\tT$” | wc -l
```

Module 3 Exam

Quiz, 24 questions

24/24 points (100%)

✓ 1 / 1 points

22.

How many entries are indels?

320

Correct Response

Similar to Q14, indels are marked with ‘INDEL’:

```
% cat out.final.vcf | grep -v “^ #” | grep -c INDEL
```

✓ 1 / 1 points

23.

How many entries have precisely 20 supporting reads (read depth)?

2

Correct Response

Similar to Q13:

```
% cat out.final.vcf | grep -v “^#” | grep -c “DP=20;”
```

✓ 1 / 1 points

24.

What type of variant (i.e., SNP or INDEL) is called at position 11937923 on Chr3?

SNP

Correct Response

Similar to Q15:

```
% cat out.final.vcf | grep –v “^#” | cut -f1-5 | grep Chr3 | grep 11937923
```

then inspect the reference and variant sequences in columns 4 and 5 to determine the type of variant.