Recruitment tasks

Please have a look at the recruitment tasks. Feel free to use the approaches you find most convenient and suitable. In case you do not know how to perform some task you still can describe how you would approach the problem. Our goal is to understand the way you think and solve problems.

P.S. make a separate repository on your git account and commit and push your solutions after each performed task.

There are 4 main tasks, the order doesn't matter. Time limitation 4-6 hours. If you will not fit in time, feel free to commit your solutions after - we would definitely have a look at them as well

Good luck!

Warm up exercises

a. Given the string x, write a function that will return a two-element list where the first element will be the string x with letters in even places changed to uppercase. The second item in the list will be uppercase in odd places.

Example:

fun ("abcdef") -> ['aBcDeF', 'AbCdEf']

Assume that the input to the function contains only letters.

Commit and push!

b. Write a function that takes a string as an argument. As output, it will return the number of letters in the string more than once. The code should not be case-sensitive.

Example:

function ("ABBA") -> 2 (a and b occur 2 times)

function ("aBcbA") -> 2 (a and b is repeated; case is ignored)

function ("RhabarbArka") -> 3 (a, bir)

Commit and push!

Task 1 - Preprocessing

Description

Having a bam file mapped to GRCh37 extract the reads corresponded to chromosome 1 and realign them to the chromosome 1 using hg38 reference.

Supplimentary data

miniMNM00065.bam

Output

1.) Bam file mapped to hg38;

2.) Description of your workflow;

Task 2 - Structural variants hands-on

Description

Having a somatic structural variants generated by manta and stored accordindg to vcf specifications please write the code(R or Python) to answer the following questions:

1.) Count the total number of variants represented in the form of breakends.

2.) Make a boxplots of the deletion length per each chromosome.

3.) Count how many variants failed to pass the filtering. Make a piechart of most frequent reasons to fail.

4.) Find the variant with the widest confidence interval around POS;

5.) What type of stractural variant represented by ID MantaBND:28842:0:1:0:0:0:0

Supplimentary data:

tumor_vs_normal.manta.somatic.vcf.gz

Output:

1.) script or notebook(jupyter or r-markdown document) with the workflow

Task 3 - Single nucleotide variants hands-on

Description

Having SNVs generated by strelka and stored accordindg to vcf specifications. Please write the code(R or Python) to answer the following questions:

1.) Discard all the variants which are failed to pass the filtering

2.) Annotate filtered variants with the SNPeff using hg19 database

3.) Count the variants that change the protein encoded by the gene in which the variant is located. The list of possible consequences of changes

4.) List a genes which are affected with the predicted Loss of function effect.

5.) (Additional, not obligatory) Make a short EDA(explaratory data analysis) of the assigned effect predictions

Supplimentary data:

tumor_vs_normal.strelka.somatic.snvs.vcf.gz

Output:

1.) Filtered and annotated vcf file

2.) Script or notebook(jupyter or r-markdown document)

3.) (optional) PDF with EDA

Task 4 - Variant Allele Frequency

Description

Strelka is a variant calling algorithm that does not provide a commonly used somatic variant statistic - VAF (Variant Allele Frequency). The manual of the tool states that VAF can be computed in a following way

Somatic SNVs:

refCounts = Value of FORMAT column $REF + “U” (e.g. if REF="A" then use the value in FOMRAT/AU)
altCounts = Value of FORMAT column $ALT + “U” (e.g. if ALT="T" then use the value in FOMRAT/TU) tier1RefCounts = First comma-delimited value from $refCounts
tier1AltCounts = First comma-delimited value from $altCounts
Somatic allele freqeuncy (VAF) is $tier1AltCounts / ($tier1AltCounts + $tier1RefCounts)

Somatic indels:

tier1RefCounts = First comma-delimited value from FORMAT/TAR tier1AltCounts = First comma-delimited value from FORMAT/TIR Somatic allele freqeuncy is (VAF) $tier1AltCounts / ($tier1AltCounts + $tier1RefCounts)

Supplied data

- VCF with somatic SNV variants.

T1_vs_N1_head.strelka.somatic.snvs.norm.vcf.gz

- VCF with somatic indel variants:

T1_vs_N1_head.strelka.somatic.indels.norm.vcf.gz

Note

Note that each file contains calls for two samples: NORMAL and TUMOR

Output

Easy mode: Tab-delimited file with variants with VAF for TUMOR and NORMAL samples

Hard mode: Modified VCF file with FORMAT/VAF field calculated as specified above for TUMOR and NORMAL samples.