Cześć!

Przed Tobą kilka zadań. Rozwiąż ile możesz zapisując skrypt/program, i grafiki/tabelki w katalogu w Twoim repozytorium GitHub. Nie uploaduj tam plików VCF większych niż 10MB. Do niektórych zadań będziesz potrzebował ściągnąć plik ok 300Mb. Zacznij ściągać już teraz aby nie tracić czasu później.

Zadania są niezależne od siebie, nie musisz ich rozwiązywać po kolei. Sposób rozwiązania jest dowolny - można skorzystać z istniejących narzędzi lub napisać własny skrypt/program. Dwa pierwsze to zadania na rozgrzewkę, a zadania z gwiazdką mogą pochłonąć więcej czasu. Pamiętaj, aby commitować (i pushować) nie rzadziej niż po każdym zadaniu (przypomnienia o tym są również poniżej).

Połamania klawiatury!

Zadania rozgrzewkowe

1. Mając ciąg znaków *x*, napisz funkcję, która zwróci dwuelementową listę w której element pierwszy będzie ciągiem znaków *x* , gdzie litery na parzystych miejscach zamienione będą na wielkie litery. Drugi element listy będzie miał wielkie litery na miejscach nieparzystych.
Przykład:
```
fun(“abcdef”) -> [‘aBcDeF’, ‘AbCdEf’]
```

Przyjmij, że input do funkcji zawiera tylko litery.

Commit and push!

2. Napisz funkcję, która jako argument przyjmuje ciąg znaków. Jako output, zwróci liczbę liter występujących w ciągu więcej niż raz. Kod nie powinien być wrażliwy na wielkość liter.
Przykład:
```
funkcja(“ABBA”) -> 2 (a i b występują po 2 razy)
funkcja(“aBcbA”) -> 2 (a i b się powtarza; wielkość liter jest
ignorowana)
funkcja(“RabarbArka”) -> 3 (a,b i r)
```

Commit and push!

Zadania główne

Zadania 3-6 wykorzystują jako dane wejściowe plik w formacie VCF skompresowany bgzip'em: vcf.gz [ok. 300Mb]. W pliku są adnotacje wariantów wygenerowane programami SnpEff/SnpSift. Jeżeli przetwarzanie pliku będzie stwarzało problemy ze względu na jego rozmiar - znajdź kreatywne rozwiązanie.

3. Stwórz plik VCF, który będzie zawierał wszystkie warianty z pliku wejściowego zawierające się pomiędzy `chr12:112,204,691` a `chr12:112,247,789`. Komendę/program oraz wynikowy plik VCF umieść w repozytorium.

Commit and push!

4. Narysuj histogramy długości insercji i delecji w pliku wejściowym dla każdego z chromosomów. Rysunek i tabelkę umieść w repozytorium.

Commit and push!

5. Z pliku wejściowego wybierz warianty, dla których pole `FILTER` zawiera wartość `PASS`. Spośród tych wariantów policz ile jest heterozygotycznych (genotyp `0/1`) z częstością występowania w populacji (Allele Frequency) mniejszą od 0.01. Do selekcji wariantów o niskiej częstości wykorzystaj adnotację `INFO:GoNLv5_AF`.

Commit and push!

6. Na podstawie wariantów w pliku wejściowym wykonaj wykres słupkowy, w którym na osi x będą wszystkie chromosomy autosomalne, a na osi y średnie pokrycie (depth of coverage) wszystkich wariantów występujących na danym chromosomie. Jeżeli napotkasz wartości brakujące (“.”), potraktuj je jako 0. Średnie pokrycia wszystkich chromosomów zapisz również w tabelce (.csv, .tsv, lub Excel).

Commit and push!

Task 8

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

- T1_vs_N1_head.strelka.somatic.snvs.norm.vcf.gz - VCF with somatic SNV variants.
- T1_vs_N1_head.strelka.somatic.indels.norm.vcf.gz - VCF with somatic indel variants.

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.

To już koniec zadań, które dla Ciebie przygotowaliśmy.

Przykładowe rozwiązania udostępnimy na początku przyszłego tygodnia.