Teste Admissional

1. **O que é um arquivo do tipo FASTQ e como posso verificar se um FASTQ é válido?**

R- FASTQ são arquivos de textos simples que, na bioinformática, são normalmente usados para armazenar sequências genéticas em conjunto com a qualidade de cada um dos seus nucleótidos. Seu uso foi disseminado com o advindo dos sequenciadores de nova geração, principalmente da illumina, pois sua estrutura permite o armazenamento dessas sequências em conjunto com a qualidade de suas bases em arquivo compacto, ocupando menos espaço de armazenamento.

Para que os arquivos do tipo FASTQ sejam considerados válidos, devem possuir uma estrutura básica que é composta por 4 linhas. A primeira linha inicia com o caracter marcador obrigatório “@”, seguido por uma descrição opcional que não tem limite de tamanho. Na segunda linha do arquivo, se encontra a sequência propriamente dita, que deve ser escrita preferencialmente de forma continua, ou seja, sem quebra de texto. A terceira linha deve conter o caracter marcador “+” e pode ser seguida de uma descrição opcional que, quando presente, comumente é a mesma descrição usada na primeira linha. Por fim, a quarta linha desse arquivo é composta por um conjunto de caracteres do tipo ASCII que são usadas para codificar os valores de qualidade das bases das sequências diminuindo o tamanho final do arquivo. Essa linha deve ter, obrigatoriamente, o mesmo tamanho da segunda linha, pois cada caracter representa o valor de qualidade de uma base da sequência. Por essa razão, é possível a presença dos caracteres @ e + nessa linha, visto que os programas devem antes de interpretar o início de um novo arquivo e/ou linha, na presença desses marcadores, contar se ambas as linhas tem o mesmo tamanho.

1. **Quais são as etapas mais comuns de um pipeline de bioinformática de NGS para DNAseq e quais ferramentas podem ser usadas em cada etapa?**

R- Quando trabalhamos com dados de NGS para o sequenciamento de DNA, podemos dividir o pipeline de bioinformática em quatro grandes etapas, são elas: Avaliação da qualidade (Pré-processamento), Alinhamento (montagem do genoma), Chamada de variantes e Anotação das variantes. Cada uma dessas etapas, tem uma função importante para a correta identificação da sequência alvo e sua utilização na investigação de mutações nos organismos.

A etapa de pré-processamento é responsável pela análise da qualidade das bases e do tamanho dos reads dos arquivos FASTQ, após o sequenciamento. Entre os principais softwares que podemos utilizar temos o Fastp, trimmomatic e cutadapt para remoção de adaptadores e de bases com baixa qualidade e o Fastp e FastQC para a avaliação dos reads após o tratamento.

Na etapa de montagem do genoma o objetivo é realizar a sobreposição dos reads sequenciados para que sejam geradas sequências maiores, chamadas de contings e, em seguida, esses contigs são organizados em Scaffolds. Essa montagem pode ser realizada de duas formas: por referência e *de novo*. A montagem *de novo*, é realizada por meio de alinhamento semi-global utilizando *Mers* (grafos de brujin). Os principais softwares para a realização dessa montagem são Velvet, SOAPdenovo e Spades. Essa estratégia é escolhida, principalmente, quando não temos genomas de referências do nosso organismo alvo. Para a realização do nosso pipeline, iremos continuar utilizando a montagem por referência, que utiliza softwares de mapeamento para alinhar nossos reads a um genoma montado e conhecido, chamado de genoma de referência, que transforma nosso arquivo FASTQ em um arquivo SAM/BAM. Os principais softwares para a realização dessa etapa são o BWA-MEM e o Bowtie2. Após esse mapeamento, devemos realizar a classificação do nosso arquivo BAM gerado e marcar as duplicatas, os principais softwares utilizados nessa etapa são o GATK e o SAMTOOLS.

Após mapeamento, nosso arquivo BAM é submetido a um software que irá comparar o genoma de referência com nossa sequência de interesse e identificar as variantes entre elas, fornecendo como output um arquivo de VCF. Essa etapa é conhecida como chamada de variantes e os principais softwares que utilizamos são o Freebayes e o GATK.

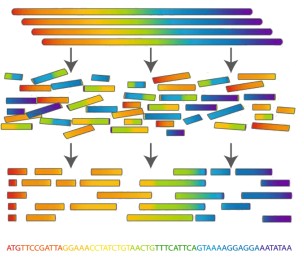
Com o arquivo VCF gerado a última etapa é a realização da anotação dessas variantes, para que possamos identificar possíveis mutações (SNP e/ou INDELS) que tenham interesse patogênico. Essa etapa pode ser realizada pelo software SNPEFF.

1. **Considerando que o alinhamento de sequências biológicas é uma das atividades mais recorrentes e importantes na área de bioinformática. Comente sobre os algoritmos computacionais de alinhamentos mais utilizados e em que cenários uns são mais indicados do que outros?**

R- O alinhamento de sequências biológicas pode ser classificado em alinhamento global, semi-global e local, todos podem ser feitos entre duas sequências (simples ou par-a-par) e entre três ou mais sequências (Múltiplo). O alinhamento global pode ser chamado de Needlman-Wunsch estende-se por toda a sequência não levando em consideração a divergência entre as sequências analisadas. O alinhamento Semi-global não penaliza gaps quando criados nas pontas das sequências e o alinhamento local (Smith-Waterman) busca a maior pontuação em um par de subsequências, ou seja, busca a maior similaridade em apenas um fragmento da sequência total.

Cada um desses algoritmos tem seu prós e contras, o alinhamento global só é indicado quando estamos trabalhando com sequências muito similares ou homólogas, de preferência com um tamanho em pares de bases semelhante, atualmente os principais softwares de alinhamento global são o ClustalW, o MAFFT e o Muscle.

O alinhamento semi-global é ideal quando estamos realizando um processo de montagem de genoma, pois nos permite encontrar sobreposições entre nossas sequências, contudo a depender do tamanho e da quantidade dessas sequências, se faz necessário utilizar um alinhamento prévio com sequências menores de tamanho “K” para diminuir o tempo de execução desse alinhamento.Já o alinhamento local é indicado quando estamos comparando sequências de tamanho diferentes, pois irá redirecionar o esforço para buscar uma região ou bloco com alta similaridade. O principal software que utiliza esse tipo de alinhamento é o BLAST.

1. **[Desafio Técnico] Todos os seres humanos compartilham aproximadamente 99,9% dos mesmos nucleotídeos em seu genoma e até a mesma ordem como são apresentados. Portanto, se apenas soubermos alguns genomas completos de uma espécie podemos ter os principais componentes para identificar o genoma de toda a espécie. Determinar o genoma completo de um organismo (chamado de sequenciamento de genoma) é uma das principais tarefas em bioinformática. Infelizmente, nós não possuímos tecnologia microscópica que consiga realizar um zoom a nível de nucleotídeo e determinar a sequência de nucleotídeos de um genoma, um por vez. Entretanto, pesquisadores podem aplicar métodos bioquímicos para gerar e identificar pequenos fragmentos de DNA que chamamos de reads. Após obter uma grande coleção de reads de múltiplas cópias do mesmo genoma, o objetivo é reconstruir o genoma a partir destes pequenos fragmentos** d**e DNA. Este processo é chamado de fragment assembly.**

**Entendido o contexto do nosso problema, vamos ao desafio: Para uma coleção de strings, uma string de maior dimensão contendo todos as strings menores como substrings é chamado de superstring. Para fins de simplicidade, vamos considerar que a superstring de menor dimensão possível sobre uma coleção de reads é um possível candidato a cromossomo (cromossomo é formado por uma sequência de bases).**

**Dado n strings de DNA cujo o tamanho não ultrapassem mais de 1000 bases e que a partir destes n reads seja possível construir um cromossomo inteiro colando pares de reads que se sobrepõem mais do que metade do seu tamanho, o objetivo é retornar a superstring de menor dimensão possível contendo todas as strings (isto equivale neste problema a conseguirmos reconstruir um cromossomo).**

**EXEMPLO de Entrada (input.txt):**

**ATTAGACCTG CCTGCCGGAA AGACCTGCCG GCCGGAATAC**

**EXEMPLO de Saída (output.txt):**

**ATTAGACCTGCCGGAATAC**

**Escreva um script em uma linguagem de sua preferência para receber um arquivo input.txt e a saída será um arquivo output.txt contendo a superstring.**

**PS1: Estaremos testando seu código com até 1000 strings de DNA, leve em consideração quesitos de desempenho também e na complexidade da sua solução. O importante neste problema é discutirmos como você solucionou o problema em questão e as escolhas para a construção da sua solução.**

1. **[DESAFIO TÉCNICO]** Neste desafio você será solicitado para construir um pipeline de chamada de variantes de bioinformática para NGS para detecção de SNVs e INDELs de uma amostra sequenciada na plataforma Illumina (DNASeq). O objetivo é montar o pipeline, executar o pipeline com uma amostra de exemplo e nos enviar os resultados encontrados e responder alguns questionamentos sobre o pipeline. O desafio se encontra no nosso repositório público github , com mais instruções e informações sobre como realizar o teste e entregar o resultado.

Link do Teste: https://github.com/Varstation/bioinfotest

PS: É necessário que você tenha uma conta no github ou se não preferir pode enviar o resultado completo em um arquivo zipado e compartilhar conosco .

1. **[DESAFIO TÉCNICO]** Um dos processos que executamos em nosso dia-a-dia é o processo de comparação de resultados das chamadas de variantes de nossos pipelines contra um dataset de referência de variantes verdadeiras (truth sets). Queremos garantir que nossos pipelines tenham boa capacidade de detecção de variantes e com baixo número de falso positivos. Neste desafio você será solicitado para realizar o processo de comparação de resultados de variantes SNVs e INDELs de uma amostra sequenciada na plataforma Illumina (DNASeq). O objetivo é que você execute a comparação de resultados a partir do **vcf resultante da etapa anterior** com o vcf que vamos disponibilizar a você (truthset). Para isto você irá utilizar a ferramenta hap.py da Illumina (https://github.com/Illumina/hap.py) com o fasta provido pela etapa anterior (hg19.fasta).

Vamos solicitar que você nos envie os resultados do arquivo \*.summary.csv e faça um comentário explicando os resultados apresentados de forma resumida. A nossa avaliação é na sua capacidade de síntese de avaliação e não em obter o resultado 100%.

Você poderá enviar o resultado em um arquivo .zip contendo apenas o summary.csv e um pequeno documento em formato txt contendo a sua avaliação dos resultados.

DICAS:

1 - Para te auxiliar providenciamos a imagem Docker do hap.py (dos autores da ferramenta) para facilitar a instalação e execução do procedimento . Pode usar o comando a seguir:

*docker run -it -v ${PWD}:${PWD} pkrusche/hap.py /opt/hap.py/bin/hap.py*

*${PWD}/VCF\_TRUTH\_SET.vcf.gz ${PWD}/VCF.vcf.gz -f ${PWD}/path\_do\_bed -r*

*${PWD}/path\_do\_fasta\_hg19 -o ${PWD}/prefixo\_saida --engine=vcfeval*

2- Foque nas seguintes métricas na sua interpretação, são os mais importantes para este teste: TRUTH.TOTAL , METRIC.Recall , Metric.Precision e METRIC.F1\_Score

3 – O vcf para ser processado pelo hap.py precisa ser comprimido e indexado usando ferramentas como bgzip e tabix.

4 – O bed de interesse das regiões que queremos avaliar está em anexo no zip truth.py

4 – O truthset VCF já está comprimido e indexado e está disponível no arquivo zip truth.zip

1. **[DESAFIO TÉCNICO]** Neste desafio , queremos entender seu conhecimento sobre análise e identificação de sequências de DNA. Temos 3 amostras recebidas de um laboratório que quer identificar devido a um surto local de algumas espécies de vírus. Necessitamos que você identifique quais vírus estão presentes nestas amostras. Para cada amostra responda as perguntas a seguir:

Amostra\_01. Qual vírus conseguiu identificar? Qual o seu genótipo? Descreva brevemente as etapas do pipeline utilizado.

A sequência que teve a maior identidade foi uma cepa de SARS-CoV-2 que apresentou um percentual de identidade 99.98% e cobertura 99% com genótipo ILMD2100705. Para realização dessa análise usamos o Fastqc para avaliar a sequência e o fastx\_trimmer e o sickle para “trimar” as sequências por tamanho e qualidade, em seguida usamos o spades com a linha de comando “spades.py -o spadesout -1 1.trimmed/R1.fastq.gz -2 1.trimmed/R2.fastq.gz -s 1.trimmed/s.fastq.gz -t 4” para realizar a montagem *de novo.* Por fim realizamos um blast com a sequência contra o banco de dados de vírus para identificar a sequência.

Amostra\_02. Qual vírus conseguiu identificar? Qual o seu genótipo? Descreva brevemente as etapas do pipeline utilizado.

Amostra\_03. Qual vírus conseguiu identificar? Qual o seu genótipo? Descreva brevemente as etapas do pipeline utilizado.

Dados das amostras: https://drive.google.com/drive/folders/16CiOLTnS1H04RCK3qaMxcxDjvIGgJ-Wk

PS: É necessário que você tenha uma conta no github com os resultados se desejar ou se não preferir pode enviar o resultado completo em um arquivo zipado e compartilhar conosco .