Processamento das Sequências Obtidas com Equipamentos de Sequenciamento de Nova Geração (NGS)

Diego Frias e Mauricio Souza Menezes

[1]Departamento de Ciências Exatas e da Terra, Campus I
Universidade do Estado da Bahia (UNEB)
Salvador, Bahia, Brasil.

mauriciosm95@gmail.com

Resumo. *Este documento tem como objetivo descrever o contexto da pesquisa, com vistas a definir melhor o escopo do projeto. Possíveis questões de pesquisa são elaboradas ao longo do texto, para uma posterior seleção e generalização.*

Abstract. *This document aims to describe the context of the research, in order to better define the scope of the project. Possible research questions are elaborated throughout the text, for later selection and generalization.*

1. Introdução

Começamos listando algumas especificidades do NGS:

- Pode ser aplicado para sequenciar DNA (dupla hélice) ou RNA (hélice única) mas, não os dois ao mesmo tempo.
- O DNA/RNA na amostra é fragmentado de forma aleatória, acoplando-se adaptadores a um ou os dois extremos de cada fragmento. Estes adaptadores fixam o fragmento à cela de sequenciamento e são parcial ou totalmente sequenciados, o que exige uma etapa inicial de processamento para retirar os adaptadores.
- A amostra pode conter diversos tipos de material genético, além do DNA/RNA que deseja ser sequenciado, como é o caso de amostras de sangue para sequenciamento de vírus, nas quais pode haver outros vírus e DNA/RNA das distintas células do sangue. De acordo com isto é necessário filtrar todas as sequências que não são do vírus em estudo.
- Nas amostras meta-genômicas existem fragmentos de DNA/RNA de genomas/-transcriptomas de múltiplos organismos, o que requer um processamento diferenciado. Neste trabalho não abordaremos sequenciamento metagenômico.
- Podemos diferenciar dois tipos de problemas de sequenciamento:
    1. Sequenciamento de novas espécies, em cujo caso é preciso construir o genoma dela a partir da sobreposição dos fragmentos. Este problema é conhecido como *de novo assembly*. Neste problema, uma vez montado o genoma, é necessário fazer a anotação do mesmo. A anotação consiste em identificar o início e fim de cada gene. No caso dos eucariotos, como os genes podem conter introns, é necessário identificar o início e o fim de cada intron, para poder concatenar os exons (regiões codificantes) e obter as sequências de aminoácido das proteínas codificada pelos genes. No caso de procariotos o problema é mais simples pois os genes não possuem (em geral) introns. No caso dos vírus, que possuem genoma muito compacto,

muitos dos genes são encadeados em quadros abertos de leitura (Open Reading Frame - ORF) que codificam vários genes. As ORFs são transcritas e traduzidas de uma única vez, ocorrendo a separação das proteínas após a tradução num processo conhecido como clivagem da cadeia polipeptídica. Este processo de clivagem é controlado por proteínas não estruturais (NSP) codificadas por outras ORFs que são traduzidas primeiro para poder produzir as proteínas estruturais virais. O resultado deste tipo de projeto de sequenciamento é um genoma anotado. Este genoma anotado pode ser usado como referência no próximo tipo de projeto de sequenciamento, descrito a seguir.

2. Sequenciamento de espécies conhecidas dispondo de um genoma de referência anotado. Este é o caso nos projetos de sequenciamento genômico de vírus para estudar a evolução dos mesmos e o surgimento de novas espécies (vigilância biomolecular de patógenos). O resultado imediato destes projetos de sequenciamento é uma listagem de mutações (mudanças pontuais de aminoácidos) entre o genoma sequenciado e o de referência. Os arquivos com a listagem das mutações tem extensão .vcf do termo em inglês *Variant Calling File*. Paralelamente, estes projetos adicionam novos genomas anotados à base de dados, o que permite fazer uma análise filogenética das cepas sequenciadas em diferentes momentos e localidades, o que é uma poderosa ferramenta de análise da evolução viral. Na figura 1 mostramos o pipeline genérico e nas figuras 2 e 3 dois pipelines mais detalhados.

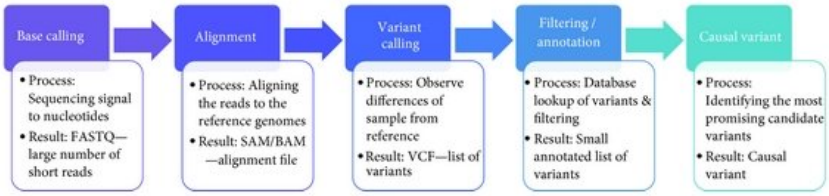

Figura 1. Pipeline simplificado

3. O de *novo assembly* é um problema NP-hard, mas tem sido desenvolvidos métodos baseados em grafos (Eulerianos `https://en.wikipedia.org/wiki/Eulerian_path`, Hamiltonianos `https://en.wikipedia.org/wiki/Hamiltonian_path` e de Bruijn (`https://en.wikipedia.org/wiki/De_Bruijn_graph` e `https://web.archive.org/web/20141030124239/http://www.homolog.us/Tutorials/index.php?p=2.1&s=1`)), bastante eficientes.
   Na figura 4 mostramos o pipeline genérico de um de novo assembly.
   Depois da montagem do genoma (contigs) se procede à anotação dos mesmos: identificação das sequências de DNA/RNA que codificam as proteínas e os RNAs funcionais. Na figura 5 mostramos o pipeline genérico desta fase. Observe o uso de métodos ab-initio, como o Universal Feature Method (UFM).

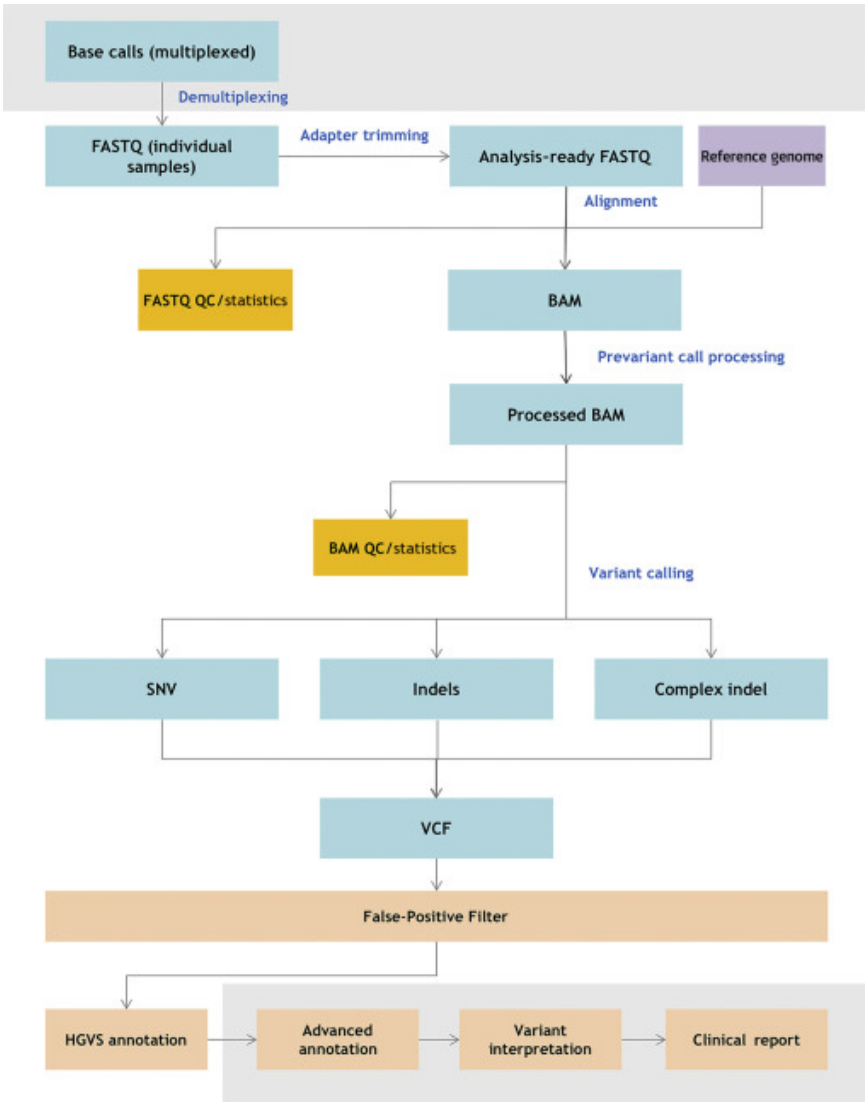

Figura 2. Typical variant calling bioinformatics pipeline

Comentários:

(1) Pelo que tenho visto, todas as ferramentas para *de novo assembly* trabalham com sequências de nucleotídeos, ou seja, alinham sequências codificantes e não codificantes. Então precisa ver se alguma ferramenta trabalha com sequências de aminoácidos (alinhando apenas sequências codificantes), porque UFM traduz os fragmentos de nucleotídeos para fragmentos de aminoácidos. Neste cenário, que percentual dos fragmentos sequenciados com diferentes técnicas NGS são considerados codificantes pelo UFM para diferentes virus, pode ser uma questão de pesquisa.

(2) Uma peculiaridade de trabalharmos com UFM é que o *de novo assembly* pode ser "focado"apenas nos genes, descartando-se automaticamente as regiões não codificantes do genoma sendo sequenciado. Quanto mais barato e efetivo seria? Acho que não vai ser uma das suas questões de pesquisa. Será de um trabalho futuro.

4. O alinhamento de fragmentos de comprimento médio f, contra um ge-

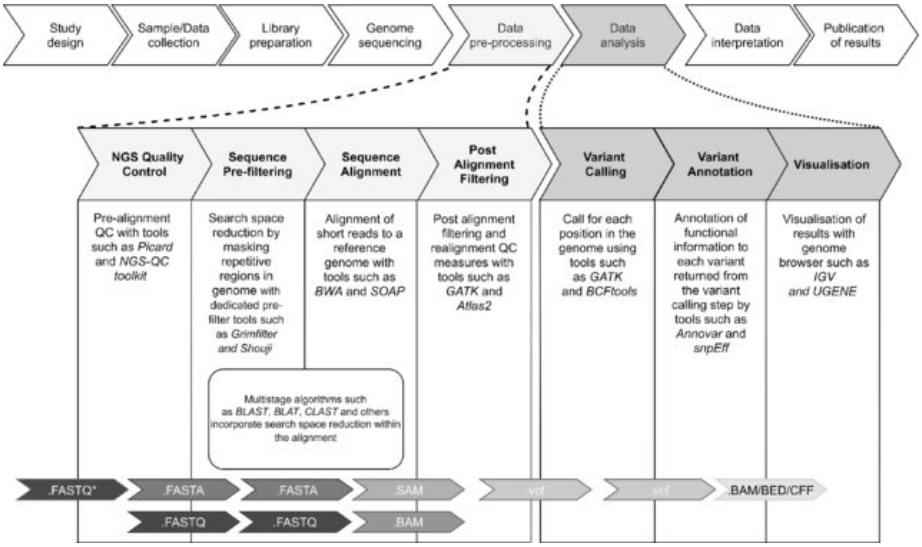

Figura 3. Typical variant calling bioinformatics pipeline composed of steps following NGS sequencing leading to the visualization of data is presented in the middle panel. The variant calling bioinformatics pipeline is contained within the data pre-processing and data analysis stages of a much larger bioinformatics-based research study as illustrated in the upper panel. Data file formats at each step are presented in the lower panel (Al Kawam et al., 2017; Lightbody et al., 2019). *Platform-specific raw sequence output either .BAM or .FASTQ or .HDF5 (NCBI, 2019).

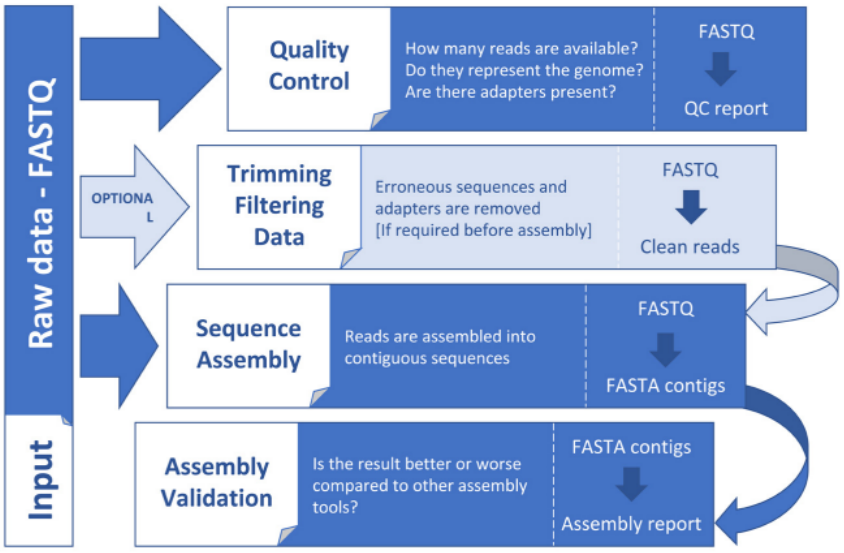

Figure 2. **General steps in a genome assembly workflow.** Input and output data are indicated for each step.

Figura 4. Pipeline genérico do *de novo assembly*

noma de referência de comprimento g é o método para processar sequências de espécies conhecidas. Neste contexto, uma das suas perguntas de pesquisa será: quais métodos são os mais utilizados e qual o custo computacional dos mesmos em função de f e g.

Comentários:

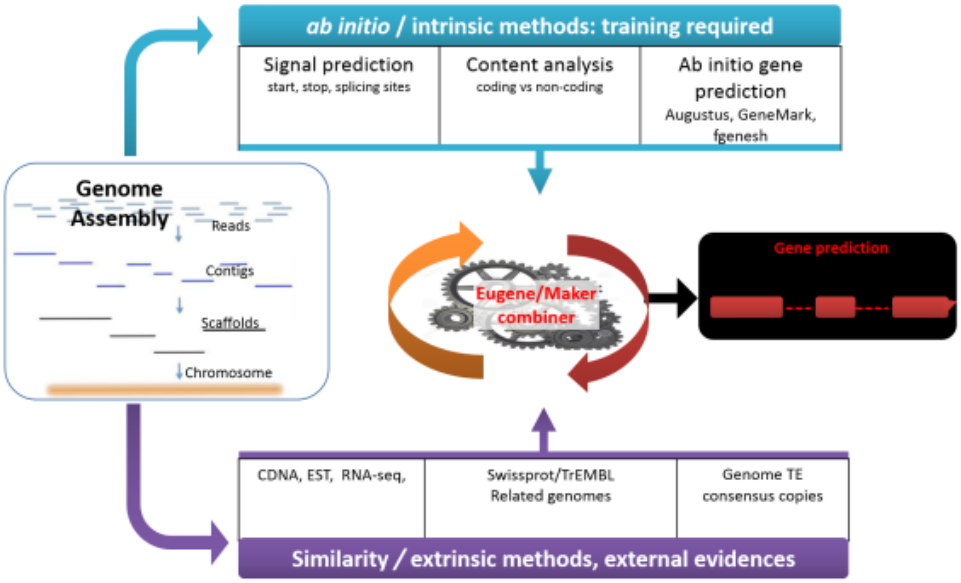

Figure 3. **Simplified Illustration of a structural genome annotation using Combiners.** On the left, the diagram shows a typical assembly process. At the end of the process, scaffolds or chromosomes ready to be annotated are obtained. These scaffolds are then annotated using two different methods. The first method is called *ab-initio* and requires a known set of training genes. Once the *ab initio* tool has been trained it can be used to predict other similarly structured genes. The second similarity-based approach relies on experimental evidence such as CDSs, ESTs, or RNA-seq to build gene models. Combiners (such as Maker or Eugene) can then incorporate all of these results, eliminate incongruences, and present gene models best supported by all methods.

Figura 5. Pipeline genérico da anotação de genomas

(1) Quando um fragmento é alinhado ao genoma de referência, é necessário consultar a anotação para determinar se o fragmento é pelo menos parcialmente codificante, e em caso positivo identificar a sequência de aminoácidos codificada pelo fragmento. Uma das suas perguntas de pesquisa pode ser: Qual é a complexidade computacional desse processo, em função de f.

(2) O UFM não é capaz de identificar partes codificantes nos fragmentos. Esperamos que se uma boa parte do fragmento for codificante, então ele irá indicar esse fragmento como codificante e gerará uma sequência de aminoácidos para todo o fragmento. Como no caso excepcional analisado, apenas uma parte do fragmento é codificante, a cadeia de aminoácidos geradas pelo UFM estará apenas parcialmente correta. Como podemos lidar com isto no caso dos sequenciamentos com referência e *de novo assembly*?.

Outra pergunta a ser respondida é qual é o limiar (número de nucleotídeos codificantes) para que UFM identifique um fragmento como codificante? De fato, o problema é mais complexo. Podemos identificar que com 200 nucleotídeos para cima ele tem uma taxa X (favorável) de acerto, mas se colocamos em um dos extremos "n" bases não codificantes, como se comporta o método? Ou seja, garantido que a parte codificante tem o número mínimo de nucleotídeos, como varia a taxa de acerto quando se acrescenta um percentual de sequência não codificante em um dos extremos? Depende do extremo, direito ou esquerdo?

5. Uma alternativa para identificar as mutações no arquivo de sequencia-

mento limpo de artefatos (obter o VCF), é alinhar cada fragmento ao genoma de referência usando blast. Uma versão do blast foi desenvolvida especificamente para esta tarefa (diamond blast). Neste contexto surgem 3 questões de pesquisa a serem respondidas:
(1) Quão exato é UFM na detecção de sequências codificantes
(2) Em caso positivo acima, quão mais rápido é UFM que diamond blast.
(3) Se usando diamond blast como pós-processamento da saída de UFM (para excluir falsos positivos) se consegue um processamento otimizado em velocidade e qualidade?.

2. Integração do UFM nos pipelines

Primeiro devemos notar que UFM é um método ab-initio para identificação de CDSs. Mas especificamente, para:

1. Classificar sequências de nucleotídeos de entrada em codificante ou não codificante, e
2. No caso codificante, identifica o quadro de leitura permitindo gerar como saída a sequência de aminoácidos (potencialmente) codificada pela sequência de entrada.

Neste ponto surgem algumas questões a serem respondidas:

1. Existem diversos outros métodos ab-initio. Porque usar UFM?
   A resposta precisa levar em conta:
   (a) A acurácia
   (b) O custo computacional, e
   (c) O custo de treinamento do modelo
   O cenário mais favorável é aquele no qual UFM tem nenhum ou menor custo de treinamento do modelo, menor custo computacional e acurácia similar ou superior à do melhor método existente.
2. Temos várias condicionantes do estudo comparativo a serem tratadas:
   (a) Quantos outros métodos (códigos) ab-initio estão disponíveis para teste?
   (b) Vamos comparar com métodos que usam mais features (promoter, TSS, etc)?
   (c) Como vamos comparar UFM com métodos que são capazes de detectar fragmentos codificantes na sequência de entrada?
   (d) Quais datasets serão selecionados para a comparação: quantos e de quais espécies? Só de vírus? de quais vírus? quantos genomas de cada vírus?

Em segundo lugar, a ideia original do trabalho não é usar UFM como substituto de outros métodos ab-initio nos pipelines com passo de anotação de genes, o que não é descartado, senão introduzir UFM como um passo de pré-processamento de sequências no início dos pipelines, transformando os passos seguintes do pipeline para tratar sequências de aminoácidos em vez de sequências de nucleotídeos.

Neste ponto precisamos responder algumas questões:

1. O que se espera com esta mudança?
   Esperamos várias coisas que simplificam o processamento, sem perda excessiva de qualidade:

(a) Redução significativa do número de reads a serem processados no pipeline, devido à filtragem de reads não codificantes?
Quanto é significativo? > 30%, 50%, 70%?
Vai depender da espécie. Por exemplo, no caso de vírus, devido à alta compactação do genoma, a maioria dos reads serão codificantes. Não serão codificantes reads nas pequenas regiões entre ORFs e no início e final do genoma. Já em eucariotos onde existem grandes regiões inter-gênicas e introns, a porção de reads não codificantes deve ser maioritária.
Surge a pergunta: Precisamos fazer um estudo comparativo do percentual de sequências filtradas por UFM em diferentes tipos de genomas (espécies)? Quantas espécies?

(b) Menor custo computacional

i. (No caso de *de novo assembly*) para fazer montagem de sequências de aminoácidos, 1/3 menores, do que de nucleotídeos?
Surgem as perguntas:

A. Quanto menor? Dá para estimar de forma analítica?
B. Os códigos de assembly atuais permitem montar sequências de aminoácidos?
C. Em caso positivo, qual é o mais rápido (mais usado deles)? Como vamos testar o ganho de performance? Quais genomas, quantos?
D. Em caso negativo, vamos construir um ensamblador com as duas variantes para o estudo? Vamos comparar o ensamblador de teste com outros de pratilheira para poder estender o resultado? BURACO NEGRO

ii. (No caso de alinhamento contra referência) para fazer alinhamento contra a referência de sequências de aminoácidos, 1/3 menores, do que de nucleotídeos?
Surgem as perguntas:

A. Quanto menor? Dá para estimar de forma analítica?
B. Qual código de alinhamento de 2 sequências vamos usar para avaliar o ganho? Por qué?
C. Como vamos testar o ganho de performance? Quais genomas, quantos?

(c) Filtragem automática de mutações sinônimas, reduzindo o custo de construção do VCF.
A redução é significativa? > 30%, 50%, 70%?
De quais fatores depende a redução?

2. Quais as desvantagens ou pontos fracos dessa abordagem que precisam ser estudados?
Numa análise preliminar vejo alguns pontos que precisamos acompanhar e avaliar no projeto:

(a) A acurácia de UFM diminui com a redução do tamanho das sequências.
Então precisamos mapear os comprimentos médios e desvio padrão dos reads das distintas técnicas NGS para avaliar a acurácia com cada técnica.

(b) UFM foi desenhado para ter parâmetros fixos para todas as espécies, mas, estudos preliminares quer fiz com 2 vírus, mostraram necessidade de ajustar os parâmetros do modelo.
Eu explorei um método simples de ajuste de parâmetros que precisaria ser avaliado antes de inclui-lo como parte do pipeline.
Perguntas:
   i. O ajuste paramétrico precisa ser realizado apenas para vírus, ou para outras espécies também. No caso de vírus, precisa ser feito para diferentes vírus?
   ii. Qual é o tamanho mínimo (número de reads) do dataset para treinamento. Como depende este número do tamanho médio dos reads?
   iii. Quais casos de teste serão usados para este estudo?
   iv. A melhora da acurácia com o ajuste de parâmetros é satisfatória, ou insuficiente? Em quais casos funciona e em quais não? O resultado depende do comprimento médios dos reads?

(c) UFM avalia o potencuial codificante do read como um todo. Por tanto, ele pode considerar não codficiante um read com uma parte codificante (Falso negativo parcial), assim como classificar como codificante um read que contem uma parte não codificante (falso positivo parcial).
Perguntas:
   i. Como podemos proceder para "detectar"e "tratar"os falsos negativo e positivo parciais?

3. Plano de Atividades

Aqui VOCÊ precisa elencar TODAS as "atividades", passo a passo em cada questão de pesquisa, na ordem, para poder fazer o planejamento do projeto.

Construa uma árvore para melhor entendimento dos ramos da pesquisa

No final escolheremos um "path"executável no seu TCC

Note que:

Cada pergunta gera ≥ 1 atividades.

Também onde disse "precisa(mos)" há atividades implícitas a serem realizadas.