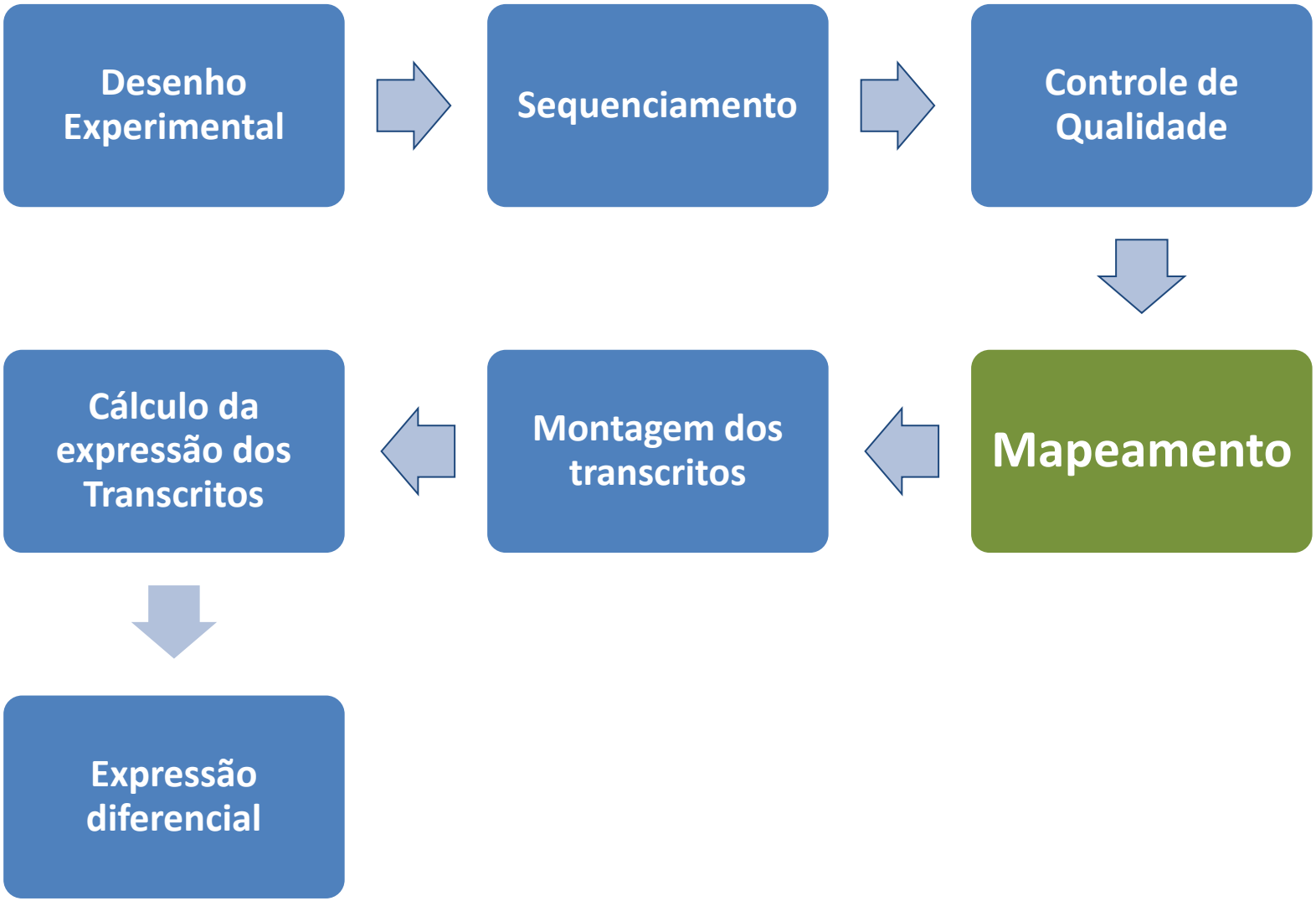
RNA-Seq Mapeamento

Vagner Okura

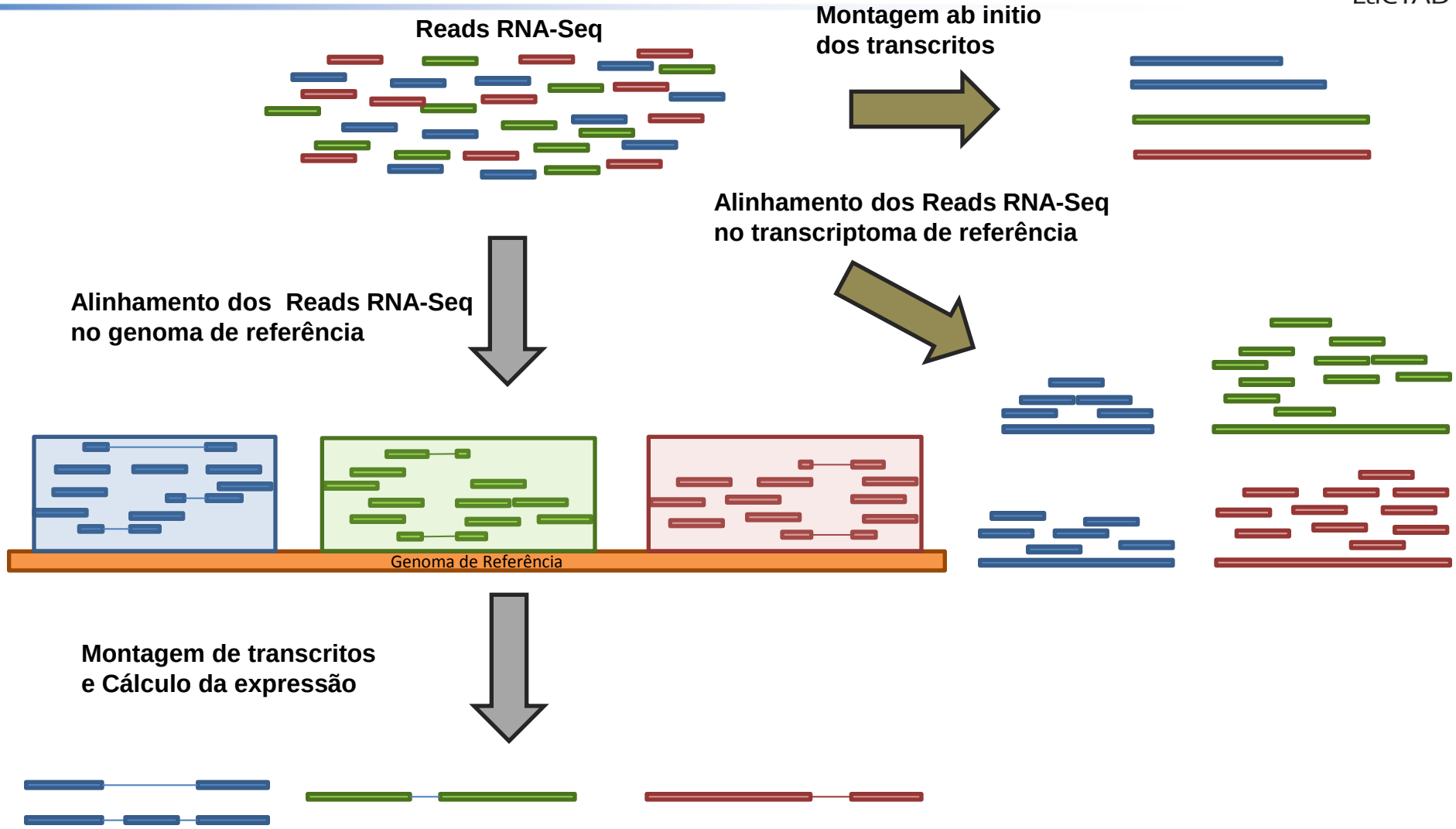
LaCTAD

vagnerko@unicamp.br

RNA-Seq


Desenho Experimental
Sequenciamento
Controle de Qualidade
Mapeamento
Montagem dos transcritos
Cálculo da expressão dos Transcritos
Expressão diferencial

Mapeamento


Reads RNA-Seq
Montagem ab initio
dos transcritos
Alinhamento dos Reads RNA-Seq
no transcriptoma de referência
Alinhamento dos Reads RNA-Seq
no genoma de referência
Genoma de Referência
Montagem de transcritos
e Cálculo da expressão

Mapeamento

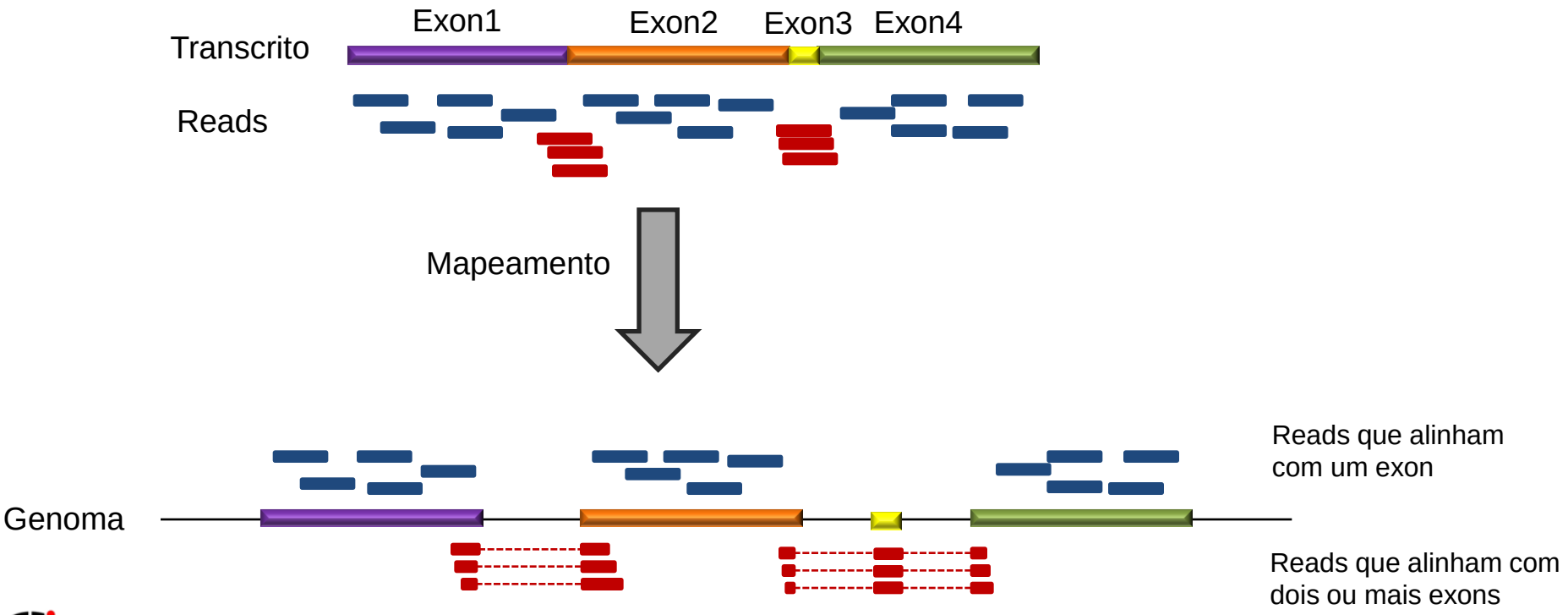
- Objetivo: determinar a localização dos reads de RNA-Seq no genoma/transcriptoma de referência.
- Para isso, os reads são alinhados no genoma/transcriptoma de referência.
  - Mapeamento ⇔ Alinhamento
- A localização dos reads na seqüência de refêrencia determina quais os genes, transcritos e/ou exons que estão associados a amostra do experimento, e o número de reads alinhados possibilita estimar sua abundância (expressão).

Desafios do Mapeamento

- Número de reads por experimento é bastante grande
  - Dezenas (a centenas) de milhões de reads por amostra
  - Software tem que ser rápido; permitir uso de múltiplas CPU's.
- Diferenças de seqüência entre reads e referência
  - Erros de sequenciamento
  - Polimorfismos, Inserções e deleções (indels)
- Reads podem alinhar em múltiplas regiões
  - Alguns genes possuem seqüências similares - família gênica; Repetições
  - Transcritos com processamento alternativo (*alternative splicing*) de um gene compartilham uma fração significatica do read.

Mapeamento usando genoma de referência

- Depende de ter genoma sequenciado.
- Alinhamento entre cDNA e DNA
  - É preciso tratar reads que alinham na junção entre exons.



Mapeamento usando transcritoma de referência

- Depende de montagem do transcriptoma
- Alinhamento entre cDNA e cDNA
  - Supõe que introns não estão envolvidos no alinhamento.

Softwares de Mapeamento

- Há dois grupos de softwares de mapeamento/alinhamento de reads de RNA-Seq:
  1. Alinhadores para reads sem junções
     - *Unspliced read aligners*
       - Não permitem alinhamentos com buracos (*gaps*) ou permitem buracos curtos.
     - São utilizados para mapeamento com transcriptoma de referência
     - Exemplo de Softwares: SHRiMP, Stampy, MAQ, BWA, SOAP2, Bowtie

  2. Alinhadores para reads com junções
     - *Spliced read aligners*.
       - Permitem alinhamentos com buracos longos.
       - Podem identificar novos exons e novas junções – novos transcritos.
     - São utilizados para mapeamento com genoma de referência.
     - Exemplo de Softwares: GSNAP, MapSplice, PALMapper, ReadsMap, STAR, TopHat

Softwares de Mapeamento

- Há dois pontos importantes no desenvolvimento dos softwares de mapeamentos de reads de RNA-Seq
    1. Alinhamento não exato
        - Ao alinhar um read a referência, o software precisa permitir um **alinhamento aproximado**. Ou seja, permitir mismatches, ou inserções e deleções.
        - Em computação, o alinhamento exato entre seqüências é eficiente. Porém, é custoso para os softwares permitirem alinhamentos não exatos.
    2. Rapidez
        - Para acelerar o processo de mapear milhões de reads, são aplicados algoritmos e estruturas de dados otimizados.
- Os softwares de mapeamento tem implementações diferentes:
    - Alinhadores baseados em *Hash Table*
    - Alinhadores baseados no método BWT (*Burrows-Wheeler transform*)

Softwares de Mapeamento

Sofwares	Basedo em Hash Table	Baseado em BWT

Comparação	• São mais lentos
	• Maior sensibilidade (*sensitivity*)
	  • Permitem número maior de *mismatches*
	• Permite detecção de indels
	• Usa mais memória RAM
		• São mais rápidos
		• Menor sensibilidade
		  • Número pequeno de *mismatches* permitidos
		• Detecção de indels limitada
		• Usa menos memória RAM

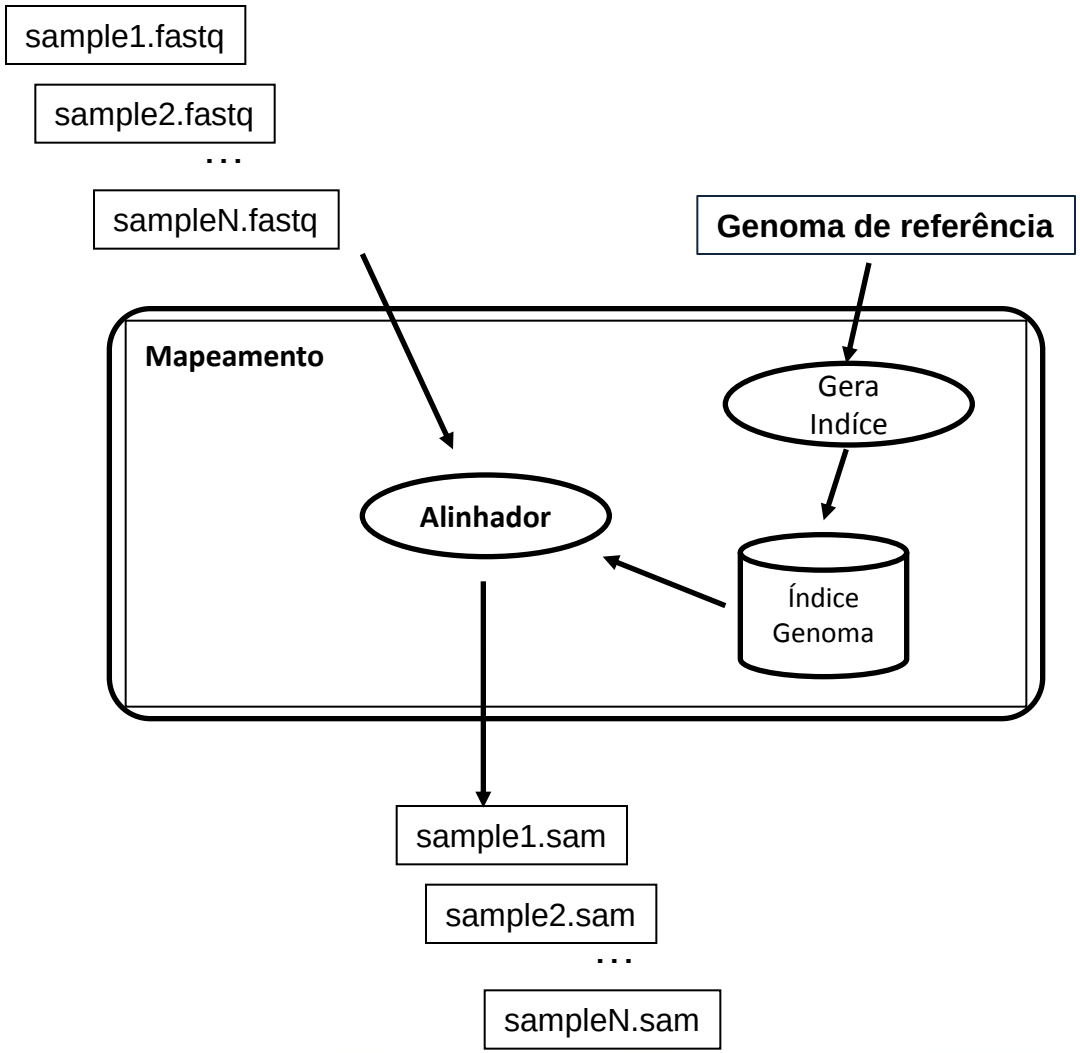
Aplicação	• Recomendado para mapeamento entre espécies mais distantes. • Pode ser usado para mapeamento de reads oriundos de células cancerigenas (podem apresentar acúmulo de polimorfismos e mutações)	• Recomendado para mapeamento entre mesma espécie ou espécies muito próximas.
Alinhadores para reads sem junções	SHRiMP, Stampy, MAQ,	Bowtie, BWA, SOAP2
Alinhadores para reads com junções	GSNAP, MapSlice	Tophat, STAR, MapSlice

Outras Considerações

- Tamanho do read: quanto menor, maior possibilidade de mapear em mais de um local.
- Reads multimapeados (mapeiam em mais de um local): são considerados para evitar perda de informação.
- Single end (SE) X Paired End (PE): PE melhora especificidade; um read repetitivo pode ter seu par mapeado unicamente.
  - SE: unicamente mapeados, multimapeado, não mapeado
  - PE: par de reads no mapeiam unicamente, um read unicamente mapeado e outro multimapeado, um read unicamente mapeado e outro não mapeado, par de reads multimapeados, um read multimapeado e outro não mapeado, nenhum dos reads mapeados.

- Principais parâmetros dos softwares de mapeamento
  - Número de mismatches
  - Número de CPUs que pode ser utilizada

Workflow Mapeamento bowtie e tophat



Bowtie

- Passo 1: gerar índice do genoma
- Uso: bowtie-build [parâmetros opcionais] <referência> <nome-índice>
  - referência: lista de arquivos fasta separados por vírgula
  - nome-índice: nome base dos arquivos de índice (nome.1.ewt, nome.2.etc, …)
- Exemplos:

```
> bowtie-build  genoma.referencia.fasta   genoma
> bowtie-build  chrX.fasta,chrY.fasta,chr20.fasta  genomaXY20
```

Bowtie

- Passo 2: alinhar dados de cada amostra
- Uso: bowtie [parâmetros opcionais] <nome-índice> {-1 <lista-arquivos-pe1> -2 <lista-arquivos_pe2> | <lista-arquivos-se}
  - nome-índice: nome base dos arquivos de índice (nome fornecido na etapa de gerar índice)
  - lista-arquivos-pe1: lista de nomes de arquivos separados por virgula da leitura 1 de reads Paired End. A ordem do nome dos arquivos deve corresponder a lista-arquivos-pe2.
  - lista-arquivos-pe2: lista de nomes de arquivos separados por virgula da leitura 2 de reads Paired End. A ordem do nome dos arquivos deve corresponder a lista-arquivos-pe1.
  - lista-arquivos-se: lista de nomes de arquivos separados por virgula de amostras Single Read.
- Alguns parâmetros
  - -v <N>: reporta alinhamentos com no máximo N mismatches (N < = 3)
  - -S ou --sam: alinhamentos são gerados no formato SAM
  - -p ou –threads <num>: divide o processamento entre num CPUs/processadores

Bowtie

- Passo 2: alinhar dados de cada amostra

- Exemplos:

```
> bowtie -v 3   -p 8  -S genoma \
    -1 amostraA_1.fastq,amostraB_1.fastq,amostraC_1.fastq \
    -2 amostraA_2.fastq,amostraB_2.fastq,amostraC_2.fastq

> bowtie -v 2   -p 8  -S genoma \
amostraD.fastq,amostraE.fastq,amostraF.fastq
```

Bowtie

- Passo 3: Converter arquivos SAM para BAM, ordenar e criar índice.

- Exemplos:

```
> samtools view -bS amostraD.sam –o amostraD.bam
> samtools sort amostraD.bam amostraD.bam.sort
> samtools index amostraD.bam.sort
```

Tophat

- Passo 1: gerar índice do genoma (usa bowtie-build)
- Uso: bowtie-build [parâmetros opcionais] <referência> <nome-índice>
  - referência: lista de arquivos fasta separados por vírgula
  - nome-índice: nome base dos arquivos de índice (nome.1.ewt, nome.2.etc, …)
- Exemplos:

```
> bowtie-build  genoma.referencia.fasta   genoma
```

Tophat

- Passo 2: Alinha reads de cada amostra
- Uso: tophat [parâmetros opcionais] <nome-índice> <lista-arquivos-pe1> <lista-arquivos_pe2>
  - nome-índice: nome base dos arquivos de índice (nome fornecido na etapa de gerar índice)
  - lista-arquivos-pe1: lista de nomes de arquivos separados por virgula da leitura 1 de reads Paired End. A ordem do nome dos arquivos deve corresponder a lista-arquivos-pe2.
  - lista-arquivos-pe2: lista de nomes de arquivos separados por virgula da leitura 2 de reads Paired End. A ordem do nome dos arquivos deve corresponder a lista-arquivos-pe1. Se não for definida, primeira lista é considerada single read.
- Alguns parâmetros
  - -r ou --mate-inner-dist <num>: define a distância entre reads paired end (50bp)
  - --mate-std-dev <int>: define os desvio padrão (20 bp)
  - -p ou –threads <num>: divide o processamento entre num CPUs/processadores
  - -G ou --GTF <arquivo-anotação>: se especificado, Tophat alinha primeiramente os reads com transcriptoma anotado, e a seguir alinha reads não mapeados buscando regiões de splicing. Formatos aceitos de anotação: GTF,GFF3
  - -o <diretório>: especifica os diretório/pasta onde serão colodados os arquivos de saída.

Tophat

- Exemplos:

```
> tophat -G genes.gff  -r 200  -p 16  -o amostraA_out  genoma \
   amostraA_rep1_1.fastq  amostraA_rep1_2.fastq
> tophat -G genes.gff  -r 200  -p 16  -o amostraA_out  genoma \
   amostraA_rep2_1.fastq  amostraA_rep2_2.fastq
> tophat -G genes.gff   -p 8  -o amostraB_out  genoma \
   amostraB_rep1.fastq,amostraB_rep2.fastq,amostraB_rep2.fastq
```

Tophat

- Arquivos de saída:
  - accepted_hits.bam: alinhamento dos reads no formato BAM
  - junctions.bed: lista de junções reportadas
  - insertions.bed: lista inserções reportadas
  - deletions.bed: lista deleções reportadas
  - Align_summary.txt

```
Reads:
          Input     :  16687898
           Mapped   :  15198242 (91.1% of input)
            of these:    461513 ( 3.0%) have multiple alignments (9 have >20)
91.1% overall read mapping rate.
```

SAM/BAM

- SAM é o formato do arquivos de alinhamento.
  - SAM - Sequence Alignment/Map.
  - É um arquivo texto delimitado por tabulações.
- BAM é a versão compactada e binária para arquivos de alinhamento.
  - Possui tamanho menor.
- Ambos formatos podem ser indexados, permitindo acesso mais rápido aos alinhamentos.
- Estes formatos são usados como entrada por outros softwares - montagem de transcritos, analise de expressão, visualização de alinhamentos (ordenado e indexado).

Samtools

- Este software fornece uma série de programas para manipular arquivos de alinhamento no formato SAM/BAM.
- Permite trabalhar com entrada e saída padrão (Linux).
- Programas mais usados:
  - **view**: usado principalmente para converter os formatos SAM ⇔ BAM
    - `samtools view amostra1.bam -o amostra1.sam`
    - `samtools view amostra1.bam > amostra1.sam`
    - `samtools view -bS amostra2.sam –o amostra2.bam`
  - **sort**: ordena arquivo BAM usando posição da seqüência de referência.
    - `samtools sort amostra1.bam amostra1.bam.sort`
  - **index** : cria um arquivo de indíce que permite busca rápida nos dados do arquivo SAM/BAM. Arquivo precisa estar ordenado para ser indexado. Gera arquivo *.sam.sai, *bam.bai.
    - `samtools index amostra1.bam.sort`