

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA

DIEGO CÉSAR BATISTA MARIANO

**SIMBA: uma ferramenta Web para gerenciamento de  
montagens de genomas bacterianos**

Belo Horizonte,  
2015

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA

DIEGO CÉSAR BATISTA MARIANO

**SIMBA: uma ferramenta Web para gerenciamento de  
montagens de genomas bacterianos**

Dissertação apresentada ao Programa de Pós-Graduação em Bioinformática da UFMG como requisito parcial para a obtenção do grau de Mestre em Bioinformática.

Orientador: Prof. Dr. Vasco Ariston Carvalho de Azevedo

Co-orientador: Prof. Dr. Rommel Thiago Jucá Ramos

Belo Horizonte,  
2015

043

Mariano, Diego César Batista.

SIMBA: uma ferramenta Web para gerenciamento de montagens de genomas bacterianos [manuscrito] / Diego César Batista Mariano. - 2015.

82 f. : il. ; 29,5 cm.

Orientador: Vasco Ariston Carvalho de Azevedo. Co-orientador: Rommel Thiago Jucá Ramos.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Biológicas.

1. SIMBA. 2. Montagem de genomas - Teses. 3. Ferramenta Web - Teses. 4. Bioinformática - Teses. 5. I. Azevedo, Vasco Ariston Carvalho de. II. Ramos, Rommel Thiago Jucá. III. Universidade Federal de Minas Gerais. Instituto de Ciências Biológicas. IV. Título.

CDU: 573:004



# DECLARAÇÃO

Declaro, para os devidos fins, que **Diego César Batista Mariano** defendeu a sua dissertação de mestrado intitulada **SIMBA: uma ferramenta Web para gerenciamento de montagens de genomas bacterianos**, defendida em 26 de fevereiro de 2015.

Declaro, ainda, que a dissertação foi aprovada pela banca examinadora e que o aluno obteve o título de mestre em Bioinformática no Programa de Pós Graduação em Bioinformática do ICB/UFMG.

Belo Horizonte, 23 de fevereiro de 2015.



Prof. Dr. Vasco Ariston de Carvalho Azevedo

Coordenador do Programa de Pós-Graduação em Bioinformática

ICB/UFMG

### ***Invictus***

*Do fundo desta noite que persiste  
A me envolver em breu - eterno e espesso,  
A qualquer deus - se algum acaso existe,  
Por mi'alma insubjugável agradeço.*

*Nas garras do destino e seus estragos,  
Sob os golpes que o acaso atira e acerta,  
Nunca me lamentei - e ainda trago  
Minha cabeça - embora em sangue - ereta.*

*Além deste oceano de lamúria,  
Somente o Horror das trevas se divisa;  
Porém o tempo, a consumir-se em fúria,  
Não me amedronta, nem me martiriza.*

*Por ser estreita a senda - eu não declino,  
Nem por pesada a mão que o mundo espalma;  
Eu sou dono e senhor de meu destino;  
Eu sou o comandante de minha alma.*

**William Ernest Henley**

*Tradução: André C. S. Masini*

## Agradecimentos

O mestrado foi o período de maiores mudanças em toda minha vida. Aprendi a escutar, a respeitar, a conviver, a pensar, a ensinar e, até mesmo, aprendi que a vida é um eterno aprendizado. Posso dizer que é imensurável o quanto cresci pessoalmente e profissionalmente nestes últimos dois anos.

Talvez seja injusto agradecer, aqui, a apenas alguns nomes em virtude da imensa quantidade de pessoas que colaboraram direta e indiretamente para que este trabalho fosse finalizado. Em todo caso, tentarei citar algumas pessoas.

Agradeço a toda equipe do Laboratório de Genética Celular e Molecular da UFMG pelo apoio e paciência. Sobretudo pela **paciência**.

Aos meus orientadores: Prof. Dr. Vasco Azevedo e Prof. Dr. Rommel Ramos.

À Universidade Federal de Minas Gerais.

A Leandro, Marcus, Edgar, Edson e Letícia pela análise textual minuciosa deste manuscrito.

Ao grupo de mestrandos da primeira turma em Bioinformática da UFMG, a todo corpo docente do Programa de Pós-graduação em Bioinformática e aos grupos de discussão.

A toda minha família, em especial minha mãe dona Vera, meu pai seu Jairo, e meu irmão Denis José.

Às meninas da secretaria de bioinformática por toda a atenção. Em especial, a Sheila.

A Oswaldo Nicácio pelo *upgrade* do *hardware* que utilizei durante o mestrado; a Gabriel Vitor por me ensinar a desenvolver com Laravel; a Felipe Pereira pela contribuição em códigos e pelas discussões sobre montagens de genomas; ao Prof. Dr. Sandro Renato Dias, por ter sido o primeiro a acreditar que eu poderia chegar onde estou; a Francislon por todas as vezes que me ajudou pelo Hangout; a Sandeep, Jamal, Hassan e Luís pelas correções nas traduções para o inglês; a Baiano pelo apoio musical durante esses dois anos de trabalho e a todos os outros baianos do LGCM; a todas as pessoas que participaram das “discussões científicas” durante o almoço na cozinha do laboratório; a Letícia Avelino por me apoiar mesmo nos momentos mais difíceis; ao Prof. Miguel Ortega pelos ensinamentos que jamais esquecerei e pelas belas palavras ditas após minha defesa; e a todos os que por algum motivo contribuíram para meu crescimento profissional e pessoal antes, durante e depois do mestrado.

## Sumário

Lista de abreviaturas.....	9
Lista de figuras.....	10
Lista de tabelas.....	12
Resumo.....	13
Abstract.....	14
1. Introdução.....	15
1.1 Plataformas de sequenciamento de próxima geração.....	16
1.2 Segunda e terceira geração de sequenciadores.....	20
1.3 Semicondutores.....	21
1.3.1 Fluxo de dados Ion Torrent™.....	24
1.4 Tratamentos iniciais de dados.....	25
1.5 Paradigmas para montagem de genomas.....	27
1.5.1 Montagem de novo.....	29
1.5.1.1 Algoritmo guloso.....	30
1.5.1.2 OLC (overlap-layout-consensus).....	30
1.5.1.3 Grafo De Bruijn.....	32
1.5.2 Montagem por referência.....	34
1.6 Finalização de montagens.....	35
1.6.1 Ordenação de contigs.....	36
1.6.2 Fechamento de gaps in silico.....	39
1.7 Preparação dos dados para depósito em bancos de dados públicos.....	40
1.8 Problemática.....	40
1.9 Justificativa.....	41
1.10 Objetivos.....	42
1.10.1 Objetivo geral.....	42
1.10.2 Objetivos específicos.....	42
2. Metodologia.....	43
2.1 Etapas do <i>pipeline</i> .....	43
2.1.1 Tratamento inicial de dados.....	44
2.1.2 Montagem de novo.....	45
2.1.3 Finalização de montagem.....	46
2.1.3.1 Ordenação de contigs por referência.....	46
2.1.3.2 Ordenação de contigs por mapeamento óptico.....	46
2.1.3.3 Mover o gene <i>dnaA</i> para a posição inicial do genoma.....	47

2.1.3.4 Construção de Supercontigs .....	48
2.1.3.5 Resolvendo regiões repetitivas .....	48
2.1.3.6 Estatísticas e curadoria manual .....	50
2.2 <i>Hardware</i> .....	51
2.3 Interface SIMBA aplicada ao <i>pipeline</i> .....	51
2.4 Estudo de caso: <i>Corynebacterium pseudotuberculosis</i> como modelo.....	52
3. Resultados e discussões .....	55
3.1 Visão geral da interface.....	55
3.1.1 Projetos .....	56
3.1.2 Montagens .....	58
3.1.2 Curadoria .....	59
3.2 Resultados do estudo de caso .....	61
3.2.1 Tratamento de dados.....	61
3.2.2 Montagem de novo .....	62
3.2.2.1 Resultados e discussões da montagem de novo usando o genoma de <i>Corynebacterium pseudotuberculosis</i> linhagem 258 .....	63
3.2.2.2 Resultados e discussões da montagem de novo usando o genoma de <i>Corynebacterium pseudotuberculosis</i> linhagem 1002 .....	64
3.2.3 Curadoria: finalização da montagem .....	65
3.2.3.1 Resultados e discussões da finalização por referência .....	65
3.2.3.2 Resultados e discussões da finalização por mapeamento óptico... ..	67
3.2.4 Comparação entre montagens .....	69
3.2.4.1 Diferença entre tamanhos de genoma.....	69
3.2.4.2 Alinhando sequências do NCBI com mapa de restrição .....	70
3.2.4.2 Comparando o genoma de Cp258 com o genoma de Cp1002 .....	71
4. Considerações finais .....	73
4.1 Perspectivas para trabalhos futuros .....	74
5. Referências bibliográficas.....	75



## Lista de abreviaturas

<b>A</b>	Adenina
<b>BAM</b>	<i>Binary Alignment/Map</i>
<b>BLAST</b>	<i>Basic local alignment search tool</i>
<b>C</b>	Citosina
<b>CDS</b>	<i>Coding DNA Sequence</i>
<b>CLI</b>	<i>Command-line interface</i>
<b>CMNR</b>	<i>Corynebacterium, Mycobacterium, Nocardia, and Rhodococcus</i>
<b>Cp258</b>	<i>Corynebacterium pseudotuberculosis 258</i>
<b>Cp1002</b>	<i>Corynebacterium pseudotuberculosis 1002</i>
<b>dATP</b>	<i>Deoxyadenosine triphosphate</i>
<b>dCTP</b>	<i>Deoxycytidine triphosphate</i>
<b>dGTP</b>	<i>Deoxyguanosine triphosphate</i>
<b>DNA</b>	<i>Deoxyribonucleic acid</i>
<b>dNTP</b>	<i>Deoxyribonucleotide triphosphate</i>
<b>dTTP</b>	<i>Deoxythymidine triphosphate</i>
<b>G</b>	Guanina
<b>Gb</b>	Gigabase
<b>GB</b>	Gigabyte
<b>HD</b>	<i>Hard disk</i>
<b>INDEL</b>	<i>Insertion or deletion</i>
<b>ISFET</b>	<i>Ion-sensitive field-effect transistor</i>
<b>Kb</b>	Kilobase
<b>Mb</b>	Megabase
<b>MB</b>	Megabyte
<b>MVC</b>	<i>Model-view-controller</i>
<b>NCBI</b>	<i>National Center for Biotechnology Information</i>
<b>NGS</b>	<i>Next-Generation Sequencing</i>
<b>OLC</b>	<i>Overlap-Layout-Consensus</i>
<b>Pb</b>	Pares de base
<b>PCR</b>	<i>Polymerase Chain Reaction</i>
<b>PGM™</b>	<i>Personal Genome Machine™</i>
<b>RNA</b>	<i>Ribonucleic acid</i>
<b>rRNA</b>	<i>Ribosomal Ribonucleic Acid</i>
<b>SAM</b>	<i>Sequence Alignment/Map</i>
<b>SFF</b>	<i>Standard flowgram format</i>
<b>SIMBA</b>	<i>Simple Manager for Bacterial Assemblies</i>
<b>SMRT</b>	<i>Single Molecule Real-Time</i>
<b>SNP</b>	<i>Single nucleotide polymorphism</i>
<b>SOLiD</b>	<i>Sequencing by Oligonucleotide Ligation and Detection</i>
<b>T</b>	Timina
<b>WGS</b>	<i>Whole Genome Shotgun</i>

## Lista de figuras

Figura 1 – Gráfico de custo por megabase sequenciada.....	16
Figura 2 - Processo de obtenção de seqüências de DNA: da extração à finalização da montagem.....	17
Figura 3 – Modelos do sequenciador Ion Torrent™.....	22
Figura 4 – Preparação do DNA para o sequenciamento.....	23
Figura 5 – Pareamento e liberação de H <sup>+</sup> .....	23
Figura 6 – Fluxo de dados do Ion Torrent™.....	24
Figura 7 – Exemplo de arquivo de seqüências: FASTA. O arquivo pode conter uma ou mais seqüências (multifasta). .....	25
Figura 8 – Exemplo de arquivo FASTQ. O arquivo contém uma seqüência de nucleotídeos concatenada a dados codificados de qualidade das bases.....	26
Figura 9 – Exemplo de gráfico gerado pelo FastQC. ....	27
Figura 10 – Alinhamento entre leituras.....	28
Figura 11 – Sobreposição/alinhamento entre leituras para formar uma seqüência consenso ( <i>contig</i> ). .....	30
Figura 12 – Construção do leiaute: sobreposição entre <i>contigs</i> repetidos. Pode-se observar o aumento relativo da cobertura da região quando B e C são sobrepostos. ....	31
Figura 13 – Leituras pareadas usadas para correta montagem de regiões repetitivas. ....	31
Figura 14 – Representação de como uma leitura é quebrada em <i>k-mers</i> no grafo De Bruijn.....	32
Figura 15 – Alinhamento entre duas seqüências no grafo De Bruijn. ....	33
Figura 16 – Alinhamento das leituras contra uma referência. ....	34
Figura 17 – Montagem por referência. Há distribuição das leituras sobre as regiões repetitivas. ....	34
Figura 18 – Comparação entre o alinhamento de leituras montadas <i>de novo</i> . Na montagem <i>de novo</i> (sem parâmetros de distribuição uniforme) as leituras idênticas serão sobrepostas permitindo a formação de <i>gaps</i> .....	35
Figura 19 – Formação de <i>supercontig</i> através da mesclagem da sobreposição entre extremidades de dois <i>contigs</i> vizinhos. ....	36
Figura 20 – Exemplo de mapeamento óptico.....	38
Figura 21 – Uso do mapa óptico para detecção de grandes inversões em <i>Xenorhabdus bovienii</i> , vistas na cor amarelo; em vermelho, pode-se perceber regiões montadas erradas ( <i>mismatches</i> ); em verde regiões corretamente montadas. ....	39
Figura 22 – Uso de leituras pareadas para fechamento de <i>gaps</i> no GapFiller. ....	39
Figura 23 – Etapas do <i>pipeline</i> para montagem de genomas procariotos. ....	44
Figura 24 – Exemplo de gráfico de alinhamento gerado pelo CONTIGuator.....	46
Figura 25 – Representação gráfica do uso do script MoveDNAA. ....	47
Figura 26 – Diferentes tipos de <i>gap</i> .....	48

Figura 27 – Funcionamento do MapRepeat. ....	49
Figura 28 – Diagrama de relacionamento entre a arquitetura da interface SIMBA aplicada ao <i>pipeline</i> e sua integração com o modelo MVC.....	52
Figura 29 – Árvore filogenética da espécie <i>Corynebacterium pseudotuberculosis</i> ...	53
Figura 30 - Diagrama de fluxo de dados no SIMBA. ....	56
Figura 31 – Interface do SIMBA. ....	57
Figura 32 – Finalização de montagens na interface SIMBA.....	61
Figura 33 - Gráfico de qualidade Phred em leituras em Cp258 (à esquerda) e Cp1002 (à direita). Figuras geradas pelo FastQC.....	62
Figura 34 - Gráfico tamanho médio de leituras de Cp258 (à esquerda) e Cp1002 (à direita). Figuras geradas pelo FastQC.....	62
Figura 35 - Alinhamento de <i>contigs</i> contra o mapa de restrição no MapSolver para Cp258. ....	67
Figura 36 – Alinhamento de <i>contigs</i> contra o mapa de restrição no MapSolver para Cp1002. ....	68
Figura 37 – Uso do MapSolver para validação da montagem realizada com SIMBA para Cp258. ....	69
Figura 38 – Uso do MapSolver para validação da montagem realizada com SIMBA para Cp1002. ....	69
Figura 39 – Comparativo no MapSolver do mapa de restrição (acima) com o genoma de <i>Corynebacterium pseudotuberculosis</i> 258 (NC_017945) depositado no NCBI (abaixo).....	70
Figura 40 – Comparativo no MapSolver do mapa de restrição (acima) com o genoma de <i>Corynebacterium pseudotuberculosis</i> 1002 (NC_017300) depositado no NCBI (abaixo).....	71
Figura 41 – Comparativo entre Cp1002 (acima) e Cp258 (abaixo) usando ACT.....	72

## Lista de tabelas

Tabela 1 - Comparação entre plataformas NGS .....	19
Tabela 2 - Probabilidade de erro para cada base segundo o algoritmo de <i>Phred</i> . ...	25
Tabela 3 – Lista de nucleotídeos não identificados e seus respectivos caracteres representantes.....	50
Tabela 4 – Tentativas de montagem para Cp258. ....	64
Tabela 5 – Tentativas de montagem para Cp1002. ....	65
Tabela 6 – Número de <i>gaps</i> ao final da cada tentativa de finalização da montagem de <i>C. pseudotuberculosis</i> 258 (Cp258) e <i>C. pseudotuberculosis</i> 1002 (Cp1002). ...	66
Tabela 7 – Comparação entre tamanho de genoma depositado no NCBI, da montagem por referencia e da montagem por mapeamento óptico para Cp258 e Cp1002. ....	70

## Resumo

A evolução das plataformas de sequenciamento em larga escala vem reduzindo o tempo gasto para o processo de decodificação do DNA a um custo reduzido. Porém, os sequenciadores possuem algumas limitações, como por exemplo, o tamanho máximo dos fragmentos de DNA que são capazes de ler. O que leva a necessidade de fragmentar o DNA em pequenos pedaços antes do sequenciamento, sendo necessário, após essa etapa, reordenar os fragmentos lidos (leituras) de forma que se possa representar o genoma original. Esse processo, conhecido como montagem de genomas, pode ser caracterizado pela sua complexidade e dependência pelas limitações dos sequenciadores, o que evidencia a necessidade do uso de diversos programas computacionais. Nos últimos anos, diversas estratégias para montagem de genomas foram propostas, mas ainda não existe um consenso sobre qual a melhor abordagem. Nesse contexto, propõe-se um *pipeline* para montagem de genomas bacterianos, que será gerenciado por uma aplicação Web com interface amigável denominada SIMBA (*Simple Manager for Bacterial Assemblies*). Para avaliar sua performance foram feitas as montagens das linhagens *Corynebacterium pseudotuberculosis 1002* (originalmente sequenciada nas plataformas 454 Roche e Sanger) e *Corynebacterium pseudotuberculosis 258* (originalmente sequenciada na plataforma SOLiD v3) através de cinco diferentes *softwares*: Mira3, Mira4, Minia, Newbler e SPAdes. Ambas as linhagens foram ressequenciadas com bibliotecas de fragmentos simples de tamanho aproximado a 200pb na plataforma de semicondutores Ion PGM™. Após a montagem, escolheu-se um dos cinco resultados para etapa de fechamento de gaps através de duas abordagens: baseada em referência e baseada em mapeamento óptico. Por fim, observou-se que a ferramenta SIMBA permitiu uma rápida e fácil execução do processo de montagem e curadoria dos genomas. O *download* da ferramenta foi disponibilizado no website: <<http://ufmg-simba.sourceforge.net>>.

**Palavras-chave:** SIMBA, montagem de genomas, sequenciamento, NGS, Webtool

## Abstract

The evolution of large-scale sequencing platforms has reduced the time taken for the process of DNA fingerprinting at a reduced cost and in less time. However, sequencers still have limitations, such as the maximum size of DNA fragments that are capable of reading. What drives the need to break the DNA into small pieces before sequencing, being necessary after this step, to rearrange the fragments (reads) so that it can represent the original genome. This process is known as genome assembly. The genome assembly is a complex process dependent on the limitations of sequencers, so there is the need to use several computer programs. In recent years, several strategies for genome assembly have been proposed, but there is still no consensus on the best approach. In this context, we propose a pipeline for the assembly of bacterial genomes, which will be managed by a web application with user-friendly interface called SIMBA (Simple Bacterial Manager for Assemblies). To evaluate its performance, we assembled the strains *Corynebacterium pseudotuberculosis* 1002 (originally sequenced in Sanger and Roche 454 platforms) and *Corynebacterium pseudotuberculosis* 258 (originally sequenced on SOLiD v3 platform) using five different assembly software: Mira3, Mira4, Minia, Newbler and SPAdes. Both strains were resequenced with simple fragments libraries of approximate size to 200pb in the semiconductor Ion PGM™ platform. After assembly, was chosen one of them to perform the closing of gaps through two approaches: reference-based and optical mapping. Finally, it was observed that the SIMBA tool allows fast and easy execution of the assembly process and curation of genomes. The download of the tool is available at the website: <<http://ufmg-simba.sourceforge.net>>.

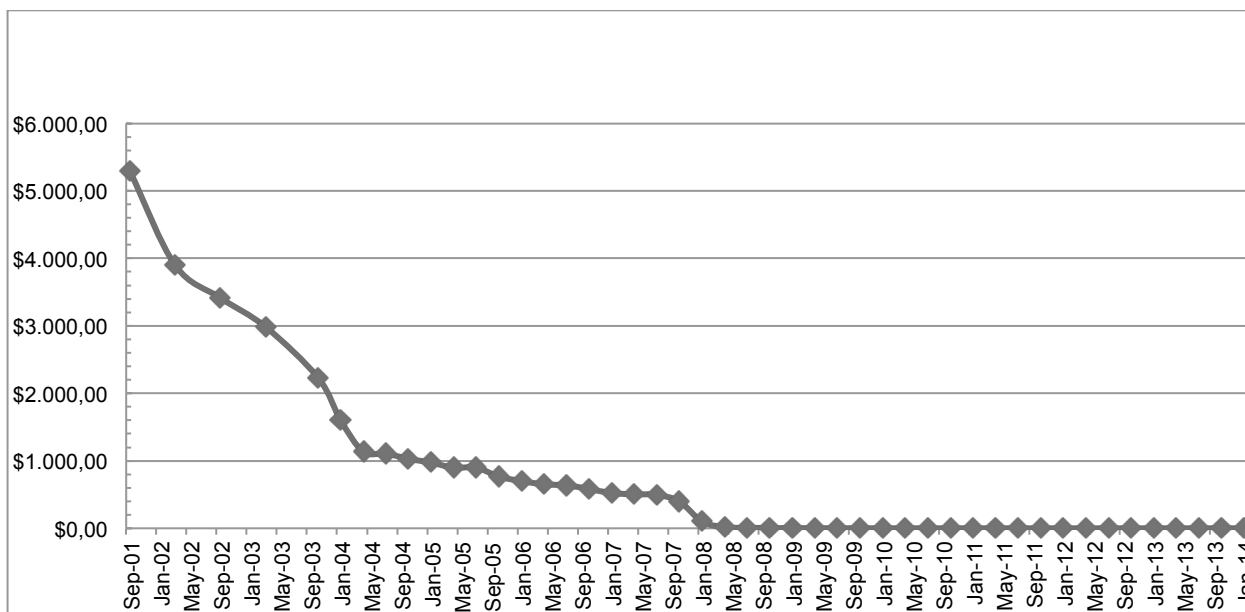
**Keywords:** SIMBA, genome assembly, sequencing, NGS, Webtool

## 1. Introdução

Define-se sequenciamento como o processo de leitura e identificação de bases de nucleotídeos (A, T, G, C) em um determinado fragmento de DNA. Através dele é possível estabelecer a estrutura primária do DNA, e assim buscar entender os fundamentos da vida em diversos organismos. Conhecer o genoma e identificá-lo com precisão é de valor fundamental para microbiologia, pois permite, por exemplo, conhecer melhor organismos patogênicos, a fim de desenvolver estratégias para controle epidemiológico (RIBEIRO *et al.*, 2012).

Nas últimas décadas, as tecnologias para sequenciamento de DNA têm revolucionado a biologia (POP, 2009). Os primeiros métodos de sequenciamento foram o método químico de degradação de bases, proposto por Allan Maxam e Walter Gilbert em 1977, e o método de sequenciamento por terminação de cadeia, proposto por Sanger e colaboradores (1977).

Com o passar do tempo, esses métodos foram aperfeiçoados, permitindo, por exemplo, o sequenciamento do primeiro genoma bacteriano, *Haemophilus influenzae*, em 1995 (FLEISCHMANN *et al.*, 1995). O sequenciamento utilizando a metodologia de Sanger produz leituras com boa acurácia e tamanho entre 1000pb e 2000pb, entretanto sua execução é lenta, incapaz de produzir grande quantidade de dados e apresenta um alto custo de execução (BONETTA, 2006; POP, 2009). Para exemplificar, o sequenciamento do genoma humano demorou aproximadamente treze anos para ser concluído (LANDER *et al.*, 2001; VENTER *et al.*, 2001), e em setembro de 2001, o custo por megabase sequenciada foi estimado em aproximadamente cinco mil dólares (Figura 1), o que representa um valor total maior que noventa e cinco milhões de dólares por genoma humano sequenciado (WETTERSTRAND, 2014).



**Figura 1 – Gráfico de custo por megabase sequenciada.**

Representa a queda do valor médio do custo por megabase sequenciada de US\$5.292,39, em janeiro de 2001, para US\$0,045, em janeiro de 2014. Assim, estima-se que o valor médio para o sequenciamento de um genoma humano em janeiro de 2001 fosse US\$95.263.072,00, entretanto o custo total do projeto genoma humano superou três bilhões de dólares. Em janeiro de 2014, o custo de sequenciamento do genoma humano foi estimado em aproximadamente US\$4.008,00. A queda de valores vista entre os anos de 2007 e 2008 está diretamente relacionada a uma maior utilização de sequenciadores NGS.

Fonte: adaptado (WETTERSTRAND, 2014).

A busca por métodos que implicassem em melhorias no processo de sequenciamento, redução de custos por base sequenciada e aceleração do processo de leitura de genomas completos, levou ao desenvolvimento dos chamados sequenciadores de próxima geração (NGS).

### 1.1 Plataformas de sequenciamento de próxima geração

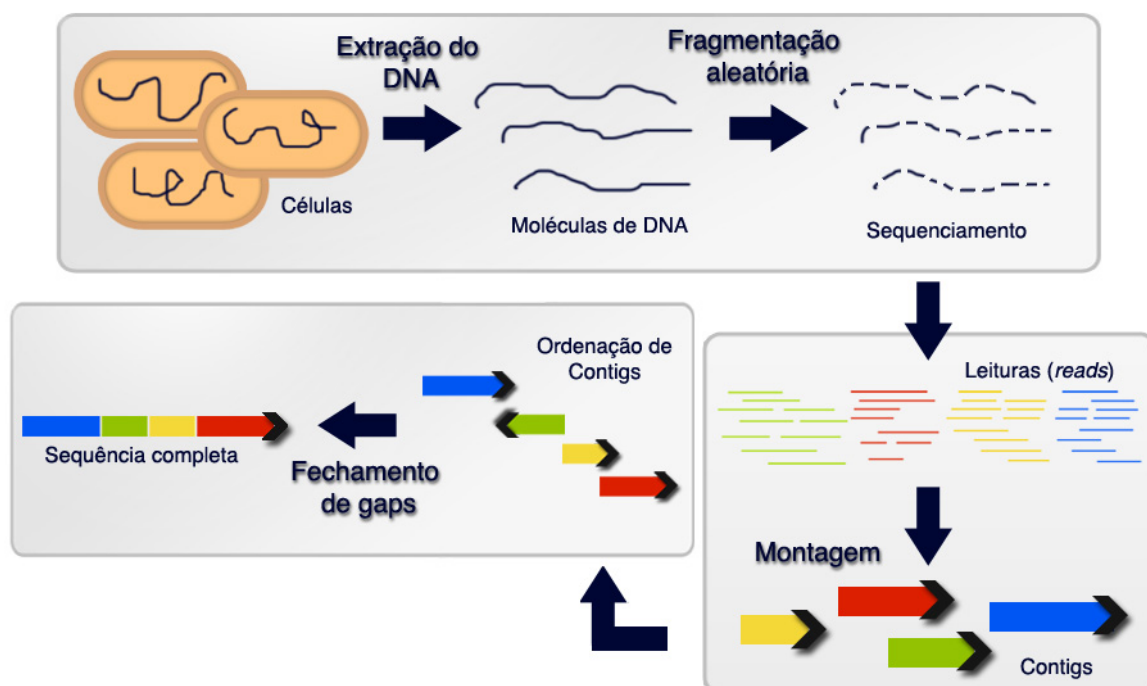
De acordo com Wetterstrand (2014), em janeiro de 2014, o custo por megabase sequenciada caiu para aproximadamente quatro centavos de dólar, ou seja, seria possível sequenciar um genoma humano por pouco mais de quatro mil dólares (Figura 1). Isso demonstra como, nos últimos anos, as plataformas de sequenciamento de próxima geração vêm permitindo leituras de DNA cada vez mais acuradas, com alto desempenho, menor tempo de execução e baixo custo por base sequenciada (EL-METWALLY *et al.*, 2013; HARISMENDY *et al.*, 2009; METZKER, 2005).



Esses novos sequenciadores podem ser caracterizados pela alta quantidade de dados gerados por corrida, além da utilização de metodologias diferentes do método de Sanger, o que possibilitou um aumento na quantidade de projetos de sequenciamento de genoma completo (WGS), principalmente em procariotos (HUSEMANN, 2011).

O sequenciamento completo de genomas procariotos através das plataformas NGS tem ajudado a transformar a microbiologia, permitindo um melhor acompanhamento da disseminação de patógenos, como visto nos estudos de GARDY e colaboradores (2011), LEWIS e colaboradores (2010) e MELLMANN e colaboradores (2011), buscando assim, compreender surtos epidêmicos e impedir a transmissão de doenças, o que pode auxiliar na melhoria de diagnósticos e na criação de novos medicamentos e vacinas (LOMAN *et al.*, 2012a).

Essas tecnologias baseiam-se em processos que podem ser basicamente divididos em cinco etapas: extração e purificação do DNA, construção de bibliotecas (fragmentação aleatória), sequenciamento, montagem e finalização da montagem (Figura 2) (HUSEMANN, 2011).



**Figura 2 - Processo de obtenção de sequências de DNA: da extração à finalização da montagem.**

Os processos de extração de DNA de células e fragmentação aleatória para o sequenciamento fazem parte das etapas *in vitro*. Enquanto os processos de montagem de leituras, ordenação de *contigs* e fechamento de *gaps* são etapas *in silico*.

Fonte: adaptado (HUSEMANN, 2011).

Após a extração e purificação, o DNA é submetido ao processo de construção de bibliotecas, na qual uma das etapas é a fragmentação aleatória, que pode ser mecânica ou enzimática (LOMAN *et al.*, 2012a). Por fim, adaptadores são ligados aos fragmentos, que em seguida são amplificados (quando necessário) e sequenciados.

Além do processo de fragmentação simples (*single-end*), em que fragmentos de DNA são cortados aleatoriamente, é possível utilizar métodos para construção das chamadas bibliotecas pareadas, que auxiliaram no processo de montagem. Elas podem ser *mate-pair* ou *paired-end*.

Nas bibliotecas *mate-pair* (inserto longo), o DNA é fragmentado em tamanhos definidos (alguns valores típicos são 3kb, 6kb, 8kb ou 20kb). Em seguida, marcadores são inseridos nas extremidades dos fragmentos. O DNA é circularizado e depois fragmentado novamente, em tamanhos adequados para leitura no sequenciador (LOMAN *et al.*, 2012a). Como a distância entre as sequências-alvo é conhecida, essa informação poderá ser utilizada no processo da montagem.

O sequenciamento com bibliotecas *paired-end* (insertos curtos) tem similaridades com o sequenciamento *mate-pair*, porém os fragmentos de DNA são sequenciados a partir de cada extremidade, sem a necessidade de passos adicionais de preparação de bibliotecas, como a circularização do fragmento (LOMAN *et al.*, 2012a). No sequenciamento com bibliotecas *paired-end* são sequenciadas uma das extremidades de um fragmento e uma das extremidades do reverso complementar obtido do mesmo fragmento, e apesar das sequências contidas entre as duas extremidades serem desconhecidas, essa informação pode ser utilizada na distância de pareamento entre leituras.

Recentemente, diversos modelos de sequenciadores foram desenvolvidos, como Illumina (*Genome Analyzer*), SOLiD (*Applied Biosystems / Life Technologies*), 454 GS FLX (*Roche*), PacBio RS (*Pacific Biosciences*) e Ion Torrent™ (*Life Technologies*), sendo esses capazes de usufruir de maneira eficiente das técnicas de construção de bibliotecas e sequenciamento (LAM *et al.*, 2012). Os sequenciadores podem ser divididos em instrumentos *high-end*, equipamentos de alto desempenho capazes de sequenciar dezenas e até milhares de genomas bacterianos por corrida (execução do sequenciamento), e instrumentos *benchtop*,

equipamentos de bancada com custo mais acessível (Tabela 1) (LOMAN *et al.*, 2012a).

**Tabela 1 - Comparação entre plataformas NGS**

Fonte: adaptado (LOMAN *et al.*, 2012a; JÜNEMANN *et al.*, 2013).

Sequenciador	Química	Tamanho das leituras (pb)	Tempo de corrida	Gb por corrida	Custo do equipamento (US\$)
<b>Instrumentos High-end</b>					
454 GS FLX+ (Roche)	Pirosequenciamento	~700	23 horas	0.7	500.000
HiSeq 2000/2500 (Illumina)	Terminador reversível	2 × 100	2-11 dias	120-600	750.000
5500xl SOLiD (Life Technologies)	Ligação	75 + 35	8 dias	150	350.000
PacBio RS (Pacific Biosciences)	Sequenciamento em tempo real	3.000 a 15.000	20 minutos	3 por dia	750.000
<b>Instrumentos Benchtop (bancada)</b>					
454 GS Junior (Roche)	Pirosequenciamento	500	8 horas	0.035	100.000
Ion PGM™ (Life Technologies)	Detecção de próton	100 – 200 (Novos kits: 300 – 400) <sup>1</sup>	3 horas	0.01 - 1	80.000
Ion Proton™ (Life Technologies)	Detecção de próton	200 (Novos kits: 300 – 400)	2 horas	10-100	220.000
MiSeq (Illumina)	Terminador reversível	2 × 150 (Novos kits: 2 x 250)	27 horas	1.5	125.000

Após o processo de sequenciamento são realizadas as etapas *in silico* de montagem de leituras, ordenação de *contigs* e fechamento de *gaps*. Entretanto, a definição de *softwares* específicos para realização dessas etapas depende da plataforma utilizada no sequenciamento (JÜNEMANN *et al.*, 2014).

Assim, as plataformas de sequenciamento baseadas no método de Sanger passaram a ser classificadas como primeira geração de sequenciadores, e as novas plataformas como segunda geração. Além disso, nos últimos anos surgiram novas plataformas, que mudaram os métodos de sequenciamento com o objetivo de eliminar o viés imposto pela amplificação e acelerar o processo de sequenciamento,

<sup>1</sup> Segundo Jünemann e colaboradores (2013), um novo *kit* de sequenciamento que produz leituras de 300pb está disponível. Além disso, um *kit* que produz até 400pb está em fases de teste.

e que passaram a ser conhecidas como terceira geração de sequenciadores (SMRT) (HUSEMANN, 2011).

## 1.2 Segunda e terceira geração de sequenciadores

As primeiras plataformas de sequenciamento NGS surgiram em 2005 (METZKER, 2005). O sequenciador Roche GS-FLX 454 foi o primeiro NGS a ser introduzido, e é baseado na técnica de pirosequenciamento (RONAGHI, 2001; MARGULIES *et al.*, 2005). As primeiras versões eram capazes de gerar leituras de até 100pb, enquanto as últimas versões são capazes de produzir leituras com tamanho médio de 700pb (RAMOS, 2013).

A plataforma Solexa da Illumina surgiu em 2006, permitindo o sequenciamento por síntese, que utiliza a DNA polimerase e nucleotídeos marcados com fluoróforos. A versão HiSeq 2000 V3, gera até 600Gb de dados, leituras pareadas com tamanho aproximado de 150pb e uma baixa taxa de erro de substituições pouco superior a 0,1% (RAMOS, 2013; SCHOLZ *et al.*, 2012).

A plataforma SOLiD da Applied Biosystems utiliza sequenciamento por ligação (LOMAN *et al.*, 2012a). Pequenas esferas magnéticas (*beads*) são utilizadas na aplicação da técnica de PCR em emulsão para amplificação de fragmentos de DNA. Em seguida, é feita a leitura dos fragmentos de dois em dois nucleotídeos, onde cada par de nucleotídeos é representado por uma cor. A quarta versão desse sequenciador é capaz de gerar até 100Gb, porém apresenta leituras menores que 50pb (RAMOS, 2013).

As plataformas NGS permitiram que diversos fragmentos fossem sequenciados ao mesmo tempo, aumentando a performance do sequenciamento (quantidade de dados produzida por corrida) e reduzindo o tempo de execução (EL-METWALLY *et al.*, 2013). Porém, algumas limitações surgiram se comparadas com os métodos de sequenciamento de primeira geração, como por exemplo, o tamanho máximo dos fragmentos de DNA lidos sem a perda de qualidade (Tabela 1).

Em 2011, foi criada a plataforma Ion Torrent™, que apesar de ainda necessitar de amplificação para o sequenciamento, é considerada uma das plataformas que marcaram a transição entre a segunda e terceira geração de sequenciadores devido ao baixo tempo gasto para sequenciamento. Essa plataforma trouxe um novo método de determinação do DNA baseado em detecção

da variação de pH gerada a cada incorporação de nucleotídeo, permitindo assim, uma grande redução nos custos por sequenciamento (ROTHBERG *et al.*, 2011).

Um exemplo de sequenciador de terceira geração é o PacBio RS (SCHADT, TURNER & KASARSKIS, 2010), uma das primeiras plataformas baseadas no princípio de sequenciamento de molécula única. Essa nova tecnologia veio com a promessa de sequenciamentos sem a dependência de amplificação (LOMAN *et al.*, 2012a). Apesar de ser capaz de produzir leituras de tamanho maiores que 1,5Kb, PacBio RS ainda apresenta uma taxa de erros de 12,86% (QUAIL *et al.*, 2012).

Assim, os sequenciadores de terceira geração implementaram novos e revolucionários métodos de determinação de sequências de DNA (RAMOS, 2013), podendo destacar como uma de suas principais características, o rápido tempo de corrida de sequenciamento e a alta quantidade de dados produzidos.

Além disso, nos últimos anos houve uma tendência de redução no tamanho de sequenciadores de alto desempenho (LOMAN *et al.*, 2012a). Surgiram assim, os chamados de sequenciadores de bancada (*benchtop*): sequenciadores menores e acessíveis para pequenos e médios laboratórios. Através do sequenciamento de uma linhagem da bactéria *Escherichia coli*, que causou um surto epidêmico na Alemanha, um estudo realizado por LOMAN e colaboradores (2012b) pode comprovar que mais de 95% das leituras geradas por sequenciadores de bancada puderam ser mapeadas em uma referência, demonstrando assim, bom desempenho dos mesmos.

O primeiro sequenciador de bancada foi 454 GS Junior, capaz de gerar 70Mb por corrida e leituras com tamanho aproximado de 500pb. Em 2007, surgiu o MiSeq (versão de bancada do Illumina HiSeq). O MiSeq é capaz de gerar leituras pareadas de 150pb e 1,5Gb de dados por corrida (LOMAN *et al.*, 2012a). Dentre os sequenciadores de bancada, também podemos destacar plataformas baseados na tecnologia de semicondutores Ion Torrent™, citadas anteriormente.

### 1.3 Semicondutores

As plataformas semicondutoras de bancada baseadas na tecnologia de sequenciamento Ion Torrent™ estão disponíveis em dois modelos: Ion PGM™ e Ion Proton™ (Figura 3), cuja principal diferença é a quantidade de dados produzida por corrida. O sequenciamento funciona com base na detecção de mudança de pH do

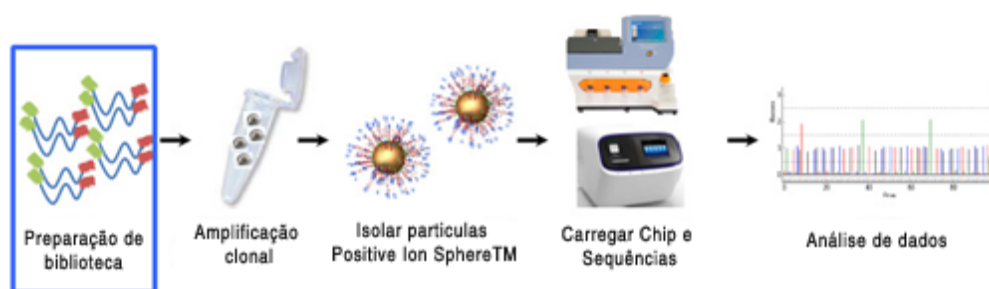
meio e trabalha com leituras de tamanho entre 100pb e 400pb (JÜNEMANN *et al.*, 2013). A versão Ion PGM™, utilizada para sequenciamentos de genoma completo, tem sua performance dada por um chip de silício: o chip 314 na versão 2 produz de 30-50Mb com leituras de 200pb e de 60-100Mb com leituras de 400pb; o chip 316 na versão 2 produz de 300-500Mb com leituras de 200pb e de 600-1000Mb com leituras de 400pb; o chip 318 na versão 2 produz de 0,6-1Gb com leituras de 200pb e de 1,2-2Gb com leituras de 400pb. O processo de sequenciamento das leituras é feito paralelamente, o que garante uma rápida execução (aproximadamente duas horas) (LIFE TECHNOLOGIES, 2014).

Plataforma de sequenciamento	 Ion PGM™ system for next-generation sequencing	 Ion Proton™ system for next-generation sequencing
Descrição	Sequenciador de bancada acurado, simples e rápido. Ideal para sequencias microbiais.	Sequenciador de bancada de alta qualidade. Ideal para sequenciamento de exoma e transcriptoma.
Benefícios	Velocidade: Tempo de corrida inferior a 2 horas.	Velocidade: Tempo de corrida inferior a 2 horas.

**Figura 3 – Modelos do sequenciador Ion Torrent™.**

Fonte: Life Technologies (adaptado). Disponível em: <<http://www.lifetechnologies.com/br/en/home/life-science/sequencing/sequencing-technology-solutions.html>>. Acesso em: 9 de dezembro de 2014.

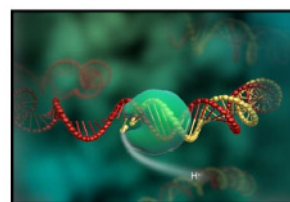
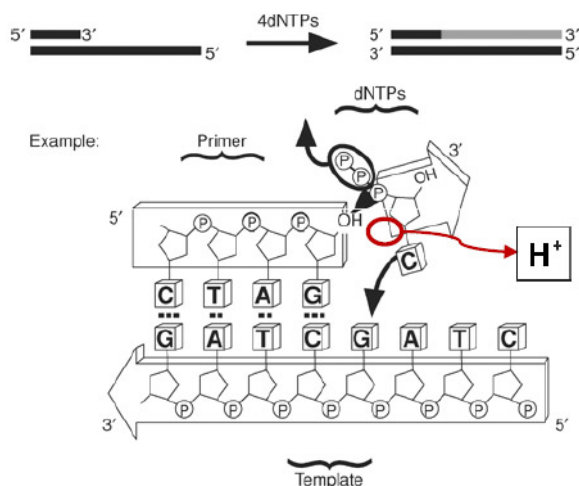
Após a extração e fragmentação do DNA, cada fragmento é ligado a uma pequena partícula e amplificado por PCR. Cada partícula é isolada e inserida em um poço específico em um *chip* de silício, onde será amplificado e sequenciado (Figura 4).



**Figura 4 – Preparação do DNA para o sequenciamento.**

Fonte: Life Technologies. Disponível em: <<http://www.lifetechnologies.com/>>. Acesso em: 9 de dezembro de 2014.

Após iniciada a corrida, o Ion liberará no *chip* uma solução contendo um dNTP específico por vez (LOMAN *et al.*, 2012a). A cada rodada, DNA polimerases farão ligações em todos os poços ao mesmo tempo, pareando a base nitrogenada timina com dATP, citosina com dGTP, adenina com dTTP e guanina com dGTP (Figura 5). A reação de polimerização libera, a cada ligação, um átomo hidrogênio, que altera o pH do meio (ROTHBERG *et al.*, 2011). Em cada poço do *chip*, um sensor ISFET detecta a mudança de pH e a converte em um sinal elétrico, que será representado por um gráfico de picos (Figura 4).



**Figura 5 – Pareamento e liberação de H+.**

Fonte: Life Technologies. Disponível em: <<http://www.lifetechnologies.com/>>. Acesso em: 9 de dezembro de 2014.

É importante ressaltar que, quando uma sequência de nucleotídeos repetitivos (homopolímeros) surge, como, por exemplo, “AAAAA”, “TTTTT”, “CCCCC” ou “GGGGG”, a DNA polimerase fará a incorporação de mais de um

dNTPs na mesma rodada. O sensor ISFET consegue processar com boa acurácia poucos nucleotídeos seguidos, perdendo a precisão com grandes sequências de homopolímeros (ROTHBERG *et al.*, 2011). Jünemann e colaboradores (2013) demonstraram que a taxa de erros de *indel* (inserção e deleção de nucleotídeos) chega a ~0,39% a cada 100pb nos sequenciamentos utilizando kits de 200pb na plataforma Ion PGM™. Apesar disso, a plataforma Ion PGM™ mostrou boa eficácia para sequenciamento de genomas procariotos (RAMOS *et al.*, 2012).

### 1.3.1 Fluxo de dados Ion Torrent™

Durante o sequenciamento, as plataformas baseadas na tecnologia Ion Torrent™ armazenam os dados em uma unidade de armazenamento local presente no sequenciador. Em seguida, os dados são transferidos para o servidor *Torrent Suite* através de uma rede *ethernet*. Por meio desse servidor é possível manipular os dados utilizando o aplicativo *Torrent Browser* (Figura 6).

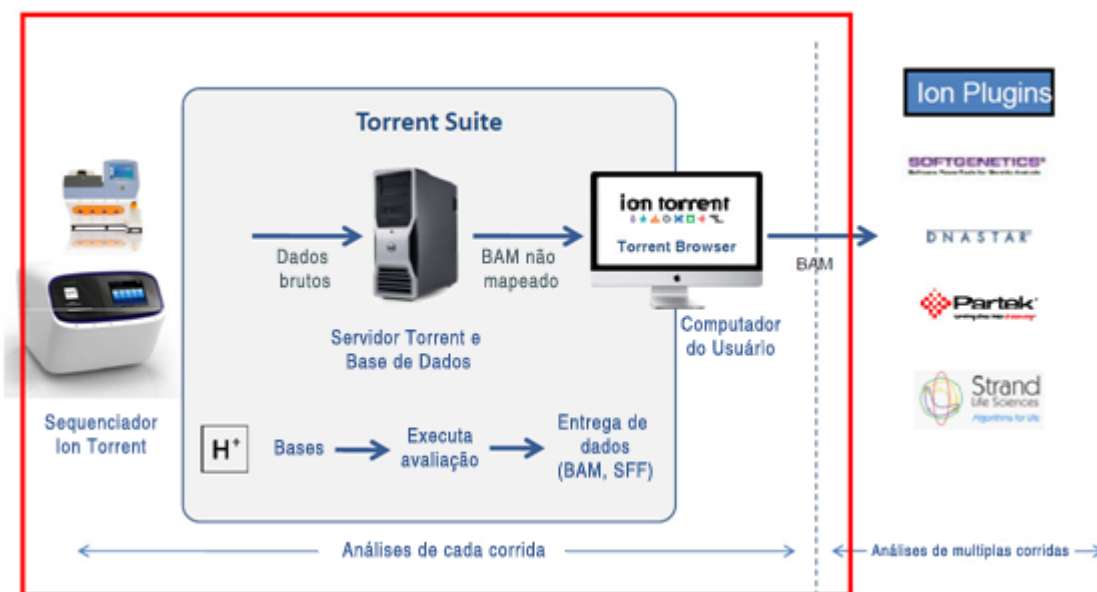


Figura 6 – Fluxo de dados do Ion Torrent™.

Fonte: *Life Technologies*. Disponível em: <<http://www.lifetechnologies.com/>>. Acesso em: 9 de dezembro de 2014.

Por fim, o *Torrent Suite*™ gera um arquivo binário de sequências (no formato BAM ou SFF), que pode ser transferido através de *Ion Plugins* para um servidor de processamento. Um segundo servidor é necessário, devido à complexidade dos processos de tratamento de dados, montagem e análise.



## 1.4 Tratamentos iniciais de dados

Antes de iniciar o processamento das leituras geradas no sequenciamento é necessário fazer conversões dos arquivos retornados pelo sequenciador. Além disso, em alguns casos é necessária a filtragem de dados de baixa qualidade.

Dentre os formatos de arquivos utilizados estão: o arquivo binário de sequências (BAM), que é uma versão da extensão SAM; o formato SFF, bastante utilizado para codificar resultados de sequenciamentos por plataformas da empresa Life Technologies; o arquivo comum de sequências (FASTA), que pode estar nas extensões FASTA, FA, FAS, FNA, dentre outros (Figura 7); além do arquivo de sequências e suas respectivas qualidades de base (FASTQ), formato mais comumente utilizado por *softwares* de montagem (Figura 8). O índice de qualidade das bases é baseado no algoritmo *Phred* (EWING *et al.*, 1998), e infere uma probabilidade de erro para os quatro nucleotídeos possíveis em cada base (Tabela 2).

**Tabela 2 - Probabilidade de erro para cada base segundo o algoritmo de *Phred*.**

Fonte: (RAMOS, 2011).

Qualidade Phred	Probabilidade de Erro	Precisão
10	1 em 10	90%
20	1 em 100	99%
30	1 em 1000	99,9%
40	1 em 10000	99,99%
50	1 em 100000	99,999%

```
>sequencia_1
ATCGATCGATGCTAGCATCGATCGATCGATCAGCTAGCTAGCTACTACGATCGATCA
GTCAGCTAGCATGCATCGATCGATCGATCGATCGATCAGCTAGCTAGCTAGCATCGA
GCTAGCTAGCATCGATCGATCGATCGATCAGTCAGCATGCATGCATCGATGCACACACA
CACACCACACACACGTGTGTCAGCTAGGCTCGCGCGCGCCCGTACGATCGGCCAC
ATCGATCGATGCTAGCATCGATCGATCGATCGATCAGCTAGCTAGCTACTACGATCGATCA
>Sequencia_2|
ATCGATCGATGCTAGCATCGATCGATCGATCAGCTAGCTAGCTACTACGATCGATCA
GTCAGCTAGCATGCATCGATCGATCGATCGATCGATCAGCTAGCTAGCTAGCATCGA
GCTAGCTAGCATCGATCGATCGATCGATCAGTCAGCATGCATGCATCGATGCACACACA
ATCGATCGATGCTAGCATCGATCGATCGATCGATCAGCTAGCTAGCTACTACGATCGATCA
GTCAGCTAGCATGCATCGATCGATCGATCGATCGATCAGCTAGCTAGCTAGCATCGA
```

**Figura 7 – Exemplo de arquivo de sequências: FASTA. O arquivo pode conter uma ou mais sequências (multifasta).**

```

@EFYTT:00004:00042
tcagtctggatgacgatGAAGATAT
+
????????????????//,,,,,
@EFYTT:00004:00043
tcagtctggatgacgatGCTCAAAGCGGGGCGAAAAATCACACTAGCGAATG
ACTTTTTTTTTCACTTTTTTTAGTCGAGGGGAGACAGTTTTTCAAATGTCTGGACA
AACCCCAT
+
????????????????
11999904:C;C:CCCCC3CBBA@ABB98859994999999&4444;;;
?':??=-BB=A:4444844,444-488811,////+/880///
@EFYTT:00005:00044
tcagtctggatgacgatCTCACGTCCCAAAGCTCCATAGACTCGAATAGCG
CGGAGAAATAGCTTGTCCGAAACCCACGCGCAATGCACCATAGGCAACG
CGTGTTTAAGTTCTA
GATCCTCGATGAGGGCTTTCGACGCGCCGCTTCAATAGCCAGGAGGTGCTC
CACTCCTCAAGGATTCAGGAGC]

```

**Figura 8 – Exemplo de arquivo FASTQ. O arquivo contém uma sequência de nucleotídeos concatenada a dados codificados de qualidade das bases.**

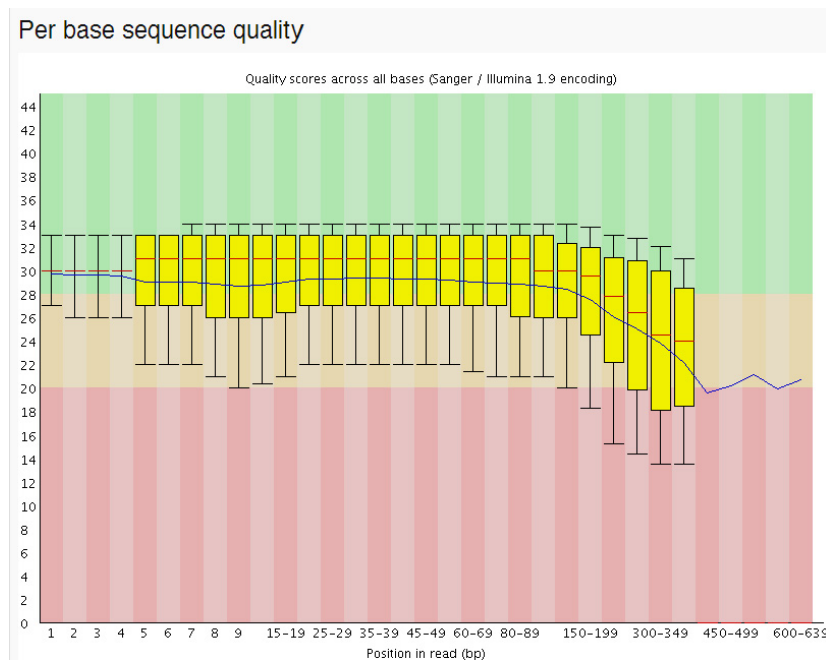
Para a conversão de formatos podem ser utilizados diversos *scripts*, como o *bam2sff*<sup>2</sup>, que converte um arquivo de extensão BAM para um arquivo no formato SFF, e *sff\_extract*<sup>3</sup>, que extrai de um arquivo SFF, outros arquivos nos formatos FASTQ, FASTA ou XML.

Nessa etapa, também é possível analisar a qualidade das bases através de gráficos usando o programa FastQC<sup>4</sup> (Figura 9). Se detectadas leituras com baixa qualidade nas extremidades, pode-se eliminá-las utilizando *in-house scripts*, ou então, utilizar essa informação como parâmetro na montagem para obtenção de melhores resultados.

<sup>2</sup> Disponível em: <<https://github.com/iontorrent/TS/blob/master/Analysis/Converter/bam2sff.cpp>>. Acesso em: 9 de dezembro de 2014.

<sup>3</sup> Disponível em: <[http://bioinf.comav.upv.es/sff\\_extract/](http://bioinf.comav.upv.es/sff_extract/)>. Acesso em: 9 de dezembro de 2014.

<sup>4</sup> Disponível em: <<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>>. Acesso em: 9 de dezembro de 2014.



**Figura 9 – Exemplo de gráfico gerado pelo FastQC.**

O gráfico de qualidade de leituras gerado pelo FastQC exibe o índice de qualidade Phred para posições específicas das leituras. Os índices de qualidade são exibidos por meio de diversas métricas, como variação de índices, maior e menor valores (blocos na cor amarela), e a média (linha azul). No eixo Y tem-se o índice de qualidade, enquanto no eixo X, o tamanho médio das leituras. A faixa na cor verde indica sequências com boa qualidade, enquanto vermelhas indicam baixa qualidade. Nessa figura é demonstrada uma perda de qualidade nas últimas bases das leituras.

## 1.5 Paradigmas para montagem de genomas

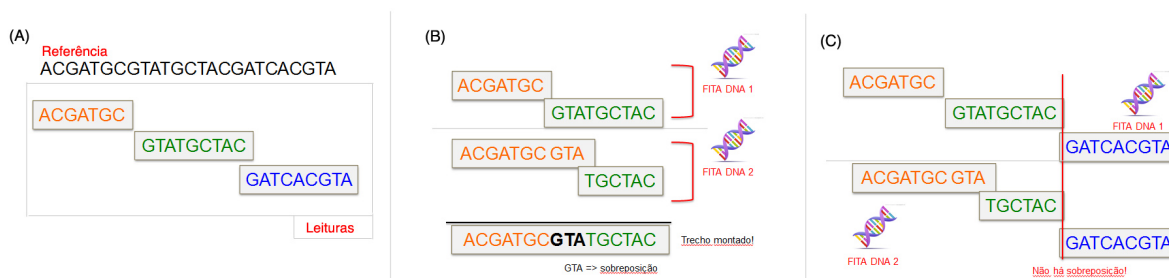
Como descrito anteriormente, o tamanho curto das leituras que os sequenciadores conseguem decodificar com precisão gerou a necessidade de fragmentar o DNA em pequenos pedaços antes do sequenciamento, sendo preciso após essa etapa, reordenar os fragmentos (leituras) de forma que possam representar o genoma original (MILLER *et al.*, 2010). Esse processo é conhecido como montagem de genomas.

Quando uma fita de DNA é quebrada em pequenos fragmentos (leituras), a ordem desses fragmentos é desconhecida. Logo, não é possível determinar a posição das leituras sem o auxílio de uma sequência referência (Figura 10A). Porém, na maioria dos casos não há uma “sequência referência” para fazer o alinhamento. Uma solução seria quebrar uma segunda molécula de DNA em pontos aleatórios e sobrepor os fragmentos de uma fita sobre os fragmentos de outra fita. Os pontos de sobreposição definirão a ordem correta das leituras, e a sequência final obtida será conhecida como “sequência consenso” ou *contig* (Figura 10B)

(MILLER *et al.*, 2010). Se não houver sobreposição entre leituras de fitas diferentes, haverá grande dificuldade em ordenar as leituras, e será considerado que o trecho não possui cobertura (Figura 10C) (POP, 2009).

Define-se como uma região com **cobertura**, uma determinada sequência de nucleotídeos de um genoma que pode ser sequenciada e representada nos dados brutos, sendo a **largura da cobertura** a porcentagem de bases do genoma original sequenciada (espera-se em um bom sequenciamento que 100% das bases sejam representadas), a **profundidade da cobertura** o número de vezes que uma base específica é representada nos dados brutos do sequenciamento e a **cobertura teórica**, ou **cobertura esperada**, o número de vezes médio que cada base é representada (SIMS *et al.*, 2014).

Assim, torna-se necessário ampliar a quantidade de fitas de DNA fragmentadas, a fim de proporcionar ligações entre todas as sobreposições de leituras e aumentar a cobertura esperada.



**Figura 10 – Alinhamento entre leituras.**

(A) Após fragmentação de uma única fita de DNA e seu sequenciamento, só é possível definir a ordem correta dos fragmentos alinhando-os contra uma referência. (B) Se houver uma segunda fita de DNA fragmentada, é possível fazer um alinhamento entre as leituras e detectar pontos de sobreposição, que indicarão a ordem das leituras. A sequência resultante será conhecida como *contig*. (C) Se uma leitura não obtiver sobreposição com outra leitura oriunda de uma segunda molécula de DNA é impossível definir a ordem de posicionamento dessa leitura, e pode-se dizer que não existe profundidade de cobertura que comprove a ligação entre os *contigs* formados.

Pop (2009) compara o processo de montagem de genomas com a montagem de um quebra-cabeça. A dificuldade em se montar um quebra-cabeças é maior quando há um grande número de peças de pequeno tamanho, e muitas delas são idênticas ou bastante similares. Assim como na montagem de genomas, a dificuldade cresce quando o genoma sequenciado possui um grande número de regiões repetitivas e o tamanho do fragmento das leituras é pequeno.

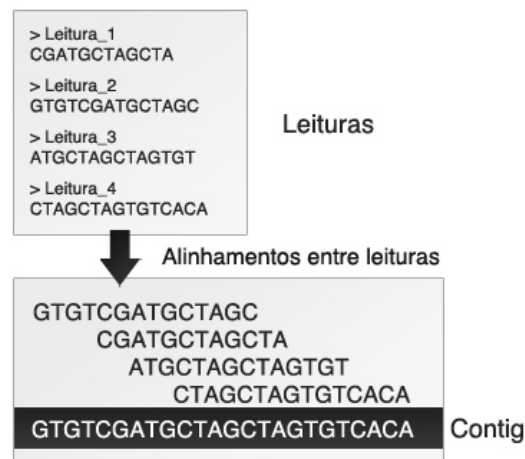
Montagem de genomas é um processo complexo, dependente das limitações dos sequenciadores e das características dos organismos sequenciados, como regiões redundantes. Por isso, requer a adoção de heurísticas, através de diversas abordagens. Apesar de terem sido propostas diversas estratégias para montagem de genomas nos últimos anos, ainda não existe um consenso sobre qual a melhor abordagem. Assim, foram desenvolvidas diversas estratégias híbridas, com diferentes plataformas e programas, buscando alcançar montagens bem sucedidas (RAMOS *et al.*, 2012; KIRCHER & KELSO, 2010; CERDEIRA *et al.*, 2011).

O uso de bibliotecas pareadas também corresponde a uma boa estratégia para auxiliar na montagem de genomas. Através dessas bibliotecas é possível conhecer a distância entre duas leituras específicas (NAGARAJAN & POP, 2013). Assim, se uma leitura montada em um determinado *contig*, possui um pareamento com outra leitura montada em outro *contig* é possível definir a posição de um em relação ao outro.

Essencialmente, os paradigmas de montagem de genomas podem ser divididos em dois tópicos: montagem *de novo* e montagem por referência (MILLER *et al.*, 2010).

### **1.5.1 Montagem de novo**

Na montagem *de novo*, também conhecida como *ab initio*, as leituras são comparadas entre si, sem a necessidade de um genoma referência, sendo então sobrepostas para formação de *contigs* (Figura 11). Os *softwares* que utilizam essa abordagem podem funcionar com base em três algoritmos distintos: algoritmo guloso, OLC (*overlap-layout-consensus*) e grafo De Bruijn (NAGARAJAN & POP, 2013; MILLER *et al.*, 2010).



**Figura 11 – Sobreposição/alinhamento entre leituras para formar uma sequência consenso (*contig*).**

### **1.5.1.1 Algoritmo guloso**

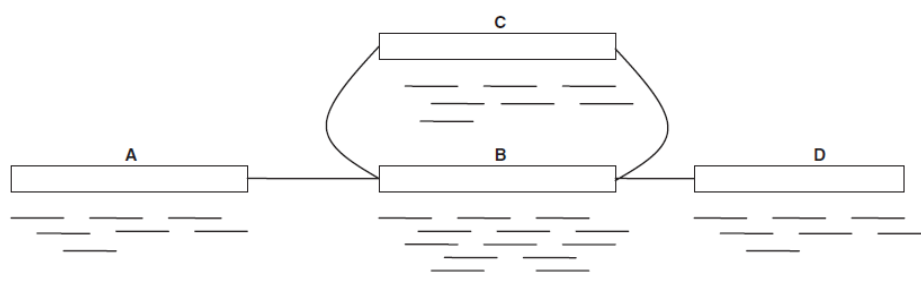
Nesse algoritmo, uma determinada leitura é alinhada com outra que possua o maior grau de similaridade (POP, 2009). Devido à existência de grandes regiões repetitivas nos genomas, esse algoritmo algumas vezes pode tomar decisões incorretas, alinhando leituras em lugares errados. Apesar disso, diversos *softwares* de montagem baseados em algoritmos gulosos incluem heurísticas projetadas para evitar montagens erradas de regiões repetitivas (NAGARAJAN & POP, 2013).

Alguns exemplos de softwares de montagem que utilizam esse algoritmo: SSAKE (WARREN *et al.*, 2007), VCAKE (JECK *et al.*, 2007) e o SHARCGS (DOHM *et al.*, 2007).

### **1.5.1.2 OLC (*overlap-layout-consensus*)**

A abordagem OLC (*overlap-layout-consensus*) é dividida em três etapas. Na primeira etapa são feitas comparações entre todas as leituras. Na segunda etapa, leituras parecidas são interligadas através de grafos de sobreposição, onde os nós representarão as leituras, e esses são ligados por arestas, quando houver sobreposição. Na terceira etapa é construído um caminho que atravesse todos os nós. Em alguns casos, a construção de um caminho único, chamado caminho hamiltoniano, é extremamente difícil, como no caso da figura 12. Nesse caso, B e C são regiões repetitivas. Uma possível solução para essa montagem seria formar dois

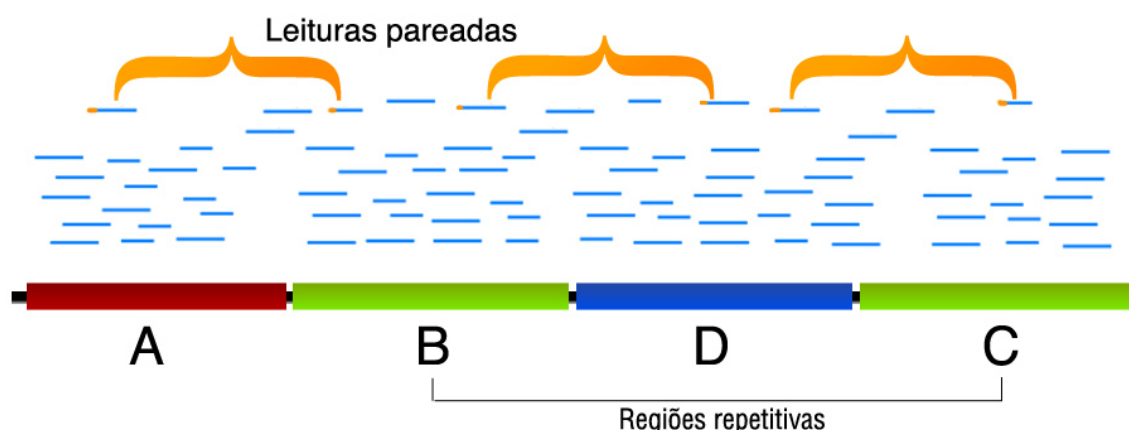
*contigs* (ABD, C), porém não é possível determinar precisamente se C ou B estão ligados a A ou a D (POP, 2009).



**Figura 12 – Construção do leiaute: sobreposição entre *contigs* repetidos. Pode-se observar o aumento relativo da cobertura da região quando B e C são sobrepostos.**

Fonte: POP (2009).

O uso de leituras pareadas pode ser uma solução para esse problema. Como visto na figura 13, a inserção de leituras pareadas, cujas distâncias são conhecidas, auxilia na correta montagem do *contig* (ABDC).



**Figura 13 – Leituras pareadas usadas para correta montagem de regiões repetitivas.**

Observa-se que a distância entre as leituras pareadas representadas na cor laranja indicam que uma determinada leitura inserida em um determinado ponto deve estar a uma distância pré-determinada de seu par. Assim, a única ligação possível entre os blocos é a ordem A, B, D, C.

Montagens com a abordagem OLC são mais precisas com leituras de maior tamanho mesmo que tenham menor acurácia (NAGARAJAN & POP, 2013), como as produzidas pelos sequenciadores 454 Roche, Sanger ou Ion PGM™. Dentre os *softwares* que utilizam OLC, pode-se citar Newbler<sup>5</sup>, Edena<sup>6</sup> (HERNANDEZ *et al.*,

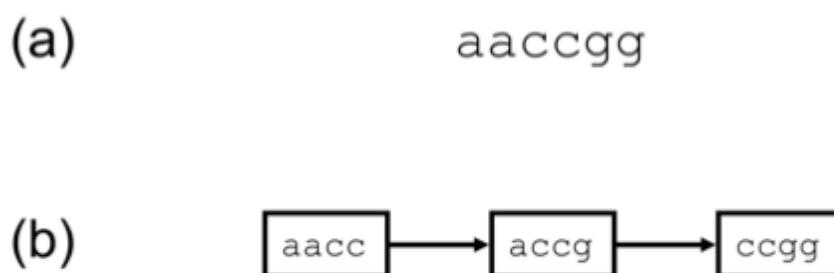
<sup>5</sup> Disponível em: <<http://www.454.com/products/analysis-software/>>. Acesso em: 7 de agosto, 2014.

<sup>6</sup> Disponível em: <<http://www.genomic.ch/edena.php>>. Acesso em: 7 de abril, 2014.

2008), e Mira (CHEVREUX *et al.*, 2004; CHEVREUX, WETTER & SUHAI, 1999). Newbler foi desenvolvido para processar dados de pirosequenciamento, sendo eficaz no tratamento de regiões de homopolímeros e pode ser executado através de linha de comando (CLI) ou através de interface gráfica (gsAssembler). Enquanto Edena foi desenvolvido para trabalhar com leituras muito curtas, Mira foi desenvolvido para ser um *software* de montagem “inteligente”, capaz de aprender com erros. Assim, Mira é capaz de descobrir e analisar possíveis erros de homopolímeros. Essa característica em especial, faz do Mira, uma boa ferramenta para tratar dados com grande taxa de erros de homopolímeros, como por exemplo, dados obtidos por sequenciamento em Ion PGM™. Além disso, Mira tem sido bastante utilizado para comparação de performance entre plataformas de sequenciamento *benchtop*, como por exemplo, nos testes realizados por Jünemann e colaboradores (2013).

### **1.5.1.3 Grafo De Bruijn**

Na abordagem do grafo De Bruijn, as leituras são quebradas em *k-mers* (*substrings* de um determinado fragmento) e interligadas através de um grafo (Figura 14). Nessa representação, só existe uma ligação se houver uma sobreposição entre dois *k-mers* de tamanho  $k-1$  (Figura 15) (MILLER *et al.*, 2010). Nessa abordagem todos os nós do grafo são interligados a fim de se construir um caminho euleriano, ou seja, um grafo em que todos os caminhos sejam percorridos.

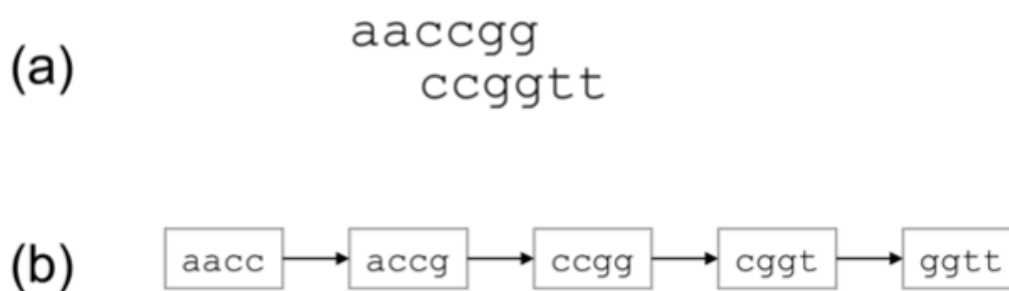


**Figura 14 – Representação de como uma leitura é quebrada em *k-mers* no grafo De Bruijn.**

(A) Leitura “aaccgg” de tamanho 6. (B) A leitura é quebrada em *substrings* de tamanho 4 ( $K = 4$ ). Cada nó é ligado a outro se os três últimos caracteres forem similares aos três primeiros caracteres do outro ( $K-1=3$ ). Nesse caso, a ligação é óbvia, pois pertence à mesma leitura. Mas esse conceito será importante para representar alinhamentos entre leituras diferentes.

Fonte: adaptado (MILLER *et al.*, 2010).





**Figura 15 – Alinhamento entre duas sequências no grafo De Bruijn.**

(A) Para o alinhamento das sequências “aaccgg” e “ccggtt” cada sequência é dividida em subsequências de tamanho 4. (B) É possível exibir um caminho único passando por ambas as sequências detectando sobreposições de tamanho  $k-1$ . É importante dizer que essa estratégia de divisão em  $k$ -mers também é utilizada na etapa de comparação par-a-par da abordagem OLC. Fonte: adaptado (MILLER *et al.*, 2010).

Essa abordagem tem sido bastante utilizada por *softwares* de montagem, pois tem um custo computacional menor e não realiza comparações entre todas as leituras. Entretanto, os grafos De Bruijn são mais sensíveis à qualidade dos dados oriundos dos sequenciadores. Na construção do grafo, cada nó só é interligado se as sequências (de tamanho  $k-1$ ) forem idênticas. Logo, erros de sequenciamento, aumentarão a complexidade de elaboração do grafo De Bruijn, podendo impedir a elaboração correta do grafo. Surge então a necessidade de algoritmos sofisticados de correção de erro (PEVZNER & TANG, 2001; CHAISSON, BRINZA & PEVZNER, 2009).

Assim, *softwares* que utilizam a abordagem do grafo De Bruijn são mais eficazes com sequências curtas de grande acurácia, como por exemplo, as obtidas nos sequenciadores da plataforma Illumina/Solexa (NAGARAJAN & POP, 2013). São exemplos de *softwares* que utilizam o grafo De Bruijn: Velvet<sup>7</sup> (ZERBINO *et al.*, 2008), SOAPdenovo<sup>8</sup> (LI *et al.*, 2010), Minia<sup>9</sup> (CHIKHI & RIZK, 2012) e SPAdes<sup>10</sup> (BANKEVICH *et al.*, 2012).

<sup>7</sup> Disponível em: <<https://www.ebi.ac.uk/~zerbino/velvet/>>. Acesso em: 7 de abril, 2014.

<sup>8</sup> Disponível em: <<http://soap.genomics.org.cn/soapdenovo.html>>. Acesso em: 7 de abril, 2014.

<sup>9</sup> Disponível em: <<http://minia.genouest.org/>>. Acesso em: 7 de abril, 2014.

<sup>10</sup> Disponível em: <<http://bioinf.spbau.ru/spades>>. Acesso em: 7 de agosto, 2014.

### 1.5.2 Montagem por referência

Na montagem por referência, ou montagem por mapeamento, todas as leituras são mapeadas no genoma de um organismo filogeneticamente próximo. Essa estratégia é uma alternativa à montagem *de novo*, e tem como um de seus objetivos a detecção de variações genéticas entre organismos, como por exemplo, detecção de substituições de nucleotídeos (SNP) (Figura 16), inserções ou deleções (*indels*) (MILLER *et al.*, 2010).

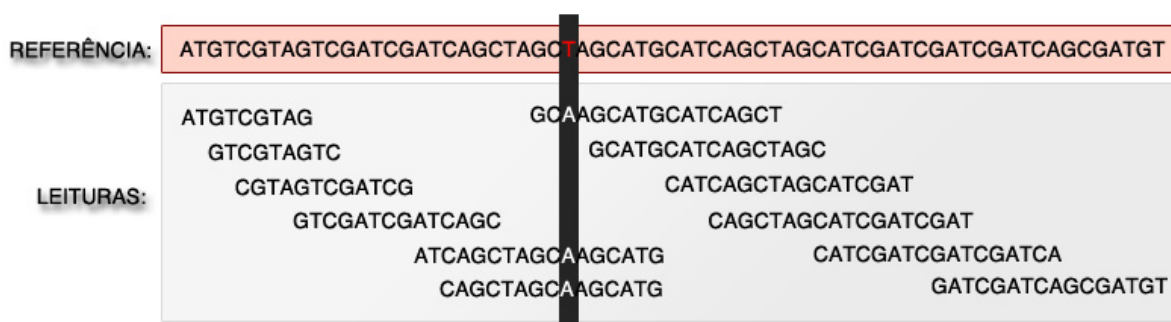


Figura 16 – Alinhamento das leituras contra uma referência.

A montagem por referência também pode ser utilizada na etapa de finalização de montagens. Na montagem por referência é possível distribuir uniformemente as leituras resolvendo o problema de regiões repetitivas (Figura 17; Figura 18). Além disso, assim como na montagem *de novo*, é possível construir *contigs* através de mapeamentos (MILLER *et al.*, 2010). Entretanto, a abordagem *de novo* permite que regiões inexistentes no organismo referência sejam detectadas. Por isso, a abordagem *de novo* é considerada mais adequada para novos projetos de sequenciamento, apesar de exigir uma maior demanda computacional e requerer um tempo maior para execução (POP, 2009).

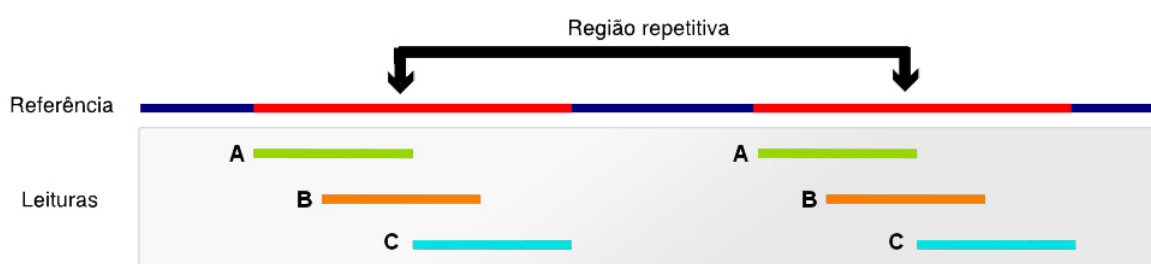


Figura 17 – Montagem por referência. Há distribuição das leituras sobre as regiões repetitivas.

Fonte: adaptado (RAMOS, 2011).

Diversos programas e algoritmos similares aos utilizados na montagem *de novo* podem ser empregados na montagem por referência. Dentre esses programas pode-se citar Mira<sup>11</sup> (CHEVREUX *et al.*, 2004) e SOAP<sup>12</sup> (LI *et al.*, 2008). Pode-se citar também *softwares* proprietários, como CLC Workbench<sup>13</sup> e SeqMan DNASTAR<sup>14</sup> (MILLER *et al.*, 2010).



Figura 18 – Comparação entre o alinhamento de leituras montadas *de novo*. Na montagem *de novo* (sem parâmetros de distribuição uniforme) as leituras idênticas serão sobrepostas permitindo a formação de *gaps*.

## 1.6 Finalização de montagens

Durante a montagem *de novo*, muitas vezes o *software* detecta pontos de baixa ou alta cobertura, optando assim, por encerrar a adição de leituras àquele *contig*. Esse é um dos possíveis motivos pelo qual um genoma tende a não ser completamente montado na etapa *de novo*. Assim, a finalização de montagens corresponde ao processo de ordenação dos *contigs* obtidos na montagem e, em seguida, na determinação de sequências que interligam um ao outro: regiões conhecidas como *gaps* (NAGARAJAN & POP, 2013; POP, 2009; RIBEIRO *et al.*, 2012).

Um dos principais fatores que dificultam a finalização de montagens são as regiões de sequências repetitivas, como por exemplo, regiões onde estão inseridos *operons* que codificam as unidades do rRNA (RNA ribossomal) 16S, 23S e 5S, além

<sup>11</sup> Disponível em: <<http://mira-assembler.sourceforge.net/>>.

<sup>12</sup> Disponível em: <<http://soap.genomics.org.cn/>>.

<sup>13</sup> Disponível em: <<http://www.clcbio.com/>>.

<sup>14</sup> Disponível em: <<http://www.dnastar.com/>>.

de regiões de elementos transponíveis. O *operon* de rRNA constitui numa região com tamanho aproximado de 5Kb, cujas sequências têm em média entre 98,04% e 99,94% de similaridade (BASHIR *et al.*, 2012). Os elementos transponíveis são outros tipos de componentes repetitivos, que podem ser: transposons ou retrotransposons (WICKER *et al.*, 2007; FINNEGAN, 1989).

Muitas vezes, *softwares* de montagem não são capazes de resolver problemas de repetições e de regiões de baixa cobertura, impedindo assim, que o genoma seja completamente montado. Logo, o processo de finalização demanda maior tempo e custo para sua realização do que outros processos, como o sequenciamento, tratamento inicial de dados ou até mesmo a montagem *de novo* (NAGARAJAN *et al.*, 2010).

Na primeira etapa do processo de finalização, as sequências contíguas (*contigs*) são orientadas e posicionadas corretamente. Esse processo é conhecido como *scaffolding*, construção de *scaffolds* ou ordenação de *contigs/supercontigs* (NAGARAJAN & POP, 2013).

Quando um *contig* é conectado a outro, através da sobreposição entre extremidades, forma-se um *supercontig* (Figura 19). A cada *contig* ou *supercontig* alinhado em sua posição final no genoma, dá-se o nome de *scaffold* (MILLER, KOREN & SUTTON, 2010). Quando a região entre dois *contigs* for desconhecida, dá-se o nome de *gap*, região tipicamente representada pelo caractere “N”.

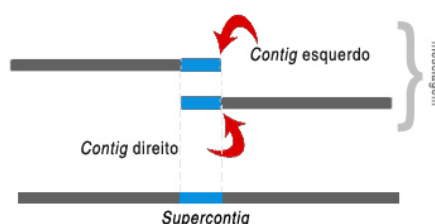


Figura 19 – Formação de *supercontig* através da mesclagem da sobreposição entre extremidades de dois contigs vizinhos.

### 1.6.1 Ordenação de contigs

Para ordenar *contigs* pode-se utilizar uma estratégia *de novo*, a qual *contigs* são comparados entre si e alinhados com base em similaridades entre extremidades (formação de *supercontigs*). A distância conhecida entre leituras pareadas também

pode ser utilizada para ordenação de *contigs*, sendo possível assim, determinar a quantidade exata de nucleotídeos desconhecidos (N) no *gap*.

Dentre os programas que realizam a construção de *scaffolds* pode-se citar o SSPACE<sup>15</sup> (BOETZER *et al.*, 2010), que utiliza informações sobre tamanho do inserto e posição do pareamento para efetuar a ordenação, e o G4ALL<sup>16</sup> (RAMOS *et al.*, 2013), que utiliza resultados de alinhamentos realizados pela ferramenta BLAST (ALTSCHUL *et al.*, 1990) para exibir sobreposições em extremidades e permitir que o usuário efetue as modificações manualmente. Alguns *softwares* de montagem possuem módulos inclusos para construção de *scaffolds*, como o Velvet e o SPAdes.

Pode-se também utilizar uma abordagem por referência para ordenação, na qual os *contigs* são alinhados com base em um genoma referência. Essa abordagem é eficaz para comparação entre organismos filogeneticamente próximos, porém perde a eficácia se não houver referências confiáveis disponíveis (MILLER *et al.*, 2010). Entretanto, a eficácia dessa estratégia tem se aprimorado à medida que montagens de dados oriundos de novos NGS têm gerado *contigs* maiores e em menor quantidade.

Dentre os *softwares* que realizam ordenação de *contigs* por referência pode-se citar o CONTIGuator<sup>17</sup> (GALARDINI *et al.*, 2011), que foi desenvolvido por meio da linguagem de programação Python e a biblioteca Biopython, e é disponibilizado sob a licença GNU/GPLv.3.0, que permite que o *software* possa ser executado, distribuído e modificado livremente.

Por fim, os *scaffolds* também podem ser construídos com o auxílio de mapeamentos ópticos (SAMAD *et al.*, 1995). Com esse tipo de mapeamento é possível determinar a localização aproximada de sítios de enzimas de restrição, e assim estimar a posição de *contigs* dentro do genoma (POP, 2009).

Mapeamento óptico de genoma completo é uma técnica que utiliza mapas de restrição de alta resolução para determinar a ordem correta de *contigs* dentro de um genoma (ANANIEV *et al.*, 2008; XAVIER *et al.*, 2014). Nessa técnica, inicialmente o DNA é estendido, imobilizado em uma lâmina e fragmentado por uma enzima de restrição. Então, uma imagem é gerada pelo equipamento e um *software* é utilizado

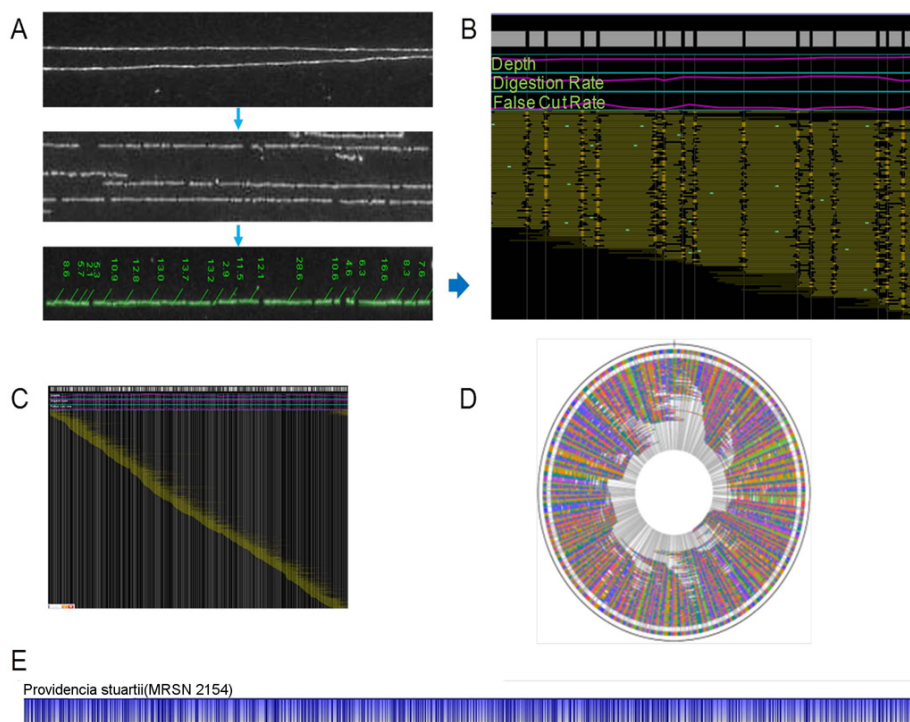
---

<sup>15</sup> Disponível em: <<http://www.baseclear.com/landingpages/basetools-a-wide-range-of-bioinformatics-solutions/sspacev12>>. Acesso em: 7 de agosto, 2014.

<sup>16</sup> Disponível em: <<http://g4all.sourceforge.net/>>. Acesso em: 7 de agosto, 2014.

<sup>17</sup> Disponível em: <<http://contiguator.sourceforge.net/>>. Acesso em: 7 de agosto, 2014.

para determinar o tamanho dos fragmentos. Todos os mapas de fragmentos são sobrepostos a fim de formar um mapa de restrição consenso do genoma completo. Por fim, é utilizado o *software* MapSolver™ da OpGen para alinhar as sequências *in silico* com o mapa de restrição (Figura 20) (ONMUS-LEONE *et al.*, 2013).

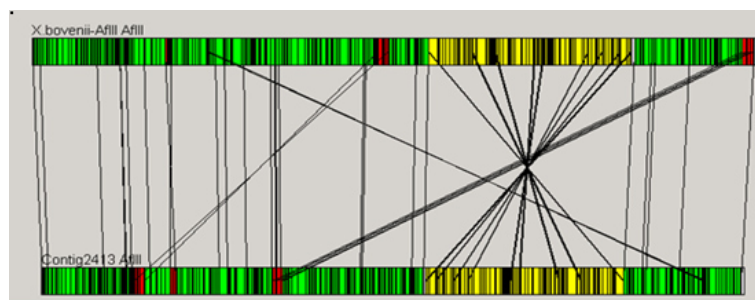


**Figura 20 – Exemplo de mapeamento óptico.**

(A) Várias fitas de DNA são imobilizadas em uma lâmina e digeridas por uma enzima de restrição. Através da imagem gerada, o *software* estima o tamanho dos fragmentos. (B) Em seguida, o *software* alinha os fragmentos a fim de gerar um mapa consenso. (C) Esse processo é repetido com todos os fragmentos, formando um mapa de restrição do genoma inteiro. (D) O mapa consenso do genoma completo é representado em forma circular. (E) Mapa do organismo *Providencia stuartii* MRSN 2154 representado pelo *software* MapSolver. Cada linha azul representa um ponto de corte da enzima de restrição.

Fonte: Onmus-Leone e colaboradores (2013).

Nos últimos tempos, a técnica de mapeamento óptico tem sido utilizada para construção de *scaffolds* com alta precisão, sendo possível, determinar grandes inversões que não seriam detectadas sem a ajuda dessa técnica (Figura 21) (LATREILLE *et al.*, 2007).

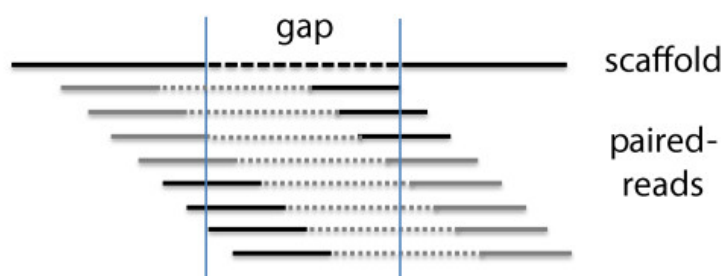


**Figura 21** – Uso do mapa óptico para detecção de grandes inversões em *Xenorhabdus bovienii*, vistas na cor amarelo; em vermelho, pode-se perceber regiões montadas erradas (*mismatches*); em verde regiões corretamente montadas.

Fonte: Latreille e colaboradores (2007).

### 1.6.2 Fechamento de gaps *in silico*

A etapa de fechamento de *gaps in silico* busca resolver regiões de nucleotídeos desconhecidos entre *contigs* através de métodos computacionais heurísticos. O uso de leituras pareadas pode facilitar a execução desse processo. O *software* GapFiller<sup>18</sup> (BOETZER & PIROVANO, 2012) é um exemplo de ferramenta que utiliza bibliotecas pareadas para fechamento automático de *gaps* (Figura 22).



**Figura 22** – Uso de leituras pareadas para fechamento de *gaps* no GapFiller.

Fonte: adaptado (BOETZER & PIROVANO, 2012).

Em projetos de sequenciamento de organismos, que possuam algum genoma filogeneticamente próximo já sequenciado, é possível utilizar técnicas de mapeamento dos dados brutos contra um genoma referência para extração da sequência consenso e posterior fechamento do *gap*. O *software* CLC Genomics Workbench permite a extensão manual de *scaffolds*, através de mapeamento dos *scaffolds* nos dados brutos (CERDEIRA *et al.*, 2011).

<sup>18</sup> Disponível em: <<http://www.baseclear.com/landingpages/basetools-a-wide-range-of-bioinformatics-solutions/gapfiller/>>. Acesso em: 7 de agosto, 2014.

## 1.7 Preparação dos dados para depósito em bancos de dados públicos

Antes do depósito do genoma completamente montado em bancos de dados públicos é necessário definir padrões para organização dos dados. Um exemplo é a definição do início da fita quando se tratar do sequenciamento de genomas circulares. Diversos genomas depositados em bancos de dados públicos, como o NCBI, utilizam o gene *dnaA* para representar o início da fita de organismos com DNA circular. O gene *dnaA* está ligado ao processo de ativação do início da replicação do DNA em organismos procariotos (FOSTER & SLONCZEWSKI, 2009).

Após a finalização da montagem é necessário realizar a predição de regiões codificadoras (CDS) e anotação automática. Para realização dessa etapa podem ser utilizados os programas: FgenesB<sup>19</sup>, RAST<sup>20</sup> (AZIZ *et al.*, 2008) ou PROKKA<sup>21</sup> (SEEMANN, 2014). Por fim, os dados são depositados no banco de dados público do NCBI.

## 1.8 Problemática

Com o advento das atuais melhorias nos sequenciadores de próxima geração e nos programas que utilizam algoritmos de montagem, genomas depositados em bancos de dados públicos puderam ser ressequenciados, gerando dados com maior acurácia. Esse fato permitiu a melhoria da qualidade das montagens em comparação aos dados depositados anteriormente, além da redução de custos por sequenciamento, e vem permitindo também um aumento no número de projetos de sequenciamento de genomas completos depositados em bancos de dados públicos.

Nesse contexto, pode-se citar dois importantes problemas que ocorrem em projetos de sequenciamento: 1) elevado custo para finalização de montagens; 2) elevado gasto de tempo para treinamento de mão-de-obra.

**Finalização de montagens:** apesar da redução dos custos por sequenciamento, finalizar montagens é uma atividade com alto custo e com grande gasto de tempo. Atualmente, diversas estratégias como o uso de bibliotecas

---

<sup>19</sup> Disponível em:

<<http://www.softberry.com/berry.phtml?topic=fgenesb&group=programs&subgroup=gfindb>>.

<sup>20</sup> Disponível em: <<http://rast.nmpdr.org/>>. Acesso em: 11 de agosto, 2014.

<sup>21</sup> Disponível em: <<http://www.vicbioinformatics.com/software/prokka.shtml>>. Acesso em: 11 de agosto, 2014.



pareadas, ordenação de *contigs* por referência e mapeamento óptico, têm sido utilizadas para finalização de montagens. Logo, seria útil uma ferramenta que permitisse o uso de diferentes estratégias para finalização de montagens.

**Elevado gasto de tempo para treinamento de profissionais especializados:** outro problema detectado é a grande quantidade de programas que devem ser executados nos processos de montagem, exigindo que profissionais responsáveis por executá-los tenham um grande domínio em informática. Cabe ressaltar que esses profissionais podem ser oriundos de diversas áreas. A adoção de uma ferramenta com boa usabilidade poderia reduzir o tempo gasto com qualificação profissional.

## 1.9 Justificativa

Assim, o seguinte trabalho justifica-se na carência de ferramentas que utilizem abordagens híbridas para tratamento, montagem e finalização de genomas sequenciados com as novas plataformas NGS; e na necessidade de criação de uma ferramenta que com rápida curva de aprendizagem, que seja capaz de oferecer montagens e finalizações de genomas de forma rápida e acurada. Além disso, há a carência de ferramentas capazes de importar resultados de mapeamento óptico, e usá-los para ordenação de *contigs*.

Para desenvolvimento dessa ferramenta propõe-se o uso do ambiente Web. Diversos trabalhos têm comprovado as vantagens do uso de ferramentas Web para solução de problemas de usabilidade em bioinformática, como Orione (CUCCURU *et al.*, 2014) e Galaxy (GOECKS *et al.*, 2010). Entretanto, essas ferramentas utilizam diversos *softwares* acoplados, sendo necessário instalá-los separadamente, tornando-se mais um fator de dificuldade.

Espera-se que a construção de uma interface amigável facilite o processo de montagem de genomas, ordenação de *contigs* e fechamento de *gaps* para usuários sem formação específica ou experiência em computação, e assim, reduzir a curva de aprendizagem para realização dessas tarefas.

## 1.10 Objetivos

### 1.10.1 Objetivo geral

Desenvolver uma ferramenta Web, que será denominada SIMBA (*Simple Manager for Bacterial Assemblies*), para o gerenciamento de um *pipeline* para a automação de montagem e finalização de genomas procariotos.

### 1.10.2 Objetivos específicos

- Determinar as etapas para o tratamento, montagem e finalização de montagens com dados obtidos a partir de sequenciamento de DNA de organismos procariotos;
- Desenvolver uma ferramenta Web de automação da execução do *pipeline*;
- Executar o *pipeline* com dados de sequenciamento dos organismos *Corynebacterium pseudotuberculosis* 258 e *Corynebacterium pseudotuberculosis* 1002;
- Testar diferentes *softwares*, que utilizem os algoritmos Grafo De Bruijn e OLC, para montagens *de novo*;
- Permitir a importação de resultados de montagens obtidas por diferentes *softwares* de montagem para a ferramenta desenvolvida;
- Comparar a finalização de montagem por mapeamento óptico e por ordenação de *contigs* por referência.

## 2. Metodologia

Para correta execução dos processos de montagem e finalização foi definido um *pipeline* específico para ser executado pela ferramenta. Assim, a ferramenta SIMBA pode ser subdividida em um *pipeline*, com um conjunto de programas e estratégias para completa finalização de genomas, e a interface SIMBA, construída para ser executada através de um *web browser*. Para o desenvolvimento do *software* foi utilizada a linguagem PHP (através do *framework* Laravel<sup>22</sup>), banco de dados SQLite, diversas ferramentas externas e *scripts* construídos na linguagem Python (que executados em sequência compõe o *pipeline*).

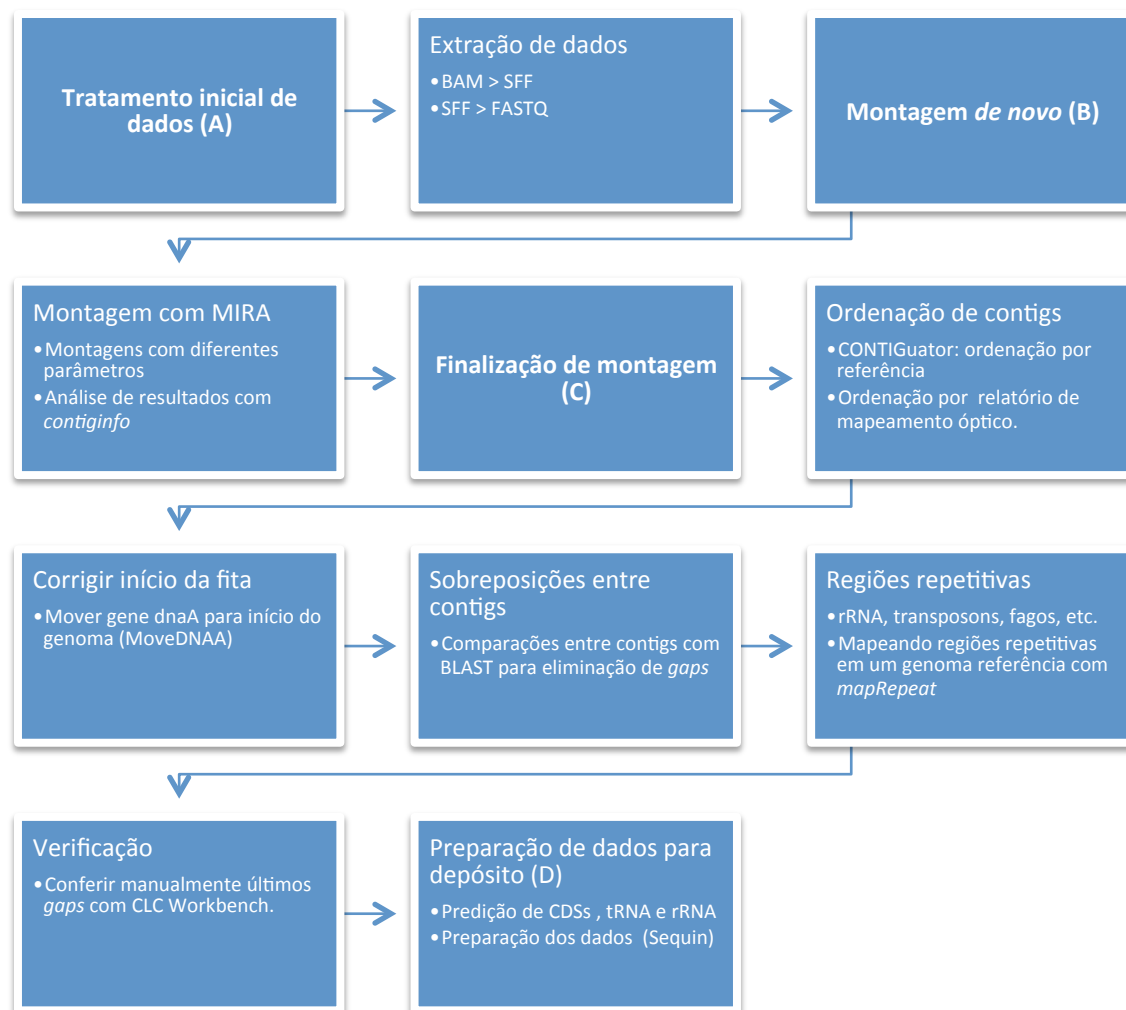
Nesta seção será apresentada a metodologia adotada em cada etapa do *pipeline*. Por fim, será descrita a infraestrutura do ambiente de produção a qual a ferramenta foi aplicada com sucesso, a metodologia para criação da interface, além de uma descrição da origem de dados que serão utilizados em um estudo de caso para atestar a eficácia da ferramenta.

### 2.1 Etapas do *pipeline*

O processo inicia-se pelo tratamento inicial de dados, seguido pela montagem *de novo*, etapas de finalização de montagem e preparação de dados para depósitos em bancos de dados públicos (Figura 23).

---

<sup>22</sup> Disponível em: <<http://laravel.com/>>. Acesso em: 11 de agosto, 2014.



**Figura 23 – Etapas do *pipeline* para montagem de genomas procariotos.**

Os processos podem ser divididos em três etapas: (A) Tratamento inicial de dados. (B) Montagem *de novo*. (C) Finalização da montagem. (D) A preparação dos dados para depósito não faz parte dos processos de montagem de genomas, e só é citado neste *pipeline* para representar o fim do processo.

### 2.1.1 Tratamento inicial de dados

Antes de iniciar o processo de montagem devem-se vincular os dados obtidos pelo sequenciamento a um determinado projeto. Uma vez obtidos os dados, é necessário analisar a qualidade das leituras. Se os dados apresentarem baixa qualidade pode-se verificar a necessidade de se refazer o sequenciamento. Essas análises podem ser feitas com o *software* FastQC.

Após essa análise é necessário converter os dados para formatos requisitados por *softwares* de montagem. Por exemplo, o *software* Mira exige um arquivo no formato FASTQ e outro no formato XML para iniciar seus procedimentos

de execução. Cada *software* tem uma exigência específica, entretanto grande parte deles aceita o formato FASTQ.

Definiu-se que o tratamento inicial de dados poderá ser feito através de dois *scripts*: BAM2SFF, que converte o arquivo binário gerado pelo sequenciador para o formato SFF, e SFF\_EXTRACT, que extrai o arquivo de sequências FASTQ. Assim, o *pipeline* proposto permite que se possa ter como entrada arquivos em três formatos distintos: FASTQ, BAM ou SFF. Porém, para iniciar a próxima etapa é necessário um arquivo no formato FASTQ, portanto os arquivos nos formatos BAM e SFF devem ser convertidos.

### **2.1.2 Montagem de novo**

Para montagem *de novo* foi definido o uso de quatro diferentes *softwares*, sendo três que utilizam o algoritmo OLC, Mira versão 3.9.18, Mira versão 4.0.2, Newbler versão 2.7, e um que utiliza o algoritmo do grafo De Bruijn, Minia versão 1.6088. Esses foram acoplados ao diretório *bin* da aplicação.

Também optou-se em permitir o uso de outros *softwares* de montagem dentro da ferramenta. Entretanto, cada *software* tem padrões específicos para entrada de dados e nomeação de arquivos e diretórios de saída, o que implica em uma dificuldade para inserção desses para execução direta por uma ferramenta Web. Entretanto, o formato FASTA é padronizado para saída de dados e exibição de *contigs*. Assim, para tentar solucionar esse problema, optou-se em utilizar sequências no formato FASTA como padrão de entrada de dados para montagens realizadas por outros *softwares* não listados acima, como por exemplo, SPAdes ou Velvet. Logo, a inserção de montagens realizadas em qualquer *software* na ferramenta pode ser feita através da opção “*Manual assembly*”, e da submissão de um arquivo no formato FASTA nomeado da seguinte forma: “tX\_out.unpadded.fasta”, onde X corresponde a versão da montagem, no diretório do projeto.

Após a finalização da montagem, o *script* CONTIGinfo<sup>23</sup> pode ser utilizado para obtenção de informações sobre a montagem, tais como: quantidade de *contigs* gerados, tamanho do menor e do maior *contig*, o tamanho total do genoma e o valor de N50. O N50 corresponde ao tamanho do *contig*, o qual corresponde a 50% do

---

<sup>23</sup> Disponível em: <<https://github.com/dcbmariano/scripts>>. Acesso em: 29 de abril, 2014.

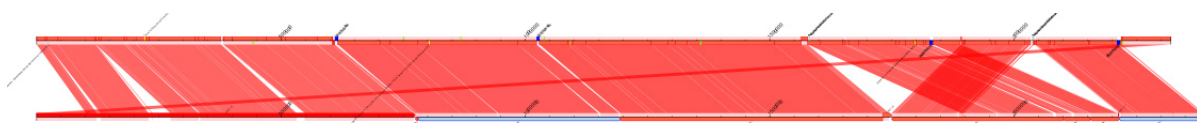
tamanho total do genoma quando todos os *contigs* são ordenados com base em seu tamanho.

### **2.1.3 Finalização de montagem**

#### **2.1.3.1 Ordenação de contigs por referência**

Para a ordenação de *contigs* baseada em referência foi utilizada uma versão modificada do programa CONTIGuator.

CONTIGuator recebe um arquivo de *contigs* gerado pelo *software* de montagem e ordena com base em um genoma referência. Por fim, CONTIGuator retorna um gráfico de alinhamento entre sequências (Figura 24), além de um arquivo com *scaffolds* e um arquivo com sequências que não puderam ser alinhadas com a referência. Foi feita uma modificação para que o CONTIGuator exiba, no gráfico de alinhamento, marcações informando a posição de aparecimento de regiões repetitivas (rRNAs, plasmídeos, transposons e fagos) no genoma referência.



**Figura 24 – Exemplo de gráfico de alinhamento gerado pelo CONTIGuator.**

A linha superior representa o genoma referência usado no alinhamento. Nela pontos em azul escuro representam regiões de RNA ribossomal; pontos em verde, regiões de fagos; pontos em amarelo, regiões de plasmídeos; e pontos em azul claro, regiões de transposons. A linha inferior representa os *contigs* do genoma recém-montado. *Contigs* em vermelho possuem sobreposição entre as extremidades com o *contig* à direita e com o *contig* à esquerda; *contigs* em laranja possuem sobreposição com apenas um dos *contigs* vizinhos; e *contigs* em azul, não possuem sobreposição entre as extremidades. As linhas que interligam as duas linhas horizontais representam sobreposições.

#### **2.1.3.2 Ordenação de contigs por mapeamento óptico**

Para ordenação de *contigs* baseada em mapeamento óptico foi criado um *parser* em PHP para ler o relatório gerado pelo OpGen MapSolver™. Esse relatório contém a lista de *contigs* ordenados, além de sua orientação.

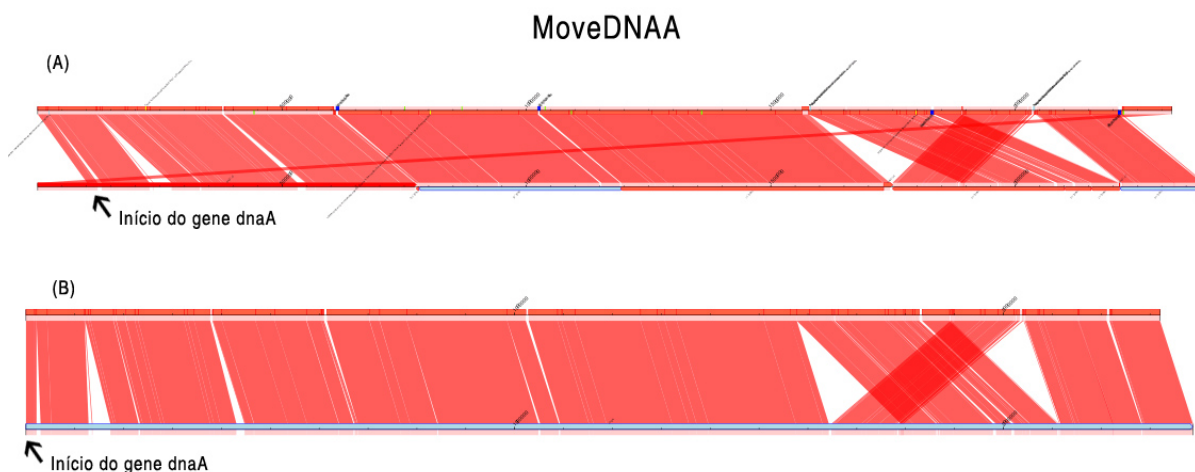
Após a leitura do relatório, o *parser* ordena, orienta e une os *contigs* para gerar um arquivo com uma única sequência, ignorando possíveis *gaps* e sobreposições entre os *contigs* orientados. Em seguida, esse arquivo é usado como referência para alinhar os *contigs* por meio do *software* CONTIGuator. Assim é

possível determinar a existência de *gaps*, gerar uma visualização gráfica de sobreposições entre *contigs* e separar *contigs* pequenos não ordenados pelo mapeamento óptico.

### **2.1.3.3 Mover o gene *dnaA* para a posição inicial do genoma**

Muitas vezes, após o processo de ordenação de *contigs*, o gene *dnaA* aparece no centro da fita. Por uma questão de padronização para depósito de dados, convém cortar a fita na posição de início do gene *dnaA* e mover o novo bloco formado para o início da fita. Para essa etapa foi criado o *script* moveDNAA<sup>24</sup>, que recebe um arquivo FASTA com *scaffold* e prediz o início da fita com base em um arquivo referência (Figura 25).

O *script* moveDNAA ajusta o início da fita com base no início da fita do genoma referência, a qual se espera encontrar o início do gene *dnaA*. Caso exista erros no início do genoma referência, será necessária uma correção na etapa de curadoria manual.



**Figura 25 – Representação gráfica do uso do script MoveDNAA.**

(A) Ordenação de *contigs* do organismo *Corynebacterium pseudotuberculosis* 31 usando como referência o organismo *Corynebacterium pseudotuberculosis* 1002. A ordenação é incapaz de determinar a posição de início do genoma. (B) MoveDNAA corta a fita na imediatamente antes do início do gene *dnaA*, movendo todo o bloco anterior ao gene para o final da fita. É possível observar a eficácia do CONTIGuator em apresentar visualizações de modificações na fita de DNA. A cor da linha inferior foi modificada devido a substituição dos *gaps* por regiões com 100 caracteres “N”, característica do *script* MoveDNAA.

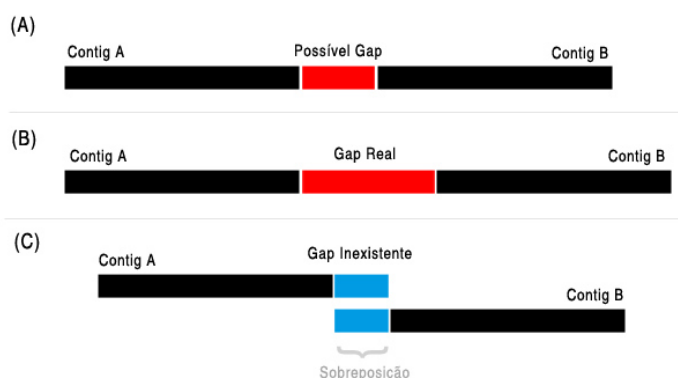
<sup>24</sup> Disponível em: <<https://github.com/dcbmariano/scripts>>. Acesso em: 29 de abril, 2014.

### **2.1.3.4 Construção de Supercontigs**

Quando a extremidade de um *contig* sobrepõe-se à extremidade de outro *contig*, pode-se detectar que ambas representam a mesma sequência, provando assim, que o *gap* não existe (Figura 26). A existência de uma sobreposição entre dois *contigs* pode ser visualizada através do gráfico de sobreposição gerado pelo CONTIGuator (Figura 24), embora só seja possível comprovar a sobreposição através de alinhamentos entre sequências, realizados com a ferramenta BLAST.

Dá-se o nome de *supercontig* quando dois ou mais *contigs* são unidos pela sobreposição de suas extremidades.

Para essa etapa foi construído um *parser* na linguagem PHP, que exhibe resultados de alinhamentos feitos com a ferramenta BLAST entre todos os *contigs* vizinhos e que permite ao usuário definir manualmente as posições de sobreposição para corte da região sobreposta redundante.



**Figura 26 – Diferentes tipos de gap.**

(A) A ordenação de *contigs* não é capaz de determinar as sequências que existem entre dois *contigs*. Quando o sequenciamento usa fragmentos simples, CONTIGuator preenche o espaço entre dois *contigs* com uma sequência de 100 caracteres “N”. (B) Quando o sequenciamento é feito com bibliotecas pareadas, a distância entre *contigs* pode ser conhecida e, pode-se dizer que realmente existe uma sequência desconhecida entre eles. Diz-se que há um *gap* real. (C) Porém, quando a extremidade de um *contig* se sobrepõe a extremidade de outro *contig*, diz-se que o *gap* é inexistente, e é possível fazer um alinhamento entre as extremidades de sequências e gerar uma sequência consenso única cortando uma das regiões sobrepostas redundante, eliminando assim, o *gap*.

### **2.1.3.5 Resolvendo regiões repetitivas**

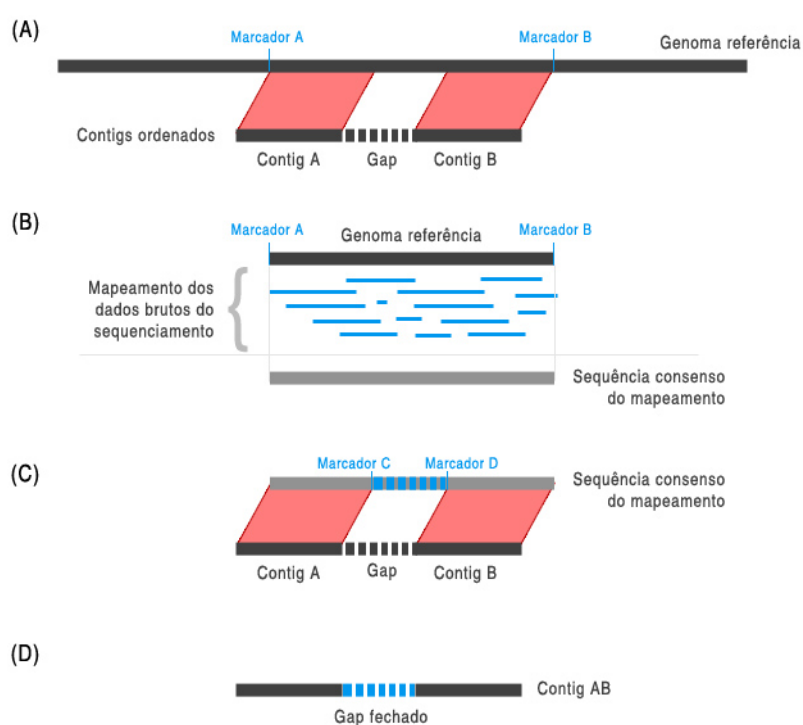
Com o intuito de tentar solucionar regiões repetitivas foi criado o *software* MapRepeat<sup>25</sup>: um *script* desenvolvido na linguagem Python integrado ao *software*

<sup>25</sup> Disponível em: <<https://github.com/dcbmariano/scripts>>. Acesso em: 29 de abril, 2014.



Mira v.4.0.2, que visa fechar regiões repetitivas com base no mapeamento de dados brutos em um genoma referência.

MapRepeat efetua um alinhamento entre dois *contigs* ordenados contra um genoma referência com objetivo de descobrir se a sequência em questão existe no genoma referência (Figura 27A). Se a sequência existir, ela é extraída e os dados brutos do sequenciamento (arquivo com leituras, por exemplo, FASTQ) são mapeados contra ela utilizando Mira v.4.0.2 (Figura 27B). Se houver cobertura de leituras que comprove a existência da sequência tanto no genoma referência quanto no genoma recém-sequenciado, um alinhamento entre *contigs* e a sequência consenso do mapeamento é feito (Figura 27C). Assim, a sequência entre *contigs*, antes desconhecida, pode ser determinada (Figura 27D).



**Figura 27 – Funcionamento do MapRepeat.**

(A) Dois *contigs* vizinhos são alinhados contra um genoma referência. Se houver similaridades entre as sequências, marcadores são utilizados para marcar a posição inicial de similaridade com o *Contig* A e a posição final de similaridade com o *Contig* B (limite de 3000pb). (B) A região delimitada pelos marcadores é extraída, e os dados brutos de sequenciamento são mapeados contra ela. (C) Se houver cobertura que comprove a existência daquela região tanto no genoma referência, quanto no genoma recém-sequenciado, a sequência consenso é alinhada contra os *contigs* A e B. Novos marcadores são utilizados para identificar a região desconhecida recém-mapeada. (D) O *gap* é fechado.

MapRepeat foi inicialmente desenvolvido como possível solução para regiões de RNA ribossomal, mas pode ser utilizado em *gaps* com outros tipos de repetição, como regiões de transposons, plasmídeos e fagos. Para determinar se os *contigs* estão próximos a regiões de repetição é necessário analisar o gráfico de alinhamento gerado pelo CONTIGuator (Figura 24).

MapRepeat também pode ser usado para mapear outras regiões de *gaps* desde que as extremidades do *gap* estejam presentes no genoma referência.

### **2.1.3.6 Estatísticas e curadoria manual**

Na última etapa é feita uma leitura do arquivo com *scaffolds* para validar se existem nucleotídeos desconhecidos (Tabela 3). Se ainda existirem *gaps* não resolvidos, o arquivo “*excluded.fsa*”, gerado pelo CONTIGuator no primeiro alinhamento, pode ser utilizado para detecção de *contigs* não ordenados de acordo com a referência. É necessário testar manualmente a adição de cada *contig* remanescente em cada *gap*, e verificar se existe sobreposição entre extremidades. Essa etapa é denominada curadoria manual e pode ser realizada com o *software* CLC Workbench.

**Tabela 3 – Lista de nucleotídeos não identificados e seus respectivos caracteres representantes.**

<b>Caractere</b>	<b>Nucleotídeo</b>
R	Purina (A ou G)
Y	Pirimidina (C, T, ou U)
M	C ou A
K	T, U, ou G
W	T, U, ou A
S	C ou G
B	C, T, U, ou G (exceto A)
D	A, T, U, ou G (exceto C)
H	A, T, U, ou C (exceto G)
V	A, C, ou G (exceto T, exceto U)
N	Qualquer base (A, C, G, T, ou U)

## 2.2 Hardware

Para testar o *pipeline* foi preparado um ambiente de produção, em que o acesso à aplicação foi permitido através de uma conexão com um Servidor Web (S.O. CentOS 64bit, processador AMD Opteron(tm) Processor 6376 (64 núcleos), 1TB RAM e 30TB HD). Considerou-se aceitável a configuração desse computador para os testes realizados, mas dependendo da complexidade do organismo sequenciado pode ser necessário um servidor com configurações mais robustas.

O processo que exige maior demanda computacional é a execução da montagem *de novo*, logo as configurações mínimas do servidor pode ser definidas pelas necessidades dessa etapa. Recomenda-se um computador com sistema Linux 64bit, no mínimo 16GB de RAM, espaço em disco rígido superior a 1TB e processador octa-core ou superior. Cabe ressaltar, que alguns *software* integrados podem exigir sistemas operacionais específicos, por isso recomenda-se o uso de sistemas operacionais baseados em Linux para hospedagem da aplicação.

## 2.3 Interface SIMBA aplicada ao *pipeline*

A fim de melhorar a usabilidade e facilitar a execução do *pipeline* proposto, desenvolveu-se uma interface Web, acessível por qualquer navegador, denominada interface SIMBA. A ferramenta foi desenvolvida com o *framework* Laravel em PHP, leiaute e navegabilidade aperfeiçoado com o *framework* Bootstrap (CSS/Javascript) e o banco de dados construído em SQLite.

Laravel utiliza a metodologia MVC, a qual a aplicação é dividida em três camadas: modelo (*model*), visão (*view*) e controlador (*controller*). O usuário acessa a aplicação através de um navegador, que faz a chamada ao controlador através de uma rota pré-definida. Dependendo do controlador requisitado, SIMBA pode requisitar o acesso a ferramentas externas, como Mira, CONTIGuator, entre outros. Pode também executar *scripts* Python, como CONTIGinfo, MapRepeat ou MoveDNAA. Informações específicas sobre montagens podem ser gravadas diretamente no banco de dados e acessadas através do modelo, ou podem ser lidas diretamente da unidade de armazenamento. Para evitar redundância de dados optou-se por não transferir todos os dados resultantes das montagens para o banco de dados SQLite. Uma montagem *de novo* com Mira4 gera em média 50GB de dados, sendo a maior parte, arquivos temporários.

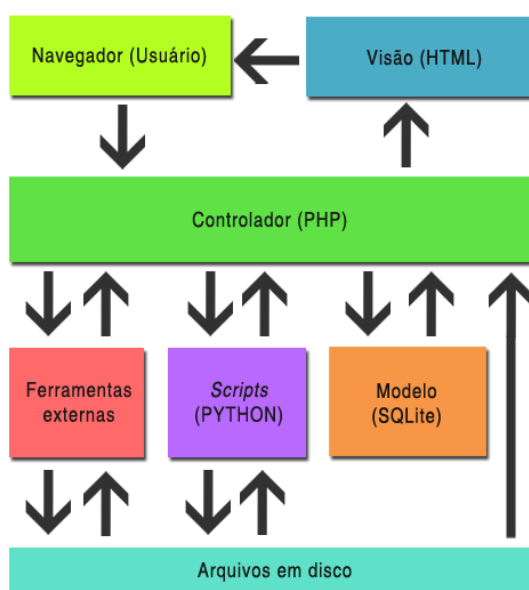
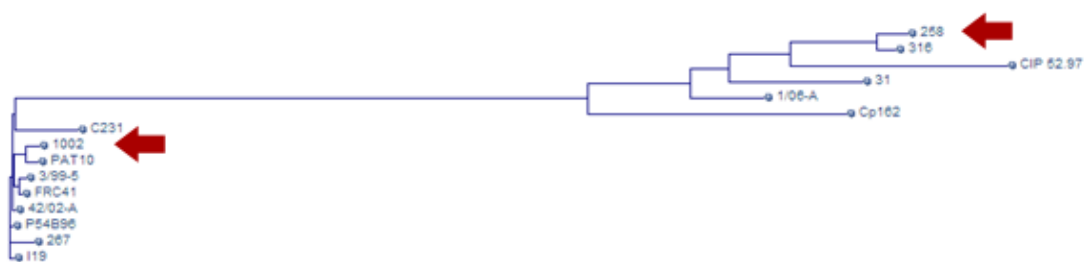


Figura 28 – Diagrama de relacionamento entre a arquitetura da interface SIMBA aplicada ao *pipeline* e sua integração com o modelo MVC.

SIMBA requer apenas a instalação de um servidor Web (recomenda-se o Apache) com PHP 5.3 ou superior e módulo SQLite ativo, Python 2.6 ou superior com a biblioteca Biopython instalada e a suíte de aplicativos BLAST+. Todos os outros *softwares* externos foram incluídos no diretório [www\_folder]/simba/app/bin.

#### 2.4 Estudo de caso: *Corynebacterium pseudotuberculosis* como modelo

Para avaliar o funcionamento do *pipeline* proposto e sua usabilidade através da interface SIMBA, propôs-se um estudo de caso através da montagem de dados de dois projetos de ressequenciamento. Foram utilizados dados de sequenciamento de duas linhagens da espécie pertencentes ao grupo CMNR: *Corynebacterium pseudotuberculosis* 258 (Cp258), pertencente ao biovar *equi*, e *Corynebacterium pseudotuberculosis* 1002 (Cp1002), pertencente ao biovar *ovis* (Figura 29) (DORELLA *et al.*, 2006). Ambas foram ressequenciadas na plataforma Ion PGM™ utilizando biblioteca de fragmentos simples. Além disso, mapas ópticos de genoma completo de ambas as espécies foram feitos pela empresa OpGen, que utilizou a enzima de restrição KpnI para fragmentação das amostras.



**Figura 29 –** Árvore filogenética da espécie *Corynebacterium pseudotuberculosis*.

À esquerda encontram-se as linhagens pertencentes ao biovar *ovis* – possuem menor plasticidade. À direita encontram-se linhagens pertencentes ao biovar *equi* – possuem maior plasticidade.

Fonte: NCBI. Disponível em: <<http://ncbi.nlm.nih.gov>>. Acesso em: 12 de abril, 2014.

*Corynebacterium pseudotuberculosis* 258 foi originalmente sequenciada utilizando a plataforma SOLiD v3 e biblioteca de fragmentos simples (SOARES *et al.*, 2013). Enquanto *Corynebacterium pseudotuberculosis* 1002 foi originalmente sequenciada utilizando pirosequenciamento (454 Roche) e Sanger (RUIZ *et al.*, 2011). As amostras foram ressequenciadas no laboratório AQUAVET da Escola de Veterinária da Universidade Federal de Minas Gerais (UFMG).

Para testar o funcionamento da ferramenta SIMBA pelo cliente foi utilizado o navegador Google Chrome, em um computador com sistema operacional Windows 7 32bit, processador Intel Dual Core 1.6GHz, 3GB RAM e 120GB de HD. O computador estava conectado a mesma rede que o servidor de processamentos.

Os dois genomas foram submetidos através da interface do SIMBA à montagem *de novo* em quatro *softwares*: Mira 3.9.18 (Mira3), Mira 4.0.2 (Mira4), Newbler 2.7 (Newbler) e Minia 1.6088 (Minia). Também foi realizada uma montagem fora da interface de SIMBA utilizando SPAdes 3.1.0 (SPAdes). Os resultados das montagens foram renomeados e movidos para a pasta correspondente ao projeto. Em seguida, a montagem foi inserida no SIMBA através da opção *manual assembly*. Para as duas versões de Mira utilizou-se os parâmetros para uso de 16 processadores e formação de *contigs* apenas se houver sobreposição mínima de 100 leituras. Para Minia foram adicionados os parâmetros de tamanho de *k-mer* 31 (parâmetro recomendado pela documentação oficial) e tamanho aproximado do genoma de 2.500.000pb. Para Newbler foi usado apenas o parâmetro para processamento em 24 núcleos de CPU. Para SPAdes apenas informou-se que a plataforma de sequenciamento foi Ion Torrent.

Por fim, foram aplicados dois métodos de finalização de montagem: o primeiro, baseado em referência, a qual foi utilizado como genoma referência os

genomas depositados anteriormente; e o segundo método, baseado no relatório obtido pelo mapeamento óptico das amostras gerado pelo *software* MapSolver™.

Por fim, foi feita uma comparação entre os resultados da finalização por referência e por mapeamento óptico, além de uma comparação com os dados originalmente depositados no NCBI.

### 3. Resultados e discussões

Apresenta-se SIMBA (*Simple Manager for Bacterial Assemblies* ou Gerenciador Simples para Montagens [de genomas] Bacterianos), uma ferramenta Web criada para gerenciar as estratégias de montagem apresentadas em um *pipeline* híbrido, que visa facilitar a execução dos processos de montagem de genomas. O código-fonte da aplicação, além de todos os scripts desenvolvidos, foi disponibilizado em <<http://ufmg-simba.sourceforge.net>>.

Nesta seção será demonstrado o funcionamento da interface da ferramenta SIMBA para gerenciamento das estratégias propostas para tratamento, montagem *de novo* e finalização de genomas bacterianos. Serão mostrados também, os resultados obtidos no estudo de caso citado anteriormente.

#### 3.1 Visão geral da interface

O acesso à ferramenta SIMBA é protegido por um sistema de *login*, que restringe a entrada de usuários não autenticados. Informações armazenadas no banco de dados são protegidas por criptografia gerada com base em uma chave de 32 caracteres aleatórios informados pelo usuário. Tal restrição visa proteger a integridade dos dados até que estejam prontos para depósito em bancos de dados públicos. Após a autenticação, o usuário é encaminhado à página inicial, que redireciona o usuário para página projetos, dando início ao fluxo de dados.

O fluxo de dados no SIMBA pode ser dividido em três módulos: projetos, montagens e curadoria (Figura 30). Em “projetos” são listados todos os projetos de sequenciamento gerenciados pelo SIMBA; em “montagens” são exibidos todas as tentativas de montagem com diferentes programas e parâmetros para um determinado projeto; enquanto em “curadoria” são exibidas cinco etapas para finalização *in silico* da montagem.

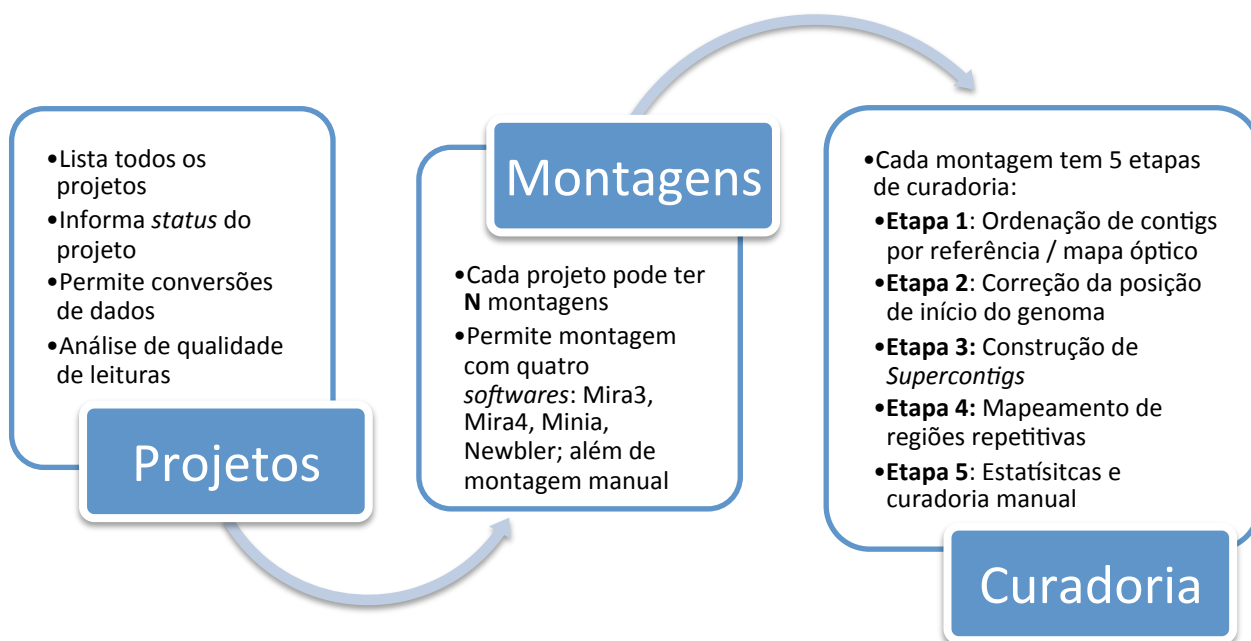


Figura 30 - Diagrama de fluxo de dados no SIMBA.

### 3.1.1 Projetos

Cada projeto corresponde a uma pasta dentro de um diretório específico no servidor (*[www\_folder]/simba/app/assembly*) em que estão inseridos os dados brutos de leituras obtidas em um sequenciamento. Há duas formas de criar um novo projeto: a primeira é criar uma pasta no diretório “*assembly*”, copiar os dados para o diretório via terminal (CLI) e na interface SIMBA utilizar o botão “*update*” (Figura 31). A segunda forma consiste em enviar os dados através do botão “*New Project*”. Apesar de a segunda forma ser mais simples, o tamanho máximo de arquivos enviados é limitado a 1.800MB e o tempo de envio depende da conexão com o servidor. É possível enviar três tipos de arquivos: BAM, SFF ou FASTQ.

Após o envio, as informações do organismo são gravadas no banco de dados SQLite, e os dados brutos são gravados na unidade de armazenamento.

Após a criação de um novo projeto algumas informações são exibidas:

- **Status:** indica o percentual de processos aplicados ao projeto;
- **Name:** chave identificadora única do projeto. Corresponde ao nome do diretório a qual os dados estão armazenados. Apresenta um *link* para a página que lista tentativas de montagem por projeto;
- **Organism:** nome científico do organismo sequenciado. Apresenta também a linhagem ou subespécie;



- **NGS:** plataforma de sequenciamento NGS utilizada. Pode ser *iontor* (Ion PGM™ ou Ion Proton™), *illumina* (MiSeq ou HiSeq), *454* (454 Roche) ou *pacbio* (PacBio RS);
- **Library:** indica a biblioteca utilizada no sequenciamento. Pode ser: *fragment*, *mate-pair* ou *paired-end*;
- **BAM:** informa a existência de um arquivo na extensão BAM no diretório raiz do projeto;
- **SFF:** informa a existência de um arquivo na extensão SFF no diretório raiz do projeto;
- **FASTQ:** informa a existência de um arquivo na extensão FASTQ no diretório raiz do projeto;
- **Assembly:** informa se já foi realizada uma montagem *de novo*;
- **Actions:** permite que diversas ações sejam realizadas, como: editar e apagar o projeto, converter um arquivo BAM em SFF, extrair o arquivo FASTQ de um arquivo SFF, gerar um relatório completo das leituras usando FastQC, fazer *download* dos dados brutos e até mesmo iniciar uma nova montagem *de novo*.

The screenshot shows the SIMBA web interface. At the top, there is a navigation bar with links for HOME, DOCS, TOOLS, and ABOUT, and a user profile section for ADMIN | LOGOUT. Below the navigation bar is the SIMBA logo. The main content area displays a table of projects with columns for Status, Name, Organism, NGS, Library, BAM, SFF, FASTQ, Assembly, and Action. A 'New project' button is located in the top right corner of the table area. Below the table, there is a large blue button labeled 'UPDATE' with a refresh icon. A dropdown menu is open from the 'Action' column of the second row, showing options: Edit project, Delete project, Generate SFF file, Extract FASTQ file, FastQC Report, Download raw data, and New assembly. A note at the bottom of the interface states: 'Note: SIMBA uses the softwares: BAM2SFF and SFF\_EXTRACT.'

(A) HOME DOCS TOOLS ABOUT ADMIN | LOGOUT

simba

Home / Projects

(B)

Status	Name	Organism	NGS	Library	BAM	SFF	FASTQ	Assembly	Action
✓	09-Cp_258	<i>Corynebacterium pseudotuberculosis 258</i>	iontor	Fragment	✗	✓	✓	✓	⌵
✓	10-Cp_1002	<i>Corynebacterium pseudotuberculosis 1002</i>	iontor	Fragment	✗	✓	✓	✓	⌵

(C) New project

(D)

- Edit project
- Delete project
- Generate SFF file
- Extract FASTQ file
- FastQC Report
- Download raw data
- New assembly

(E) UPDATE

Note: SIMBA uses the softwares: BAM2SFF and SFF\_EXTRACT.

SIMBA version beta by LCGM | Universidade Federal de Minas Gerais | 2014

Figura 31 – Interface do SIMBA.

Nesse exemplo são mostrados dois projetos registrados: 09-Cp\_258 (*Corynebacterium pseudotuberculosis* 258) e 10-Cp\_1002 (*Corynebacterium pseudotuberculosis* 1002). O exemplo mostra também apenas a existência de arquivos SFF, FASTQ, além de pelo menos uma montagem já realizada no diretório dos projetos 09-Cp\_258 e 10-Cp\_1002. (A) Menu para rápida navegação com controle de sessão de usuário. (B) Tela principal (conteúdo varia de acordo com a página). (C) Botão para criação de novos projetos. (D) Menu para controle de ações, como edição e deleção de projetos, conversões de arquivos, geração de relatórios de qualidade de leituras com FastQC, *download* de arquivos brutos e iniciação de novas montagens. (E) Atualização de informações sobre projetos.

### 3.1.2 Montagens

No módulo “montagens” é exibida uma lista com todas as tentativas de montagem para o projeto. Cada projeto pode possuir diversas tentativas de montagem.

Uma nova tentativa de montagem pode ser feita através do botão “*New assembly*”. Há dois modos para criação de montagens: modo simples (*default assembly*) e modo avançado (*advanced assembly*).

No modo simples, o *software* que será utilizado e seus parâmetros são pré-definidos (montagem *de novo* com Mira3 e dados sequenciados com fragmentos simples na plataforma Ion PGM™), sendo necessário apenas pressionar o botão “*Run assembly with default parameters*”. Os parâmetros do modo simples podem ser alterados no código-fonte de acordo com as necessidades dos usuários ou da equipe, desde que o administrador local tenha conhecimentos de programação em linguagem PHP.

No modo avançado é possível definir qual *software* será utilizado na montagem *de novo*. Há cinco opções: Mira3, Mira4, Minia, Newbler e *manual assembly*. A última opção permite que uma montagem feita com outro *software* não embutido no SIMBA seja inserida no sistema. Nessa página também é possível definir parâmetros para montagem. Se nenhum parâmetro for passado, SIMBA automaticamente aplica parâmetros recomendados pela documentação de cada *software*.

Após iniciado o processo, um novo item é inserido na lista de tentativas de montagens. Enquanto a montagem é executada, ou caso o processo falhe, um “x” é exibido. A lista pode ser atualizada clicando no botão “*update*”. Finalizada a montagem, informações referentes aos resultados são exibidas:

- ***Trial***: versão da tentativa de montagem;
- ***Contigs***: exibe a quantidade de *contigs* obtidos no resultado da montagem *de novo*;

- **Length**: exibe o tamanho em pares de base do genoma;
- **Min**: exibe o tamanho em pares de base do menor *contig*;
- **Max**: exibe o tamanho em pares de base do maior *contig*;
- **N50**: exibe o valor do N50 do resultado da montagem;
- **Date**: exibe a data de início de execução da montagem;
- **Assembly info**: exibe informações completas sobre a montagem, como por exemplo, a cobertura teórica;
- **Parameters**: exibe os parâmetros utilizados na montagem;
- **Actions**: botão “Actions” permite que diversas ações sejam realizadas, como: apagar o registro da montagem no banco de dados (caso ela falhe), fazer *download* do arquivo de *contigs* e efetuar a curadoria.

A cada tentativa de montagem é possível iniciar um novo procedimento de curadoria. Os procedimentos de curadoria são necessários para finalização da montagem.

### 3.1.2 Curadoria

No módulo “curadoria” é exibida uma lista com etapas para tentar finalizar a montagem. Para cada etapa é exibido: (i) um botão que permite visualização do gráfico de sintonia contra a referência (é possível visualizar e fazer o *download* do arquivo em PDF com o gráfico em alta qualidade); (ii) um botão que permite o *download* do arquivo de *scaffolds*; (iii) um botão que permite o *download* do arquivo de *scaffolds* separado em um arquivo FASTA de várias sequências; e (iv) o botão “action”, que permite a execução da etapa.

A ferramenta SIMBA divide os processos de curadoria em cinco etapas (Figura 32):
















1. **Ordenação de *contigs***: pode ser por referência ou por mapeamento óptico:
  - a. **Por referência**: exibe dois campos para inserção do endereço de um arquivo FASTA (contendo o genoma completo de um organismo referência), e do arquivo GENBANK (contendo informações sobre genes e proteínas), que será utilizado para inserir marcações no gráfico informando a posição de regiões de repetição. Inseridas essas informações, SIMBA efetua ordenação dos *contigs* obtidos pela montagem *de novo* mapeando-os no genoma referência por meio de

uma versão modificada do *software* CONTIGuator. Em seguida, retorna um arquivo com *scaffolds*, além de um gráfico de sintenia entre os dois genomas.

- b. **Por mapeamento óptico:** exibe um campo para inserção do relatório gerado pelo MapSolver. Com base nesse relatório, SIMBA ordena os *contigs*. Em seguida, a versão modificada do CONTIGuator é executada, para que o arquivo com sequências não alinhadas possa ser gerado, além do arquivo com *scaffolds* e um gráfico de sintenia que pode ser utilizado para detectar sobreposições entre *contigs*.
2. **Correção da posição inicial do genoma:** nessa etapa é feita a correção da posição inicial do genoma. Para isso é utilizado o *script* MoveDNAA. Essa etapa não funciona se for escolhido o alinhamento por mapeamento óptico. Novamente é feito um alinhamento dos *contigs* contra a referência.
3. **Construção de Supercontigs:** nessa etapa são feitos alinhamentos entre todas as extremidades de *contigs* com o objetivo de detectar sobreposições. SIMBA exibe o gráfico de sintenia obtido na etapa anterior, além de uma lista com resultado de alinhamento de todas as extremidades. Caso um alinhamento seja detectado, um sinal é exibido na coluna “*Is there overlap?*”, sendo então possível utilizar o botão “*BLAST*” na coluna “*Action*” para analisar o resultado do alinhamento e efetuar cortes de regiões duplicadas em sobreposições, eliminando um *gap*. Terminada a análise de todas as sobreposições pode-se gravar o resultado no banco de dados e gerar um novo alinhamento contra a referência.
4. **Mapeamento de regiões repetitivas:** nessa etapa é exibido um novo gráfico de sintenia e uma lista com todos os *gaps* ainda existentes. Se um *gap* corresponder a uma região de repetição conhecida na referência, é possível utilizar o botão “*map*” na coluna “*actions*”, e assim, usar técnicas de mapeamento para tentar fechá-lo. Nessa etapa, marcações coloridas no gráfico de sintenia auxiliam. Pontos azuis representam regiões onde se encontra o *operon* de rRNA; pontos na cor azul claro representam regiões de transposons; amarelo representa regiões de plasmídeos; e pontos verdes representam regiões de fagos. Terminada essa etapa, as alterações feitas podem ser gravadas no banco de dados e um novo alinhamento contra a referência é feito.

5. **Estatísticas e curadoria manual:** nessa última etapa são exibidas diversas estatísticas como tamanho final do genoma, quantidade de nucleotídeos não identificados, total de *contigs* que não puderam ser ordenados. Também é possível efetuar o *download* de arquivo com *scaffolds*, além do arquivo com *contigs* excluídos, e curar manualmente com outras ferramentas de acordo com a preferência do usuário, caso necessário. Ao final é possível submeter o arquivo curado manualmente. Um novo alinhamento é feito contra a referência e são exibidos os resultados das etapas de curadoria.

Home / Projects / 10-Cp\_1002 / Trial 14 (A)

(B) Step	(C) Status	(C) Action	(D) Gaps	(F) Synteny chart	(G) Download	(H)	(I) Action
1	✓	Set reference	6		<input type="checkbox"/>		
2	✓	Move dnaA	6		<input type="checkbox"/>		
3	✓	Building Supercontigs	4		<input type="checkbox"/>		
4	✓	Analyze repetitive regions	4		<input type="checkbox"/>		
5	✓	Statistics and manual curation	0		<input type="checkbox"/>		

(E) Organism: *Corynebacterium pseudotuberculosis* 1002  
Date: 2014-05-07 14:00:00

Figura 32 – Finalização de montagens na interface SIMBA.

(A) Informações sobre o projeto e versão da tentativa de montagem. (B) Etapa da curadoria. (C) Estado da ação atual. (D) Quantidade de *gaps* restantes. (E) Informações sobre organismo e data do projeto. (F) Gráfico de alinhamento gerado pelo CONTIGuator. (G) Opção para fazer o *download* do arquivo com *scaffolds*. (H) Opção para fazer o *download* do arquivo com *contigs*. (I) Executar a etapa.

## 3.2 Resultados do estudo de caso

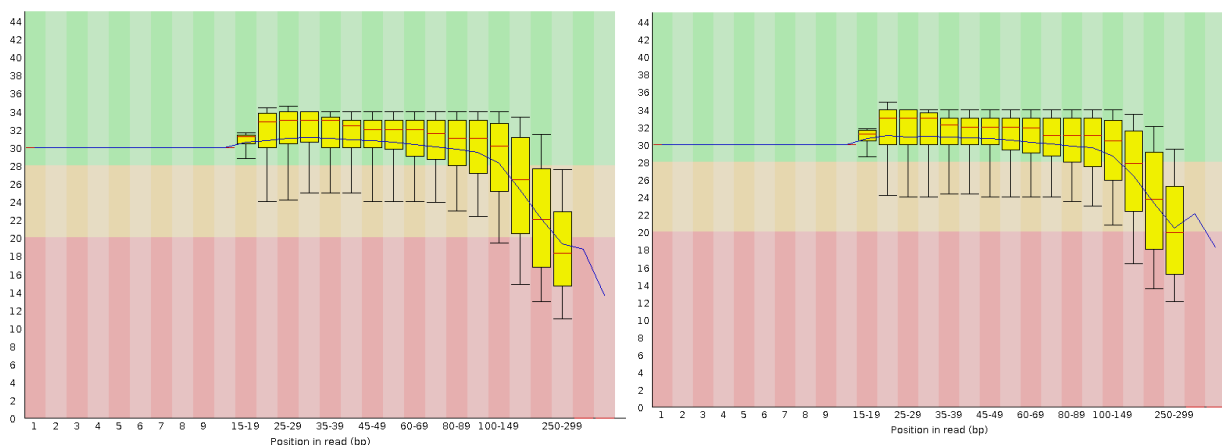
### 3.2.1 Tratamento de dados

A criação dos projetos ocorreu pela interface gráfica. Os projetos receberam os nomes: 09-cp\_258 (*Cp258*) e 10-cp\_1002 (*Cp1002*). Para cada projeto foi enviado um arquivo BAM com dados brutos dos sequenciamentos. O tempo de submissão de cada arquivo foi de aproximadamente cinco minutos, sendo limitado pela banda de conexão da rede interna (100Mbps).

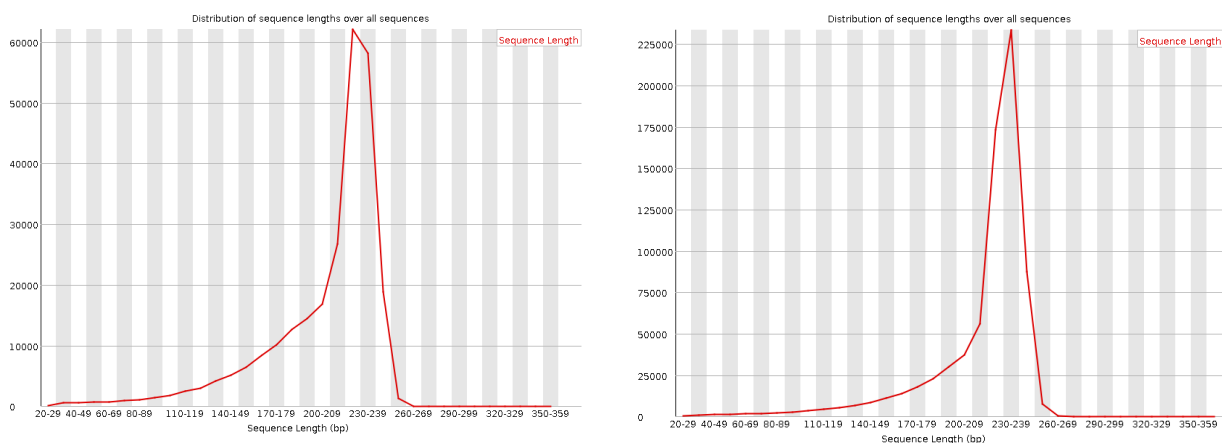
A conversão de arquivos no formato BAM para SFF foi facilmente realizada através do botão “Convert BAM > SFF” na coluna “Actions”. O mesmo pode ser feito

para extração dos arquivos FASTQ de arquivos SFF através do botão “*Extract FASTQ*”.

A análise de leituras com FastQC dos projetos 09-cp\_258 e 10-cp\_1002 mostrou leituras com boa qualidade (algoritmo Phred), mas com uma leve perda de qualidade na extremidade direita (Figura 33). A maior parte das leituras possuem tamanhos entre 230-239pb (Figura 34).



**Figura 33 - Gráfico de qualidade Phred em leituras em Cp258 (à esquerda) e Cp1002 (à direita). Figuras geradas pelo FastQC.**



**Figura 34 - Gráfico tamanho médio de leituras de Cp258 (à esquerda) e Cp1002 (à direita). Figuras geradas pelo FastQC.**

### 3.2.2 Montagem de novo

Foram feitas cinco tentativas de montagem *de novo* para cada projeto. A primeira com Mira3, seguido por Mira4, Minia, Newbler e *manual assembly* (SPAdes). Para isso, em cada projeto utilizou-se a opção “*New assembly*”. Na

página de criação de novas montagens, optou-se por utilizar o modo “*Advanced assembly*”, pois esse permite a alteração dos parâmetros de montagem.

Considera-se o melhor resultado para uma montagem *de novo*, o teste que apresentar: o menor número de *contigs* formados e maior valor de N50. O tamanho total de genoma em pares de base também pode ser considerado uma variável importante para definir a qualidade da montagem, mas não primordial, pois alguns *softwares* podem erroneamente gerar *contigs* a mais, além disso, esse valor pode variar nas etapas de curadoria. Valores mais altos nos campos “menor *contig*” e “maior *contig*” também podem ser considerados um fator de qualidade.

### **3.2.2.1 Resultados e discussões da montagem de novo usando o genoma de *Corynebacterium pseudotuberculosis* linhagem 258**

Para Cp258 as montagens apresentaram entre 41 e 675 *contigs* (Tabela 4). Newbler apresentou o menor número de *contigs*, um alto valor de “N50”, além do “menor *contig*” com o maior tamanho se comparado aos outros *softwares*. Porém, apresentou o menor tamanho de genoma. O fato do resultado da montagem *de novo* com Newbler apresentar um valor de “menor *contig*” alto, comparado aos outros *softwares* (Tabela 4), pode ter impactado diretamente no tamanho final do genoma, e pode indicar uma tendência do Newbler, em desprezar a formação de *contigs* pequenos (menores do que cinco mil pares de base). Isso pode representar um problema nas etapas de construção de *supercontigs* e mapeamento de regiões repetitivas. Newbler apresentou rápido tempo de execução (oito minutos), sendo inferior apenas ao Minia. Apresentou também um alto valor de “N50”, inferior apenas ao resultado do SPAdes.

Mira3 apresentou uma menor quantidade de *contigs* e valores maiores nos campos: “maior *contig*” e “N50” do que Mira4. Em compensação, Mira4 formou um “menor *contig*” maior do que sua versão anterior, além de um “tamanho do genoma” maior. Ambas as montagens foram executadas em quarenta e três minutos.

As duas versões do Mira apresentaram resultados distintos, mesmo sendo executadas com os mesmos parâmetros. Isso demonstra mudanças nos padrões de execução do algoritmo interno entre as versões. Mira4 apresentou uma quantidade maior de *contigs*, mas foi capaz de gerar um genoma com um tamanho maior. Por outro lado, Mira3 gerou um “N50” maior. Os tempos de execução foram idênticos.

Minia apresentou a maior quantidade de *contigs*, os menores tamanhos de “menor *contig*”, “maior *contig*” e “N50”. Apesar disso, apresentou também o maior tamanho total de genoma. O tempo total desde o registro da tarefa até a geração do arquivo de *scaffolds* foi aproximadamente um minuto. Por não desprezar *contigs* pequenos, Minia pode obter o maior tamanho de genoma estimado na montagem *de novo* para *C. pseudotuberculosis 258*. Minia apresentou um tempo de execução inferior a um minuto, um fato excepcional, apesar de que, isso possivelmente deve-se a baixa cobertura média do sequenciamento.

SPAdes apresentou o maior tempo de execução, o maior valor de “N50”, o maior valor de “maior *contig*” formado, além do segundo mais alto “número de *contigs*”.

**Tabela 4 – Tentativas de montagem para Cp258.**

Tentativa	Software	Número de <i>contigs</i>	Tamanho do genoma	Menor <i>contig</i>	Maior <i>contig</i>	N50	Tempo de execução
1	Mira3	41	2.353.419	837	249.766	93.650	0h43
2	Mira4	56	2.357.501	1.233	188.183	71.762	0h43
3	Minia	675	2.361.252	63	22.211	8.272	0h01
4	Newbler	27	2.348.526	5.309	316.245	218.551	0h08
5	SPAdes	58	2.357.282	87	405.530	272.654	0h52

### **3.2.2.2 Resultados e discussões da montagem de novo usando o genoma de *Corynebacterium pseudotuberculosis* linhagem 1002**

Para Cp1002 foram realizadas cinco tentativas de montagem *de novo*, que apresentaram entre 9 e 2.425 *contigs* (Tabela 5). Devido à alta profundidade de cobertura em Cp1002 (~2,5x superior a de Cp258) os *softwares* demandaram tempo superior de execução, se comparados à montagem de Cp258.

Mira3 apresentou o menor “número de *contigs*”, maior valor de “menor *contig*”, além do segundo maior valor de “N50”. Obteve também o maior tempo de execução. Mira4 obteve resultados próximos aos de Mira3, tendo resultado pouco superior nos campos “maior *contig*” e “tamanho do genoma”.

Newbler apresentou resultados pouco distintos aos obtidos com Mira3 e Mira4. Minia apresentou os piores resultados, como por exemplo, uma quantidade de *contigs* até 240x superior aos outros *softwares*. Minia apresentou um tamanho de



genoma superior em relação aos outros *softwares*, o que pode indicar erros de montagem.

SPAdes gerou 15 contigs em um “tempo de execução” de 1h46, além do maior valor de “N50” e “maior *contig*”.

**Tabela 5 – Tentativas de montagem para Cp1002.**

Tentativa	Software	Número de <i>contigs</i>	Tamanho do genoma	Menor <i>contig</i>	Maior <i>contig</i>	N50	Tempo de execução
1	Mira3	9	2.319.231	4.133	542.891	402.955	3h33
2	Mira4	12	2.320.091	660	542.903	276.749	3h18
3	Minia	2.425	2.366.290	63	11.622	2.613	0h04
4	Newbler	10	2.317.746	5.991	542.754	402.788	0h19
5	SPAdes	15	2.317.579	87	818.938	486.047	1h46

### **3.2.3 Curadoria: finalização da montagem**

Verificou-se bons resultados obtidos tanto na montagem com Mira3, Mira4, Newbler e SPAdes. Entretanto, para finalização da montagem escolheu-se as montagens realizadas com Mira3 por apresentar um baixo número de *contigs* na montagem de ambos os organismos.

Para essa etapa foi necessário duplicar os resultados da montagem de Mira3 para que fosse possível realizar duas tentativas de finalização com os mesmos dados iniciais: por referência e por mapeamento óptico.

#### **3.2.3.1 Resultados e discussões da finalização por referência**

Foi realizado o alinhamento dos *contigs* com base nos respectivos genomas de *C. pseudotuberculosis* 258 e *C. pseudotuberculosis* 1002 depositados no NCBI. Esperava-se que, com os dados obtidos com o novo sequenciamento, seria possível melhorar a montagem, inserindo pequenos *contigs* antes não detectados ou verificar a existência de “*mismatches*”, sequências montadas em regiões erradas. Poucas diferenças foram notadas em Cp1002, entretanto em Cp258 foi possível detectar novas regiões não presentes no primeiro sequenciamento.

Assim, as cinco etapas do processo de finalização foram aplicadas (Tabela 6) (gráficos de sintenia em anexo). Para Cp258, dos 41 *contigs*, 39 puderam ser alinhados contra a referência, gerando 38 *gaps*. Enquanto, que para Cp1002, do total de 9 *contigs*, 8 puderam ser alinhados, gerando 7 *gaps*. Na segunda etapa é

feita apenas a correção do início da fita, logo não há alteração na quantidade de *gaps*. Na terceira etapa, foram analisadas as extremidades de cada *gap*. Assim, em Cp258, foi possível gerar 16 *supercontigs*, reduzindo a quantidade de *gaps* para 22. Em Cp1002, foi criado 2 *supercontigs*, e a quantidade de *gaps* reduziu-se a 5. Na quarta etapa, os *gaps* remanescentes foram fechados com base na extração do consenso do mapeamento das leituras na referência. Nessa etapa todos os *gaps* de Cp1002 foram fechados e em Cp258 restou apenas um *gap*.

**Tabela 6 – Número de *gaps* ao final da cada tentativa de finalização da montagem de *C. pseudotuberculosis* 258 (Cp258) e *C. pseudotuberculosis* 1002 (Cp1002).**

<b>Etapa</b>	<b><i>Gaps</i> Cp258</b>	<b><i>Gaps</i> Cp1002</b>
<b>1</b>	38	7
<b>2</b>	38	7
<b>3</b>	22	5
<b>4</b>	1	0
<b>5</b>	0	0

Na última etapa, foram exibidas estatísticas para auxiliar na curadoria manual. Foi feito o *download* do arquivo de sequências de ambos os projetos, além do arquivo de sequências não ordenadas na primeira etapa (*excluded.fsa*). Detectou-se que havia 2 *contigs* (6.141pb) não ordenados no genoma de Cp258 e 1 *contig* (4.133pb) não ordenado no genoma de Cp1002.

Os arquivos com sequências foram analisados manualmente com CLC Workbench. Para Cp258, foi possível utilizar um dos *contigs* remanescentes para fechar o último *gap*. Assim, em ambas as montagens não havia mais *gaps*, entretanto ambas possuíam um *contig* não inserido, com tamanho aproximado de 4.000pb. A ferramenta online BLAST<sup>26</sup> do NCBI foi utilizada para analisar regiões codificadoras presentes nesses últimos *contigs* de ambos os projetos. Constatou-se que ambos os *contigs* se tratavam de *operons* de rRNA. Na espécie *Corynebacterium pseudotuberculosis* é possível detectar quatro cópias desse *operon*. Cada *operon* possui três regiões codificadoras: 16s, 23s e 5s, que são altamente conservadas. Entretanto, a região intergênica entre elas não é tão conservada. Essas características fazem com que a montagem das quatro cópias dessa região do genoma seja um problema computacional de difícil solução. O que

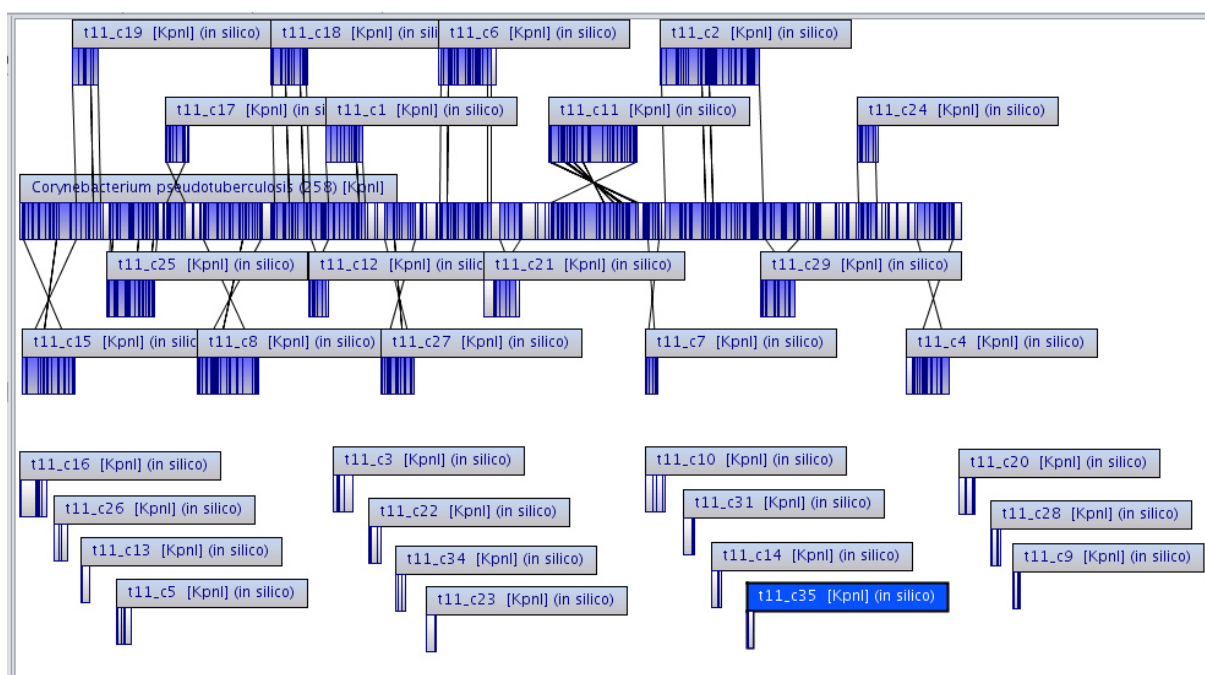
<sup>26</sup> Disponível em: <<http://blast.ncbi.nlm.nih.gov/Blast.cgi>>. Acesso em: 16 de agosto, 2014.

explica o fato do *software* de montagem só ter gerado um único *contig* para representar quatro regiões em um mesmo genoma.

Como os *gaps* em regiões de *operons* de rRNA foram resolvidos com base na extração da sequência consenso do mapeamento, pode-se eliminar esse último *contig* remanescente e considerar o genoma completamente fechado. Assim, o genoma de Cp258 apresentou 2.370.835pb e o de Cp1002 apresentou 2.335.972pb.

### **3.2.3.2 Resultados e discussões da finalização por mapeamento óptico**

Na finalização da montagem por mapeamento óptico, utilizou-se o MapSolver para ordenar os *contigs* com base no mapa de restrição (Figura 35; Figura 36). MapSolver não foi capaz de ordenar pequenos *contigs* (menores que 70Kb).



**Figura 35 - Alinhamento de *contigs* contra o mapa de restrição no MapSolver para Cp258.**

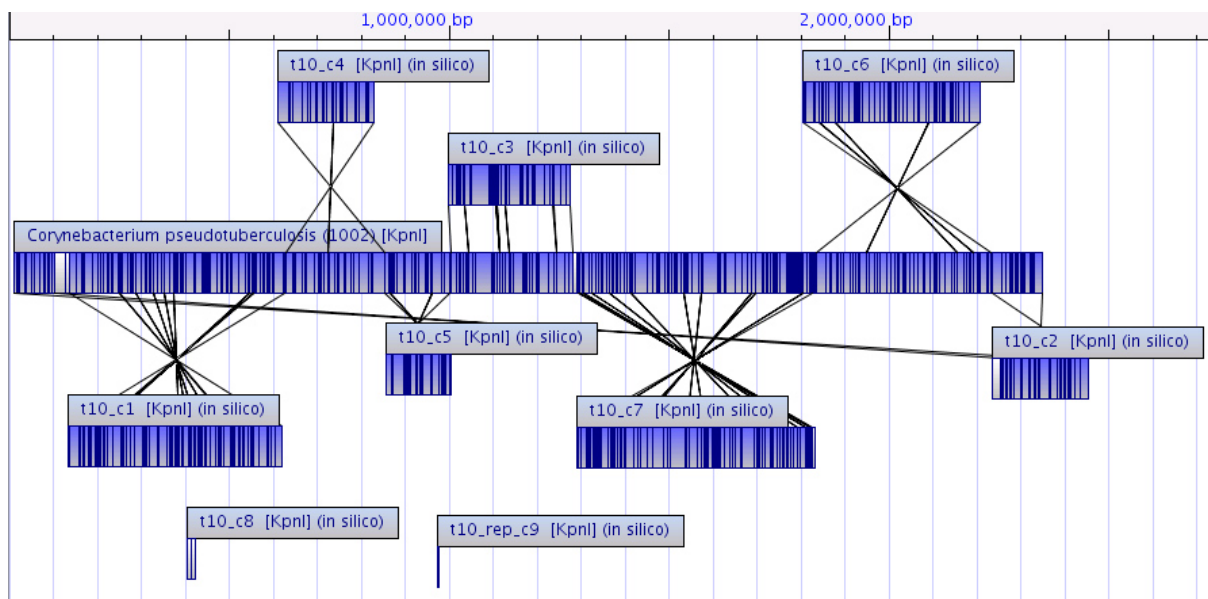


Figura 36 – Alinhamento de *contigs* contra o mapa de restrição no MapSolver para Cp1002.

Em seguida, o relatório com a ordem dos *contigs* para ambos os projetos foi exportado. Esse relatório foi importado na primeira etapa do processo de curadoria. Assim, em Cp258, dos 41 *contigs*, 17 foram ordenados, produzindo 16 *gaps*; e em Cp1002, dos 9 *contigs*, 7 puderam ser ordenados, produzindo 6 *gaps*. A segunda e quarta etapas não puderam ser realizadas por não se ter um genoma referência. Na terceira etapa, em Cp258 a quantidade de *gaps* foi reduzida para 14, e em Cp1002 para 4. Na quinta etapa, foi feito o *download* tanto o arquivo de sequências não alinhadas quanto o arquivo com o *scaffold* formado.

Constatou-se que 24 *contigs* (537.383pb) não estavam ordenados no genoma de Cp258, e 2 *contigs* (21.900pb) não estavam ordenados no genoma de Cp1002.

Para inserir os últimos *contigs* optou-se por utilizar uma estratégia híbrida: primeiro separou-se o arquivo com o *scaffold* gerado em um arquivo com múltiplas sequências, em seguida o arquivo de sequências excluídas foi unido a ele. Com o novo arquivo gerado foi feito um alinhamento contra referência utilizando CONTIGuator. Com o gráfico de alinhamento gerado pelo CONTIGuator foi possível analisar manualmente onde os pequenos *contigs* se alinhavam, respeitando o princípio que a ordem dada pelo resultado do mapeamento óptico estava sendo mantida. Assim, utilizando CLC Workbench, os pequenos *contigs* foram inseridos na fita final. Por fim, restou em cada projeto apenas um *contig* que continha um *operon*

de rRNA. Esse *contig* foi desprezado, pois as regiões de *operon* de rRNA foram mapeadas no genoma referência, assim como todos os *gaps* remanescentes.

Para validar o resultado dessa montagem, a sequência final foi importada pelo MapSolver e alinhada com o mapa de restrição (Figura 37; Figura 38). Ambos os resultados mostraram que as montagens realizadas apresentam boa sintonia com o mapa de restrição, demonstrando assim que, as estratégias adotadas para inserção de pequenos *contigs* não ordenados pelo MapSolver foram eficazes.

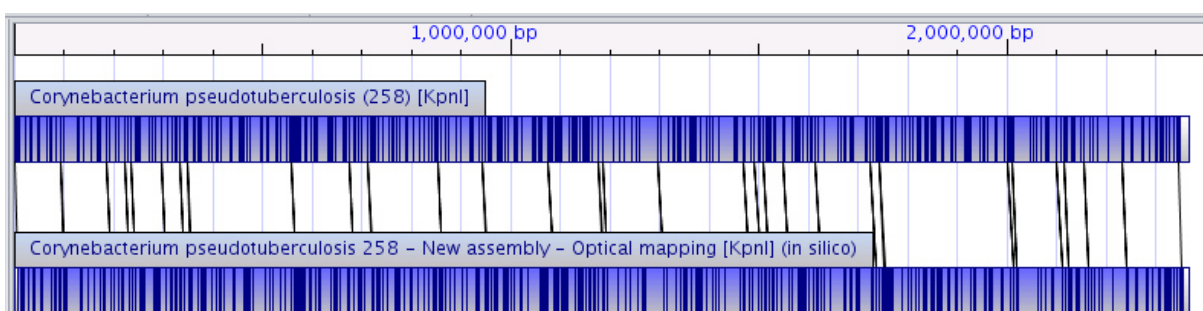


Figura 37 – Uso do MapSolver para validação da montagem realizada com SIMBA para Cp258.

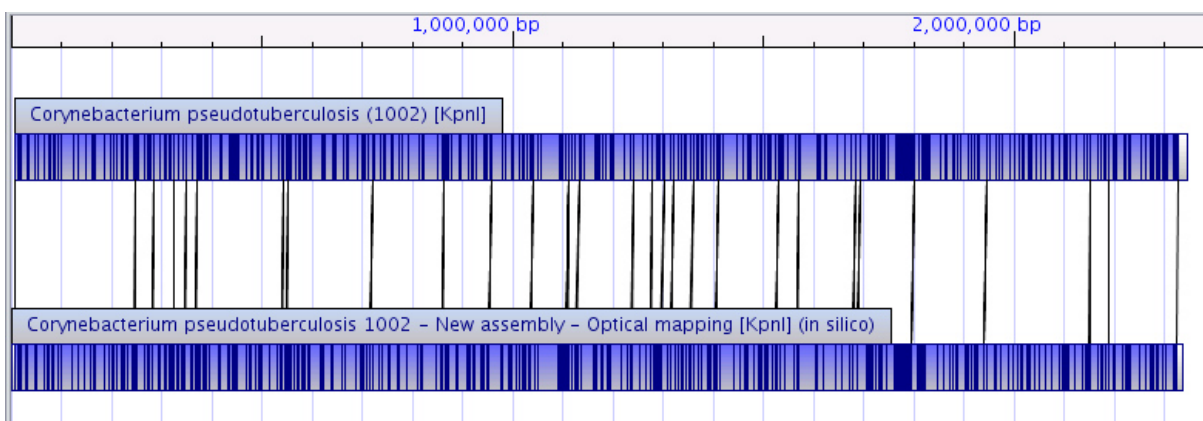


Figura 38 – Uso do MapSolver para validação da montagem realizada com SIMBA para Cp1002.

O genoma de Cp258 apresentou 2.368.328pb e o de Cp1002 apresentou 2.334.892pb.

### 3.2.4 Comparação entre montagens

#### 3.2.4.1 Diferença entre tamanhos de genoma

Na montagem por referência de Cp1002, o genoma obtido foi 859pb maior do que o depositado no NCBI, enquanto na montagem por mapeamento óptico foi 221pb menor (Tabela 7). Com base no estudo de Jünemann e colaboradores (2013) é possível inferir que erros de inserções e deleções em dados oriundos de Ion

PGM™ (leituras de 200pb) podem causar pequenas variações no tamanho final do genoma dependendo da forma como os dados são processados. Assim, pode se considerar que essas variações não representam diferenças significativas de tamanho.

Entretanto, em Cp258 há uma elevada diferença entre o tamanho do genoma depositado no NCBI e o tamanho dos genomas nas montagens finalizadas por referência e por mapeamento óptico (Tabela 7). A diferença, superior a 55.000pb, indica que a nova montagem apresenta uma vasta quantidade de genes não identificados anteriormente.

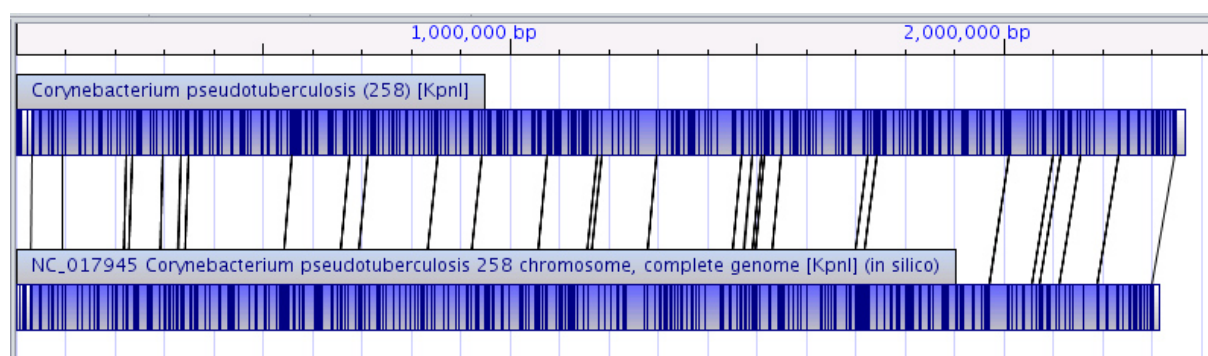
**Tabela 7 – Comparação entre tamanho de genoma depositado no NCBI, da montagem por referência e da montagem por mapeamento óptico para Cp258 e Cp1002.**

	Cp258	Cp1002
Genoma depositado no NCBI	2.314.404pb	2.335.113pb
Montagem por referência	2.370.835pb	2.335.972pb
Montagem por mapeamento óptico	2.368.328pb	2.334.892pb

### **3.2.4.2 Alinhando sequências do NCBI com mapa de restrição**

O mapa de restrição foi utilizado para avaliar a qualidade da montagem dos genomas de Cp258 e Cp1002 depositados no NCBI. Para isso, foi feito o *download* dos genomas, importação desses dados para o MapSolver e alinhamento das sequências com os mapas de restrição.

O genoma de *C. pseudotuberculosis* 258 (NC\_017945) não apresentou grandes “*mismatches*” quando alinhado ao mapa de restrição (Figura 39). Entretanto, é possível visualizar diferenças entre o tamanho do genoma e do mapa de restrição.



**Figura 39 – Comparativo no MapSolver do mapa de restrição (acima) com o genoma de *Corynebacterium pseudotuberculosis* 258 (NC\_017945) depositado no NCBI (abaixo).**

Entretanto a mesma análise para o genoma de *C. pseudotuberculosis* 1002 (NC\_017300) mostrou uma região superior a um milhão de pares de base, em posição inversa (Figura 40). Essa inversão no genoma depositado foi propagada na montagem finalizada por referência.

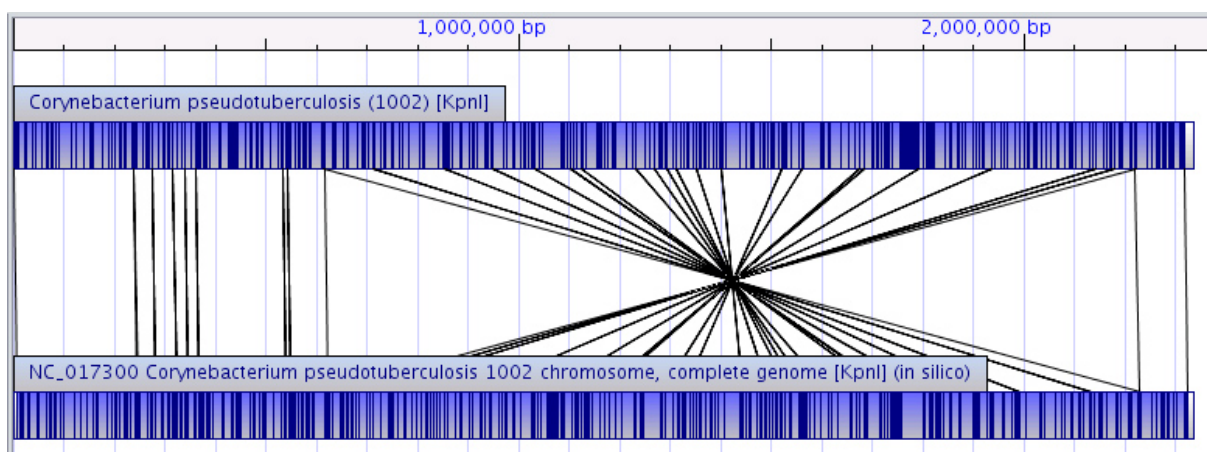


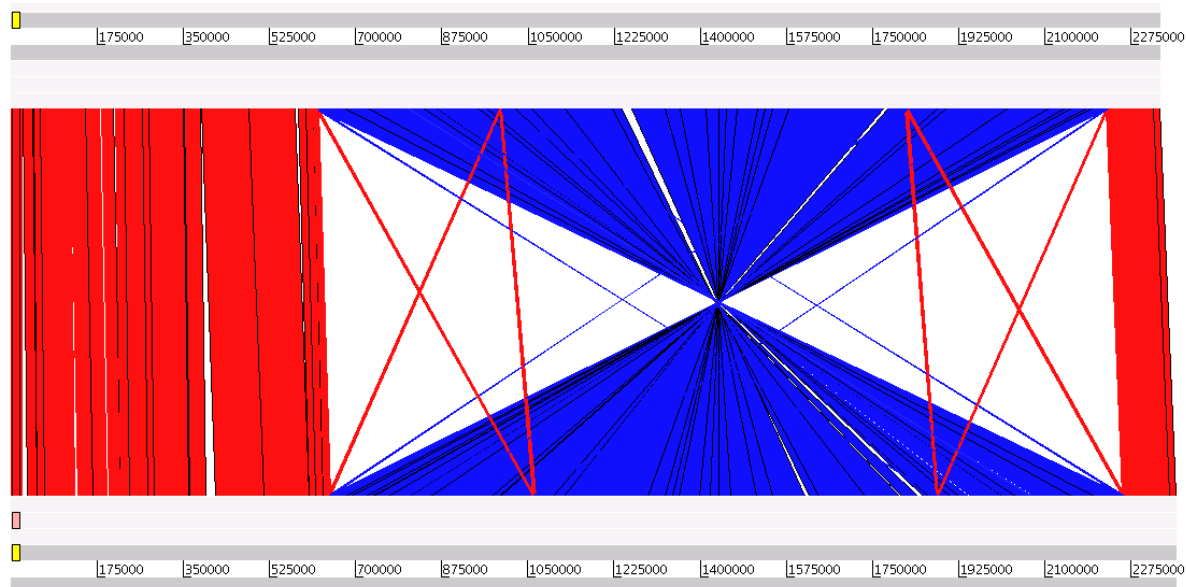
Figura 40 – Comparativo no MapSolver do mapa de restrição (acima) com o genoma de *Corynebacterium pseudotuberculosis* 1002 (NC\_017300) depositado no NCBI (abaixo).

#### **3.2.4.2 Comparando o genoma de Cp258 com o genoma de Cp1002**

A detecção de uma grande inversão em Cp1002 através do mapeamento óptico, inversão que não ocorre em Cp258, levantou a hipótese de não haver uma grande sintonia entre linhagens de biovars diferentes na espécie *Corynebacterium pseudotuberculosis*. Para testar essa hipótese, os dois genomas obtidos pela montagem finalizada com base no mapeamento óptico foram submetidos a uma comparação no WebACT<sup>27</sup>. Em seguida os resultados foram analisados com a ferramenta ACT<sup>28</sup>. O gráfico gerado (Figura 41) pode ser utilizado para comprovar a existência da inversão entre elas. É possível visualizar linhas que interligam quatro diferentes pontos (duas em vermelho e duas em azul para cada região), as quais representam regiões que codificam rRNA. Com base nisso é possível dizer que as inversões ocorrem entre duas regiões de *operons* de rRNA, entretanto não foi possível entender os motivos pelo qual essas inversões ocorreram.

<sup>27</sup> Disponível em: <<http://www.webact.org/WebACT/home>>. Acesso em: 16 de agosto, 2014.

<sup>28</sup> Disponível em: <<http://www.sanger.ac.uk/resources/software/act/>>. Acesso em: 16 de agosto, 2014.



**Figura 41 – Comparativo entre Cp1002 (acima) e Cp258 (abaixo) usando ACT.**

Linhas vermelhas indicam regiões homólogas, enquanto linhas em azul indicam inversões. É possível observar linhas vermelhas ligando diferentes regiões próximas ao bloco de linhas azuis. Essas regiões são codificadoras de rRNA.

Assim, conclui-se que a realização de experimentos de mapeamento óptico em outras linhagens da espécie *Corynebacterium pseudotuberculosis* é de grande importância, pois permite a detecção de erros de montagem em grandes regiões repetitivas. Entretanto, cabe ressaltar que essas grandes inversões podem ocorrer aleatoriamente sob condições ainda desconhecidas. O que indica a possibilidade de mudanças na estrutura do genoma entre dois sequenciamentos distintos.



## 4. Considerações finais

A interface de SIMBA demonstrou grande usabilidade, facilitando assim a execução de diversos processos de montagem de genomas procariotos através de diferentes estratégias. Descomplicar os processos de montagem permite que bioinformatas se preocupem menos com atividades técnicas, e assim, possam se dedicar em estudar, analisar e buscar compreender os processos biológicos.

Durante o estudo de caso, em que houve uma comparação entre diferentes tipos de finalização de montagens, pode-se perceber a inversão de uma grande região no genoma de Cp1002, que não existe em Cp258. Isso levantou diversas questões: quais as vantagens para o organismo obtidas por essa inversão em seu genoma? Isso causou alguma alteração em seu estilo de vida? Quais seriam as condições biológicas para que a inversão ocorresse? Poderiam organismos com genomas tão distintos ser da mesma espécie? Será que essa inversão ocorre em outras linhagens de *C. pseudotuberculosis* do mesmo biovar? Este trabalho não tem por objetivo tentar responder essas questões, entretanto é importante levá-las para futuros trabalhos.

Quanto aos *softwares* de montagem *de novo*, era esperado que aqueles que utilizassem a abordagem OLC obteriam um melhor resultado de montagem com dados de Ion PGM™, devido à menor sensibilidade do algoritmo a erros de homopolímero. Isso destaca os bons resultados de Mira3, Mira4 e Newbler. Entretanto, SPAdes também apresentou um bom resultado devido à presença de uma etapa de correções de leituras em seu *pipeline*. Mira3 foi superior a Mira4 em quase todos os parâmetros avaliados, o que pode se considerar um resultado atípico por Mira4 se tratar de uma versão posterior a Mira3. Apesar da rápida execução, Minia não apresentou bons resultados, o que não está totalmente ligado ao algoritmo utilizado por ele, mas possivelmente ao fato de ter sido desenvolvido para trabalhar com genomas de eucariotos.

O mapeamento óptico oferece mapas visuais de genomas de alta definição, porém não é capaz de ordenar com precisão *contigs* muito pequenos (em alguns casos, menores que 70Kb). Nesse caso, a abordagem por referência foi eficiente para auxiliar na ordenação desses *contigs*. Entretanto, pode-se perceber que a finalização de montagens apenas por referência pode ser ineficaz em alguns casos,

como em Cp1002, por permitir a propagação de erros ocorridos em montagens anteriores.

O código-fonte de SIMBA foi disponibilizado à comunidade desenvolvedores pela plataforma Sourceforge. Através de <<http://ufmg-simba.sourceforge.net>> pode-se fazer o *download* do código-fonte, que poderá ser livremente modificado e distribuído.

#### 4.1 Perspectivas para trabalhos futuros

- Inserir o *software* de montagem SPAdes ao SIMBA;
- Desenvolver um módulo para adição automática de novos *softwares* de montagem;
- Permitir que as definições padrões de montagem *de novo* sejam alteradas pelo usuário na interface;
- Inserir um módulo de finalização de montagem *de novo*, comparando similaridades apenas entre *contigs*;
- Realizar um estudo para compreender os motivos que levaram a inversão entre os genomas de *C. pseudotuberculosis* 258 e *C. pseudotuberculosis* 1002;
- Realizar o mapeamento óptico de outras linhagens da espécie *C. pseudotuberculosis* a fim de verificar se a inversão ocorrida em Cp1002 existe em outras linhagens.

## 5. Referências bibliográficas

ALTSCHUL, S. F.; GISH, W.; MILLER, W.; MYERS, E. W.; LIPMAN, D. J. **Basic local alignment search tool**. J Mol Biol. 1990 Oct 5; 215(3):403-10.

ANANIEV, G. E.; GOLDSTEIN, S.; RUNNHEIM, R.; FORREST, D. K.; ZHOU, S.; POTAMOUSIS, K.; CHURAS, C. P.; BERGENDAHL, V.; THOMSON, J. A.; SCHWARTZ, D. C. **Optical mapping discerns genome wide DNA methylation profiles**. BMC Mol Biol.; 7:68. 2008.

AZIZ, R. K.; *et al.* **The RAST Server: rapid annotations using subsystems technology**. BMC Genomics 9, 75, 2008.

BANKEVICH, A.; NURK, S.; ANTIPOV, D.; GUREVICH, A. A.; DVORKIN, M.; KULIKOV, A. S.; LESIN, V. M.; NIKOLENKO, S. I.; PHAM, S.; PRJIBELSKI, A. D.; PYSHKIN, A. V.; SIROTKIN, A. V.; VYAHHI, N; TESLER, G.; ALEKSEYEV, M. A.; PEVZNER, P. A. **SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing**. Journal of Computational Biology. Volume 19, Number 5, 2012.

BOETZER, Marten; HENKEL, Christiaan V.; JANSEN, Hans J.; BUTLER, Derek; PIROVANO, Walter. **Scaffolding pre-assembled contigs using SSPACE**. Oxford University Press. Dezembro, 2010.

BOETZER, Marten; PIROVANO, Walter. **Toward almost closed genomes with GapFiller**. Genome Biology, 13:R56, 2012.

BONETTA, L. **Genome sequencing in the fast lane**. Nat Methods, 3:141-147, 2006.

CERDEIRA, L. T.; CARNEIRO, A. R.; RAMOS, R. T. J.; DE ALMEIDA, S. S.; D'AFONSECA, V.; SCHNEIDER, M. P. C., *et al.* **Rapid hybrid de novo assembly of a microbial genome using only short reads: Corynebacterium pseudotuberculosis I19 as a case study**. J Microbiol Methods 86, p. 218–223, 2011.

CHAISSON, M. J.; BRINZA, D.; PEVZNER, P. A. **De novo fragment assembly with short mate-paired reads: Does the read length matter?** Genome Res, 19:336–46, 2009.

CHEVREUX, B.; WETTER, T.; SUHAI, S. **Genome Sequence Assembly Using Trace Signals and Additional Sequence Information**. Computer Science and Biology: Proceedings of the German Conference on Bioinformatics (GCB) 99, pp. 45-56. 1999.

CHEVREUX, B.; PFISTERER, T.; DRESCHER, B.; DRIESEL, A. J.; WETTER, T. **Using the miraEST Assembler for Reliable and Automated mRNA Transcript Assembly and SNP Detection in Sequenced ESTs**. Genome Research, 14: 1147–1159, 2004.

CHIKHI, R.; RIZK, G. **Space-efficient and exact De Bruijn graph representation based on a Bloom filter**. Disponível em: <<http://minia.genouest.org/files/minia.pdf>>. WABI, 2012.

CUCCURU, Gianmauro; ORSINI, Massimiliano; PINNA, Andrea; SBARDELLATI, Andrea; SORANZO, Nicola; TRAVAGLIONE, Antonella; UVA, Paolo; ZANETTI, Gianluigi; FOTIA, Giorgio. **Orione, a web-based framework for NGS analysis in microbiology**. Oxford University Press. Bioinformatics. 2014.

DOHM, J.; LOTTAZ, C.; BORODINA, T.; HIMMELBAURER, H. **SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing**. Genome Research, 17:1697–1706, 2007.

DORELLA, F. A.; PACHECO, L. G. C.; OLIVEIRA, S. C.; MIYOSHI, A.; AZEVEDO, V. **Corynebacterium pseudotuberculosis: microbiology, biochemical properties, pathogenesis and molecular studies of virulence**. Veterinary Research 37, 201–218. 2006.

EL-METWALLY, Sara; HAMZA, Taher; ZAKARIA, Magdi; HELMY, Mohamed. **Next-Generation Sequence Assembly: Four Stages of Data Processing and Computational Challenges**. PLOS Computational Biology, 2013.

EWING, B.; HILLIER, L.; WENDL, M.; GREEN, P. **Base-Calling of Automated Sequencer Traces Phred. I. Using Accuracy Assessment**. Genome Research, 8:175–185, 1998.

FINNEGAN, D. J. **Eukaryotic transposable elements and genome evolution**. Trends Genet. 5, 103–107. 1989.

FLEISCHMANN, R. D.; *et al.* **Whole-genome random sequencing and assembly of *Haemophilus influenzae***. Rd. Science 269, 496–512, 1995.

FOSTER, J. B.; SLONCZEWSKI, J. **Microbiology: an evolving science**. New York: W.W. Norton & Co. ISBN 0-393-97857-5. 2009.

GALARDINI, Marco; BIONDI, Emanuele G.; BAZZICALUPO, Marco; MENGONI, Alessio. **CONTIGuator: a bacterial genomes finishing tool for structural insights on draft genomes**. Disponível em: <<http://www.scfbm.org/content/6/1/11>>. Source Code for Biology and Medicine, 6:11 doi:10.1186/1751-0473-6-11, 2011.

GARDY, J. L. *et al.* **Whole-genome sequencing and social-network analysis of a tuberculosis outbreak**. N. Engl. J. Med. 364, 730–739, 2011.

GOECKS, J. *et al.* **Galaxy: a comprehensive approach for supporting accessible, reproducible and transparent computational research in the life sciences**. Genome Biol., 11, R86. 2010.

HARISMENDY O.; NG P. C.; STRAUSBERG, R. L.; WANG, X.; STOCKWELL, T. B. **Evaluation of next generation sequencing platforms for population targeted sequencing studies.** *Genome Biology* 10 R32, 2009.

HERNANDEZ, D.; FRANCOIS, P.; FARINELLI, L.; OSTERAS, M.; SCHRENZEL, J. **De Novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer.** *Genome Res*, 18:802-809, 2008.

HUSEMANN, P. **Bioinformatic Approaches for Genome Finishing.** Bielefeld University, Alemanha, 2011.

JECK, W.; REINHARDT, J.; BALTRUS, D.; HICKENBOTHAM, M.; MAGRINI, V.; MARDIS, E.; DANGL, J.; JONES, C. **Extending assembly of short DNA sequences to handle error.** *BMC Bioinformatics*, 23:2942–2944, 2007.

JÜNEMANN, Sebastian; PRIOR, Karola; ALBERSMEIER, Andreas; ALBAUM, Stefan; KALINOWSKI, Jörn; GOESMANN, Alexander; STOYE, Jens; HARMSSEN, Dag. **GABenchToB: A Genome Assembly Benchmark Tuned on Bacteria and Benchtop Sequencers.** *PLOS ONE*, volume 9, 2014.

JÜNEMANN, Sebastian; SEDLAZECK, Fritz Joachim; PRIOR, Karola; ALBERSMEIER, Andreas; JOHN, Uwe; KALINOWSKI, Jörn; MELLMANN, Alexander; GOESMANN, Alexander; HAESSELER, Arndt von; STOYE, Jens; HARMSSEN, Dag. **Updating benchtop sequencing performance comparison.** *Nature Biotechnol*, 31, 294–296, 2013.

KAUR, Ritesh; MALIK, Chander Parkash. **Next Generation Sequencing: a Revolution in Gene Sequencing.** *CIBTech Journal of Biotechnology* ISSN: 2319-3859 (Online). Disponível em: <<http://www.cibtech.org/cjb.htm>>. Vol. 2 (4) October-December, pp.1-20, 2013.

KIRCHER, M.; KELSO, J. **High-throughput DNA sequencing - concepts and limitations.** *Bioessays* 32: 524–536, 2010.

KOREN, S.; SCHATZ, M. C.; WALENZ, B. P.; MARTIN, J.; HOWARD, J. T.; GANAPATHY, G.; WANG, Z.; RASKO, D. a.; MCCOMBIE, W. R.; JARVIS, E. D.; PHILLIPPY, A. M. **Hybrid error correction and de novo assembly of single-molecule sequencing reads.** *Nature biotechnology*, Nature Publishing Group, v. 30, n. 7, p. 693–700, 2012.

LAM, H. Y. K.; *et al.* **Performance comparison of whole-genome sequencing platforms.** *Nature biotechnology* 30: 78–82, 2012.

LANDER, E. S.; LINTON, L. M.; BIRREN, B.; *et al.* **Initial sequencing and analysis of the human genome.** *Nature*, v. 409, n. 6822, p. 860–921, 2001.

LEWIS, T.; *et al.* **High-throughput whole-genome sequencing to dissect the epidemiology of *Acinetobacter baumannii* isolates from a hospital outbreak.** *J. Hosp. Infect.* 75, 37–41 (2010).

LI R.; *et al.* **De novo assembly of human genomes with massively parallel short read sequencing.** 265–272. 2010.

LI R., LI Y.; KRISTIANSEN K.; WANG, Jun. **SOAP: short oligonucleotide alignment program.** *Bioinformatics* (Oxford, England) 24: 713–714, 2008.

LIFE TECHNOLOGIES. **Ion PGM™ and Ion Proton™ System Chips.** Disponível em: <<http://www.lifetechnologies.com/br/en/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-workflow/ion-torrent-next-generation-sequencing-run-sequence/ion-pgm-ion-proton-system-chips.html>>. Acesso em: 12 de setembro, 2014.

LOMAN, N. J.; CONSTANTINIDOU, C.; CHAN, J. Z. M.; HALACHEV, M.; SERGEANT, M.; PENN, C. W.; ROBINSON, E. R.; PALLAN, M. J. **High-throughput bacterial genome sequencing: an embarrassment of choice, a world of opportunity.** *Nature reviews. Microbiology*, Nature Publishing Group, v. 10, n. 9, p. 599–606, set. 2012a.

LOMAN, N. J. *et al.* **Performance comparison of benchtop high-throughput sequencing platforms.** *Nature Biotech.* 30, 434–439, 2012b.

MARGULIES, M.; *et al.* **Genome sequencing in microfabricated high-density picolitre reactors.** *Nature* 437, 376–380, 2005.

MELLMANN, A.; *et al.* **Prospective genomic characterization of the German enterohemorrhagic *Escherichia coli* O104:H4 outbreak by rapid next generation sequencing technology.** *PLoS ONE* 6, e22751, 2011.

METZKER, M. L. **Emerging technologies in DNA sequencing.** *Genome Res.* 15, 1767–1776, 2005.

MILLER, J. R., KOREN, S.; SUTTON, G. **Assembly algorithms for next-generation sequencing data.** *Genomics* 95: 315-327, 2010.

NAGARAJAN, N.; POP, M. **Sequence assembly demystified.** *Nature reviews. Genetics*, Nature Publishing Group, v. 14, n. 3, p. 157–67, 2013.

NAGARAJAN, N.; COOK, C.; Di Bonaventura, M.; GE, H.; RICHARDS, A.; BISHOP-LILLY, K. a.; DESALLE, R.; READ, T. D.; POP, M. **Finishing genomes with limited resources: lessons from an ensemble of microbial genomes.** *BMC genomics*, v. 11, p. 242, 2010.

ONMUS-LEONE, Fatma; HANG, Jun; CLIFFORD, Robert J.; YANG, Yu; RILEY, Matthew C.; KUSCHNER, Robert A.; WATERMAN, Paige E.; LESHO, Emil P. **Enhanced De Novo Assembly of High Throughput Pyrosequencing Data Using Whole Genome Mapping.** *PLoS ONE* 8 (4). 2013.

POP, M. **Genome assembly reborn: recent computational challenges.** *Briefings in bioinformatics*, v. 10, n. 4, p. 354–66, 2009.

PEVZNER P. A.; TANG, H.; WATERMAN, M. S. **An Eulerian path approach to DNA fragment assembly**. Proc Natl Acad Sci USA, 98:9748–53, 2001.

QUAIL, Michael A.; SMITH, M.; COUPLAND, P.; OTTO, T. D.; HARRIS, S. R.; CONNOR, T. R.; BERTONI, A.; SWERDLOW, H. P.; GU, Y. **A tale of three next generation sequencing platforms: comparison of Ion Torrent™, Pacific Biosciences and Illumina MiSeq sequencers**. BMC genomics, 13: 341, 2012.

RAMOS, Rommel Thiago Jucá; CARNEIRO, Adriana Ribeiro; SOARES, Siomar de Castro; SANTOS, Anderson Rodrigues dos; ALMEIDA, Sintia; GUIMARÃES, Luis; FIGUEIRA, Flávia; BARBOSA, Eudes; TAUCH, Andreas; AZEVEDO, Vasco; Silva, Artur. **Tips and tricks for the assembly of a Corynebacterium pseudotuberculosis genome using a semiconductor sequencer**. Microbial biotechnology, 2012.

RAMOS, Rommel Thiago Jucá. **Desenvolvimento de um “suíte” de aplicativos computacionais para suporte à montagem de genomas bacterianos a partir de leituras curtas**. Dissertação de mestrado. Programa de Pós-Graduação em Genética e Biologia Molecular da UFPA. Belém (PA), 2011.

RAMOS, R. T.; CARNEIRO, A. R.; CARACCILO, P. H.; AZEVEDO, V.; SCHNEIDER, M. P.; BARH, D.; SILVA, A. **Graphical contig analyzer for all sequencing platforms (G4ALL): a new stand-alone tool for finishing and draft generation of bacterial genomes**. Bioinformatics, 9(11): 599-604, 2013.

RAMOS, Rommel Thiago Jucá. **Pipeline para obtenção de genomas bacterianos por sequenciamento semicondutor**. Tese de doutorado. Programa de Pós-Graduação em Genética e Biologia Molecular da UFPA, Belém (PA), 2013.

RIBEIRO, F. J.; PRZYBYLSKI, D.; YIN, S.; SHARPE, T.; GNERRE, S.; ABOUELLEIL, A.; BERLIN, A. M.; MONTMAYEUR, A.; SHEA, T. P.; WALKER, B. J.; YOUNG, S. K.; RUSS, C.; NUSBAUM, C.; MACCALLUM, I.; JAFFE, D. B. **Finished bacterial genomes from shotgun sequence data**. Genome research, v. 22, n. 11, p. 2270–7, 2012.

ROTHBERG, Jonathan M.; HINZ, Wolfgang; REARICK, Todd M.; *et al.* **An integrated semiconductor device enabling non-optical genome sequencing**. Nature, 475(7356):348–352, 2011.

RONAGHI, M. **Pyrosequencing sheds light on DNA sequencing**. Genome Res. 11, 3–11, 2001.

RUIZ, J. C. *et al.* **Evidence for reductive genome evolution and lateral acquisition of virulence functions in two Corynebacterium pseudotuberculosis strains**. PLoS ONE 6, e18551. 2011.

SAMAD, A.; HUFF, E. F.; CAI, W.; *et al.* **Optical mapping: a novel, single-molecule approach to genomic analysis**. Genome Res, 5:1–4, 1995.

SCHADT, E. E.; TURNER, S.; KASARSKIS, A. **A window into third-generation sequencing. Human molecular genetics**, v. 19, n. R2, p. R227–40, 2010.

SEEMANN, Torsten. **Prokka: rapid prokaryotic genome annotation**. Oxford University Press, 2014.

SIMS, David; SUDBERY, Ian; ILOTT, Nicholas E.; HEGER, Andreas; PONTING, Chris P. **Sequencing depth and coverage: key considerations in genomic analyses**. Nature Reviews. Volume 15, p. 121-132. 2014.

SOARES, Siomar C.; *et al.* **Genome sequence of *Corynebacterium pseudotuberculosis* biovar equi strain 258 and prediction of antigenic targets to improve biotechnological vaccine production**. Journal of Biotechnology. Volume 167, p. 135–141. 2013.

VENTER, J. C.; ADAMS, M. D.; MYERS, E. W.; *et al.* **The sequence of the human genome**. Science (New York, N.Y.), v. 291, n. 5507, p. 1304–51, 2001.

WARREN, R. L.; *et al.* **Assembling millions of short DNA sequences using SSAKE**. Bioinformatics (Oxford, England), 23, 500-501, 2007.

WETTERSTRAND, K. A. **DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)**. Disponível em: <[www.genome.gov/sequencingcosts](http://www.genome.gov/sequencingcosts)>. Acesso em: 10 de junho, 2014.

WICKER, Thomas; SABOT, François; HUA-VAN, Aurélie; BENNETZEN, Jeffrey L.; CAPY, Pierre; CHALHOUB, Boulos; FLAVELL, Andrew; LEROY, Philippe; MORGANTE, Michele; PANAUD, Olivier; PAUX, Etienne; SANMIGUEL, Phillip; SCHULMAN, Alan H. **A unified classification system for eukaryotic transposable elements**. Nature Reviews, volume 8. 2007.

XAVIER, B. B.; SABIROVA, J.; PIETER, M.; HERNALSTEENS, J. P.; DE GREVE, H.; GOOSSENS, H.; MALHOTRA-KUMAR, S. **Employing whole genome mapping for optimal de novo assembly of bacterial genomes**. BMC Research Notes, 7:484, 2014.

ZERBINO, D. R.; BIRNEY, E. **Velvet: algorithms for de novo short read assembly using De Bruijn graphs**. Genome research, 18: 821–829, 2008.



## **Anexos**

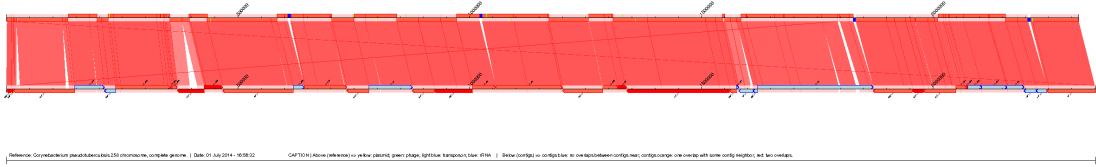
## Prêmios

1º lugar apresentação de pôsteres no ISCB - Latin America X-Meeting on Bioinformatics with BSB and SoiBio 2014, ISCB - LA / X-Meeting / BSB / SoiBio.

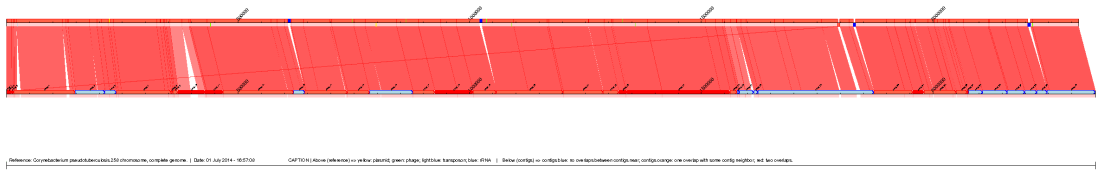


# Gráficos de sintenia gerado pelo CONTIGuator2 para finalização por referência - *C. pseudotuberculosis* 258 (Mira 3.9)

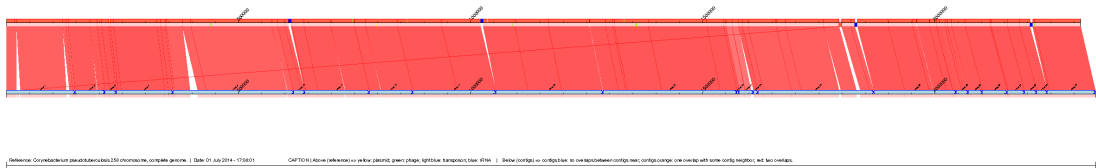
## Etapa 1



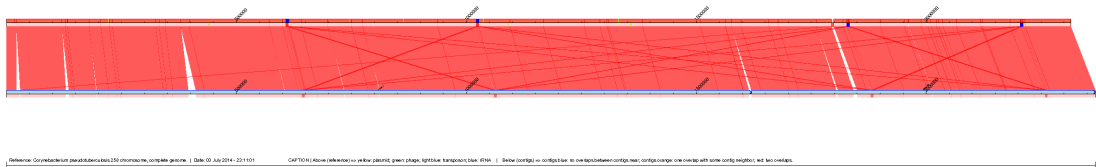
## Etapa 2



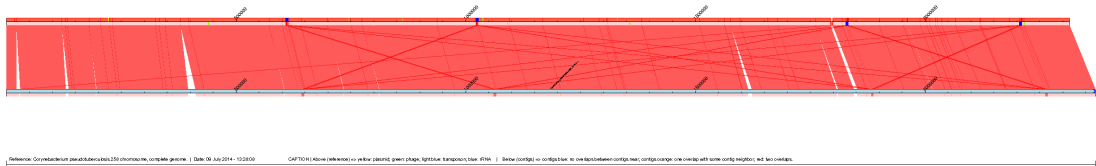
## Etapa 3



## Etapa 4

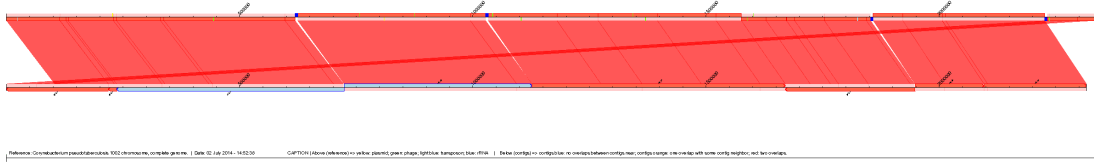


## Etapa 5

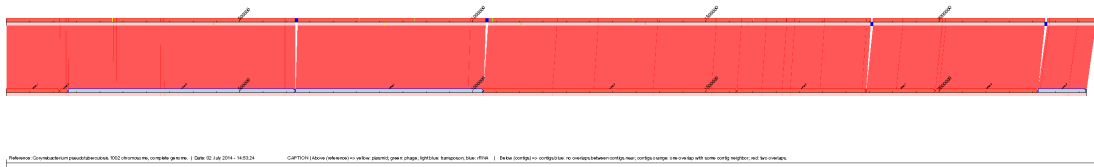


Gráficos de sintenia gerado pelo CONTIGuator2 para finalização por referência - *C. pseudotuberculosis* 1002 (Mira 3.9)

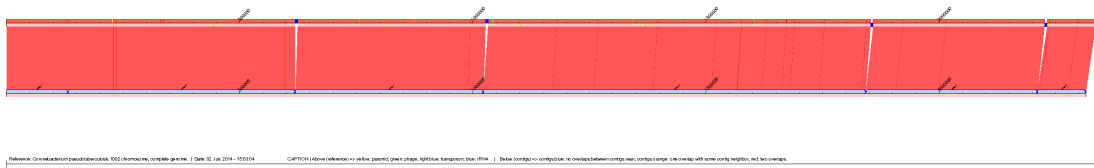
Etapa 1



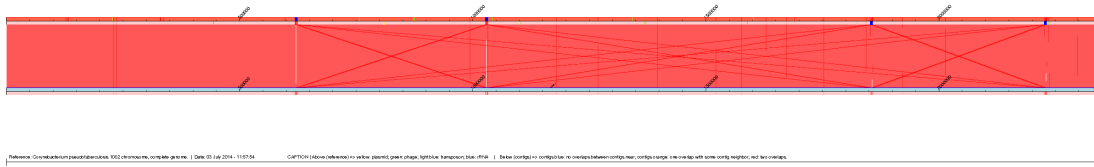
Etapa 2



Etapa 3



Etapa 4



Etapa 5

