

UNIVERSIDADE FEDERAL DO PARANÁ

DIEVAL GUIZELINI

**G-FINISHER: Uma nova estratégia para refinar
e finalizar montagens de genomas bacterianos**

**CURITIBA
2016**

UNIVERSIDADE FEDERAL DO PARANÁ

DIEVAL GUIZELINI

**G-FINISHER: Uma nova estratégia para refinar
e finalizar montagens de genomas bacterianos**

Tese apresentada ao Curso de Pós-Graduação
em Bioquímica da Universidade Federal do
Paraná como requisito parcial para obtenção
do título de Doutor em Ciências-Bioquímica.

Orientador: Prof. Dr. Fábio de Oliveira Pedrosa
Coorientadores:
Prof.^a Dr.^a Maria Berenice Reynaud Steffens
Prof. Dr. Roberto Tadeu Raittz

**CURITIBA
2016**

TERMO DE APROVAÇÃO

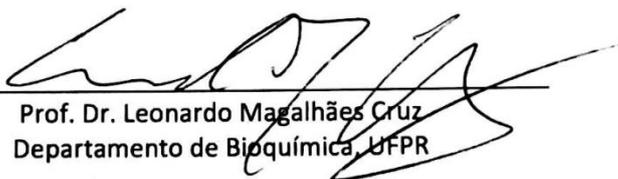
DIEVAL GUIZELINI

G-FINISHER: Uma nova estratégia para refinar e finalizar montagens de genomas bacterianos

Tese aprovada como requisito parcial para a obtenção do grau de Doutor no curso Pós-Graduação em Ciências-Bioquímica, Setor de Ciências Biológicas, Universidade Federal do Paraná, pela seguinte banca examinadora:



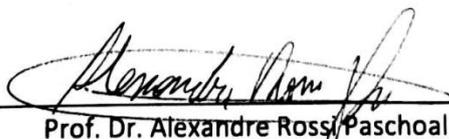
Prof. Dr. Fabio de Oliveira Pedrosa – Orientador
Departamento de Bioquímica, UFPR



Prof. Dr. Leonardo Magalhães Cruz
Departamento de Bioquímica, UFPR



Prof. Dr. Emanuel Maltempi de Souza
Departamento de Bioquímica, UFPR



Prof. Dr. Alexandre Rossi Paschoal
PPG – Bioinformática, UTFPR



Dr. Helisson Faoro
ICC, FIOCRUZ – PR

Curitiba, 27 de julho de 2016.

*À minha mãe, Alice.
À minha esposa, Juliana.
Aos meus filhos, Lucas e João.
Aos meus irmãos, Dayane e Diges.*

AGRADECIMENTOS

À minha mãe, Alice Feltrin, pela vida, pela educação que me propiciou e por toda ajuda que me presta;

à minha esposa Juliana, que tanto amo, pelo acolhimento e cumplicidade. Que nosso amor e nosso tempo não sejam eternos, mas que sejam infinitos enquanto durem;

aos professores Fabio de Oliveira Pedrosa e Maria Berenice Reynaud Steffens, pela amizade, pela oportunidade, pelas orientações, pela paciência e dedicação nesta jornada. Não tenho palavras suficientes para descrever a sincera gratidão, particularmente pela confiança depositada em meu trabalho;

ao professor Roberto Tadeu Raittz, meu amigo, pelo apoio e ajuda que tornou as encruzilhadas inerentes a este trabalho menos solitárias;

à amiga Jeroniza Nunes Marchaukoski, pelo incentivo e discussões sobre temas da bioinformática, banco de dados e políticas universitárias;

ao companheiro Mario de Paula Soares Filho, pela parceria na representação de nossas carreiras junto ao Conselho de Planejamento e Administração (COPLAD) e pelas experiências vividas com as invasões, ao gás lacrimogêneo..., e nas contribuições que fizemos para reverter a extinção de nossa carreira na UFPR, no meio de nossos doutorados;

aos professores Emanuel Maltempi de Souza e Leonardo Magalhães Cruz, pela amizade, sugestões e considerações no desenvolvimento desta tese;

aos professores Glaucia Regina Martinez e Emanuel Maltempi de Souza, pelos trabalhos frente à Coordenação do Programa de Pós-Graduação em Ciências-Bioquímica;

ao colegiado do Programa de Pós-Graduação em Ciências-Bioquímica pela acolhida, oportunidade e ensinamentos;

a Thiago Vello, Secretário do PPG em Ciências-Bioquímica, pelos atendimentos e e-mails, que nos auxiliaram nos processos burocráticos;

aos amigos Helisson, Michelle, Valter, Eduardo, Roseli pelos sequenciamentos, esclarecimentos, contribuições e pelo convívio;

ao amigo Rodrigo Cardoso, pelas conversas, revisões e contribuições nesta tese;

a todos os colegas que me auxiliaram durante a realização dos créditos;

aos amigos do Colegiado do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo apoio e pelo afastamento concedido;

aos professores Luiz Antônio Passos Cardoso, Silvana Maria Carbonera e Adriano Moraes, diretores do Setor de Educação Profissional e Tecnológica, pelo apoio e pela concessão do afastamento para realização deste doutorado;

aos amigos do Setor de Educação Profissional e Tecnológica, por me apoiarem nesta trajetória;

a todos que, direta ou indiretamente, contribuíram para eu enfrentar este imenso desafio acadêmico, sou, sinceramente,

muito grato!

*“As invenções são, sobretudo,
o resultado de um trabalho teimoso.”*

Alberto Santos Dumont

*“A imaginação é mais importante que o
conhecimento. O conhecimento é limitado,
enquanto a imaginação abraça o mundo inteiro,
estimulando o progresso, dando à luz à
evolução”*

Albert Einstein

RESUMO

O processo de reconstrução completa da sequência de DNA dos genomas bacterianos ainda é complexo. Apenas 13% dos projetos de sequenciamento de genomas procarióticos são concluídos. Versões rascunho da sequência do genoma são depositadas nos bancos de dados públicos, na forma fragmentada de contigs e com prováveis perdas de informações gênicas. Esta tese tem o objetivo de identificar erros de montagem e melhorar o processo de montagem de genomas de bactérias. Padrões biológicos observados em sequências genômicas e a utilização de informação *a priori* permitem a identificação de regiões com erros de montagem, reorganizar as sequências e melhorar a montagem do genoma. Com a finalidade de melhorar a finalização das montagens, os contigs são quebrados nos pontos de máximo e mínimo local da curva Fuzzy-GC-Skew e armazenados em nós de um grafo sem bordas. Esses nós são ordenados com base na sequência de referência e submetidos para fechamento das lacunas pelo jFGap. No método desenvolvido neste trabalho – G-Finisher –, os contigs são quebrados nos pontos críticos da curva Fuzzy GC Skew, reordenados e as lacunas fechadas com o jFGap. O G-Finisher foi testado nas 96 montagens obtidas pelo GAGE-B e reduziu na média 86% o número de contigs. G-Finisher pode facilmente melhorar os projetos de montagens de genomas de procariotos, de modo que os programas de montagem podem ser melhorados com a incorporação do G-Finisher ou com a utilização de padrões de sequências biológicas. O software e o código-fonte, escrito em Java, foram licenciados na forma do software livre e disponibilizados em <http://gfinisher.sourceforge.net/>.

ABSTRACT

The process of reconstruction of complete genome from DNA sequences is still complex. Only 13% of the prokaryotic genome sequencing projects are completely finished. Draft genome sequences deposited in public databases are fragmented in contigs and may lack the full gene content. To identify assembly errors and improve the assembly process of bacterial genomes are the purpose of this work. The biological patterns observed in genomic sequences and the application of *a priori* information allows the identification of misassembled regions, and the reorganization and improvement of the overall genome assembly. In order to improve the finishing of genome assemblies the contigs are broken down at the peaks (all critical points) of a Fuzzy-GC-Skew-Moving-Average graph and stored in computer nodes in a graph data structure without edges. These nodes are ordered following a reference and submitted to the gap closing software jFGap. In the proposed new method – GFinisher – critical peaks in Fuzzy GC skew graphs are broken down, reassembled and closed using jFGap. The number of contigs decreases by up 86%. This has been successfully applied to the 96 genome assemblies described and provide by GAGE-B. GFinisher can easily optimize assemblies of prokaryotic draft genomes and can be used to improve the assembly programs using biological genome sequence patterns. The software was written em Java, licensed in open-source and the binaires and source code are available at <http://gfinisher.sourceforge.net/>.

Keywords: genome finisher, gap close, contig order, genome assembly

LISTA DE FIGURAS

Figura 1 - Protocolo de montagem <i>De Novo</i> da AB SOLiD 3.....	29
Figura 2 – Pipeline de montagem A5	31
Figura 3 - Exemplo do grafo de bruijn	34
Figura 4 - (a) P é consistente para $P_{x,y1}$, mas é inconsistente para $P_{x,y2}$; (b) P é inconsistente para ambos os $P_{x,y1}$ e $P_{x,y2}$ e; (c) P é consistente para os dois $P_{x,y1}$ e $P_{x,y2}$	37
Figura 5 - Diagrama esquemático das três categorias de erros no Velvet	44
Figura 6 - Ciclos são caminhos que convergem em si mesmos, induzidos por repetições curtas na sequência	44
Figura 7 - Captura da Tela do FastQC do relatório da análise das leituras da <i>H. hiltneri</i> N3 fornecidas pelo illumina MiSeq.....	58
Figura 8 - captura do relatório do percentual da ocorrência de base por coluna nAS leituras da <i>H. hiltneri</i> N3	58
Figura 9 - Viés se repete em outros organismos e tecnologias de sequenciamento	59
Figura 10 - Diferenças de alinhamentos de sequências graficamente demonstradas no Dotplot	66
Figura 11 - Comparação entre três versões de montagens da <i>Herbaspirillum hiltneri</i> N3 feita pelo QUAST, sem indicação de referência.....	68
Figura 12 - Esquema de segmentação e identificação das regiões comparadas.....	76
Figura 13 - efeito da indução do comportamento do GC Skew com a padronização da representação dos <i>contigs</i> baseada na relação de frequência G>C	81
Figura 14 - código-fonte do “todosMer.sh”	84
Figura 15 - Fluxograma do G-Finisher	86
Figura 16 - Exemplo da diferença de sensibilidade na detecção dos pontos críticos nas curvas GC-Skew e Fuzzy-GC-Skew.....	91
Figura 17 - aplicação do “finder” nas curvas (F)GC Skews após escolha do sentido e preservando a ordem original dos <i>contigs</i> obtidos na montagem de <i>Aeromonas hydrophila</i> pelo MaSuRCA	92

Figura 18 - Ampliação da Figura 19 na região de 4.350.000 pb e 4.650.000 pb que demonstra as diferenças de sensibilidades dos métodos (F)GC Skew e os pontos críticos identificados pelo “Finder” para quebra dos <i>contigs</i>	93
Figura 19 - Comparação das médias dos números de <i>contigs</i> das quinze montagens por organismo entre os resultados do GAGE-B (azul) e do G-Finisher (vermelho).....	94
Figura 20 - Redução de erro na escolha pelos montadores na ocorrência do dilema do caminho, através da preservação do contexto GC	97
Figura 21 - Captura das telas do FastQC da análise da qualidade das leituras SOLiD originais e depois do tratamento	97
Figura 22 - GC Skew da montagem do genoma da <i>Herbaspirillum hiltneri</i> N3 em 02/04/2014	99
Figura 23 - GC Skew da montagem do genoma da <i>Herbaspirillum hiltneri</i> N3 em 03/04/2014	99
Figura 24 - Captura da imagem produzida pelo g-finisher do Dotplot dos dois <i>contigs</i> da montagem do 4º plasmídeo da <i>A. brasilense</i> FP2 (x) comparados com o 3º plasmídeo da <i>A. brasilense</i> Sp7	103

LISTA DE TABELAS

Tabela 1 - Exemplo da escala phred de qualidade e a relação dos valores com a precisão e probabilidade de erros na leitura de base.	47
Tabela 2 - Representação Simbólica dos valores de qualidade no formato fastq.....	47
Tabela 3 - Características das leituras e do sequenciamento	56
Tabela 4 - Número de <i>contigs</i> maiores que 199 pb obtidas pelo gage-B.....	70
Tabela 5 - Comparação das frequências das subsequências das leituras <i>H. hiltneri</i> N3 (MiSeq, fragmentos provenientes do arquivo R1).	77
Tabela 6 - Análise das subsequências exclusivas em cada.....	77
Tabela 7 - Comportamento da bias observado nos outros conjuntos de sequenciamentos	78
Tabela 8 - Preferência dos genes para cada fita em cada uma das metades dos genomas completos de procariotos.....	82
Tabela 9 - Relação da distribuição de Guanina e Citosina nos genes codificantes de procariotos.....	83
Tabela 10 - Comparação da distribuição do conteúdo GC nas duas metades dos genes codificantes de procariotos	84
Tabela 11 - Média do número de <i>contigs</i> por montador e taxa de redução obtidas pelo G-Finisher.	95
Tabela 12 - Montagens obtidas com o sequenciamento da <i>H. hiltneri</i> N3	98
Tabela 13 - Comparação entre os tamanhos das sequências nas montagens das estirpes de <i>A. brasilense</i>	102
Tabela 14 – Síntese das comparações realizadas pelo QUASt da melhor montagem da <i>A. brasilense</i> FP2 com a <i>A. brasilense</i> Sp7.....	104
Tabela 15 - Mapeamento dos genes da <i>A. brasilense</i> Sp7 nos <i>contigs</i> da <i>A. brasilense</i> FP2.....	105
Tabela 16 - montadores estudados no gage-B	125
Tabela 17 - Informações dos Organismos e dos sequenciamentos.....	126
Tabela 18 - Redução do número de <i>contigs</i> nas 96 montagens fornecidas pelo GAGE-B	127
Tabela 19 - Comparação dos resultados pelas medidas do N50, N75, L50 E L75.	139

LISTA DE SÍMBOLOS

pb	pares de bases
kpb	quilo de pares de bases (ordem de grandeza - 10^3)
Mpb	mega de pares de bases (ordem de grandeza - 10^6)
Gpb	giga de pares de bases (ordem de grandeza - 10^9)

LISTA DE SIGLAS

ASCII	- acrônimo para Código Padrão Americano para o Intercâmbio de Informação (<i>American Standard Code for Information Interchange</i> , em inglês).
DNA	- ácido desoxirribonucléico
FASTA	- formato usado para armazenar sequências de bases e de aminoácidos em arquivo texto
FASTQ	- formato utilizado para armazenar sequências e qualidade de bases e de em arquivo texto
FTP	- Protocolo de Transferência de Arquivo (<i>File Transfer Protocol</i> , em inglês)
GAGE	- Avaliação Ouro dos programas de montagem de genomas (do inglês <i>Genome Assembly Gold-Standard Evaluations</i>).
GenBank	- banco de dados público do National Center for Biological Information, do Instituto de Saúde dos Estados Unidos da América.
IO ou I/O	- entrada e saída (<i>Input/Output</i> , em inglês), pode ser empregado para dispositivos ou processos executados pelos programas de computador.
NCBI	- <i>National Center for Biotechnology Information</i>
NFN	- Núcleo de Fixação de Nitrogênio
NIH	- <i>National Institutes of Health</i>
NLM	- <i>National Library of Medicine</i>
OLC	- sobreposição-disposição-consenso (do inglês <i>overlap-layout-consensus</i>)
PCR	- reação em cadeia da polimerase (em inglês <i>Polymerase Chain Reaction</i> - PCR)
PFGE	- eletroforese em gel de campo pulsado.
RAM	- memória de acesso aleatório (<i>Random Access Memory</i> , em inglês)
RNA	- ácido ribonucleico
UFPR	- Universidade Federal do Paraná
WGS	- método de sequenciamento de genoma completo (em inglês, <i>Whole Genome Shotgun</i>)

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Objetivo Geral	25
1.2	Objetivos específicos	26
1.3	Metodologia da pesquisa	26
2	REVISÃO BIBLIOGRÁFICA	27
2.1	O Processo de montagem de genomas	27
2.2	Algoritmos, programas, pipeline e protocolos de montagem de genomas	27
2.2.1	Visão geral dos protocolos tradicionais	28
2.2.2	Um exemplo de <i>pipeline</i> atual	30
2.2.3	Análise das leituras - FastQC	31
2.2.4	Montadores gulosos	32
2.2.5	Montadores de <i>detecção de sobreposição, layout dos fragmentos e decisão da sequência consenso</i>	32
2.2.6	Montadores baseados em grafo De Bruijn	33
2.2.7	O grafo de String	35
2.2.8	O dilema do caminho	36
2.2.9	K-mer	37
2.3	GC Skew	38
2.4	Avaliação e comparação de montagens	38
2.4.1	Dotplot	38
2.4.2	QUAST	39
2.4.3	GAGE e GAGE-B	39
2.4.4	Principais unidades de medidas utilizadas na comparação de montagens	40
2.5	Ferramentas de finalização	41
2.5.1	Opera	41
2.5.2	SSPACE	41
2.5.3	jContigsort	42
2.5.4	Fechamentos de lacunas	42
2.5.5	Correção e detecção de erros de montagens	43
2.6	Qualidade de base	45
2.7	Fuzzy	47
2.8	Organismos	48
2.8.1	Sequenciados pelo Núcleo de Fixação de Nitrogênio	48
2.8.2	Estudados pelo GAGE-B e utilizados na validação do método	50
2.9	Ferramentas utilizadas nos processos de montagens	53
2.9.1	NCBI BLAST	53
2.9.2	MUMmer	53
2.9.3	Gepard	54
2.9.4	Artemis	54
3	MATERIAIS E MÉTODOS	55
3.1	Conjunto de dados	55
3.1.1	Características das leituras obtidas nos sequenciamentos	55
3.2	Análise e Tratamentos das leituras	57
3.2.1	Análise das bases	57

3.2.2	Análise da qualidade de bases	60
3.2.3	Parâmetros considerados para aplicação de filtros	60
3.3	Desenvolvimento das ferramentas de pré-processamento	61
3.3.1	Desenvolvimento da ferramenta de filtragem	61
3.3.2	Desenvolvimento da ferramenta de verificação de emparelhamento	62
3.3.3	Obtenção do conjunto k-mer	62
3.4	Etapa de montagem	63
3.5	Etapa de pós-montagem	64
3.6	Análise com o Dotplot	65
3.7	Avaliação da montagem pelo QUASt	66
3.8	Desenvolvimento do G-finisher	68
3.9	Validação do G-finisher	69
3.10	Ferramenta de avaliação de montagem	71
3.11	Ferramenta de uso geral para auxílio das montagens	71
4	RESULTADOS	75
4.1	Conjunto de dados	75
4.1.1	Análise do viés da frequência de base observado nos relatórios do FastQC	75
4.1.2	Experimento: Estudo das tendências das frequências de bases iniciais nas leituras	75
4.1.3	Estudo da frequência de k-mer	79
4.2	Ferramentas Desenvolvidas para a Etapa de Pré-processamento: Análise e Tratamento dos dados	79
4.2.1	jTrimmer	79
4.2.2	Verificação do pareamento das leituras	80
4.3	Análise do contexto GC nas sequências dos genes e dos genomas completos	80
4.4	Automatizando a variação de parâmetros na etapa de montagem	84
4.5	G-Finisher	85
4.5.1	Aperfeiçoamento do <i>jContigsort</i>	86
4.5.2	Desenvolvimento do <i>jFGAP</i>	87
4.5.3	Fuzzy GC Skew	89
4.5.4	Desenvolvimento da função "Finder"	90
4.5.5	Considerações do uso do Fuzzy GC Skew no processo de remontagem	91
4.6	Validação da estratégia do G-Finisher	93
4.7	Redução dos erros de montagem pela preservação do contexto do GC Skew	96
4.8	Estudos de Casos	97
4.8.1	<i>Herbaspirillum hiltneri</i> N3	97
4.8.2	<i>Azospirillum brasilense</i> FP2	101
4.8.3	<i>Burkholderia contaminans</i> LTEB	106
4.8.4	<i>Herbaspirillum seropedicae</i> Z67	107
5	DISCUSSÃO	109
6	CONCLUSÃO	113
	REFERÊNCIAS	115
7	ANEXO 1 – LISTA DE MONTADORES E ENDEREÇOS ELETRÔNICOS	123

8	ANEXO 2 – TABELAS SUPLEMENTARES DOS ORGANISMOS E MONTADORES UTILIZADOS NO GAGE-B	125
9	ANEXO 3 – RESULTADOS INDIVIDUAIS DAS MONTAGENS DO GAGE-B	127
10	ANEXO 4 – CERTIFICADO DE REGISTRO DO JTRIMMER	151

1 INTRODUÇÃO

O processo para descobrir a ordem dos nucleotídeos que constituem uma molécula de DNA é denominado sequenciamento de DNA. O método predominante de sequenciamento consiste em quebrar aleatoriamente todas as moléculas de DNA (do inglês, *Whole Genome Shotgun-WGS*), produzindo fragmentos com vários tamanhos (POP *et al.*, 2004). O equipamento de sequenciamento lê de 100 a 1.000 bases a partir de uma ou das duas extremidades dos fragmentos e fornece, respectivamente, uma leitura (fragmento) ou o par de leituras (*pair-end/mate-pair*). O processo utilizado na primeira geração de sequenciadores precisa que o DNA seja clivado, amplificado e clonado, produzindo as bibliotecas que são usadas na reação de sequenciamento. O método Sanger foi o primeiro processo aplicado em sequenciamento automático de DNA, o qual fornece leituras (reads) com mais de 800 pb. O processo de sequenciamento de segunda geração não utiliza clonagem; usa PCR em emulsão, cujas leituras são obtidas com comprimento entre 50 e 400 pb. A terceira geração do processo de sequenciamento não utiliza o processo de “amplificação”, de maneira que diversas leituras são produzidas a partir de uma mesma molécula de DNA. Um exemplo dessa geração é o PacBio NGS, que fornece leituras com tamanhos maiores que 15 kpb (LIU *et al.*, 2012; METZKER, 2010).

A reconstrução da sequência original do genoma a partir dessas leituras é denominada montagem. Os diferentes algoritmos desenvolvidos para o processo de montagem buscam encontrar as melhores combinações possíveis de alinhamentos entre as leituras (algoritmos gulosos), ou organizar as leituras em forma de grafo (onde as leituras representam os vértices e as sobreposições as arestas) ou no método de *detecção de sobreposição - layout dos fragmentos - decisão da sequência consenso* (em inglês, *overlap-layout-consensus* ou OLC) (LI *et al.*, 2011; MILLER; KOREN; SUTTON, 2010). Esses algoritmos são modelagens matemáticas e/ou computacionais, em que muito pouco da informação biológica é explorada para orientar o processo.

O *National Center for Biotechnology Information* (NCBI) é uma divisão do National Library of Medicine (NLM) do National Institutes of Health (NIH). O NCBI tem como missão manter e padronizar banco de dados biológicos e fomentar o intercâmbio de informações e experiências entre os pesquisadores em âmbito

mundial. O NCBI mantém e disponibiliza o GenBank, um banco de sequências, genomas e anotações (NCBI, 2016). O NCBI Blast é uma ferramenta computacional para comparação de sequências baseadas em alinhamentos e possui um serviço na internet com mesmo nome para consulta aos dados de genomas que são atualizados diariamente.

Ainda que se observe a evolução dos programas de montagens e das técnicas de sequenciamento, os resultados dos processos de montagem continuam longe de fornecer genomas completos. Segundo (LAND *et al.*, 2015), apenas 13% dos projetos de sequenciamento de genomas são finalizados, os demais projetos depositam as sequências com uma média de 190 *contigs* (fragmentos contíguos de sequências de nucleotídeos). Para (LI *et al.*, 2011; POP, 2009), a tarefa de obter a sequência completa do genoma utilizando as leituras fornecidas pelos sequenciadores e pelos programas de montagem continua sendo um dos maiores desafios da bioinformática.

Genomas incompletos ou muito fragmentados são depositados na forma de rascunho ou “draft” e dificultam a realização de estudos comparativos, a identificação de vias metabólicas e de prospecção biotecnológica. A classificação taxonômica de alguns organismos pode ser prejudicada pela ausência de genes, especialmente nos casos em que a sequência do gene rRNA 16S não é suficiente para identificar a espécie.

O Núcleo de Fixação de Nitrogênio (NFN) enfrenta essas dificuldades nos últimos dez anos. Dos mais de 60 organismos sequenciados, apenas os genomas de *Herbaspirillum seropedicae* SmR1, *Herbaspirillum hiltneri* N3, *Herbaspirillum rubrisubalbicans* M1, *Herbaspirillum seropedicae* Z67 se encontram depositados no NCBI com classificação de genoma completo. Segundo ALNOCH (dados não publicados), durante o estudo de um isolado do gênero *Burkholderia*, para isolamento de uma lipase de interesse, enfrentou-se o problema de classificação, em que inicialmente a bactéria foi identificada como *B. lata*, na sequência como *B. cepacia* e apenas com o sequenciamento completo foi possível classificar o isolado como *Burkholderia contaminans* LTEB e selecionar corretamente a região de interesse.

Os protocolos para montagem *De Novo* da Applied Biosystems (SOLID) e da Illumina (2014) foram estudados e aplicados nos genomas sequenciados no NFN, bem como em dados públicos. Essencialmente, os protocolos são semelhantes e

seguem as mesmas estratégias, que podem ser resumidas em: (1) tratamento das leituras, (2) montagem e (3) finalização. Na primeira tarefa, as leituras são podadas para remover bases de baixa qualidade e adaptadores. Na segunda, as leituras são combinadas para produzir *contigs* ou scaffolds. Na terceira tarefa, são removidas sequências menores que um limite determinado pelo pesquisador e os dados estatísticos da montagem são coletados. O sistema fornecido pela Illumina no sequenciador MiSeq realiza o processo de filtragem e fornece as leituras do sequenciamento previamente tratadas. O sistema realiza, ainda, uma montagem utilizando o montador Velvet (ZERBINO e BIRNEY, 2008) em seu fluxograma. No fluxograma da Applied Biosystems, uma versão modificada do Velvet para trabalhar com as leituras codificadas no formato *color-space* também é utilizada.

A etapa de montagem pode ser realizada por mais de 40 programas diferentes, entretanto, uma parte significativa não tem sido atualizada ou possui licenças comerciais ou não são boas soluções para as tecnologias de sequenciamento atuais. Os principais programas gratuitos atualmente em uso, segundo Magoc et al. (2013), estão relacionados na Tabela 16 (Anexo 2), acompanhados da indicação do algoritmo, última versão e data da última atualização. Esses programas usam a estratégia dos algoritmos baseados em grafos ou de sobreposição de consenso de disposição (OLC).

A terceira etapa, geralmente, é a coleta de dados estatísticos e/ou a aplicação de algum filtro sobre os resultados. Contudo, alguns protocolos começaram a incluir etapas de finalização de montagem, tais como o GapCloser presente no protocolo SOAPDenovo2. O processo de finalização busca identificar erros de montagem e fechamento de lacunas (gaps).

Os dados estatísticos fornecidos pelos montadores são insuficientes para indicar o quanto falta no processo de montagem para obter um genoma completo. Geralmente, as métricas fornecidas são o número de *contigs*, o N50 e o total de bases. Apesar de constarem na maioria das publicações, essas métricas são ineficientes para comparar duas montagens e ainda são influenciáveis por erros de montagens (NARZISI e MISHRA, 2011). Em “Assemblathon 2” (BRADNAM et al., 2013) podem ser encontradas mais de 100 métricas utilizadas para avaliar três projetos de montagens de genomas de eucariotos com tamanhos estimados em mais de 1 Gpb. Por sua vez, no programa QUASt (GUREVICH et al., 2013) são encontradas mais de 60 métricas que foram utilizadas no trabalho “Genome

Assembly Gold-Standard Evaluations” (GAGE) por MAGOC et al. (2013). Em ambos os trabalhos, os autores concluem que os montadores produzem montagens úteis, contendo uma representação significativa dos genes e da estrutura geral do genoma; porém, uma abordagem que funciona bem para uma espécie pode não necessariamente funcionar bem para outra.

No artigo “Complexidade de montagem dos genomas procarióticos usando leitura curta”, os autores Kingsford, Schatz e Pop (2010) concluem que

[...] embora existam um número astronômico de reconstruções consistentes na montagem de genomas com leituras curtas, mesmo nas leituras extremamente curtas (25 pb), são suficientes para reconstrução corretas da maioria dos genes (...) se os resultados práticos não impressionam, estes resultados devem ser **consequências dos dados** ou da **deficiência nos algoritmos**, [...]. (grifo nosso)

Para Nagarajan et al. (2010), a etapa pós-montagem (finalização) serve para ultrapassar as duas maiores limitações do processo de sequenciamento WGS. Primeiro, para reduzir o número de *contigs*, que são uma consequência da existência das regiões de repetição nos genomas e das falhas de leitura de bases causadas pela clonagem e amplificação e; segundo, validar os *contigs* montados que “frequentemente contêm erros, quer devido ao sequenciamento de artefatos ou pela reconstrução incorreta de repetições”.

Entre os programas de finalização, o IMAGE (TSAI; OTTO; BERRIMAN, 2010) alinha as leituras nas bordas dos *contigs* e procura o par complemento das leituras para o devido preenchimento da lacuna. O GapFiller (BOETZER; PIROVANO, 2012) tenta estender as pontas dos *contigs* fazendo uma nova montagem de Bruijn. Para isso, precisa de leituras em pares para formar os scaffolds. O GapCloser (LUO *et al.*, 2012) é um módulo do montador SOAPDenovo2 utilizado para resolver lacunas causadas por repetições maiores que o tamanho da leitura. O FGAP (PIRO *et al.*, 2014) se diferencia dos demais programas por utilizar um conjunto de *contigs* de montagens alternativas (chamada de *dataset* pelo programa) para fechar as lacunas. O FGAP usa o BLAST para alinhar as pontas dos *contigs* da montagem alvo aos *contigs* do *dataset*; quando encontra um contig no *dataset* que sobrepõe à lacuna, esse contig passa a ser candidato de fechamento da lacuna. Após a comparação de todos os candidatos, aquele que apresentar melhor resultado de alinhamento em ambas as pontas é escolhido para fechar a lacuna.

Esses programas funcionam melhor quando os *contigs* estão organizados e formando scaffolds. Essa condição pode ser obtida com o uso do programa OPERA - *Optimal Paired-End Read Assembler* (GAO; SUNG; NAGARAJAN, 2011). Entretanto, os pares de leitura precisam estar a uma distância superior as áreas de repetição, o que raramente ocorre nos sequenciamentos realizados com o protocolo *pair-end* (kit Illumina Nextera XT), em que os fragmentos são de tamanhos inferiores a 1,5 kpb (ILLUMINA, 2014). Ainda, em “Dicas e resolução de problemas” a Illumina diz que os fragmentos produzidos pelo Kit Nextera XT não clusterizam na lâmina de sequenciamento, quando o tamanho dos fragmentos acrescidos dos adaptadores forem maiores que 1,2 a 1,5 kpb (ILLUMINA, 2015).

Ainda na etapa de finalização, os programas que se propõem a fazer a correção de erros dependem da disponibilidade de duas situações: leituras em pares ou genomas de referências. Em todos os casos, informações de tendências (bias) biológicas não são utilizadas para identificar potenciais erros de montagens dos *contigs*. Collyn et al. (2007) associaram uma inversão da sequência cromossômica a variações locais no gráfico do GC Skew e utilizaram essa informação para resolver problemas de ambiguidade na montagem do genoma da bactéria da *Idiomarina loihiensis* L2TR. Esse fato fortaleceu o indício de que variações no gráfico do GC Skew possibilitam a identificação de eventuais falhas de montagens.

Nesse trabalho, desenvolvemos uma nova aplicação computacional para finalização de montagem de sequências de genomas de procariotos que mescla *contigs* de diferentes montagens e utiliza informação *a priori* para identificar erros na montagem dos *contigs*.

1.1 OBJETIVO GERAL

Desenvolver uma metodologia para montagem de genomas de procariotos, sequenciados com a tecnologia que obtém leituras curtas, explorando informações *a priori*.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta tese são:

- analisar sequências curtas obtidas com a tecnologia SOLiD (Applied Biosystems), MiSeq ou HiSeq (Illumina) ou Ion Torrent PGM ou Roche 454;
- realizar pré-montagens;
- criar uma estratégia de organizar os *contigs*;
- identificar erros de montagens;
- identificar padrões biológicos que orientem a montagem;
- construir ferramenta para visualização e analisar as montagens; e
- aplicar a estratégia para montar os genomas das bactérias *Herbaspirillum hiltneri* N3 e *Azospirillum brasilense* FP2.

1.3 METODOLOGIA DA PESQUISA

Neste trabalho, foi aplicado o método da indução experimental, mais especificamente o método de engenharia que consiste em observar o fenômeno, projetar um modelo, simular com os dados conhecidos e avaliar os resultados. O modelo permite identificar as variáveis, compreender e explicar o(s) fenômeno(s) observado(s). A simulação, por sua vez, permite aprimorar o modelo e estudar o comportamento de causa e efeito das variáveis envolvidas. A avaliação permite confirmar a aderência do modelo aos casos observados (especificidade) e projetar a aplicação do modelo para os demais casos (generalização). A precisão do modelo em explicar o fenômeno ainda pode ser medida em relação à especificidade e à sensibilidade.

Esta pesquisa pode ser classificada, de acordo com Silva e Menezes (2005), quanto:

- a) à natureza: aplicada;
- b) à abordagem: qualitativa;
- c) ao objetivo do estudo: explicativa;
- d) ao método científico: indutivo; e
- e) ao procedimento técnico: estudo de caso.

2 REVISÃO BIBLIOGRÁFICA

2.1 O PROCESSO DE MONTAGEM DE GENOMAS

O processo de montagem de fragmentos de DNA é classificado em **De Novo** e **por referência** (POP, 2009; ZHANG et al., 2011; PASZKIEWICZ e STUDHOLME, 2010). Na montagem *De Novo*, os programas utilizam apenas as leituras e a sobreposição para reconstruir os fragmentos de DNA. Na montagem “por referência”, os programas mapeiam as leituras em um genoma conhecido e com alta similaridade de sequência. Posteriormente, calcula a sequência consenso do alinhamento. Na montagem por referência, são perdidas as diferenças na organização estrutural do genoma; nas montagens *De Novo*, frequentemente não se consegue distinguir regiões que apresentem repetições maiores que o tamanho da leitura.

Para Nagarajan e Pop (2013), a escolha dos montadores deve ser feita em função das características das leituras obtidas no sequenciamento. Para dados com maior qualidade de base e leituras curtas (até 150 pb), são frequentemente utilizados montadores baseados no algoritmo do grafo de Bruijn. Para dados de menor qualidade e de fragmentos mais extensos, é recomendada a abordagem do OLC ou *grafo de String*.

2.2 ALGORITMOS, PROGRAMAS, PIPELINE E PROTOCOLOS DE MONTAGEM DE GENOMAS

Neste trabalho, o termo **Protocolo** é empregado para descrever o processo de montagem de genomas como um todo. O processo demanda etapas de análise, interpretações e decisões humanas, bem como de etapas que são realizadas por aplicativos computacionais. Partes do processo podem ser descritas ou realizadas por **pipelines**. Identificamos na literatura dois tipos de pipelines: os automatizados e os manuais ou descritivos. Em todos os casos, os *pipelines* descrevem um fluxo de execução de diferentes programas computacionais para atingir uma determinada meta no processo. Na Ciência da Computação, **programas** são *algoritmos* escritos segundo a regra de uma linguagem de programação/computacional e codificada em um formato apropriado para um determinado sistema operacional ou *hardware*. Os

algoritmos são a descrição de procedimentos formados por um conjunto de instruções bem definidas, executadas por alguém, cujo objetivo é resolver um problema.

2.2.1 Visão geral dos protocolos tradicionais

Os protocolos de montagem *De Novo* fornecidos pelas empresas que fornecem as tecnologias de sequenciamento, geralmente, seguem os mesmos passos: análise das leituras, montagem e finalização.

A etapa da “análise das leituras” implica avaliar a qualidade das bases e o tamanho médio das leituras com uma qualidade mínima, bem como estimar a cobertura do sequenciamento, remover eventuais contaminantes (como adaptadores), além de adaptar os formatos dos arquivos para os montadores. Mantido pelo Babraham Institute, o programa FastQC é uma das principais ferramentas de análise e se encontra disponível em <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.

O processo de montagem tem a finalidade de construir longas sequências de nucleotídeos (*contigs*), a partir da análise das leituras. Vale destacar que diferentes estratégias foram desenvolvidas, com a predominância de três algoritmos: guloso, baseado em grafo e sobreposição de consenso de disposição (do inglês: *overlap-layout-consensus* - OLC). A relação dos principais programas com os respectivos algoritmos está listada na Tabela 16 do Anexo 2.

A etapa de finalização, ainda em consolidação na literatura, realiza a análise da montagem fornecida pelos montadores e tenta identificar os erros de montagem e o fechamento das lacunas (*gaps*).

Há um número enorme de ferramentas de software desenvolvidas para auxiliar na montagem de genomas (TRITT et al, 2012). Compatibilizar os formatos dos dados para as diferentes ferramentas e definir os valores para os parâmetros, a fim de obter os melhores desempenhos dessas ferramentas e bons resultados de montagens nem sempre são tarefas triviais. Esse contexto favorece o surgimento de propostas de automação da integração dessas ferramentas, que, inicialmente, dá-se na forma de “arquivos em lote” (do inglês: *script*) e na sequência na forma de *pipelines*.

Protocolos particulares ou especializados surgem das peculiaridades de cada projeto de montagem que estão diretamente associadas às tecnologias de sequenciamento utilizadas e aos recursos computacionais disponíveis. Pode-se observar que os fornecedores dos equipamentos e de químicas de sequenciamento fornecem um protocolo básico e de uso geral para a aplicação das suas tecnologias. Os protocolos particulares ou especializados tendem a derivar desses modelos básicos.

A Applied Biosystems fornecia no manual da plataforma de sequenciamento SOLiD 3 um protocolo de uso geral. A Figura 1 foi adaptada da “Figura 3 Diagrama esquemático do fluxo de execução do protocolo de montagem *de novo*” (tradução livre do inglês: *Figure 3 Schematic of execution flow in de novo assembly pipeline*) do manual “Applied Biosystems SOLiD 3 Plus System - De Novo Assembly Protocol” página 10.

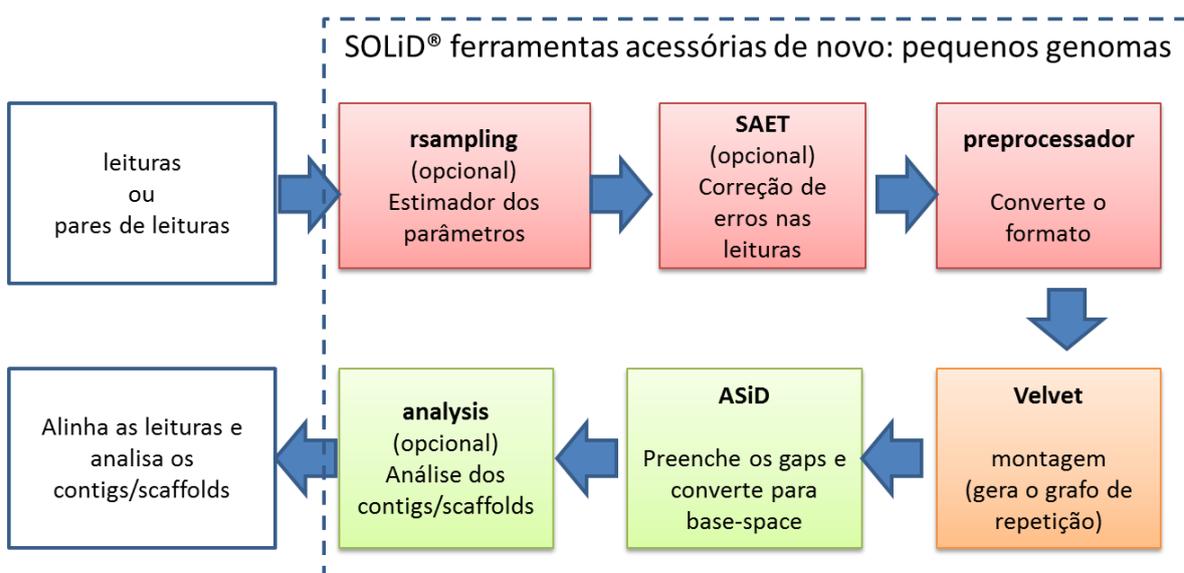


FIGURA 1 - PROTOCOLO DE MONTAGEM *DE NOVO* DA AB SOLiD 3

FONTE: Adaptado do manual “Applied Biosystems SOLiD 3 Plus System - De Novo Assembly Protocol”, página 10 (APPLIED BIOSYSTEMS, 2010).

NOTA: Em vermelho, estão as etapas do pré-processamento; em laranja, o processo de montagem realizado pelo Velvet, e em verde, as etapas do pós-processamento ou finalização de montagem.

Nesta figura, temos em vermelho a etapa de pré-processamento, em laranja o programa de montagem (nesse caso o Velvet) e em verde a etapa da finalização.

Conforme o manual (p. 9) “o pipeline de montagem foi desenhado para simplificar e aperfeiçoar os parâmetros de forma a facilitar o uso e obter o melhor desempenho.”. Algumas das etapas do pipeline eram estratégias para lidar com o volume de dados fornecidos pela plataforma, outros para transformar as leituras que

inicialmente eram codificadas em color-space para um formato compatível com o Velvet. Destaca-se no protocolo a necessidade de tratamento das leituras antes da montagem, a execução do montador e as etapas de finalização para melhorar os resultados.

2.2.2 Um exemplo de *pipeline* atual

O A5 é um exemplo de *pipeline* independente publicado de início em 2012 (ANDREW TRITT et al, 2012) e com uma atualização em 2015 (COIL et al, 2015). Segundo os autores, o A5 é um acrônimo em inglês para “Andrew And Aaron's Awesome Assembly pipeline” e tem a finalidade de

[...] simplificar todo o processo de montagem de genoma, automatizando esses estágios, através da integração de vários algoritmos publicados anteriormente como novos algoritmos para controle de qualidade e seleção de parâmetros de montagem automatizada. (ANDREW TRITT et al., 2012).

A Figura 2 demonstra que o *pipeline* é formado por cinco etapas. A primeira etapa realiza a correção das leituras. A segunda etapa implica o programa de montagem IDBA (acrônimo do inglês, *Iterative de Bruijn Graph De Novo Assembler*, PENG et al., 2010). A terceira etapa se relaciona à execução do SSPACE, um programa que faz a combinação dos *contigs* e constrói *scaffolds*. A quarta etapa é a entrada do algoritmo de análise da montagem, “quebra os *scaffolds*” gerados pelo IDBA e pelo SSPACE. Na quinta etapa, o SSPACE é chamado novamente para produzir o conjunto de *contigs/scaffolds* final.

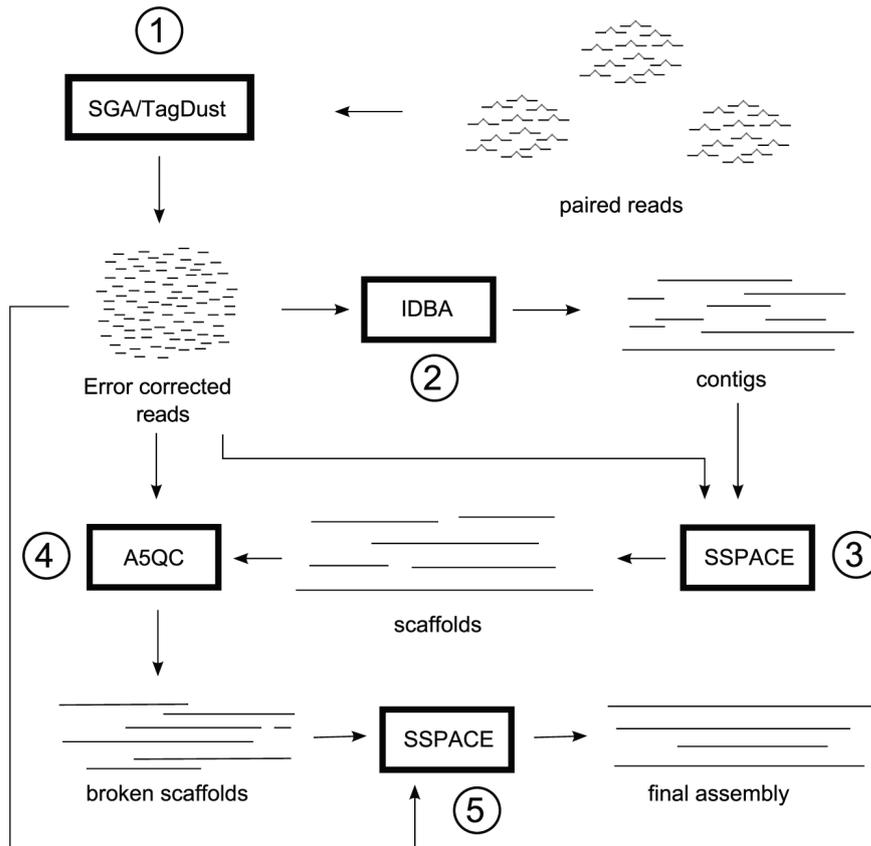


FIGURA 2 – PIPELINE DE MONTAGEM A5

FONTE: Adaptado de ANDREW TRITT et al., 2012, Figura 2.

2.2.3 Análise das leituras - FastQC

O *Babraham Institute* mantém e disponibiliza o programa FastQC. De acordo com o Instituto, o FastQC é “uma ferramenta de controle de qualidade de dados de sequências de alto rendimento” (BABRAHAM INSTITUTE, 2016). O programa analisa o arquivo de leitura no formato FASTQ e fornece onze relatórios. O FastQC foi desenvolvido em Java e fornece os relatórios em formato HTML. O programa foi incorporado no CLC Genomics Workbench como ferramenta de análise de leituras.

Dos relatórios, destacam-se: a) as medidas de qualidade por base em que são apresentadas graficamente as medidas do boxplot, a média e a mediana; b) a frequência de base por coluna; c) o conteúdo GC por coluna e; d) as subsequências sobre representadas, onde o tamanho e a variabilidade das repetições podem ser detectados. O programa pode ser baixado da internet no endereço <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.

Com base nos relatórios do FastQC, o pesquisador pode optar em filtrar ou não as leituras.

2.2.4 Montadores gulosos

Os montadores baseados nesse paradigma escolhem aleatoriamente uma leitura para iniciar o processo de montagem. A leitura escolhida é comparada com todas as demais e o método escolhe a segunda leitura que apresentar o “maior” e “melhor” alinhamento. Essas leituras são combinadas e formam um novo fragmento contínuo (denominado contig), que terá as extremidades novamente comparadas com as demais leituras. A estratégia de “escolha gulosa” caracteriza e nomeia a técnica (SCHATZ et al., 2010; MILLER et al., 2010; PASZKIEWICZ e STUDHOLME, 2010). Essa estratégia foi utilizada por montadores que ficaram famosos, como Phrap e CAP3 e, mais recentemente, pelo VCAKE (JECK et al., 2007); porém, gradativamente, está sendo abandonada em função de não ser suficientemente adequada para trabalhar com pares de leitura e por não resolver o problema das regiões repetidas (NAGARAJAN e POP, 2013).

2.2.5 Montadores de *detecção de sobreposição, layout dos fragmentos e decisão da sequência consenso*

O método de *detecção de sobreposição, layout dos fragmentos e decisão da sequência consenso* (em inglês, *overlap-layout-consensus* ou OLC), também encontrado na literatura brasileira com o nome de “método de grafos de sobreposição”, é o método em que o processo de montagem é dividido em etapas. Na primeira etapa, são calculadas todas as sobreposições; na segunda, são verificados os arranjos que contemplem o maior número de leituras e sobreposições; já na terceira etapa, é calculado o consenso. Vários autores descrevem que cada leitura é vista como um vértice de um grafo, e as sobreposições como arestas (POP, 2009; SCHATZ et al., 2010; MILLER e al., 2010; CONWAY e BROMAGE, 2011). A identificação das áreas de sobreposições é influenciada diretamente pelos parâmetros do tamanho da semente, tamanho mínimo da sobreposição e grau mínimo de semelhança. Para reduzir o número de comparações, os montadores empregam a estratégia de “semente” semelhante à do NCBI Blast, que busca por

regiões idênticas de tamanho mínimo e, depois, tentam ampliar o alinhamento a partir delas. A construção da “disposição” (em inglês, *layouts*) se assemelha ao problema do caminho hamiltoniano em grafo (PASZKIEWICZ e STUDHOLME, 2010), em que a busca é realizada para encontrar o caminho que visite cada vértice do grafo uma única vez. A terceira etapa, denominada consensos, é semelhante ao problema do alinhamento múltiplo (do inglês, *multiple sequence alignment – MSA*), em que os alinhamentos são organizados e a frequência das bases é calculada para determinar a mais recorrente. São dois os problemas essenciais nessa técnica: o primeiro é que a ordem dos alinhamentos mudam os resultados e o segundo, que não existe solução ótima viável (PASZKIEWICZ e STUDHOLME 2010). Os montadores baseados em OLC têm apresentado melhores resultados em leituras de tamanho médio, maiores que 400 pb e com baixas coberturas.

2.2.6 Montadores baseados em grafo De Bruijn

Em 1995, Idury e Waterman (1995, citado por ZERBINO, 2009) introduziram o conceito de representar a montagem de sequência na forma de grafos e um algoritmo de montagem inspirado no método de hibridização de sequências. Nascia o conceito de k-mer, em que palavras com tamanho fixo de k nucleotídeos eram obtidas das leituras. As palavras eram representadas como nós (ou vértices) no grafo e as sobreposições entre as palavras eram representadas na forma de arestas. Eles foram capazes de produzir cadeias de sobreposições de k-mers que formavam sequências contíguas (*contigs*), por causa da ausência de ramificações (adaptado de ZERBINO, 2009, tese, p. 19).

Pevzner et al. (2001) ampliaram esse conceito e definiram o grafo de Bruijn:

Dado um conjunto de leituras $S=\{s_1,\dots,s_n\}$, define-se o grafo de Bruijn $G(S_l)$ para o conjunto de vértices S_{l-1} (para todas as $(l-1)$ tuplas de S). Uma $(l-1)$ -tupla $v \in S_{l-1}$ juntamente com uma aresta dirigida com origem em $(l-1)$ -tupla $w \in S_{l-1}$, se S_l contém uma l -tupla para os primeiros $l-1$ nucleotídeos de v coincidem com os últimos $l-1$ nucleotídeos de w . Cada l -tupla de S_l corresponde a uma aresta em G . Se S contém uma única sequência S_1 ; então esta sequência corresponde a um caminho que visita cada aresta do grafo de Bruijn – o caminho do carteiro chinês. (PEVZNER et al., 2001)

Compeau et al. (2011) representaram graficamente os conceitos acima, que adaptamos e reproduzimos na Figura 3. Na Figura 3A é representado um conjunto

com seis leituras alinhadas e uma sequência do “genoma” reconstruído a partir dessas leituras. Na Figura 3B, o grafo de Bruijn para $k=3$ é demonstrado tendo os k -mers representados pelos arcos e os vértices pelos círculos. Os números indicam os passos que o algoritmo pode seguir para reconstruir a sequência. Na Figura 3C, são apresentados os passos que podem ser seguidos pelo algoritmo de travessia do grafo com o alinhamento mostrando a sobreposição de duas bases ($k-1$) e a sequência final. A parte da sequência em cinza demonstra a subsequência que identifica o início e o término do círculo. Nos casos de genomas lineares, ocorrerá um vértice que terá apenas uma aresta saindo e um vértice que terá apenas uma aresta chegando.

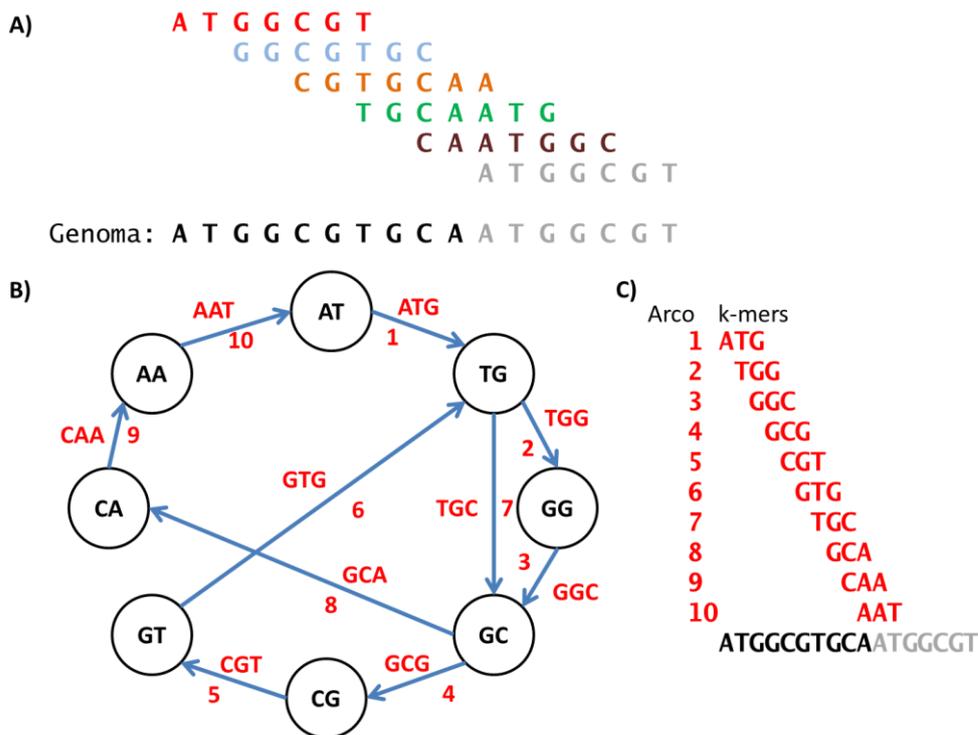


FIGURA 3 - EXEMPLO DO GRAFO de Bruijn

FONTE: Adaptado de COMPEAU et al. (2011)

O grafo de Bruijn é uma estratégia que reduz o consumo de memória porque os vértices não precisam ser armazenados e os arcos são facilmente mantidos em uma tabela indexador por uma função de hash. A reconstrução da sequência do genoma é obtida pelo algoritmo que faz o percurso no grafo. Nesse sentido, Pevzner et al. (2001) descrevem que “o problema do carteiro chinês está intimamente relacionado com o problema de encontrar um caminho de visitar cada aresta de um

grafo exatamente uma vez.(...) Pode-se transformar o problema do carteiro chinês no problema do caminho Euleriano ao introduzir o valor das multiplicidades de arestas no grafo de Bruijn”. Em teoria dos grafos (Ciência da Computação), o caminho Euleriano é o caminho que percorre o grafo passando uma única vez em cada aresta.

De acordo com Zerbino (2009, p. 20), autores do mesmo grupo publicaram uma sequência de artigos descrevendo algoritmos para construir e corrigir os erros no grafo de Bruijn, para uso de pares de leituras e leituras curtas.

Zerbino (2009, p. 22) afirma que

[...] embora tenha sido inspirado pelas ideias de Pevzner et al. (2001), a estrutura do grafo implementada no Velvet se diferencia por manter os nós ao invés dos arcos e que cada nó tem associado a sequência complementar reversa que fornece um grafo bidirecional implícito, permitindo estender as sequências em ambos os sentidos.

Para Nagarajan e Pop (2013), essa abordagem deve provavelmente perder a importância na medida em que fragmentos fiquem mais longos e com melhor acurácia. Segundo os autores, a estratégia não representa diretamente as leituras; sobreposições são exatas, exigindo correção dos erros antes e durante a montagem, para que se obtenha uma sequência consenso de qualidade.

2.2.7 O grafo de String

O método do grafo de String simplifica o grafo de sobreposição-disposição-consenso pela eliminação de redundância. Segundo Paszkiewicz e Sstudholme (2010), a execução é feita em quatro passos:

- a) geração do grafo de sobreposição através da comparação de todas as leituras (vértices);
- b) conversão para o grafo de String, através da fusão e redução de sobreposições e arestas redundantes;
- c) redução de arestas e vértices falsos através do algoritmo de fluxo de rede;
- d) busca do caminho ou circuito Euleriano.

Essa abordagem foi proposta por Myers (1995) e utilizada no montador SGA (Simpson e Durbin, 2012). Essa abordagem é recomendada para montagens de fragmentos mais longos e com dados de menor qualidade.

2.2.8 O dilema do caminho

Ao proporem a abordagem do caminho Euleriano para o problema de montagem de genomas, Pevzner et al. (2001) descrevem o problema do Caminho Euleriano, o qual consiste em encontrar um caminho em que cada aresta do grafo seja visitada uma única vez. Em outro ponto do mesmo artigo, os autores definem que “Uma repetição é chamada de emaranhado se não houver caminhos de leituras que contenham esta repetição”; ou seja, denomina-se emaranhado quando não existirem leituras com tamanhos maiores que o da região repetida e que sobreponham ou atravessem a sequência repetida.

Os autores afirmam que os “emaranhados” criam um problema de montagem que não pode ser resolvido pela análise das leituras e propõem o problema do Supercaminho Euleriano. O Supercaminho Euleriano é definido pelos autores como: “dado um grafo Euleriano e uma coleção de caminhos neste grafo, encontrar um caminho Euleriano neste grafo que contenha todos esses caminhos como subcaminhos”. No processo de simplificar o grafo e de tratar o efeito dos erros de leituras, regiões repetidas tendem a serem representadas por um único caminho, conectadas a vários outros caminhos de entrada e a vários outros caminhos de saída. Para resolver essa situação, os autores defendem uma estratégia de “descolamento” dos caminhos. Os novos caminhos são consistentes se forem compatíveis com todos os subcaminhos que transpassem a região em questão. As combinações que tornam um caminho consistente, ou não, são representadas pela Figura 4, que reproduz a Figura 5 original.

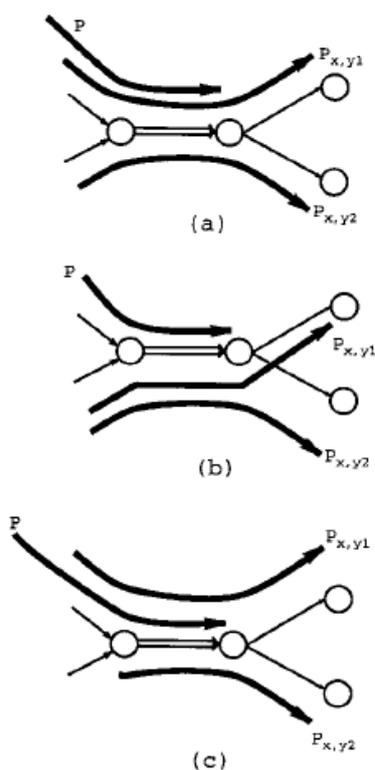


FIGURA 4 - (a) P é consistente para $P_{x,y1}$, mas é inconsistente para $P_{x,y2}$; (b) P é inconsistente para ambos os $P_{x,y1}$ e $P_{x,y2}$ e; (c) P é consistente para os dois $P_{x,y1}$ e $P_{x,y2}$

FONTE: Figura 5 de PEVZNER et al. (2001)

NOTA: Os círculos representam os nós no grafo de Bruijn, as linhas finas as arestas no grafo. As linhas mais grossas são os caminhos obtidos a partir das leituras.

Os autores concluem que os descolamentos e cortes provaram ser poderosas técnicas para “desembaraçar” o grafo de Bruijn e reduzir o problema da montagem de genomas. No entanto, “ainda existe uma lacuna na análise teórica para o problema do supercaminho Euleriano, **no caso em que o sistema de caminhos não possibilite qualquer descolamento ou cortes**”. (negrito nosso).

2.2.9 K-mer

Segundo Zerbino (2009), em 1995, Idury e Waterman (IDURY e WATERMAN, 1995) introduziram os grafos para representar a montagem de genomas e demonstraram um algoritmo com palavras formadas por k nucleotídeos, também

conhecidos por k-mers, que eram representadas por nós (ou vértices) e os arcos (ou arestas) formados pelas sobreposições.

O termo foi popularizado pelo Velvet e adotado como sinônimo para subsequências de tamanho k em vários outros programas e estudos.

2.3 GC SKEW

Lobry (1996) identificou a segmentação genômica das frequências de base e a frequência equivalente entre adenina e timina ou entre citosina e guanina. O bias pode ser medido/representado pela relação $[(G-C)/(G+C)]$. Posteriormente, a origem e o término da região de replicação foram associados aos pontos de máximos e mínimos, respectivamente, da curva formada pelos valores acumulados da relação GC. Denomina-se GC Skew Acumulada a representação dessa curva (GRIGORIEV, 1998; FRANK e LOBRY, 2000; ROTEN et al., 2002). A análise do GC Skew Acumulada de mais de 400 sequências de cromossomos bacterianos revelaram que rearranjos são raros (ROTEN et al., 2002). COLLYN et al (2007) descrevem inversão cromossômica na montagem de *Idiomarina loihiensis* L2TR, resultando em inversão da tendência das curvas de inclinação cumulativas e localmente perturbar a forma de V invertido simétrico.

2.4 AVALIAÇÃO E COMPARAÇÃO DE MONTAGENS

2.4.1 Dotplot

Para Jan Krumsiek et al. (2007), a análise matricial é um método popular para criar comparações de duas sequências de nucleotídeos ou de proteínas. Os Dotplots são representações gráficas dos resultados da análise matricial e podem ser usados para interpretar e analisar as relações evolutivas das sequências, dos domínios conservados e das repetições. O programa 'Dotter' foi a primeira ferramenta interativa com interface gráfica para construção de Dotplots (SONNHAMMER e DURBIN, 1995).

De acordo com os autores, o principal problema do cálculo dotplot é a complexidade computacional $\Theta(m.n)$, onde m e n são os tamanhos respectivos das

duas sequências. Para comparar sequências do tamanho dos genomas, o tempo seria inaceitável. Uma solução para esse problema é o uso de um índice de palavras para a rápida identificação de combinações de subsequências. Vários programas empregam essas estratégias (HUANG e ZHANG, 2004; SZAFRANSKI et al., 2006). Uma estrutura de sufixos para pesquisar palavras e gerar dotplots já foi utilizada pelo MUMmer (KURTZ et al., 2004).

Um dos programas mais recentes para produção de dotplots é o Gepard (KRUMSIEK et al., 2007).

2.4.2 QUAST

O QUAST foi desenvolvido entre as duas publicações do GAGE e foi usado no GAGE-B. Para os autores, as “limitações das técnicas de sequenciamento levaram ao desenvolvimento de dezenas de algoritmos de montagens, nenhum dos quais é perfeito.” (GUREVICH et al., 2013), o que justifica a necessidade de comparação de diferentes montagens. O QUAST fornece um conjunto de relatórios e indicadores e pode ser usado na presença ou ausência de um genoma de referência. O QUAST incorporou um conjunto de métricas existentes em programas como Plantagora, GAGE, GeneMark, Glimmer, entre outros, e criou algumas métricas novas. Além disso, o QUAST utiliza o Nucmer do pacote MUMmer para fazer o alinhamento das sequências. O QUAST agrupou as métricas em: a) tamanho dos *contigs*; b) erros de montagens e variações estruturais; c) representação do genoma e os elementos funcionais e d) variações do N50.

O número de genes e operons é contado pelo QUAST quando informado um genoma de referência ou o Glimmer é utilizado para prever e poder indicar em seu relatório o número de genes preditos.

2.4.3 GAGE e GAGE-B

Conforme o site do projeto¹, GAGE é uma avaliação dos mais recentes algoritmos de montagem de genoma de larga escala. Eles organizam esse modelo

¹ Site do projeto GAGE do *Center For Computational Biology* da Universidade John Hopkins: http://ccb.jhu.edu/gage_b/

de “competição” para produzir uma avaliação realista dos softwares de montagem de genomas e o comportamento dos montadores frente à rápida evolução dos sequenciadores de alto desempenho. (adaptado de <http://gage.cbcb.umd.edu/>) Sua primeira “edição” teve os critérios e resultados publicados na revista *Genome Research*, sob o título *GAGE: Uma avaliação crítica dos conjuntos do genoma e algoritmos de montagem* (SALZBERG et al., 2012).

A segunda “edição” foi publicada sob o título “GAGE-B: uma avaliação dos montadores do genoma de organismos bacterianos” (MAGOC et al., 2013) e teve os critérios de comparação ampliados, assim como a incorporação do QUILT como avaliador dos resultados de montagem e disponibilizaram todos os dados utilizados nas montagens e nas comparações.

Em “Genome Assembly Gold-Standard Evaluations” (GAGE-B) foram produzidas 96 montagens com o uso de oito montadores (Tabela 16, Anexo 2) gratuitos e doze sequenciamentos de oito organismos (Tabela 17, Anexo 2). A relação dos montadores e endereços eletrônicos está disponível no Anexo 1.

2.4.4 Principais unidades de medidas utilizadas na comparação de montagens

Existem centenas de unidades de medidas utilizadas por diferentes trabalhos para comparar as montagens. Relacionamos abaixo as definições das medidas mais amplamente utilizadas e que são citadas nesse trabalho. As definições foram obtidas no descritivo (ou ajuda), disponibilizadas pelo programa QUILT (GUREVICH et al., 2013). São elas:

- a) **número de *contigs*** (sequências contíguas): o número de *contigs* obtidos pelo montador. Quanto menor o número de *contigs* melhor tende ser a montagem;
- b) **tamanho do contig mais extenso**: o tamanho do maior contig obtido na montagem;
- c) **tamanho total**: o somatório do número de bases representadas nos *contigs/scaffolds*;
- d) **N50**: identifica o tamanho do contig que a soma dos comprimentos de todos os *contigs* de comprimentos maiores ou iguais a ele representem 50% das bases da montagem. Em geral, não existe um valor que produza exatamente 50%, de modo que a definição técnica é o comprimento máximo x , tal que o

uso dos *contigs* de comprimento igual ou superior a x representa pelo menos 50% do comprimento total de bases da montagem;

- e) **N75**: similar ao N50, o N75 identifica o comprimento do *contig* x em que a soma de todos os comprimentos dos *contigs* maiores ou iguais a x representem 75% das bases da montagem;
- f) **L50**: o número mínimo de *contigs* necessários para representar 50% das bases da montagem;
- g) **L75**: o número mínimo de *contigs* necessários para representar 75% das bases da montagem.

2.5 FERRAMENTAS DE FINALIZAÇÃO

2.5.1 Opera

Para Gao et al. (2011), o problema de ordenar e orientar os *contigs* em pares de leituras é um passo crucial para os projetos de montagem de genomas de alta qualidade. O Opera é apresentado pelos autores como a primeira solução para esse problema. O Opera faz o alinhamento dos pares de leitura nos *contigs* e cria o grafo de scaffolds e descreve um procedimento para redução do grafo. O Opera inclui uma proposta para estimar o tamanho das lacunas (gaps) entre os *contigs*.

2.5.2 SSPACE

Boetzer et al. (2010) consideram o SSPACE uma ferramenta para construção de scaffolds a partir de pares de leituras e *contigs* montados por diferentes montadores. Esses autores afirmam que, com base nas leituras realizadas, o SSPACE é capaz de avaliar a ordem, a distância e a orientação dos *contigs*. Com a adoção de bibliotecas do tipo mate-pair, o SSPACE é capaz de reduzir em até 75% o número de *contigs*/scaffolds.

2.5.3 jContigsort

O jContigsort (GUIZELINI, 2011) é uma aplicação escrita em Java para determinar a sequência (ou ordem) dos *contigs* e o sentido com base no mapeamento dos *contigs* em um genoma de referência.

No jContigsort são criados “índices remissivos” de subsequências (kmer) do genoma de referência. As subsequências e respectivas posições formam a tupla (chave-valor), armazenada em uma coleção do tipo HashMap. Para cada contig a ser alinhado, são obtidas as subsequências e procuradas na coleção HashMap, identificando assim as posições correspondentes da subsequência no genoma de referência. Os valores das posições são ordenados e quando a distância entre os valores extremos forem maiores que o tamanho do contig, o algoritmo de agrupamento (clusterização) é invocado. O algoritmo de agrupamento é baseado na técnica de k-means, restrito a dez iterações. A restrição existente é que cada grupo não pode apresentar uma diferença entre os valores extremos intergrupos maiores que o tamanho do contig. No final, o grupo com maior número de posições é escolhido e o valor médio do grupo é associado ao contig. No final, os *contigs* são organizados com base nas posições associadas.

2.5.4 Fechamentos de lacunas

2.5.4.1 IMAGE

O IMAGE (TSAI et al., 2010) utiliza sequências de leituras Illumina para melhorar os projetos de montagens de genomas. O programa alinha as leituras contra as extremidades dos *contigs* e realiza montagens locais para estender e preencher a lacuna.

2.5.4.2 GapFiller

O GapFiller é uma estratégia de preenchimento de lacunas presentes em scaffolds, usando pares de leituras (BOETZER; PIROVANO et al., 2012). O algoritmo do GapFiller alinha as leituras nas bordas das lacunas, remove as regiões de baixa qualidade, seleciona os pares de leituras que ancoram nas bordas e preenche a lacuna com a montagem baseada na estratégia do *grafo de kmer*.

2.5.4.3 GapCloser

O GapCloser é um módulo integrado ao SOAPDenovo que foi desenvolvido para substituir e identificar erros utilizando os pares de leituras. De acordo com Luo et al. (2012), o método foi aperfeiçoado para identificar regiões de repetições com poucas divergências de similaridade, melhorando a resolução de bases conflitantes e a precisão no fechamento das lacunas.

2.5.4.4 FGap

O FGap é uma ferramenta de fechamento de lacunas (gaps) que alinha as regiões adjacentes às lacunas, utilizando o NCBI BLAST em um banco formado por *contigs* provenientes de montagens alternativas. O FGap identifica regiões nos *contigs* de montagens alternativas que sobrepõe a região do gap e preenche o gap com a sequência. O algoritmo escolhe automaticamente a sequência com melhor alinhamento nas extremidades adjacentes a lacuna (PIRO et al., 2014).

2.5.5 Correção e detecção de erros de montagens

Salzberg (2012, GAGE) afirma que um dos passos mais importantes em qualquer montagem é o processo de limpeza de dados. Também entende que a alta qualidade dos dados produz diferenças dramáticas nos resultados: por exemplo, uma montagem do genoma da *Rhodobacter sphaeroides* apresentou o valor de N50 de apenas 233 pb. Após a correção de erros, o mesmo montador alcançou um valor de N50 de 7793 pb, ou seja, 30 vezes melhor.

Os montadores baseados em grafos e/ou OLC tendem a gerar *contigs* com três padrões de erros em consequência das limitações das técnicas de sequenciamento (WGS). Daniel Zerbino (2009) classifica os erros em (Figura 5): a) desalinho (tradução livre para o termo *tip*, do inglês); b) bolha e c) construção quimérica. A Figura 5, criada por Zerbino (2009) em sua tese, demonstra os três erros que ocorrem no grafo de Bruijn criado pelo Velvet.

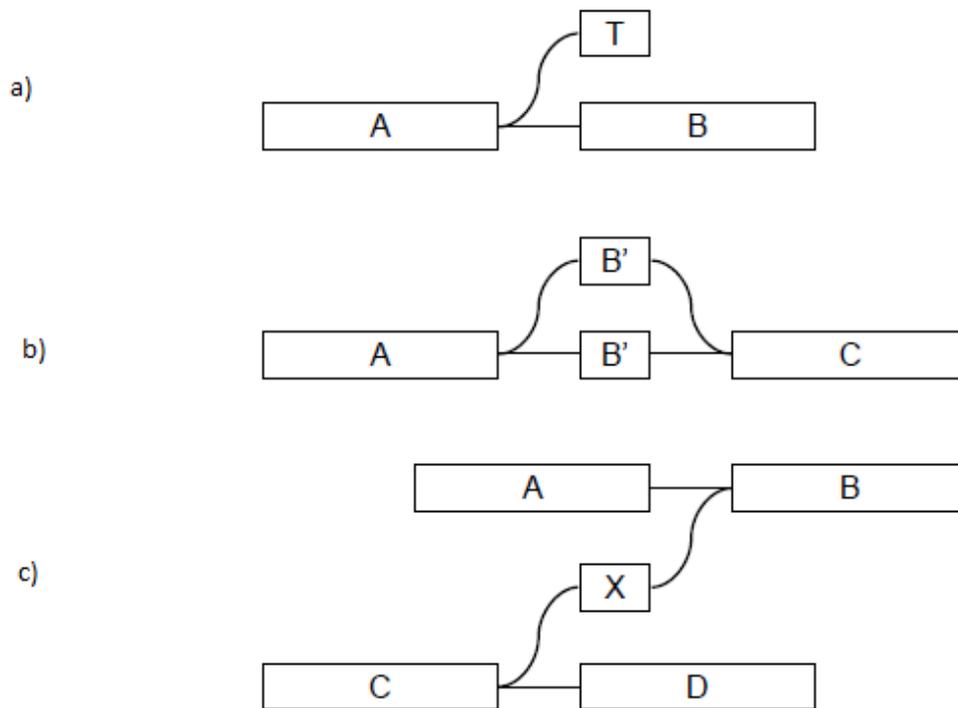


FIGURA 5 - DIAGRAMA ESQUEMÁTICO DAS TRÊS CATEGORIAS DE ERROS NO VELVET

FONTE: adaptado de ZERBINO, 2009 (Figura 3.1, p. 48)

NOTA: a) Um caminho alternativo é criado no grafo, porém esse caminho não pode ser ampliado (desalinho). b) Existem duas possibilidades de caminhos para conectar os segmentos A e C, as regiões B divergem em sua composição de bases (bolha). c) O nó X indica uma conexão espúria do nó C com o nó B e causando concorrência com os caminhos CD e AB, impedindo a correta montagem dos segmentos AB e CD (quimera).

Miller et al. (2010) denominam o erro “desalinho” como “espora” e apresenta uma versão alternativa (Figura 6) para a representar o erro representado em “c” na Figura 5. Para o autor, a quimera é mais frequente em sequências com repetições curtas, que produzem construções cíclicas.

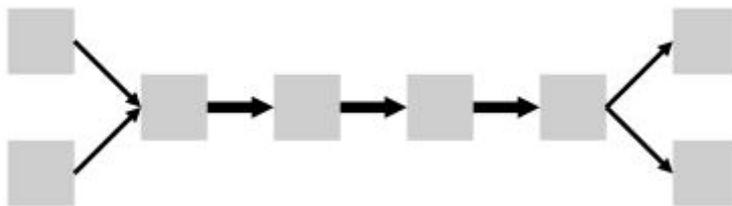


FIGURA 6 - CICLOS SÃO CAMINHOS QUE CONVERGEM EM SI MESMOS, INDUZIDOS POR REPETIÇÕES CURTAS NA SEQUÊNCIA

FONTE: adaptado de MILLER et al. (2010)

Para Miller et al. (2010), o “efeito bolha” são caminhos que divergem e em seguida convergem. As bolhas são induzidas por erros de sequenciamento, especialmente no meio da leitura e por polimorfismos decorrentes dos processos de amplificação (PCR). Ainda segundo os autores, a detecção da bolha não é um processo trivial.

Os erros de montagem existem tanto em projetos rascunhos (draft) quanto em genomas completos (PHILLIPPY et al., 2008). Os autores relatam falhas identificadas no processo de revisão do genoma humano que incluem a omissão de algumas regiões, a reorganização de trechos ou regiões deformadas. Os autores destacam que os erros tendem a ser de duas categorias: a) regiões repetidas que são fundidas ou expandidas e; b) rearranjo de sequência ou inversão. Para eles, divergência de bases com uma cobertura mínima é “assinatura” típica de erro de montagem. Métodos de validação baseados em *mate-pair* são eficientes para identificar os quatro tipos de erros citados. Leituras quebradas são pistas a serem observadas, de maneira que segundo os autores “um mapeamento que alinha a primeira metade de uma leitura em uma região e a segunda metade em outra com 100% de identidade é preferível que um mapeamento de leitura completa com identidade de 80%”.

Para identificar os erros, Phillipy et al. (2008) afirmam que a análise forense da montagem deve recorrer à análise de leituras quebradas ou não alinháveis à montagem e à validação da montagem por alinhamento dos pares de leituras (*mate-pair*). Ainda, entendem que as duas classes de erros mais comuns são a combinação de regiões repetidas em uma única região e/ou expansão da sequência.

O grupo do GAGE fez correções nas leituras por meio do QUAKE, um programa que leva em conta as estatísticas de frequência de kmers para remover das leituras os elementos de baixa frequência. Essa estratégia melhorou a montagem de alguns montadores. No GAGE-B, foram empregados filtros para remover leituras com qualidade inferior a Q10 (MAGOC, 2013), sob o uso do aplicativo *ea-utils* (ARONESTY, 2011).

2.6 QUALIDADE DE BASE

Para cada nucleotídeo representado nas leituras, existe um valor de qualidade associado. A escala padrão foi definida originalmente pelo programa

PHRED e representava a probabilidade estimada de erro na leitura (COCK et al., 2009):

$$Q_{PHRED} = -10 \times \log_{10}(P_e)$$

Ao criar o formato FASTQ, Jim Mullikin, do Instituto Wellcome Trust Sanger, converteu a notação de qualidade para ser representada por um símbolo ASCII na faixa de 33 a 126. Em 2004, a Solexa introduziu uma versão do FASTQ com uma codificação de valores na faixa de 59 a 126 e uma nova fórmula:

$$Q_{SOLEXA} = -10 \times \log_{10}\left(\frac{P_e}{1 - P_e}\right)$$

Apesar dos diferentes sistemas de sequenciamento e da forma de estimar o erro, é possível fazer um mapeamento entre os dois sistemas por meio das fórmulas abaixo. Essa conversão foi difundida pelo programa MAQ. (Adaptado de COCK et al., 2009).

$$Q_{PHRED} = 10 \times \log_{10}\left(10^{\frac{Q_{SOLEXA}}{10}} + 1\right)$$

$$Q_{SOLEXA} = 10 \times \log_{10}\left(10^{\frac{Q_{PHRED}}{10}} - 1\right)$$

Ao arredondar os resultados dessas equações, os valores de qualidade são praticamente iguais. Quanto à interpretação da probabilidade da leitura da base estar errada, esta se encontra de forma simplificada na Tabela 1.

TABELA 1 - EXEMPLO DA ESCALA PHRED DE QUALIDADE E A RELAÇÃO DOS VALORES COM A PRECISÃO E PROBABILIDADE DE ERROS NA LEITURA DE BASE.

Valor PHRED	Probabilidade da identificação da base estar errada	Precisão da identificação da base
10	1 em 10	90%
20	1 em 100	99%
30	1 em 1000	99,9%
40	1 em 10.000	99,99%
50	1 em 100.000	99,999%
60	1 em 1.000.000	99,9999%

FONTE: O autor

Atualmente, os sistemas de codificação de qualidade, as faixas de símbolos existentes e a escala que corresponde à PHRED podem ser consultados a seguir:

TABELA 2 - REPRESENTAÇÃO SIMBÓLICA DOS VALORES DE QUALIDADE NO FORMATO FASTQ

Descrição	Símbolos ASCII				Qualidade
	Faixa	Deslocamento	Iniciais	Finais	Faixa
SANGER	33 a 73	33	! " # \$	F G H I	0 a 40
Solexa	59 a 104	64	; < = >	e f g h	-5 a 40
Illumina 1.3+	64 a 104	64	@ A B C	e f g h	0 a 40
Illumina 1.5+		64	C D E F	e f g h	3 a 40 *
Illumina 1.8+		33	! " # \$	G H I J	0 a 41

FONTE: O autor

NOTA: No sistema Illumina 1.5+ os valores 0 e 1 são reservados e o valor 2 indicador de controle de qualidade e não podem ser mapeados diretamente para valores PHRED.

2.7 FUZZY

Zadeh (1965) criou a *lógica fuzzy* e, com isso, ampliou a teoria dos conjuntos clássicos para conjuntos nebulosos (ou vagos). Anteriormente, um elemento qualquer podia ser comparado a um conjunto e classificado como pertencente ou

não. Com fuzzy foi possível atribuir graus de pertinência, bem como obter a resposta do problema de classificação.

Em seu artigo, Zadeh exemplifica o conceito ao classificar uma pessoa jovem em relação aos conjuntos criança e adulto. Uma pessoa que tenha de 12 a 18 anos, segundo a teoria clássica de conjuntos, não pertenceria ao conjunto dos adultos e a pertinência ao conjunto das crianças poderia ser questionada ou no mínimo desconfortável para o classificador. A lógica fuzzy permite, por exemplo, afirmarmos que aos 12 anos uma pessoa pertence em 90% ao conjunto das crianças e 10% ao conjunto dos adultos e aos 17 anos será o inverso.

2.8 ORGANISMOS

2.8.1 Sequenciados pelo Núcleo de Fixação de Nitrogênio

2.8.1.1 *Herbaspirillum hiltneri* N3 (DSM 17495)

Herbaspirillum hiltneri N3 (DSM 17495) é um Betaproteobacteria isolado a partir da superfície esterilizada de raízes de trigo, descrito inicialmente por Rothballer et al. (2006). Rothballer classificou *H. hiltneri* N3 como cepa tipo. Embora o gênero *Herbaspirillum* contenha espécies diazotróficas, tais como *H. seropedicae*, *H. rubrisubalbicans*, e *H. frisingensis*, com o potencial de colonização endófito e sistêmica de uma variedade de plantas, *H. hiltneri* estirpe N3 carece de todos os genes do cluster NIF, confirmando que este organismo é incapaz de fixar nitrogênio, corroborando com o estudo anteriormente descrito (ROTHBALLER et al. 2006). No entanto, os genes envolvidos na regulação global do metabolismo do nitrogênio (*glnA*, *glnB*, *glnE*, *glnK*, *ntrB*, *ntrC*, *ntrY* e *ntrX*) estão presentes (GUIZELINI et al., 2015).

A sequência completa do genoma de *Herbaspirillum hiltneri* N3 tem um cromossomo circular de 4.965.474 pb de comprimento com um percentual de GC de 61,84%. A anotação do genoma usando RAST e nossa plataforma (in-house) SILA previu 4.581 sequências codificadoras (CDS), o tRNAScan previu 49 tRNAs, e NCBI Blast identificou 4 cópias do operons ribossomal 5S-16S-23S.

2.8.1.2 *Azospirillum brasilense* FP2

Azospirillum brasilense é uma alfa-proteobactéria gram-negativa, diazotrófica, microaerófila, capaz de viver livremente no solo ou em associação com raízes de diversas gramíneas de importância econômica, onde promove crescimento vegetal e tem sido indicado como um inoculante para o trigo e milho (HAUER, 2012). *A. brasilense* FP2 é um mutante espontâneo de *Azospirillum brasilense* Sp7 (ATCC 29145) resistente à estreptomicina e ácido nalidíxico.

Bactérias da espécie *Azospirillum brasilense* têm um genoma de multicomponentes que sofre frequentes rearranjos espontâneos, obtendo-se alterações nos perfis dos plasmídeos de estirpes. Especificamente, as variantes (CD, Sp7.K2, Sp7.1, Sp7.4, Sp7.8, etc.) da estirpe *A. brasilense* Sp7 que tinha perdido um plasmídeo 115-MDA (KATSY EI e PETROVA LP, 2015).

A caracterização da estrutura genômica e do gênero foi produzida por Martin-Didonet et al., (2000), quem descreveu o genoma da espécie composto por um cromossomo e seis plasmídeos e o tamanho estimado variando de 6,7 a 6,9 Mb.

2.8.1.3 *Herbaspirillum seropedicae* SmR1

Herbaspirillum seropedicae é uma bactéria endofítica capaz de colonizar espaços intercelulares de gramíneas, como arroz e cana-de-açúcar. O genoma do *H. seropedicae* estirpe SmR1 foi sequenciado e anotado. O genoma é composto de um cromossomo circular de 5.513.887 pb e contém um total de 4,804 genes. A sequência do genoma revelou que o *H. seropedicae* é um microrganismo altamente versátil, com a capacidade de metabolizar uma ampla gama de fontes de carbono e nitrogênio, com a posse de quatro oxidases terminais distintas. Além disso, o genoma contém uma infinidade de sistemas de secreção de proteínas, incluindo sistemas de secreção de tipo I, tipo II, tipo III, tipo V, tipo VI e tipo IV pili, sugerindo um elevado potencial para interagir com plantas hospedeiras. *H. seropedicae* é capaz de sintetizar o ácido indolacético, um gene que codifica para ACC-deaminase. Os genes para hemaglutininas/hemolisinas/adesinas foram encontrados e podem desempenhar um papel na adesão da superfície celular da planta. (PEDROSA et al., 2011)

2.8.1.4 *Herbaspirillum seropedicae* Z67

A estirpe *Herbaspirillum seropedicae* Z67 (ATCC 35892) foi sequenciada no NFN.

2.8.1.5 *Burkholderia contaminans* LTEB

Burkholderia contaminans LTEB é uma Betaproteobacteria, Gram-negativa, isolada da contaminação em meio de cultura suplementada com óleo vegetal. *B. contaminans* LTEB produz a lipase LipBC de interesse comercial que pode ser usada em meios orgânicos para obtenção de produtos de interesse biotecnológicos. A fim de compreender os mecanismos e vias metabólicas para a produção da enzima e contribuir para o desenvolvimento de abordagens para a diferenciação dentro taxonômica do gênero *Burkholderia*, o genoma foi sequenciado no NFN (ALNOCH et al., dados não publicados).

Segundo Alnoch et al. (dados não publicados), o genoma é composto por três cromossomos com tamanhos 3,66 Mpb, 3,55 Mpb e 1,4 Mpb. O número de cópias do operon ribossomal ainda não foi determinado.

2.8.2 Estudados pelo GAGE-B e utilizados na validação do método

2.8.2.1 *Aeromonas hydrophila*

Aeromonas hydrophila é uma proteobacteria, heterotrófica, gram-negativa, que pode sobreviver em ambientes aeróbicos ou anaeróbicos. *A. hydrophila* emergiu como um importante agente patogênico humano, uma vez que causa gastroenterite e infecções extra-intestinais (NAGAR et al., 2016)

A. hydrophila ATCC 7966 (GenBank número de acesso NC_008570) possui um cromossomo constituído de 4.744.448 pb com conteúdo GC de 61,5% e 5.195 regiões codificantes (CDS), apresenta 128 tRNA preditos e dez cópias do operon ribossomal. Algumas cópias se diferenciam em 1 pb (SESHADRI et al., 2006).

2.8.2.2 *Bacillus cereus*

Bacillus cereus é uma bactéria do filo Firmicutes, gram-positiva, beta hemolítica, de forma cilíndrica, endêmica, que vive no solo e é adaptada para o crescimento no trato intestinal de insetos e mamíferos. A partir desses habitats se

espalham para alimentos, de maneira que podem causar diarreia (ARNESEN et al., 2008).

O gênero da estirpe *B. cereus* ATCC 10987 se encontra depositada no Genbank (*Genbank Access number* NC_003909 e NC_005707). A estirpe apresenta um cromossomo com 5.224.283 pb, conteúdo GC de 35%, a anotação indica 5.772 genes, sendo 5.603 codificantes (CDS), 97 tRNAs e 36 rRNA (12 cópias do operon ribossomal). O único plasmídeo possui 208.369 pb e 241 genes codificantes (CDS).

2.8.2.3 *Bacteroides fragilis*

Bacteroides fragilis é uma bactéria gram-negativa, anaeróbia, em forma de haste que compreende cerca de 1 a 2% da microflora do cólon humano e é fundamental para a imunidade da mucosa e sistêmica e a nutrição do hospedeiro. No entanto, também é um agente patogênico oportunista, sendo o principal isolado anaeróbio em espécimes clínicos, infecções na corrente sanguínea, e abscessos abdominais (NIKITINA et al., 2015).

Bacteroides fragilis 638R (*Genbank access number* NC_016776) apresenta um cromossomo linear com 5.373.121 pb, conteúdo GC de 44,5% e a anotação indica 4.417 genes, sendo 4326 codificantes, 19 rRNA e 72 tRNA.

2.8.2.4 *Mycobacterium abscessus*

Mycobacterium abscessus pertence a um grupo de micobactérias de crescimento rápido (RGM). *M. abscessus* faz parte de um grupo de micobactérias ambientais encontradas na água, solo, e poeira. Conhecida por contaminar medicamentos e produtos, incluindo os dispositivos médicos. *M. abscessus*, é a espécie de micobactéria mais resistente aos medicamentos conhecidos e exibe taxas de resposta de tratamento insatisfatória, especialmente para pacientes com doença pulmonar. Conforme Dymova et al., (2016), “[...] determinou-se a sequência de todo o genoma, pelo sequenciador GS Júnior, da estirpe NOV0213, isolado a partir de um paciente com a doença semelhante à tuberculose e com um elevado nível de resistência a vários antibióticos”.

Mycobacterium abscessus 6G-0125-R (*Genbank access number* NC_010397 e NC_010394) apresenta um cromossomo com 5.067.172 pb, conteúdo GC de 64,5% e uma anotação que identifica 4.970 genes, sendo 4920 codificantes, 3 cópias de rRNAs e 47 tRNAs.

2.8.2.5 *Rhodobacter sphaeroides*

Rhodobacter sphaeroides é uma bactéria púrpura não sulfurosa (BPNS) pertencente à classe de alfa-proteobacteria. *R. sphaeroides* tem um metabolismo alternado que a torna capaz de crescer de forma fotoheterotrófico ou quimioheterotrófico ou autotroficamente e fixa nitrogênio. O genoma de *R. sphaeroides* 2.4.1 foi sequenciado originalmente em 2001 e dois cromossomos (I e II) e cinco plasmídeos (A, B, C, D e E). Recentemente, após o sequenciamento do genoma de uma estirpe mutante, foi identificado um número de potenciais erros na sequência originalmente publicada da *R. s. 2.4.1*. KONTUR et al. (2012) revisaram as sequências dos cromossomos I e II e revisaram toda a anotação, inclusive dos cinco plasmídeos. O genoma é constituído de 4.6 Mb e um conteúdo GC de 69%.

Neste trabalho, foram utilizados as sequências depositadas no Genbank e identificadas pelos *Access Numbers* NC_007488, NC_007489m NC_007490, NC_007493, NC_007494, NC_009007 e NC_009008.

2.8.2.6 *Staphylococcus aureus*

Staphylococcus aureus, do filo Firmicutes, é uma das principais causas de infecções hospitalares e uma das principais causas de mortalidade (NCBI Genome, 2016). Recentemente, foram sequenciadas quatro estirpes da *S. aureus* isoladas de vacas de dois rebanhos de Minas Gerais, que apresentavam infecções persistentes (SILVA et al., 2016). *S. aureus* apresenta um cromossomo e dois plasmídeos (*Genbank Access Number* NC_010063, NC_010079, NC_012417).

2.8.2.7 *Vibrio cholerae*

Vibrio cholerae é uma proteobactéria que pode colonizar a superfície da mucosa do intestino delgado de seres humanos, onde provocará a cólera, doença diarreica grave e aparecimento repentino. Tem-se demonstrado que *V. cholerae* permanece no ambiente entre os surtos. Vale mencionar que tal proteobactéria foi detectada em regiões mais amplas do que se pensava anteriormente (WU et al., 2016). Quanto à sua dimensão, o genoma é formado por dois cromossomos com tamanho total de 4 Mb e um conteúdo GC de 48%. Um exemplar do genoma se encontra depositado no Genbank sob os números de acessos NC_002505 e NC_002506.

2.8.2.8 *Xanthomonas axonopodis*

Xanthomonas axonopodis é uma espécie que causa cancro cítrico, uma infecção localizada que afeta plantas cítricas em todo o mundo. A virulência pode aumentar na presença do inseto *Phyllocnistis citrella* (NCBI Genome, 2016, Xa).

X. anthomonas é um gênero de gammaproteobactéria que inclui numerosas espécies fitopatogênicas, cada espécie caracterizada por uma estreita gama de hospedeiros. No entanto, os membros do gênero são capazes de infectar uma ampla variedade de plantas, distribuídas ao longo de 124 monocotiledôneas e 268 espécies de plantas dicotiledôneas (ALBUQUERQUE et al., 2011). O genoma é formado por um cromossomo com 2.9mb e um conteúdo GC de 33%. O exemplar do genoma obtido no Genbank tem o número de identificação NC_016010.

2.9 FERRAMENTAS UTILIZADAS NOS PROCESSOS DE MONTAGENS

2.9.1 NCBI BLAST

O *Basic Local Alignment Search Tool* (BLAST) busca regiões com similaridade entre duas sequências (ALTSCHUL et al, 1990). Para funcionar o blast, cria um banco de dados com um conjunto de sequências a serem comparadas, de modo que a sequência de interesse (*query*) é confrontada contra o banco. O algoritmo utiliza uma heurística de procurar as sequências candidatas ao alinhamento com base em sementes (*seeds*) e, posteriormente, amplia o “alinhamento”. A comparação das sequências é realizada pelo algoritmo de alinhamento local, criada originalmente por Smith-Waterman.

2.9.2 MUMmer

O MUMmer é um software livre para comparação de genomas com diferentes distâncias evolutivas. O MUMmer permite encontrar correspondências entre as sequências com ou sem repetições, com comparações exatas e similares. O MUMmer organiza os dados em uma estrutura de árvores de sufixos e o uso de memória pelo MUMmer está próximo do mínimo possível, mantendo o tempo ótimo

ou próximo do ótimo no pior dos casos executados. Os fontes do programa MUMmer estão disponíveis em <http://www.tigr.org/software/mummer> (KURTZ et al., 2004).

2.9.3 Gepard

Gepard ("chita" em alemão) é um acrônimo para a expressão inglesa "GENome PAir – Rapid Dotter". Gepard permite o cálculo de Dotplots para grandes sequências cromossômicas. Realiza o alinhamento local entre duas sequências de nucleotídeos ou aminoácidos a partir de arquivos especificados pelos usuários. O Gepard pode ser executado a partir da linha de comandos e é escrito em Java. O programa utiliza matrizes sufixo para a heurística de cálculo do Dotplot. Para grandes Dotplots, ele procura correspondências de palavras exatas de um comprimento determinado pelo usuário (o valor 10 é definido como padrão) e prossegue a busca na matriz de sufixo da segunda sequência. Essa estratégia, segundo os autores, reduz a complexidade de $O(m \times n)$ para $O(m \times \log n)$, onde m é o tamanho da maior sequência e n o tamanho da menor sequência. Para pequenos Dotplot, o cálculo é realizado através da clássica janela (KRUMSIEK et al., 2007)

2.9.4 Artemis

Artemis é uma ferramenta para visualização de genomas e anotações, que permite diferentes análises baseadas em sequência e também a tradução e visualização dos seis *frames*. Artemis foi escrito em Java e se encontra disponível para os sistemas operacionais UNIX, Macintosh e Windows. Ele pode ler arquivos de dados no formato EMBL, GenBank, FASTA, entre outros (RUTHERFORD et al., 2000).

3 MATERIAIS E MÉTODOS

Essa seção foi estruturada na mesma ordem que ocorrem as tarefas do processo de montagem. Estão descritos: primeiramente, o conjunto de dados e a fase de pré-processamento ou análise do conjunto de dados que levou ao desenvolvimento de três experimentos: a) filtragem das leituras; b) verificação de emparelhamento e c) análise de k-mer. Depois, estão descritas a fase de processamento, que levou ao desenvolvimento do script todosMer, e a fase de pós-montagem, que levou ao desenvolvimento do programa para finalização e refinamento das montagens – G-Finisher.

3.1 CONJUNTO DE DADOS

Foram utilizados dois conjuntos de dados, de maneira que o primeiro foi proveniente de sequenciamentos de organismos de interesse e realizados pelo Núcleo de Fixação de Nitrogênio (NFN) e o segundo conjunto de dados foi obtido no site do projeto GAGE-B. O primeiro foi utilizado para obter a montagem dos genomas dos organismos de interesse do NFN e para o desenvolvimento e teste das ferramentas decorrentes desse trabalho. O segundo conjunto foi aplicado para validar e comparar as soluções desenvolvidas.

Em todos os experimentos foi realizada ao menos uma comparação com conjunto de dados de leituras e montagens do GAGE-B, como um controle externo e para reduzir eventuais influências dos equipamentos e dos kits utilizados pelo NFN ou pelo protocolo interno de sequenciamento.

3.1.1 Características das leituras obtidas nos sequenciamentos

A Tabela 3 apresenta os dados brutos que descrevem os resultados obtidos pelo sequenciamento. O genoma da *A. brasilense* FP2 foi sequenciado por três grupos distintos, sendo o primeiro sequenciamento foi realizado pela empresa FASTERIS (Genebra, Suíça), utilizando uma biblioteca do tipo *pair-end*, na plataforma Illumina, com distância média de aproximadamente 300 pb entre as leituras. O segundo sequenciamento foi realizado no NFN com a tecnologia de sequenciamento SOLiD 3 (Life Technologies), utilizando uma biblioteca do tipo

mate-pair com distância média de 1.750 pb (SILVEIRA, 2012). O terceiro sequenciamento foi realizado no NFN utilizando a tecnologia Illumina MiSeq e a construção da biblioteca com o kit Nextera do tipo *pair-end*. O quarto sequenciamento foi realizado no laboratório da professora doutora Cyntia Maria Telles Fadel Picheth, integrante do NFN, utilizando um equipamento GS Junior (Roche 454). O sequenciamento do genoma da *A. brasilense* Sp7 foi desenvolvido no NFN com o equipamento Ion Proton S5 utilizando uma biblioteca de fragmento. A bactéria *Burkholderia contaminans* LTEB foi sequenciada no NFN com o equipamento Ion Proton S5, utilizando uma biblioteca de fragmento, e com Illumina MiSeq, sob o emprego do kit para preparação da biblioteca Nextera XT do tipo *pair-end*. O sequenciamento da *Herbaspirillum hiltneri* N3 teve inicialmente a construção de duas bibliotecas para o sequenciamento do tipo fragmento e *mate-pair* no SOLiD 4 e posteriormente uma nova biblioteca foi construída com o kit Nextera para o sequenciamento no Illumina MiSeq. O genoma da *Herbaspirillum seropedicae* Z67 foi sequenciado pelo NFN no Illumina MiSeq utilizando uma biblioteca do tipo *pair-end* e o kit Nextera XT.

TABELA 3 - CARACTERÍSTICAS DAS LEITURAS E DO SEQUENCIAMENTO

Organismo	Plataforma de sequenciamento	Tipo de biblioteca	Leitura		Cobertura
			Quantidade	Tamanho	
<i>Azospirillum brasilense</i> FP2	Illumina	<i>pair-end</i>	5.768.466	2x38	32,2
	SOLiD	<i>mate-pair</i>	112.628.482	2x50	828,2
		Órfão	288.468	50	2,1
	Illumina MiSeq	<i>pair-end</i>	777.652	2x250	28,6
	Roche 454	fragmento	128.954	450	8,5
<i>Azospirillum brasilense</i> Sp7	Ion Proton S5	fragmento	3.913.846	145	70,9
<i>Burkholderia contaminans</i> LTEB	Illumina MiSeq	<i>pair-end</i>	2.698.078	2x211,1	71,2
	Ion Proton	fragmento	5.661.193	125	88,5
<i>Herbaspirillum hiltneri</i> N3	SOLiD	<i>mate-pair</i>	105.987.246	50	1059,9
		Órfão	754.321	50	7,5
		Fragmento	8.382.369	50	83,8
	Illumina MiSeq	<i>pair-end</i>	5.088.454	2x250	254,4
<i>Herbaspirillum seropedicae</i> Z67	Illumina MiSeq	<i>pair-end</i>	4.797.230	2x300	266,5

FONTE: O autor

NOTA: Os fragmentos “órfãos” são leituras obtidas do sequenciamento de bibliotecas de pares de leituras que perderam durante o processo o respectivo par.

3.2 ANÁLISE E TRATAMENTOS DAS LEITURAS

3.2.1 Análise das bases

A análise inicial das leituras permite estabelecer as estimativas de %GC, cobertura – se tiver algum indício do tamanho esperado do genoma – e a frequências de bases ambíguas que, juntamente com a informação de qualidade, fornecem uma expectativa da complexidade do processo de montagem. É esperado que o sequenciamento de genomas com mais de 30x de cobertura e com qualidade média superior a Q15 resultem em uma representação superior a 99% do genoma. Da mesma forma, é esperado que genomas com percentuais de GC próximos de 50% sejam mais fáceis de sequenciar. Se de um lado a análise serve para realizar especulações, por outro é um fator determinante para identificar a necessidade e identificar quais tratamentos serão necessários. Importante observar que o sistema da Illumina que acompanha o MiSeq já faz um tratamento prévio que remove adaptadores e sequências de baixa qualidade. Contudo, tanto nos dados de sequenciamento do NFN quanto do GAGE-B foram encontrados fragmentos de adaptadores e outras “impurezas” nos dados.

O programa que usamos para fazer o processo de análise foi o FastQC (Item 2.2.3). Exemplos de telas e de relatórios de dados bons e ruins estão disponíveis na internet, por meio do site do projeto, mantido pelo Instituto Babraham (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>).

O programa trabalha com arquivos no formato FASTQ (COCK et al., 2010), considerando que os dados do SOLiD são fornecidos em formato fasta e codificados em *color-space* e do Roche 454 no formato SFF ou FASTA/QUAL e precisam ser convertidos. Existem diversos scripts e programas na internet, mas encontramos algumas dificuldades nesse campo, tanto que desenvolvemos em Java uma ferramenta para essas operações. No NFN existe um equipamento com o software comercial CLC Genomics Workbench que permite fazer as conversões.

A Figura 7 apresenta a captura da tela do relatório emitido pelo FastQC, onde o ícone verde indica que o FastQC não identificou problemas naquela análise; amarelo indica uma atenção necessária e o vermelho a presença de problemas. O tratamento aplicado nas leituras pelo pesquisador depende dos resultados das análises realizadas pelo FastQC.

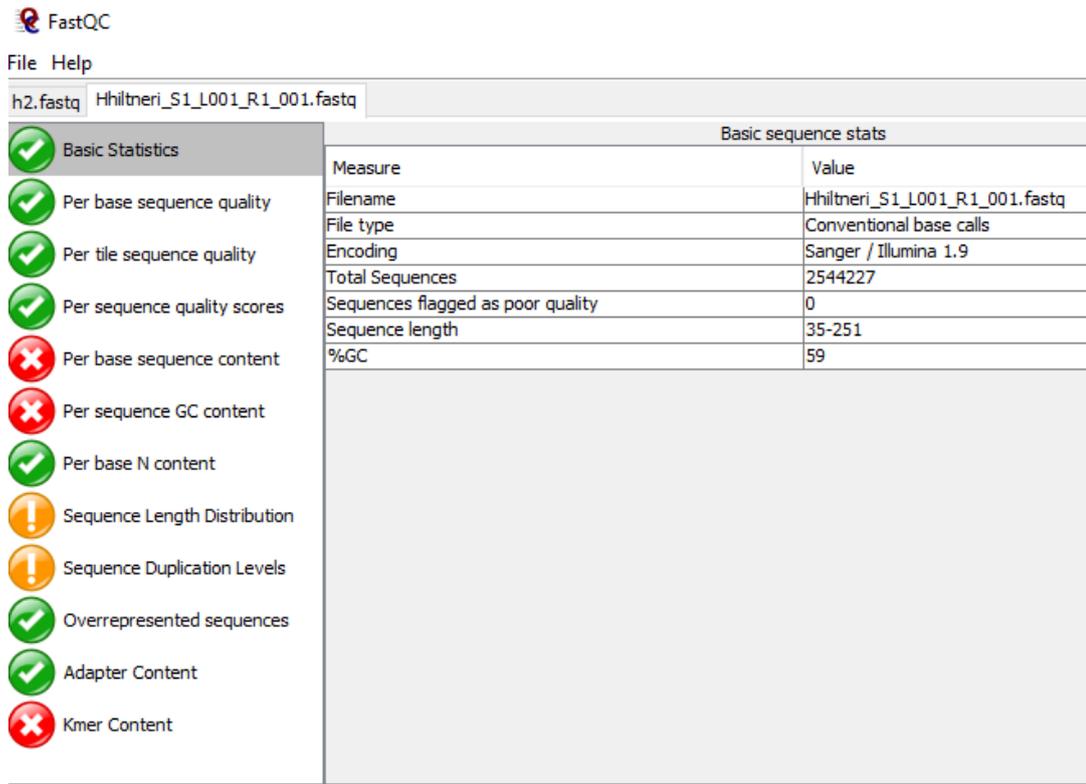


FIGURA 7 - CAPTURA DA TELA DO FASTQC DO RELATÓRIO DA ANÁLISE DAS LEITURAS DA *H. hiltneri* N3 FORNECIDAS PELO ILLUMINA MiSeq

FONTE: O autor

A Figura 8 mostra a captura de tela do relatório “Per base sequence content” e equivale ao primeiro sinal vermelho do relatório apresentado na Figura 7.

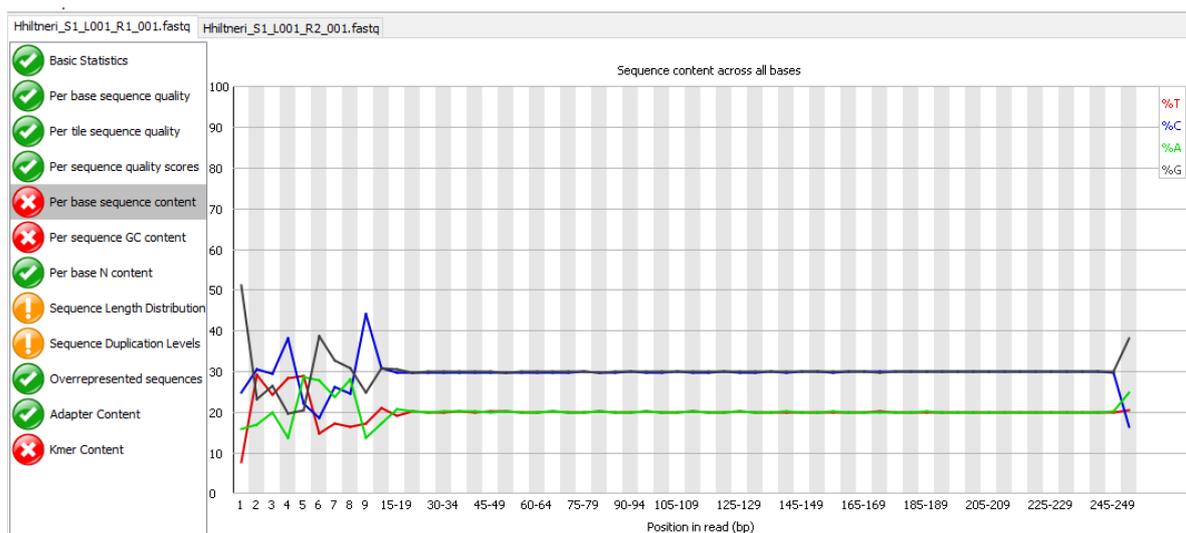


FIGURA 8 - CAPTURA DO RELATÓRIO DO PERCENTUAL DA OCORRÊNCIA DE BASE POR COLUNA NAS LEITURAS DA *H. hiltneri* N3

FONTE: O autor

Observa-se que o percentual de bases G ($\approx 30\%$) e C ($\approx 30\%$), indicado entre as bases 15 e 245 presentes no relatório capturado na Figura 8, corresponde ao valor estimado de 59% registrado na Figura 7. Essa análise indica um viés forte nos sequenciamentos MiSeq em relação às 15 primeiras bases das leituras, confirmado com o mesmo estudo realizado em outros conjuntos. A primeira base em todas as leituras do arquivo R1 inicia com Guanina (50% dos casos) ou Citosina (25%); a segunda base é Citosina ou Timina (30%); a terceira base das leituras é C(30%), G(27%) ou T(25%); a quarta base é C(38%) ou T(28%); a quinta base é A ou T (28%); sexta base é G(38%) e assim por diante. Apenas após a 15ª ou 16ª base é que os percentuais refletem a frequência de bases esperada se as leituras fossem iniciadas em posições aleatórias do genoma.

A Figura 9 mostra os resultados da mesma análise apresentada na Figura 8, mas com dados do sequenciamento da *H. hiltneri* N3 na plataforma SOLiD (A), do *A. brasilense* Sp7 no Ion S5 (B), *A. brasilense* FP2 (C) no SOLiD e *H. seropedicae* Z67 (D) no Illumina MiSeq.

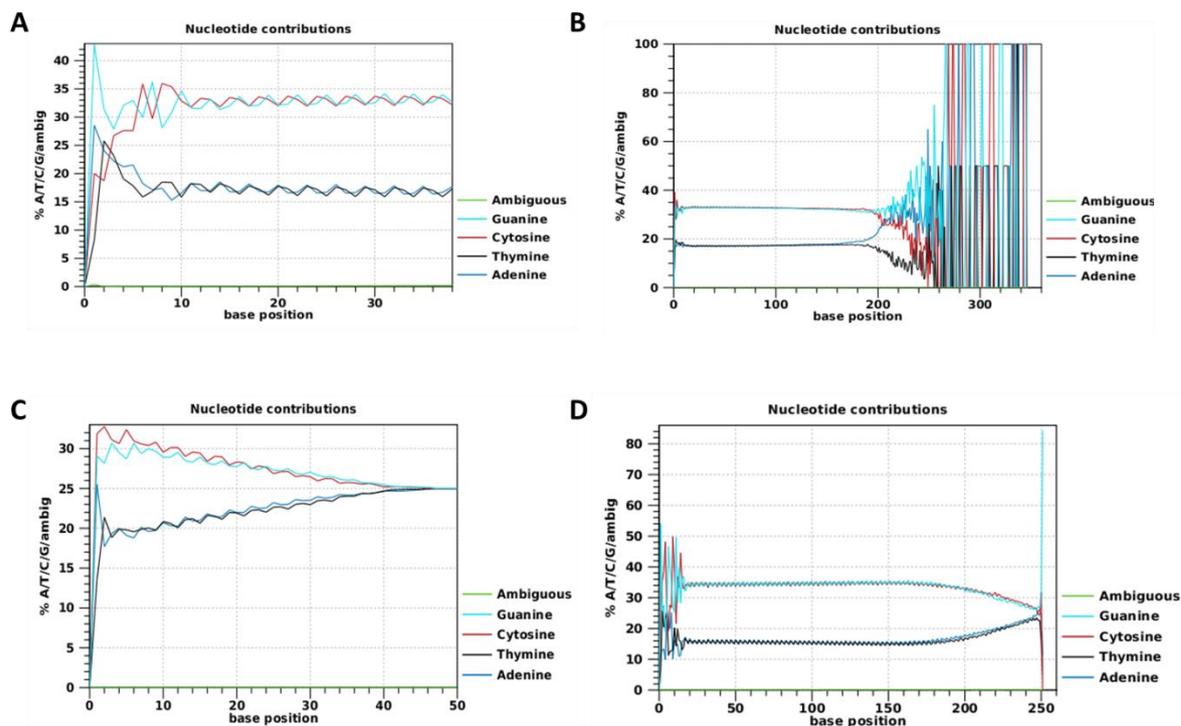


FIGURA 9 - VIÉS SE REPETE EM OUTROS ORGANISMOS E TECNOLOGIAS DE SEQUENCIAMENTO

FONTE: O autor

NOTA: (A) *H. hiltneri* N3 na plataforma SOLiD, (B) *A. brasilense* Sp7 no Ion, (C) *A. brasilense* FP2 no SOLiD e (D) *H. seropedicae* Z67 no Illumina MiSeq.

3.2.2 Análise da qualidade de bases

Sequenciamentos realizados na plataforma Illumina MiSeq apresentam uma qualidade média superior a Q30 o que praticamente descarta os processos de filtragem por qualidade. Na plataforma Ion, a qualidade média fica entre Q15 e Q20 e, no sequenciador SOLiD, a frequência de bases com qualidade inferior a Q15 é muito alta. Nos dados SOLiD, o processo de filtragem chega a descartar de 75% a 90% das bases em função da qualidade inferior a Q15. Mesmo assim, os dados restantes devidamente tratados permitem a montagem de genoma utilizando a estratégia *De novo*.

Os nossos experimentos corroboram com os resultados e interpretação de DEL FABBRO *et al* (2013) que analisou os efeitos da filtragem de dados em sequenciamentos MiSeq e concluiu em relação a montagem *de novo* que a “filtragem rigorosa por qualidade (Q30) tende a remover dados aceitáveis e diminuir a qualidade geral da montagem”.

A variação de qualidade e a necessidade de assegurar leituras com qualidade mínimas em torno de Q15, justificaram o desenvolvimento do jTrimmer.

3.2.3 Parâmetros considerados para aplicação de filtros

A relação entre filtragem e cobertura é muito tênue, pois existem regiões de genomas que apresentam valores extremos de percentuais de GC que irão influenciar a qualidade e que tendem a ter menor cobertura. Mudanças bruscas de alto teor para baixo teor de GC são pontos críticos de sequenciamento.

Neste trabalho, em todas as montagens, tentamos assegurar uma cobertura mínima de 30x, na impossibilidade reduzimos esse limite de 5 em 5 até o limite de 15x. A cobertura determina a estratégia de montagem, para leituras curtas com menos de 15x de cobertura conseguimos realizar apenas montagens baseadas “em referência”.

Realizamos montagens com dados com qualidade alta (Q20 ou superior) e depois com qualidade Q15 ou superior, de modo que somente para os dados SOLiD utilizamos qualidade Q10 ou superior para preenchimento de lacunas.

3.3 DESENVOLVIMENTO DAS FERRAMENTAS DE PRÉ-PROCESSAMENTO

3.3.1 Desenvolvimento da ferramenta de filtragem

As principais dificuldades que encontramos com as ferramentas de filtragem disponíveis foram: a) plataforma e versão do sistema operacional ou do distribuidor dependente; b) dependência de formato específico; c) codificação específica de qualidade; d) os parâmetros eram informados por valores em diferentes escalas ou notações; e) as estratégias não eram universais.

Durante a montagem da *H. hiltneri* N3 e da *A. brasilense* FP2, identificamos nas montagens do Velvet uma redução do percentual de GC para valor muito inferior ao calculado com base nas leituras e mais baixo do que para as montagens obtidas no CLC. Ao investigar o motivo, encontramos no código do Velvet o tratamento para bases ambíguas (N) que são substituídas por A. Esse fato fez que substituíssemos as ocorrências de múltiplos Ns por apenas um N ou podássemos a leitura preservando a maior subsequência íntegra.

Também verificamos que a ordem de aplicação dos filtros modificava os resultados do conjunto filtrado. Por exemplo, o filtro para remoção de sinais ou adaptadores (quatro primeiras bases em corridas 454) deve ser realizado antes de outras estratégias para evitar perda de dados.

O jTrimmer foi escrito em java e projetado para executar os filtros na ordem que são apresentados os parâmetros na linha de comando. Os filtros básicos são: a) poda de n bases da extremidade 5'; b) poda de n bases da extremidade 3'; c) poda a partir da base que apresentar qualidade inferior a n; d) poda a partir da base que apresentar probabilidade de erro maior do n., onde n são valores informados pelo utilizador do programa. Além dos filtros básicos, foi criado um algoritmo novo, denominado "melhor subsequência maior que n", que busca o melhor subconjunto que apresenta a melhor média de qualidade, dado um parâmetro de tamanho mínimo. O algoritmo aplica o conceito de média móvel, criando "janelas deslizantes" de tamanhos variáveis, calculando a média da qualidade e preservando a subsequência de maior qualidade.

3.3.2 Desenvolvimento da ferramenta de verificação de emparelhamento

Constatamos que as leituras em pares obtidas no MiSeq e no SOLiD não estavam na mesma ordem nos dois arquivos fornecidos pelo sequenciamento e que os montadores não verificam a relação de correspondência das leituras com base nos cabeçalhos.

Escrevemos o programa “verificaPair” em java que recebe como parâmetros os nomes dos arquivos de leituras, os nomes dos três arquivos de saída (os dois arquivos que preservam os pares em correspondência e o arquivo das leituras órfãs) e o padrão do cabeçalho. O programa possui alguns padrões pré-definidos, mas aceita, como alternativa, uma expressão regular em que capture a parte do texto que deve coincidir nos pares de sequências e distinguir dos demais pares.

O programa varre os dois arquivos de entrada, sequencialmente, e avalia os cabeçalhos das sequências e no caso de correspondência salva as sequências nos arquivos correspondentes de saída, e no caso de disparidade, os identificadores são verificados no banco de sequências orfas mantidas em memória. Caso encontre o par correspondente no banco, as leituras são salvas e removidas da memória, caso contrário, são incluídas no banco.

3.3.3 Obtenção do conjunto k-mer

A obtenção de todas as substrings de comprimento k (k-mer) de sequências de DNA é um passo comum em diferentes análises que ocorrem no processo de montagem. Os k-mers podem ser obtidos das leituras, dos contigs e/ou das sequências de um genoma. As contagens de k-mer permitem estimar o tamanho de um genoma a partir das leituras, identificar as regiões de repetição ou indexar um conjunto de sequências.

O JELLYFISH (Marçais; Kingsford, 2011) é uma ferramenta para contagem rápida de k-mers em sequências de DNA. É um dos programas mais eficientes para contagem e faz parte de outros programas, como o MaSuRCA e o QuorUM (Marçais et al., 2015).

Percorrer o conjunto de leituras, ou de contigs ou do genoma e extrair as subsequências é um processo com complexidade linear ($O(1)$). Entretanto a implementação nas linguagens de programação deparam com o problema de não

ser possível conhecer previamente o tamanho real do número de k-mers diferentes. O custo do redimensionamento de matrizes impacta no tempo de execução, dessa forma investigamos como os programas tratam esse procedimento. O Velvet resolve o problema com o uso de hash (função CRC) para uma matriz que aponta para uma segunda estrutura em árvore espalhada (splaytable). O Jellyfish utiliza a estrutura tabela de hash e uma estratégia de tempo quadrático para resolver as colisões. O Jellyfish apresenta uma estratégia para evitar o bloqueio, necessário em situações de paralelismo (multi-thread).

Nesse trabalho, optamos em utilizar a estrutura *Hashmap* e em definir o tamanho inicial do vetor de chaves com o mesmo tamanho das sequências e quando não for possível com o valor inicial de 20.000.000 e fator de crescimento padrão. A função hash utilizada foi baseada na codificação binária das sequências de nucleotídeos dos k-mers, uma transformação inversível, de forma que k-mers de comprimento menores que 16 são armazenados em 32 bits, os k-mers com tamanhos de 32 a 64 são armazenados em 64 bits e com tamanhos acima de 32 em um vetor de bytes, onde cada byte armazena a representação de 4 nucleotídeos. O valor da função hash é o próprio valor numérico obtido para os dois primeiros casos e uma função adicional foi criada para o array de bytes.

3.4 ETAPA DE MONTAGEM

Neste trabalho partimos do protocolo descrito no manual da *Applied Biosystems* e representado na Figura 1, onde o processo de montagem é constituído das etapas de análise e tratamento das leituras (pré-montagem), a etapa de montagem e a avaliação e análise dos resultados do montador (etapa de pós-montagem ou finalização).

As montagens realizadas nesse trabalho foram realizadas com o Velvet, CLC Genomics Workbench 6.5, Edena V3, MIRA, MaSuRCA, Allpaths-LG, SOAPDenovo, e Newbler.

Os montadores, de forma geral, têm como entrada um ou mais arquivos com as leituras organizadas nos formatos FASTA e FASTQ. Entretanto, algumas combinações apresentam maiores restrições, como combinar bibliotecas de pair-end com bibliotecas de fragmento. Os montadores partem da premissa que os dados estão corretos, ou seja, que são apresentados na forma esperada pelo montador,

dessa forma quase não existe verificações, validações ou retorno de eventuais problemas nas leituras ou na forma que as mesmas são submetidas aos montadores. Cabe ao usuário verificar os dados antes da montagem e averiguar a codificação aceita pelo montador, tanto para a base quanto para a qualidade.

Para os dados da *H. hiltneri* N3 e *A. brasilense* FP2, foram realizadas várias montagens alternando entre o tratamento dos dados e a variação dos parâmetros dos montadores. Posteriormente, os parâmetros foram comparados com os passos previstos nos pipelines disponíveis pelo GAGE-B (http://ccb.jhu.edu/gage_b/recipes/recipes.pdf, consultado em 24/05/2016).

Cabe destacar, inicialmente, que o montador principal era o Velvet e que outros montadores eram utilizados apenas para testes, entre eles Edena V3, Newbler e MIRA. No período de 2012 ao primeiro semestre de 2013, as montagens realizadas em outros montadores apresentavam resultados inferiores aos obtidos com o Velvet.

As montagens realizadas com o Velvet dos dados de sequenciamento da *H. hiltneri* N3 e da *A. brasilense* FP2 não permitiram estabelecer uma relação boa entre o tamanho do K-mer e o tamanho médio das leituras, mesmo o uso do script VelvetOptimiser não garantia boa relação entre o número de contigs, N50 e o tamanho esperado do genoma. Dessa forma, optamos em desenvolver o script *todosMer.sh*.

Uma vez que no final de 2013 o NFN adquiriu a licença do CLC Genome Workbench e, em decorrência da publicação do GAGE-B, começamos a testar os montadores MaSuRCA, o SOAPdenovo e o SPAdes. Por causa do ambiente Linux instalado em nossos servidores, o MaSuRCA foi o mais compatível, considerando que o SOAPdenovo apresentou alguns problemas de instabilidade e compatibilidade, em especial com as versões das bibliotecas básicas do Sistema Operacional Linux (a CLIBC) instalados no servidor SGI Altix UV.

3.5 ETAPA DE PÓS-MONTAGEM

A etapa de pós-montagem consiste na análise dos *contigs/scaffolds* para avaliar o avanço da montagem e identificar elementos que auxiliem no fechamento do genoma. Nessa etapa, são coletados dados estatísticos, realizadas anotações automáticas e a análise mais aprofundada nas bordas dos *contigs*.

A parte estatística e o estudo comparativo da montagem com o genoma de referência e entre as diferentes montagens estão gradativamente sendo melhoradas no QUAST, dessa forma, substituímos quase todos os procedimentos domésticos previamente criados pelo QUAST. A anotação automática foi realizada no SILA e no RAST, e a anotação dos tRNAs foi feita por meio do programa tRNAScan.

Com frequência, a análise do Dotplot e do GC Skew foi considerada para acompanhar o processo de montagem, a ponto de que essas ferramentas começaram a inspirar a automação dos processos que culminaram na construção do G-Finisher.

Após a obtenção de uma montagem viável, realizamos alinhamento das leituras com a sequência obtida, para calcular o percentual de leituras presentes na sequência, à distância entre os pares e a cobertura final. Esse procedimento tem sido realizado com os programas CLC Genome Workbench e Bowtie.

3.6 ANÁLISE COM O DOTPLOT

O Dotplot permite uma análise global e a identificação dos três fenômenos decorrentes do alinhamento de duas sequências: inserção, exclusão e inversão exemplificadas na Figura 10.

As primeiras análises de dotplot foram feitas com os recursos do pacote MUMmer; posteriormente, adotamos o Gepard. O MUMmer é lento, exige vários passos para obter a imagem, o resultado é uma imagem de baixa resolução e não é incomum ocorrerem erros inesperados em sua execução. O Gepard é rápido, tem uso mais simples, produz dotplot de maior resolução e permite visualizar o alinhamento local de uma região de interesse, além de poder ser executado em qualquer sistema operacional por ser escrito em Java.

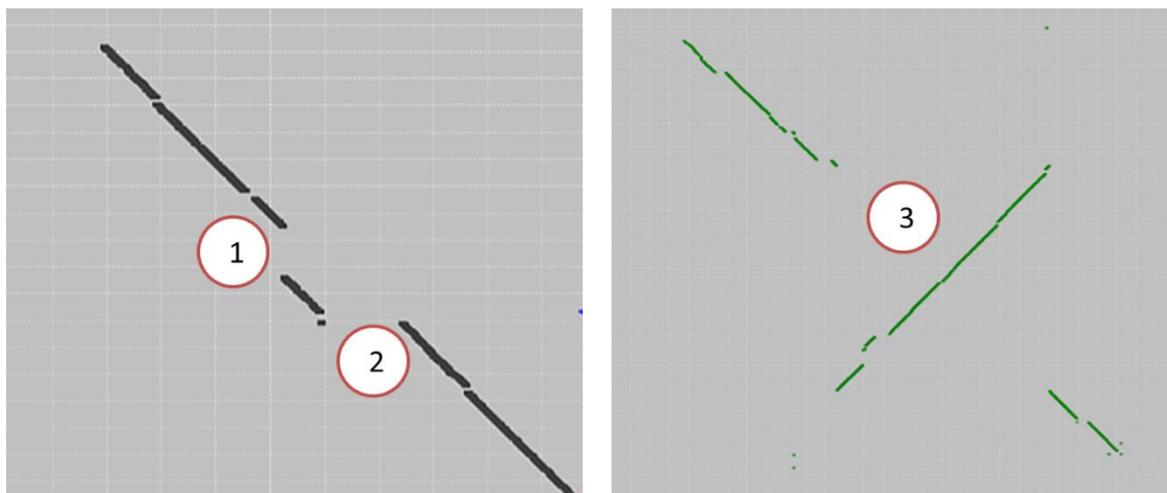


FIGURA 10 - DIFERENÇAS DE ALINHAMENTOS DE SEQUÊNCIAS GRAFICAMENTE DEMONSTRADAS NO DOTPLOT

FONTE: O autor

NOTA: Em (1) exemplo de uma ocorrência de exclusão na montagem (eixo-x) em relação à referência (eixo y), em (2) uma inserção. No rótulo (3), há uma inversão interna em um contig em relação à sequência de referência.

3.7 AVALIAÇÃO DA MONTAGEM PELO QUAST

Um dos objetivos desta tese é identificar métricas e desenvolver um programa que realize a avaliação e comparação das montagens, gerando indicação para tomada de decisão por parte do pesquisador e, posteriormente, para orientação e automação do processo de montagem. Antes que tivéssemos mapeado as variáveis e medidas possíveis, foram publicados os trabalhos do Assemblathon e do QUAST. O primeiro foi destinado mais para eucariotos e o segundo apresentava a maior parte dos conceitos que estávamos avaliando, especialmente no que se refere à identificação de genes completos e parciais nas montagens. Essa informação nós já tínhamos na anotação automática do SILA, de modo que já sintetizávamos os scores de alinhamentos.

Quando surgiu o QUAST, abandonamos essa linha de trabalho e o adotamos como ferramenta de avaliação e comparação das montagens. Para facilitar a utilização do QUAST e preservar as condições (parâmetros) para realizar as comparações entre as diferentes montagens de um mesmo organismo, bem como entre os diferentes sequenciamentos, incorporamos no G-Finisher a preparação dos

arquivos de *contigs* e de referência e automatizamos a criação do script de execução do QUASt para cada conjunto de dados tratados pelo G-Finisher.

Após a execução do QUASt, diversos relatórios são disponibilizados, incluindo uma versão para consulta em formato HTML, que apresenta uma síntese das comparações conforme apresentado na Figura 11.

Neste exemplo, três montagens distintas da *Herbaspirillum hiltneri* N3 são comparadas pelo QUASt sem o uso de um genoma de referência. Nesse modo, a versão rotulada com “26_08” indica melhores resultados em todos os critérios, em que se destacam a predição de onze genes a mais em relação às versões anteriores, sendo que destes, três são novos (campo *unique* do relatório).

QUAST report

27 August 2014, Wednesday, 08:56:45

All statistics are based on contigs of size ≥ 500 bp, unless otherwise noted (e.g., "# contigs (≥ 0 bp)" and "Total length (≥ 0 bp)" include all contigs.)

[Extended report](#)

worst.....best

Genome: 4 965 474 bp, G+C content: 60.76 %

Statistics without reference	≡ 26_08	≡ 15_06	≡ 30_04
# contigs	1	1	1
Largest contig	4 965 474	4 945 633	4 945 633
Total length	4 965 474	4 945 633	4 945 633
N50	4 965 474	4 945 633	4 945 633
Misassemblies			
# misassemblies	0	2	2
Misassembled contigs length	0	4 945 633	4 945 633
Mismatches			
# mismatches per 100 kbp	0	0.24	0.24
# indels per 100 kbp	0	0	0
# N's per 100 kbp	0	0	0
Genome statistics			
Genome fraction (%)	100	99.594	99.594
Duplication ratio	1	1	1
NGA50	4 965 474	4 786 693	4 786 693
Predicted genes			
# predicted genes (unique)	4484	4481	4481
# predicted genes (≥ 0 bp)	4546	4535	4535
# predicted genes (≥ 300 bp)	4153	4141	4141
# predicted genes (≥ 1500 bp)	637	633	633
# predicted genes (≥ 3000 bp)	61	59	59

FIGURA 11 - COMPARAÇÃO ENTRE TRÊS VERSÕES DE MONTAGENS DA *Herbaspirillum hiltneri* N3 FEITA PELO QUAST, SEM INDICAÇÃO DE REFERÊNCIA

FONTE: O autor

NOTA: A melhor montagem encontra-se representada na primeira coluna (26_08), as outras duas são comparadas utilizando a primeira como referência.

3.8 DESENVOLVIMENTO DO G-FINISHER

Inicialmente, desenvolvemos um pipeline para testar as combinações de soluções para cada etapa, aperfeiçoar os programas e modelar o processo. Quando os resultados começaram a apresentar desempenho médio superior a 68% nos

conjuntos de validação, decidimos integrar as ferramentas e estratégias em uma única solução, que denominamos G-Finisher.

O G-Finisher é uma aplicação escrita na linguagem Java e depende apenas do BLAST como ferramenta externa; ou seja, o G-Finisher prepara os dados, executa as ferramentas do Blast e interpreta os resultados (parse do relatório do BLAST). Todos os demais recursos fazem parte da aplicação. A aplicação foi desenvolvida para execução na linha de comando e, posteriormente, foi incluída uma interface simplificada para o uso em ambiente gráfico. Um ajuste mais refinado da aplicação para casos específicos implicará o uso do programa na linha de comando, uma vez que um número maior de parâmetros está disponível.

O G-Finisher foi executado nas 96 montagens do GAGE-B e em mais de 10 montagens do NFN sem a exigência de grandes recursos de hardware. Vale destacar que a maioria dos testes foi rodada em um notebook. Apesar disso, as tentativas de uso das ferramentas isoladamente no processo de montagens de dois genomas de eucariotos exigiram muita memória, sendo possível a sua execução de fato apenas nos servidores.

O G-Finisher é a junção dos programas *jContigsort*, *FGap*, *FGCSkew* e várias rotinas para manipulação dos dados, análise dos resultados e relatórios dessas ferramentas.

3.9 VALIDAÇÃO DO G-FINISHER

A estratégia de validação do G-Finisher foi a mesma desenvolvida por Al-okaily (2016) e Sá *et al.* (2016) que consiste em comparar os resultados obtidos com os resultados do GAGE-B (MAGOC *et al.*, 2013), ambos medidos com o QUAST.

As 96 montagens obtidas e fornecidas pelo GAGE-B foram utilizadas neste trabalho para validar a solução desenvolvida e comparar os resultados obtidos com outras aplicações. O número de *contigs* com pelo menos 200 pb das 96 montagens está relacionado na Tabela 4.

Os números na Tabela 4 demonstram que nenhum montador apresenta média do número de *contigs* inferior a 100 e que o MaSuRCA apresenta em 66% dos casos o menor número de *contigs* e em 84% dos casos o valor do N50 maior que os demais montadores. O CABOG e o SOAPDenovo2 apresentaram menor

número de *contigs* em um caso cada, mas foi o SPAdes que obteve o segundo melhor desempenho no critério do N50.

TABELA 4 - NÚMERO DE *CONTIGS* MAIORES QUE 199 pb OBTIDAS PELO GAGE-B.

Organismo	ABySS	CABOG	MIRA	MaSuRCA	SGA	SOAPdenovo	SPAdes	Velvet	Média
<i>A. hydrophila</i> (H)	75	105	1048	32	201	61	312	65	237,4
<i>B. cereus</i> (M)	115	78	153	90	3335	105	49967	404	6780,9
<i>B. cereus</i> (H)	472	164	676	250	961	410	1041	376	543,8
<i>B. fragilis</i> (H)	158	137	400	119	487	246	146	213	238,3
<i>M. abscessus</i> (H)	124	127	3241	66	377	91	96	155	534,6
<i>M. abscessus</i> (M)	209	857	1751	326	1114	113	908	279	694,6
<i>V. cholerae</i> (H)	206	127	728	105	484	139	205	261	281,9
<i>V. cholerae</i> (M)	267	241	430	173	1721	244	1475	201	594,0
<i>R. sphaeroides</i> (H)	603	537	1224	130	838	859	298	696	648,1
<i>R. sphaeroides</i> (M)	485	146	867	63	986	437	185	415	448,0
<i>S. aureus</i> (H)	103	56	207	52	259	70	68	70	110,6
<i>X. axonopodis</i> (H)	182	99	2750	155	313	202	191	214	513,3
Média	249,9	222,8	1122,9	130,1	923,0	248,1	4574,3	279,1	968,8

FONTE: O autor

Nota: (H) indica sequenciamento por HiSeq e (M) por MiSeq.

O G-Finisher foi aplicado com o mesmo conjunto de parâmetros em todas as montagens. A exceção foi em relação às montagens provenientes do MIRA, em que os resultados obtidos pelo G-Finisher foram inferiores à média observada nas demais montagens. Por motivos que não foram investigados detalhadamente, o montador MIRA produziu um grande número de repetições, em particular nas pontas dos *contigs*. Essas repetições superaram os valores de 5.000 pb que foram definidos como valor padrão para o parâmetro do tamanho máximo da borda do contig, considerada no jFGap. Portanto, foram adotados dois conjuntos de parâmetros, um para o MIRA especificamente nas duas corridas de *M. abscessus* 6G-0125-R e na montagem *V. cholerae* CO1032, e o segundo conjunto de parâmetros, considerado parâmetros padrão pelo programa, para os demais casos. As demais montagens do MIRA foram testadas com os parâmetros ampliados; porém foram obtidas melhoras significativas apenas nas corridas HiSeq, de forma que mantivemos o resultado do conjunto padrão de parâmetros.

O GAGE-B utilizou o QUASt para comparar as montagens construídas pelos diferentes montadores dos doze conjuntos de leituras. Neste trabalho, utilizamos o QUASt para comparar os resultados do GAGE-B com os resultados do G-Finisher. Os dados extraídos dos 96 relatórios do QUASt foram organizados e sistematizados no Excel. Comparamos o número de *contigs*, N50, N75, L50, L75, entre outros, das montagens do GAGE-B com as montagens intermediárias e finais do G-Finisher. Os dados foram sistematizados utilizando as tabelas dinâmicas e analisados.

3.10 FERRAMENTA DE AVALIAÇÃO DE MONTAGEM

A ferramenta jAnalyzer foi desenvolvida e utilizada entre 2012 e final de 2013, quando foi substituída pelo QUASt. O programa analisa a montagem e fornece, para cada *contig*, a distribuição de bases, percentual de GC, quantidade de bases ambíguas (no caso de scaffolds), o tamanho do maior *contig*, o total de bases da montagem, o N50 e o tamanho médio dos *contigs*.

3.11 FERRAMENTA DE USO GERAL PARA AUXÍLIO DAS MONTAGENS

O DnaTools, escrito em Java, foi criado para auxiliar nas tarefas que fizemos repetidas vezes e que quase sempre apresentavam alguma dificuldade. A última atualização do programa foi registrada em 31/10/2013, cujos recursos disponíveis são:

- a) complementar reversa – obtém a sequência complementar reversa de todas as sequências de um arquivo no formato fasta;
- b) separar sequências – dado um arquivo fasta contendo leituras ou *contigs* e um arquivo texto contendo o cabeçalho da sequência de interesse em cada linha, separa as sequências do primeiro arquivo que case o cabeçalho presente no segundo arquivo, produzindo um terceiro arquivo;
- c) multi-fasta para fasta único – separa um arquivo que contém várias sequências (multi-fasta) em vários arquivos com uma sequência cada;
- d) remove múltiplos N – substitui as múltiplas ocorrências de um símbolo por apenas um símbolo, por padrão o símbolo “N”, para identificar base ambígua;

- e) formato fasta de linha única – o formato fasta geralmente quebra as sequências em 60 nucleotídeos por linhas; veja-se que as versões mais antigas do Velvet não eram compatíveis com esse formato e a busca com expressões regulares era mais fácil em sequências únicas;
- f) formato FastQ de linha única – para compatibilizar com versões mais antigas do Velvet, sequências maiores que 60 nucleotídeos precisavam ser convertidas de múltiplas linhas para uma única linha;
- g) formato fasta padrão – o arquivo multi-fasta com sequências maiores que 60 nucleotídeos são quebrados em várias linhas;
- h) extrair as pontas dos *contigs* – ao informar um arquivo de sequências e um tamanho como parâmetro, as pontas dos *contigs* ou scaffolds eram copiados para o novo arquivo;
- i) indexar sequências – o cabeçalho era modificado e recebia no início a expressão “id=N”, onde N representava o número da sequência dentro do arquivo iniciado no valor informado pelo usuário;
- j) quebrar scaffolds – quebra as sequência de scaffolds em dois ou mais *contigs*;
- k) identificar por padrão regex - localiza as sequências que atendiam determinado parâmetro. Esse procedimento separava as leituras que atendiam uma semente codificada no formato de expressão regular, e foi utilizado para separar leituras para montagens de genes de interesse como o rRNA 16S.
- l) separar sequências intercaladas – o arquivo que tivemos acesso das leituras Illumina de *A. brasilense* sequenciado pela FASTERIS apresentava os pares de leituras intercaladas em um mesmo arquivo, programas como Bowtie e SHRiMP não reconheciam esse formato para alinhamentos em pares.
- m) conversor FastQ para Fasta – conversão de formatos, principalmente em decorrência das mudanças de codificação das qualidades.
- n) conversor Fasta para FastQ – conversão de formatos para uso geral.

Geralmente, o Blast reconhece o cabeçalho das sequências até a posição do primeiro espaço. Para análise sistemática dos resultados do alinhamento pelo Blast, é necessário assegurar uma identificação única para as sequências envolvidas.

Dessa forma, todas as montagens tiveram os *contigs* e, em alguns casos, as leituras foram renomeadas com a opção 9 “indexar sequências”.

4 RESULTADOS

4.1 CONJUNTO DE DADOS

4.1.1 Análise do viés da frequência de base observado nos relatórios do FastQC

Com base nas análises descritas na seção 3.2.1, concluímos que todas as tecnologias de sequenciamento, estudadas nesse trabalho, apresentam alguma evidência de tendências (bias), algumas mais fortes que se estendem por mais de 10 bases e outras que ficam limitadas as quatro ou cinco primeiras bases. Fizemos podas nas leituras para remover essa tendência inicial e verificar o comportamento dos montadores. De forma geral, as montagens pioraram causando um aumento significativo no número de *contigs*.

4.1.2 Experimento: Estudo das tendências das frequências de bases iniciais nas leituras

Para avaliar se esse padrão nas leituras (seção 3.2.1) podia ser uma fonte de problemas para as montagens, realizamos uma análise de ocorrência das subsequências iniciais comparadas às subsequências do restante da leitura e, ainda, das iniciais com as subsequências extraídas na quinquagésima posição da leitura. A hipótese considerada é que subsequências presentes na extremidade que não ocorrem em outras partes das leituras tendem a gerar caminhos sem possibilidade de continuidade (erro do tipo *tip*, descrito por ZERBINO, tese, 2009) ou “pontas secas” – situações que causam lacunas em que não existem leituras que cobrem a região. Por outro lado, sequências que ocorrem somente no início ou término das leituras podiam indicar artefatos da técnica de sequenciamento, como, por exemplo, adaptadores.

A Figura 12 apresenta o esquema de segmentação das leituras, em que todas as subsequências da região em vermelho constituem o conjunto A; todas as subsequências da região em verde constituem o conjunto B e todas as subsequências da região azul constituem o conjunto C. Para cada conjunto, foram identificadas a frequência de únicas e diferentes e a quantidade de subsequências

exclusivas em cada conjunto. O tamanho 15 para as subsequências foi escolhido em função da região do padrão observado na Figura 8 e foram consideradas nessa análise apenas as leituras com pelo menos 70 pb.

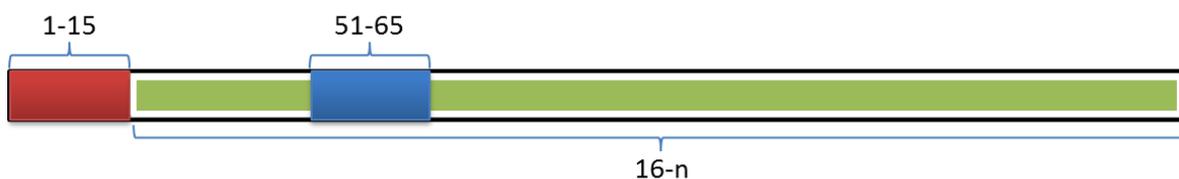


FIGURA 12 - ESQUEMA DE SEGMENTAÇÃO E IDENTIFICAÇÃO DAS REGIÕES COMPARADAS

FONTE: O autor

A Tabela 5 apresenta os dados obtidos no experimento com o arquivo de leituras da ponta R1 da *H. hiltneri* N3, de maneira que essas observações se referem à análise interna de cada conjunto. Ocorrências únicas não são esperadas e tendem a representar erros de leituras. Em teoria, em um sequenciamento livre de erros, a frequência mínima de quaisquer subsequências é igual à menor cobertura. O que permite a junção das leituras e formação dos *contigs* e do genoma são as sobreposições de subsequências. Não tem como existir sobreposição exata (ou alinhamento total) de subsequências únicas, particularmente em motores que utilizam a estratégia de *Brujn*. As presenças de subsequências únicas, especialmente nas extremidades das leituras, inviabilizam o crescimento dos *contigs*. Portanto, parece-nos lógico afirmar que observações de subsequências únicas são ruins para o processo. Pior quando a análise indica 70,2% de subsequências únicas na primeira base das leituras da *H. hiltneri* N3, no sequenciamento MiSeq. O padrão observado na Figura 8 poderia ser a causa desse fenômeno, mas descartamos ao comparar com o valor de 70,5% obtido no conjunto formado pelas subsequências iniciadas na quinquagésima e 58,2% de subsequências únicas, observadas nas demais posições das leituras.

TABELA 5 - COMPARAÇÃO DAS FREQUÊNCIAS DAS SUBSEQUÊNCIAS DAS LEITURAS *H. hiltneri* N3 (MiSeq, fragmentos provenientes do arquivo R1).

Conjunto	Todas as subsequências				N. de subsequências após o filtro ($f > 2$)			
	Únicas		Diferentes		Únicas		Diferentes	
	N.	%	N.	%	N.	%	N.	%
A (vermelho)	1104574	70,2	467803	29,8	99160	34,6	187590	65,4
B (verde)	15029475	58,2	10786276	41,8	218658	2,2	9673642	97,8
C (azul)	1119567	70,5	467396	29,5	99084	34,9	184788	65,1

FONTE: O autor

A interpretação inicial para esses dados indica um percentual de erro muito elevado ocorrendo a cada 15 pb. Resolvemos repetir o experimento, aplicando um filtro (f) de frequência mínima, e separamos as subsequências que ocorrem três ou mais vezes ($f > 2$). O filtro representou um descarte de 3,7% das subsequências. Um filtro menor (2 ou mais vezes) causou uma perda de 3,2% das subsequências e 9,4% de únicas no conjunto B. Para as montagens que possuem cobertura superior a 100x é possível aumentar o filtro e a perda de dados, mas para as montagens com menos de 50x de cobertura a perda de 5% a 10% dos dados é inviável.

Ainda para verificar se a tendência observada na Figura 8 poderia afetar as montagens foi identificado o percentual de subsequências que ocorrem em A (vermelho) e que não ocorrem em C (azul) ou em B (verde), com base nos dados filtrados. A Tabela 6 apresenta os valores obtidos.

Das 187.590 subsequências (Tabela 6) observadas em A apenas 97 (0,05%) não são localizadas no restante da leitura (região verde), entretanto 98% das subsequências de A não ocorrem na quinquagésima coluna. O zero de C em B se justifica porque C é um subconjunto de B, ou seja, C está incluído em B.

TABELA 6 - ANÁLISE DAS SUBSEQUÊNCIAS EXCLUSIVAS EM CADA CONJUNTO ($U \notin V$)

Conjunto (U)	Distintas	Conjunto (V)		
		A (vermelho)	B (verde)	C (azul)
A (vermelho)	187.590	0	97 (0,05%)	183.208 (97,66%)
B (verde)	9.673.642	9.486.149 (98,06%)	0	9.488.854 (98,08%)
C (azul)	184.788	180.406 (97,62%)	0	0

FONTE: O autor

Esses resultados indicam que as leituras Illumina precisam de tratamento prévio para redução do percentual de erro e que a região em vermelho não representa uma contaminação inesperada, uma vez que ocorrem nas demais regiões das leituras. Porém, a ausência de representação dos 98,21% de subsequências da área verde na área vermelha permite especular (de forma *in silico*) que a tecnologia da Illumina tem uma especificidade muito forte para ligação do primeiro adaptador, potencialmente um efeito colateral da transposase. Assim, permanece a dúvida se esse viés não pode causar baixa cobertura ou ausência de cobertura se a sequência do genoma apresentar regiões maiores que o tamanho das leituras.

Os tratamentos para as outras tecnologias também são necessários por apresentarem média de qualidade inferior a média obtida pelo MiSeq.

Esses experimentos foram repetidos com os dados da *A. brasilense* FP2, *B. contaminans* LTEB e *A. hydrophila*. Os resultados são apresentados a seguir.

TABELA 7 - COMPORTAMENTO DA TENDÊNCIA OBSERVADO NOS OUTROS CONJUNTOS DE SEQUENCIAMENTOS

Organismo / Filtro	Únicas A	Únicas B	Únicas C	A \neq B	B \neq A
<i>H. hiltneri</i> N3 (f>2)	99.160 34,58%	218.658 2,21%	99.084 34,90%	97 0,05%	9.486.149 98,06%
<i>A. brasilense</i> FP2 (f>1)	18.302 46,69%	2.389.899 18,69%	18.073 47,12%	3.070 14,69	10.379.699 99,83%
<i>B. contaminans</i> LTEB (f>2)	15.884 43,56%	989.035 6,47%	15.474 43,82%	157 0,76%	14.272.396 99,86%
<i>A. hydrophila</i> (f>2)	218.326 42,93%	457.038 4,36%	205.950 42,97%	151 0,05%	9.735.627 97,11%

FONTE: O autor

NOTA: Os valores (n) para a frequência mínima (f) foram determinados de forma a reduzir o número de únicas e preservar o número de leituras.

Frente a esses resultados optou-se em realizar o tratamento das leituras para correção da frequência de subsequências e redução do número das bases com baixa qualidade.

4.1.3 Estudo da frequência de k-mer

A análise de k-mer consiste em obter todas as subsequências de tamanho k do conjunto de leituras, ou do conjunto de *contigs* ou de um genoma e calcular o número de subsequências únicas, o número de subsequências diferentes e o total de ocorrências de cada subsequência. Com base nessas medidas, é possível estimar o tamanho do genoma (LI e WATERMAN, 2003).

O programa MaSuRCA faz uso do programa Jellyfish para criar e contar os k-mers e estimar o tamanho do genoma. Além disso, o MaSuRCA faz a correção das leituras utilizando o Quorum, que trabalha com base na frequência e no banco de kmers.

Entretanto, ao solicitar ao MaSuRCA, a montagem do genoma da *H. hiltneri* N3, sequenciada pelo MiSeq, observamos no arquivo de histórico (log) que o programa tinha estimado o genoma em 14.777.676 pb, enquanto o valor esperado era próximo de 5 Mpb.

Fizemos uma filtragem de alta qualidade (Q20) e a estimativa mudou para 8 Mpb.

Vale ressaltar que tentativas de estimar a quantidade de repetições e a cobertura média não apresentavam valores aceitáveis.

Nas montagens realizadas pelo MaSuRCA, é fundamental observar o tamanho estimado do genoma, um vez que essa estimativa é calculada com base na distribuição de k-mer, porque a subestimação ou superestimação do tamanho pode indicar uma eventual falha no sistema de correção das leituras com base no espectro de frequências dos k-mers.

4.2 FERRAMENTAS DESENVOLVIDAS PARA A ETAPA DE PRÉ-PROCESSAMENTO: ANÁLISE E TRATAMENTO DOS DADOS

4.2.1 jTrimmer

O programa de filtragem permitiu a extração de subconjuntos de leituras com qualidades mínimas controladas e redução da perda de dados, além do benefício da portabilidade proporcionada pela linguagem Java. A combinação de uso de janelas móveis, semelhante às utilizadas em médias móveis e no GC Skew, e tamanho

ajustável, ainda não foram observadas em nenhum outro programa do gênero. A estratégia de encadeamento dos algoritmos de filtragem, de forma semelhante ao conjunto de classes da *IO Stream* ou do padrão de projetos *Decorator*, possibilitam a ampliação de métodos e fácil integração com outras aplicações.

Este programa e a técnica mencionada tiveram o registro de software solicitado em 2014, recebendo número provisório, de modo que obteve certificado de registro definitivo em 2015 (Anexo 4).

4.2.2 Verificação do pareamento das leituras

Quando a empresa Illumina modificou o formato do cabeçalho e os montadores não causaram erros de interpretação na carga dos arquivos de leituras, levantamos a suspeita de que o cabeçalho não estava sendo considerado para identificar o pareamento das leituras. Para verificar, modificamos a identificação do arquivo R1, embaralhamos as leituras e realizamos montagens comparativas no Velvet e no Edena v3. Os dois montadores não alertaram sobre as diferenças de cabeçalhos entre os dois arquivos R1 e R2 e aceitaram os arquivos como leituras em pares. O resultado das montagens apresentavam diferenças numéricas e de constituição dos *contigs*. A partir desse momento, fizemos um programa para verificar o pareamento das leituras fornecidas nos diversos arquivos e assegurar que a ordem de ocorrência das leituras no arquivo R1/F3 correspondesse à mesma posição do par no arquivo R2/R3 e para separar as leituras órfãs (desemparelhadas) em um terceiro arquivo.

4.3 ANÁLISE DO CONTEXTO GC NAS SEQUÊNCIAS DOS GENES E DOS GENOMAS COMPLETOS

Neste trabalho, foi considerado como pressuposto que os genomas tendem a apresentar uma curva do GC Skew com maior probabilidade na forma de “V” invertido ou com menor probabilidade na forma de V., em que são esperadas raras variações na curva.

Estudamos o comportamento da formula do GC Skew nos *contigs* obtidos nas montagens e identificamos que podíamos “induzir” o comportamento da curva ao escolher a representação do contig com base na frequência de guanina e citosina.

Na Figura 13, apresentamos as quatro curvas do GC Skew, em que as linhas vermelhas representam o sentido do contig (5'-3') e, as azuis, a versão complementar reversa de cada contig (5'-3'). Em "A", os *contigs* estão na ordem e no sentido obtido pelo montador; em B a ordem foi preservada e o sentido foi escolhido na versão em que a frequência de guanina é maior que a frequência de citosina. Desta forma, as variações internas nos *contigs* são preservadas e evidenciadas, uma vez que ruído proporcionado pela aleatoriedade da distribuição do sentido é eliminado.

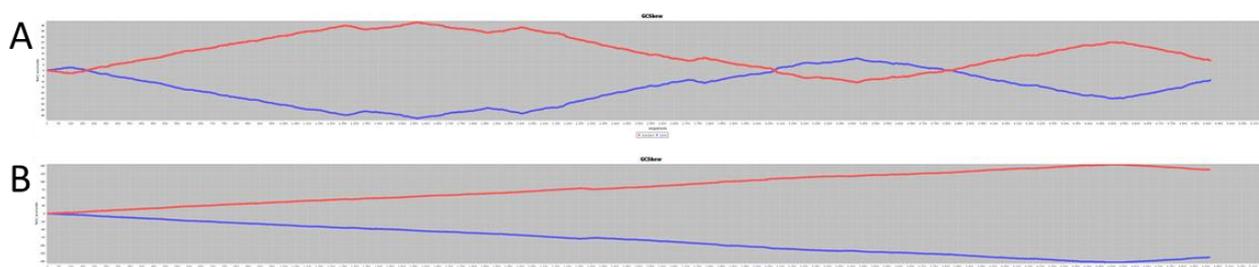


FIGURA 13 - EFEITO DA INDUÇÃO DO COMPORTAMENTO DO GC SKEW COM A PADRONIZAÇÃO DA REPRESENTAÇÃO DOS *CONTIGS* BASEADA NA RELAÇÃO DE FREQUÊNCIA G>C

FONTE: O autor

NOTA: Em (A) os *contigs* estão na ordem e sentido fornecidos pelo montador e em (B) os mesmos *contigs* aparecem na mesma ordem e com o sentido escolhidos com base na frequência de guanina e citosina.

Paralelamente, investigávamos dois padrões descritos na literatura; a compartimentalização do GC Skew (LOBRY, 1996), onde uma metade do genoma apresenta um sentido e a outra metade o sentido inverso em relação à primeira e a preferência dos genes por uma fita.

Para medir a frequência de genes por fita em cada uma das metades dos genomas, realizamos a contagens dos genes anotados nos genomas completos de procariotos depositados no NCBI. Ao combinar duas metades do genoma com as duas possibilidades de fita, identificamos quatro possibilidades para medir a preferência do gene por uma fita e a relação da preferência da fita para cada metade do genoma. As possibilidades combinatórias são: a) mais de 50% dos genes anotados nas duas metades do genoma na fita padrão, b) mais de 50% dos genes anotados na fita padrão na primeira metade do genoma e na fita complementar na

segunda metade do genoma; c) mais de 50% dos genes anotados na fita complementar na primeira metade do genoma e na fita padrão na segunda metade do genoma e; d) mais de 50% dos genes anotados nas duas metades do genoma na fita complementar.

A Tabela 8 apresenta os resultados que demonstram que 100% dos organismos apresentam a preferência de ter mais de 50% dos genes anotados e distribuídos em uma das quatro situações identificadas e que em pelo menos 65% dos casos existe uma inversão de preferência da primeira metade para a segunda metade do genoma.

TABELA 8 - PREFERÊNCIA DOS GENES PARA CADA FITA EM CADA UMA DAS METADES DOS GENOMAS COMPLETOS DE PROCARIOTOS

Reino	Padrão ¹ / Padrão ²	Padrão ¹ / Complementar ²	Complementar ¹ / Padrão ²	Complementar ¹ / Complementar ²	Total Geral
Archaea	48 (19,3%)	75 (30,1%)	76 (30,5%)	50 (20,1%)	249
Bactéria	851 (17,3%)	594 (12,1%)	2649 (53,8%)	834 (16,9%)	4928
Total Geral	899 (17,4%)	669 (12,9%)	2725 (52,6)	884 (17,1%)	5177

FONTE: O autor

NOTA: ^{1,2} – Números de organismos com frequência de genes superior a 50% observados na (¹) primeira metade do genoma ou na (²) segunda metade do genoma.

Apesar da evidência, devemos considerar que o banco de genomas completos contém cromossomos e plasmídeos, uma parte dos genomas depositados não iniciam a anotação no gene *dnaA*, existem sequências de genoma depositadas classificadas como genomas completos mas que apresentam lacunas e pode existir genomas depositados com erros de montagem.

Avaliamos o banco de dados NCBI NT para verificar o comportamento do percentual de GC no banco de nucleotídeos. Identificamos 22.552.407 sequências com 60 ou mais nucleotídeos; destes, 66,89% (15.085.626) apresentam o percentual de guanina maior do que de citosina; 32,26% (7.274.810) apresentam maior frequência de citosina do que guanina e 0,85% (191.971) apresentam o mesmo número de guanina e citosina.

Para identificar se o comportamento da distribuição era o mesmo ao longo de toda sequência, contamos e comparamos as ocorrências das bases nas duas metades de cada gene. Em 71,19% (16.055.031), os genes apresentaram a mesma relação nas duas parcelas das sequências, de maneira que, em 14,34% (3.232.977),

a ocorrência de guanina é menor do que a ocorrência de citosina, na primeira metade, e o inverso na segunda metade; e em 14,47% (3.264.399) a ocorrência de guanina é maior do que a ocorrência de citosina na primeira metade e o inverso na segunda metade.

Analisamos o comportamento da distribuição de guanina e citosina nos genes codificantes anotados nos genomas completos, depositados no banco "Bacteria" do GenBank. Dos 5.182 registros do Genbank, 2.168 estão identificados como plasmídeos e os 3.014 foram considerados cromossomos. As Tabelas 9 e 10 demonstram os valores numéricos e percentuais encontrados.

TABELA 9 - RELAÇÃO DA DISTRIBUIÇÃO DE GUANINA E CITOSINA NOS GENES CODIFICANTES DE PROCARIOTOS

Molécula	Num. genes	G=C	G>C	C>G
Todas	10255786	181753	6851653	3209268
		1,77%	66,89%	31,33%
Cromossomo	9971216	176707	6681235	3100399
		1,77%	67,09%	31,13%
Plasmídeo	284570	5046	170418	108869
		1,77%	59,94%	38,29%

FONTE: O autor

A distribuição da composição de guanina e citosina nos genes dos plasmídeos é um pouco menor do que os genes anotados nos cromossomos; porém, não muda a média geral de 67% dos genes apresentarem mais guanina que citosina.

Os dados da tabela 10 indicam que 70% dos genes apresentam uma mesma distribuição em relação à frequência de guanina e citosina ao longo do gene, praticamente os mesmos valores medidos no banco NT do NCBI.

Esses dados indicam uma tendência dos genomas apresentarem o GC Skew crescente quando o gene está na fita padrão e decrescente quando o gene está na fita complementar.

TABELA 10 - COMPARAÇÃO DA DISTRIBUIÇÃO DO CONTEÚDO GC NAS DUAS METADES DOS GENES CODIFICANTES DE PROCARIOTOS

Molécula	Num. genes	GG	GC	CG	CC
Cromossomo	9958341	5179598	1383166	1604316	1791261
	100%	52,01	13,89	16,11	17,99
Plasmídeo	284333	122744	45528	53264	62797
	100%	43,17	16,01	18,73	22,09

FONTE: O autor

NOTA: GG – maior frequência de guanina nas duas metades do gene, GC – maior frequência de guanina na primeira metade do gene e de citosina na segunda metade, CG – maior frequência de citosina na primeira metade do gene e de guanina na segunda metade e. CC – maior frequência de citosina nas duas metades do gene.

4.4 AUTOMATIZANDO A VARIAÇÃO DE PARÂMETROS NA ETAPA DE MONTAGEM

Para o Velvet, foi criado um script em Bash, batizado de “todosMer” que executava o Velvet para todos os valores de k-mers definidos no programa (Figura 14). No exemplo, os parâmetros foram definidos para os dados da *Burkholderia contaminans* LTEB após o tratamento realizado pelo jTrimmer e pela verificação de pareamento. Os arquivos bc01c_pair e bc02_pair estavam com a ordem dos pares averiguada e as leituras órfãs estavam no arquivo bc02unpair. Nesse caso, o Velvet foi executado para os valores de 21 a 171 para o parâmetro do k-mer e de dois em dois para utilizar apenas os valores ímpares.

```
#!/bin/sh
#
#
for kmer in {21..171..2}
do
  echo "processando kmer $kmer"
  outdir="/storage/dieval/burko/k$kmer"
  mkdir -p $outdir
  outlog="$outdir/frag_orig_.log"
  analis="$outdir/analysis"
  contigfile="$outdir/contigs.fa";

  /usr/local/bin/velveth $outdir $kmer -fastq -shortPaired bc01c_pair.fastq bc02c_pair.fastq \
    -fastq -short bc02unpair.fastq
  /usr/local/bin/velvetg $outdir -shortMatePaired yes -scaffolding yes -min_contig_lgth 100
done
```

FIGURA 14 - CÓDIGO-FONTE DO “todosMer.sh”

FONTE: O autor

NOTA: Script criado para executar o Velvet alternando o valor do k-mer

4.5 G-FINISHER

O G-Finisher é o principal produto deste trabalho e representa uma nova estratégia para refinar e finalizar a montagem do genoma bacteriano. Contudo, o emprego do G-Finisher requer que o usuário tenha realizado diversas montagens previamente e que tenha um genoma de referência disponível. Dessa forma, os principais parâmetros de G-Finisher são uma montagem do genoma de interesse (montagem alvo), um conjunto de *montagens alternativas* do genoma de interesse, um genoma de referência filogeneticamente próximo ou espécie-específicas e um caminho computacional para salvar os resultados (Figura 15).

O programa começa determinando a posição e o sentido da fita para cada contig, com base no mapeamento dos *contigs* ao genoma de referência (Figura 15-[1]), gerando a montagem “A”. Na segunda tarefa (Figura 15-[2]), o jFGap busca nas montagens alternativas os *contigs* que apresentem regiões que alinhem nas bordas da lacuna e complementem a lacuna, produzindo a montagem “B”. Na terceira tarefa (Figura 15-[3]), o Fuzzy GC Skew é calculado a partir das sequências da montagem B, de modo que os *contigs* são quebrados nos picos de máximos e mínimos, originando a montagem C.

Na quarta etapa (Figura 15-[4]), os *contigs* são novamente ordenados e é realizada uma nova tentativa de fechar as lacunas, resultando na montagem “D”. Na quinta etapa (Figura 15-[5]), a montagem “D” é comparada com a montagem “B” e as lacunas são criadas pela quebra dos *contigs* e não tratadas. No passo anterior, são recuperadas, fornecendo a montagem “E”. Os *contigs* da montagem “E” são novamente submetidos ao tratamento do jFGap com os parâmetros modificados e produzindo a montagem final “F” (Figura 15-[6]). Os relatórios de acompanhamento e dados dos *contigs* são salvos em cada etapa do processo, incluindo os gráficos do GC Skew e dos Dotplots. Na última tarefa, os dados são preparados e o arquivo do roteiro (script) é criado para a execução do QUAST (Figura 15-[7]).

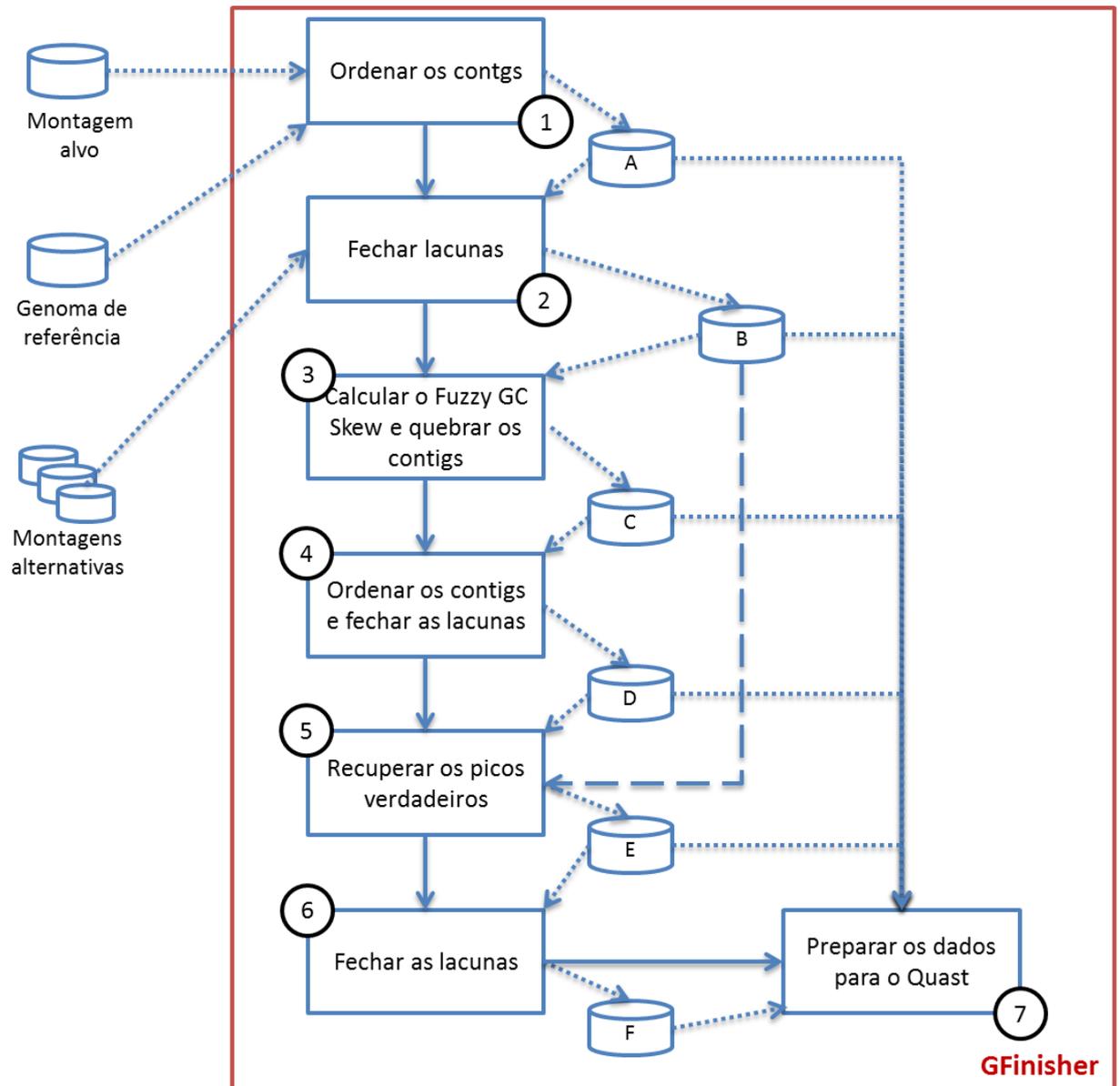


FIGURA 15 - FLUXOGRAMA DO G-FINISHER

FONTE: O autor

NOTA: Fluxograma do G-Finisher (caixa vermelha). Do lado esquerdo estão representados os conjuntos de dados necessários para execução do G-Finisher. As caixas identificadas pelos números de 1 a 7 representam as tarefas e as montagens intermediárias produzidas. Estão identificadas pelas letras de A-E e a montagem final pela letra F. Linhas sólidas indicam o fluxo do processamento e linhas tracejadas o fluxo das montagens.

4.5.1 Aperfeiçoamento do *jContigsort*

O conceito do *jContigsort* foi desenvolvido em 2011, mas durante o emprego e a avaliação dos resultados foram detectadas situações em que o algoritmo não

apresentava bons resultados, especialmente em *contigs* pequenos e constituídos de sequências repetidas. Então, a técnica de agrupamento foi completamente refeita, de maneira que, além de ordenar os *contigs* com base na interseção de k-mers, ele começou a gerar informações para predizer os parâmetros do FGap. Dessa forma, os parâmetros que limitam o tamanho da inserção, o máximo de remoção e o tamanho das pontas dos *contigs* foram automatizados; todavia, ainda existem máximos e mínimos para cada parâmetro estabelecido internamente e que podem ser modificados no arquivo de configurações do G-Finisher.

O algoritmo de agrupamento inicia ordenando todas as posições do genoma de referência que possuem os mesmos k-mers dos *contigs*. Para criar os grupos, as posições são distribuídas de forma que a diferença entre os valores máximos e mínimos em cada grupo não supere o tamanho do contig. Em no máximo dez iterações os elementos dos grupos são rearranjados e os centroides recalculados. A cada iteração, os grupos são divididos quando apresentarem distâncias entre os pontos extremos maiores que o tamanho do contig e são unidos quando a distância entre os centroides forem menores que o tamanho do contig.

Essa estratégia reduziu o tempo de processamento do algoritmo de agrupamento original, que era uma implementação mais fidedigna do algoritmo de k-means; porém, tinha problemas em pré-definir o número de grupos e precisava de muitas iterações para convergir para uma distribuição razoável.

O grupo com maior densidade de pontos é escolhido para ancorar o contig, entretanto o programa de análise dos resultados do *jContigsort* agora emite alertas quando partes distintas dos *contigs* alinham em diferentes posições e quando as mesmas regiões do contig apresentam um mapeamento em regiões distintas do genoma de referência.

4.5.2 Desenvolvimento do jFGAP

O algoritmo do FGap é perfeito para o cenário em que lacunas são criadas em consequência de regiões de repetição ou nos casos em que as pontas dos *contigs* apresentam montagens espúrias. Entretanto, nossas análises para determinar a ordem dos *contigs* ou o arranjo genômico com base na combinação dos *contigs*, que podem ser “emendados” pelo FGap, não apresentaram bons resultados. A estratégia de força bruta para testar todas as combinações dos *contigs*

foi realizada na montagem da *H. hiltneri* N3 e da *A. brasilense* FP2 com insucesso. Nos dois organismos existiam lacunas que não eram totalmente cobertas por algum contig e/ou a combinação de operações do FGap resultava em um mesmo número de *contigs* com significativa diferença de bases entre as montagens. Ou seja, o uso indiscriminado do FGap pode causar o “inchaço” ou “compressão” dos genomas. O “inchaço” ocorre com mais frequência quando o limite de inserção é alto (gap positivo) e a “compressão” ocorre quando os limites de remoção (gap negativo) são muito permissivos.

A técnica apresenta, contudo, resultados significativamente melhores quando a ordem dos *contigs* está estabelecida, o que determinou sua inclusão nesta proposta de solução. Entretanto, o FGap foi escrito em Matlab e dependia do NCBI Blast. Optamos, assim, em reescrever o FGap em Java e avaliar a necessidade da dependência do Blast. A reescrita permitiu aprofundar os algoritmos e melhorar os controles de pontuação da combinação de alinhamentos resultantes do Blast, bem como aplicar a estratégia quando as lacunas não são apresentadas na forma de scaffolds. O jFGap apresenta maior velocidade, menor consumo de memória e processamento e pode ser executado em qualquer Sistema Operacional que tenha a máquina virtual Java. Os resultados não são idênticos, porque fomos mais restritivos no tratamento de lacunas próximas, quando a distância entre as lacunas é inferior ao tamanho da ponta do contig combinada com o mínimo de identidade.

No ambiente gráfico do G-Finisher, é possível executar o jFGap sem realizar os demais passos do protocolo aqui desenvolvido.

As tentativas de remoção da dependência do Blast no jFGap implicaram em perda significativa nos resultados, de forma que optamos em mantê-lo por enquanto. Os algoritmos de alinhamento testados consomem, de forma geral, muita memória e/ou perdem muito em sensibilidade quando comparados aos resultados do Blast. A estratégia que apresentou o melhor resultado foi a dos algoritmos baseados em SAIS (acrônimo em inglês para *suffix array induced order*), onde dado uma montagem alvo com tamanho m (ou mesmo considerando apenas as pontas dos *contigs*) e o conjunto de montagens alternativas n , o alinhamento exige $(m+n) \times 8$ bytes de memória e pode ser construído em tempo linear. Trata-se de uma quantidade de memória muito inferior ao teórico $m \times n \times (4 \text{ ou } 8)$ do algoritmo de alinhamento local. No entanto, perde-se a sensibilidade local dos alinhamentos, de maneira que soluções alternativas para esse problema são propostas pelo LAST

(KIEŁBASA et al, 2011) e KMACS (LEIMEISTER e MORGENSTERN, 2014). Ainda assim, não conseguimos aprofundar essa linha de pesquisa, ficando como proposta para trabalhos futuros.

4.5.3 Fuzzy GC Skew

A análise automática do GC Skew apresentava imprecisão na identificação da localização dos picos de máximos e mínimos proporcionais ao tamanho da janela utilizada. No início dos estudos adotamos janelas grandes de 10k, 15k, 30k e 70k, o que aumentava o problema da imprecisão.

Para resolver esse problema, o professor doutor Roberto Tadeu Raittz propôs e desenvolveu o modelo conceitual e um protótipo para calcular o *Fuzzy GC Skew*. Partimos do protótipo, aperfeiçoamos os algoritmos a ponto de podermos calcular janelas com tamanhos de 5k a 75k em poucos minutos.

A função *Fuzzy GC Skew* é definida na equação 6 e os passos dos cálculos são demonstrados pelas equações de 1 a 5.

Dado uma sequência S de DNA com tamanho n :

$$S = \{S_{i=1}^n | S_i \in ['a', 'c', 'g', 't']\} \quad \text{eq.1}$$

e $2d+1$ o tamanho da janela móvel, onde “ d ” é o valor definido pelo pesquisador da extensão da vizinhança a montante e a jusante de um resíduo de nucleotídeo específico.

A função triangular é dada por:

$$T_j = 1 - \frac{d}{|(i-j)|}, \quad \max(i-d, 1) \leq j \leq \min(i+d, n) \quad \text{eq.2}$$

Definimos as funções G e C como:

$$G = \begin{cases} G_j = 1, & \text{se } S_j = 'g' \\ G_j = 0, & \text{otherwise} \end{cases} \quad \text{eq.3}$$

$$C = \begin{cases} C_j = 1, & \text{se } S_j = 'c' \\ C_j = 0, & \text{otherwise} \end{cases} \quad \text{eq.4}$$

A função Fuzzy GC Skew U proposta é dada pela equação 5:

$$U_i = \left| \frac{\left(\sum_{j=\max(i-d,1)}^{\min(i+d,n)} (G_j \cdot T_j) - \sum_{j=\max(i-d,1)}^{\min(i+d,n)} (C_j \cdot T_j) \right)}{\left(\sum_{j=\max(i-d,1)}^{\min(i+d,n)} \text{Max}(G_j, C_j) \cdot T_j \right)} \right| \quad \text{eq.5}$$

E a função acumulada do Fuzzy GC Skew é dada pela equação 6:

$$A_i = \sum_{k=1}^i U_k \quad \text{eq.6}$$

A função *Fuzzy GC Skew* (U, eq.5) é uma média móvel semelhante à GC Skew, onde cada elemento de U corresponde a uma base específica da sequência original S (Eq.1). A mesma propriedade pode ser vista na função acumulada da Fuzzy GC Skew (Eq.6). Observa-se, no entanto, que a propriedade não é verificada numa média móvel convencional, onde o número de elementos calculados é menor do que o tamanho da matriz de origem.

Neste trabalho, denominamos *FGC-Skew* para os valores obtidos da função acumulada *Fuzzy GC Skew* e adotamos *(F)GC-Skew* para identificar as duas equações, respectivamente, *Fuzzy GC Skew* e *GC Skew*.

4.5.4 Desenvolvimento da função “Finder”

No G-Finisher, desenvolvemos a função denominada “finder”, para identificar os picos de máximos e mínimos locais nas curvas do GC Skew e do Fuzzy GC Skew.

Os picos de variação nas curvas (F)GC Skew ocorrem nas posições i , em que o sinal de $S(i)$ é diferente do sinal de $S(i+1)$ e o sinal é calculado conforme descrito na eq.7 e os picos identificados para todo P_i diferente de 0.

Seja S a o sinal do diferencial de A

$$S(i) = \text{ sinal}(A_i - A_{i-1})_{i=2}^n \quad \text{eq.7}$$

Os índices dos pontos de picos são dados pelos índices dos valores verdadeiros da variação de sinal de S :

$$P_i = (S_{i+1} <> S_i)_{i=1}^{N-1} \quad \text{eq. 8}$$

de modo que $P_i=1$, quando i for um ponto de pico e zero nos demais casos. Ao final, os pontos que apresentam distância inferior a 1000 pb de outro ponto ou dos limites dos *contigs* são removidos da lista.

4.5.5 Considerações do uso do Fuzzy GC Skew no processo de remontagem

O contig (ctg7180000001875) de *Aeromonas hydrophila* SSU com 1.100 kpb foi escolhido para demonstrar as diferenças entre as curvas do GC Skew e do Fuzzy GC Skew. Na Figura 16, combinamos o Dotplot com os gráficos do GC Skew e Fuzzy GC Skew. O alinhamento demonstrado pelo Dotplot indica que o contig em questão apresenta três grupos de alinhamentos (Figuras 16 A, B e C).

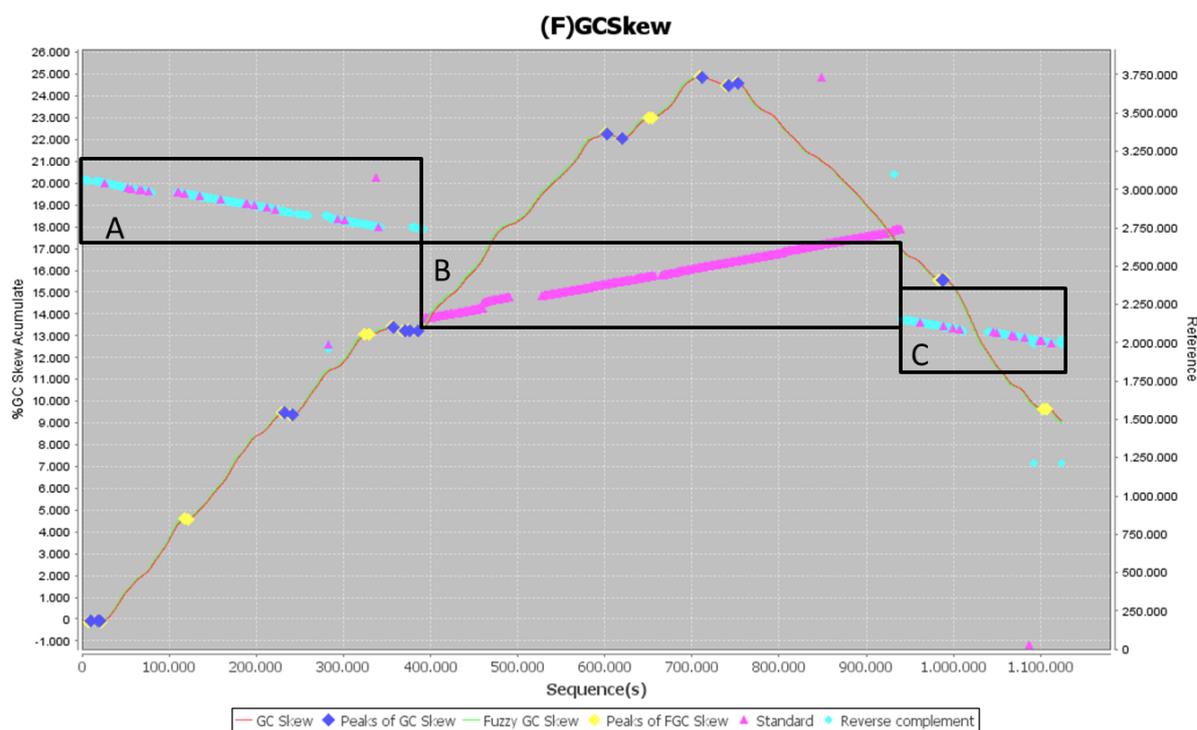


FIGURA 16 - EXEMPLO DA DIFERENÇA DE SENSIBILIDADE NA DETECÇÃO DOS PONTOS CRÍTICOS NAS CURVAS GC-SKEW E FUZZY-GC-SKEW

FONTE: O autor

NOTA: As curvas foram calculadas com base na sequência do contig ctg7180000001875 de 1,1mb da montagem de *Aeromonas hydrophila*. O gráfico do GC Skew foi obtido pela equação clássica (linha vermelha) ou pelo método fuzzy (linha verde), utilizando uma janela de 10 kpb para o cálculo. Os pontos críticos na curva GC Skew clássica são identificados pelos losangos azuis e na versão fuzzy pelos losangos em amarelo. Regiões de divergência do gráfico Dotplot (linhas em rosa e ciano) são uma das causas das variações na curva GC Skew.

Os grupos de alinhamentos A e C estão no sentido complementar reverso em relação à sequência do genoma de referência e o grupo intermediário B apresenta o

mesmo sentido da referência. As marcações no formato de losango (azul e amarelo) representam os pontos indicados pelo algoritmo “finder”. A montagem do contig é melhorada quando o mesmo é quebrado nos pontos indicados e os novos fragmentos são realinhados com base na referência.

A identificação desses pontos é feita com base na curva GC Skew; conseqüentemente, a estratégia pode ser aplicada na ausência de referência. Para reduzir o efeito da fragmentação e aleatoriedade do sentido dos *contigs* resultante dos montadores, os mesmos podem ser organizados no sentido em que a frequência de guanina seja maior que a frequência de citosina (G>C).

Para demonstrar como os erros de montagem podem ser identificados a partir da variação do GC Skew, preservando a ordem e apenas modificando o sentido, aplicamos a estratégia de modificar o sentido dos *contigs* com base na frequência (G>C) e realizar o tratamento de calcular o GC Skew e identificar os picos de máximos e mínimos na montagem do genoma *Aeromonas hydrophila* SSU feita pelo MaSuRCA (Figura 17). Essa estratégia não resolve o problema de ordem, mas acentua o efeito das variações espúrias observáveis na curva do CG SKew e induz o sentido crescente para o cálculo do GC Skew.

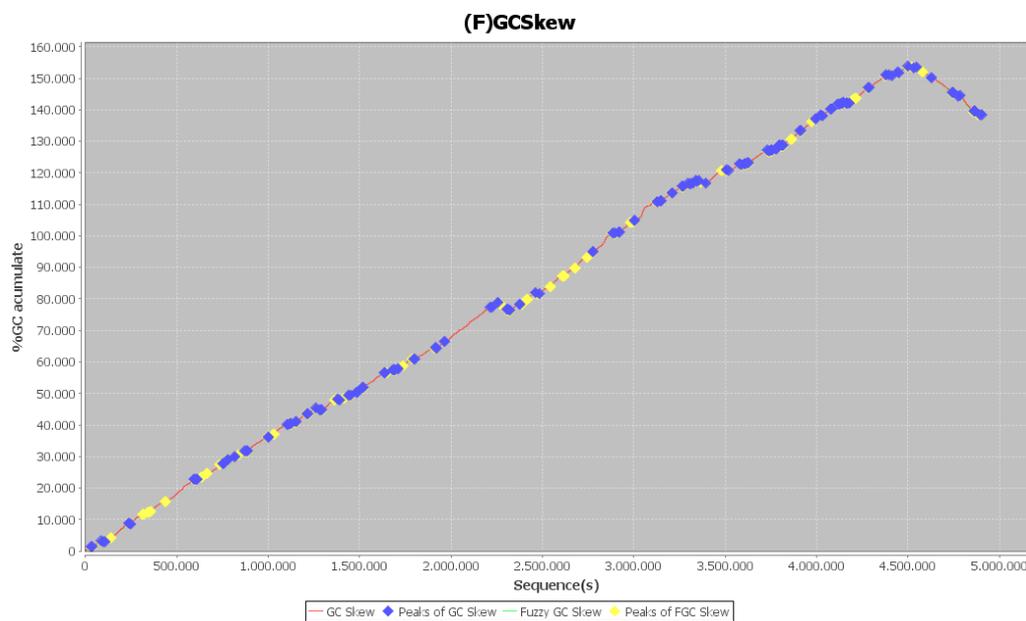


FIGURA 17 - APLICAÇÃO DO “FINDER” NAS CURVAS (F)GC SKEWS APÓS ESCOLHA DO SENTIDO E PRESERVANDO A ORDEM ORIGINAL DOS *CONTIGS* OBTIDOS NA MONTAGEM DE *Aeromonas hydrophila* PELO MASURCA

FONTE: O autor

A diferença do número de ocorrências de marcas amarelas para o número de marcas azuis é justificada pela diferença de precisão obtida pelo Fuzzy GC Skew acumulada em relação ao GC Skew acumulado. A Figura 18 é uma captura da tela do G-Finisher ampliando a imagem da Figura 17 na região próxima de 4.500.00 pb, onde se destacam quatro pontos amarelos (FGC Skew) que não são identificados pelos azuis (GC Skew) e que o efeito de suavização da curva GC Skew é maior que a Fuzzy GC Skew.

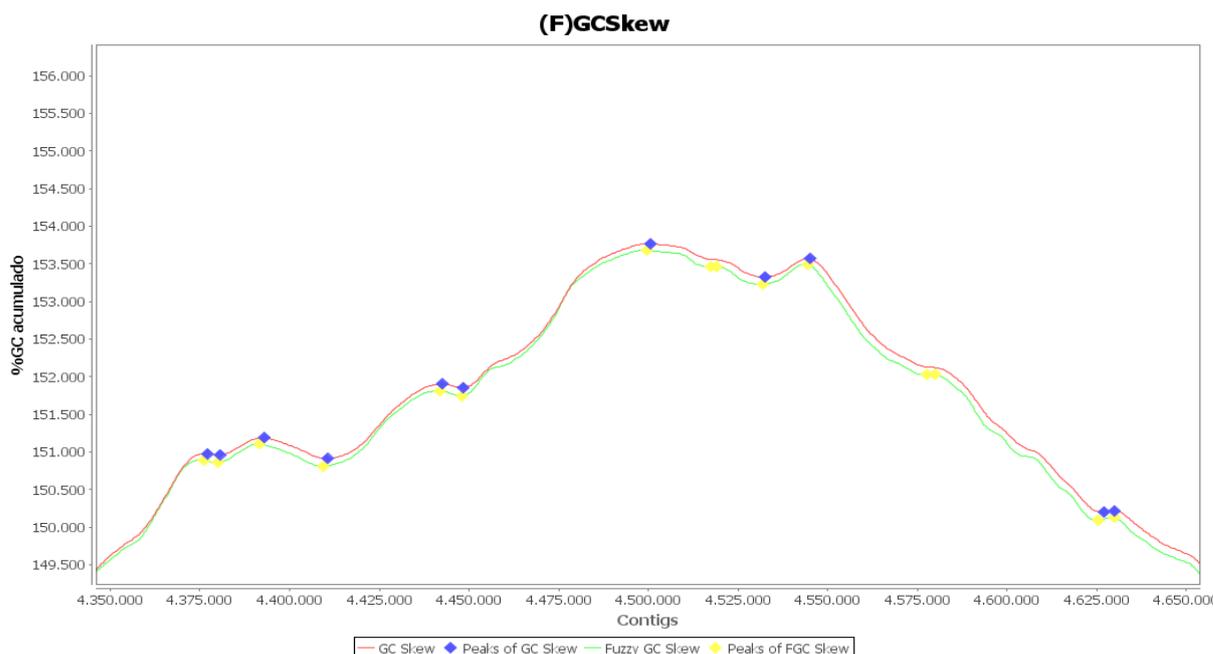


FIGURA 18 - AMPLIAÇÃO DA FIGURA 17 NA REGIÃO DE 4.350.000 pb E 4.650.000 pb QUE DEMONSTRA AS DIFERENÇAS DE SENSIBILIDADES DOS MÉTODOS (F)GC SKEW E OS PONTOS CRÍTICOS IDENTIFICADOS PELO "FINDER" PARA QUEBRA DOS CONTIGS

FONTE: O autor

4.6 VALIDAÇÃO DA ESTRATÉGIA DO G-FINISHER

Os resultados de validação do G-Finisher revelaram melhoras significativas em todos os casos testados.

A Figura 19 apresenta a redução da média do número de *contigs* de 172,95 para 23,46, isso representa uma taxa de 86% de redução. Os relatórios do QUASt ainda indicam o aumento médio do valor de N50 de 123.584,12 para 654.320,94, do valor de N75 de 62.033,91 para 30.7613,61 e da redução média dos valores de L50

de 28,84 para 4,37 e dos valores de L75 de 58,13 para 8,44. Os valores individuais de todas as montagens estão relacionados na Tabela 19.

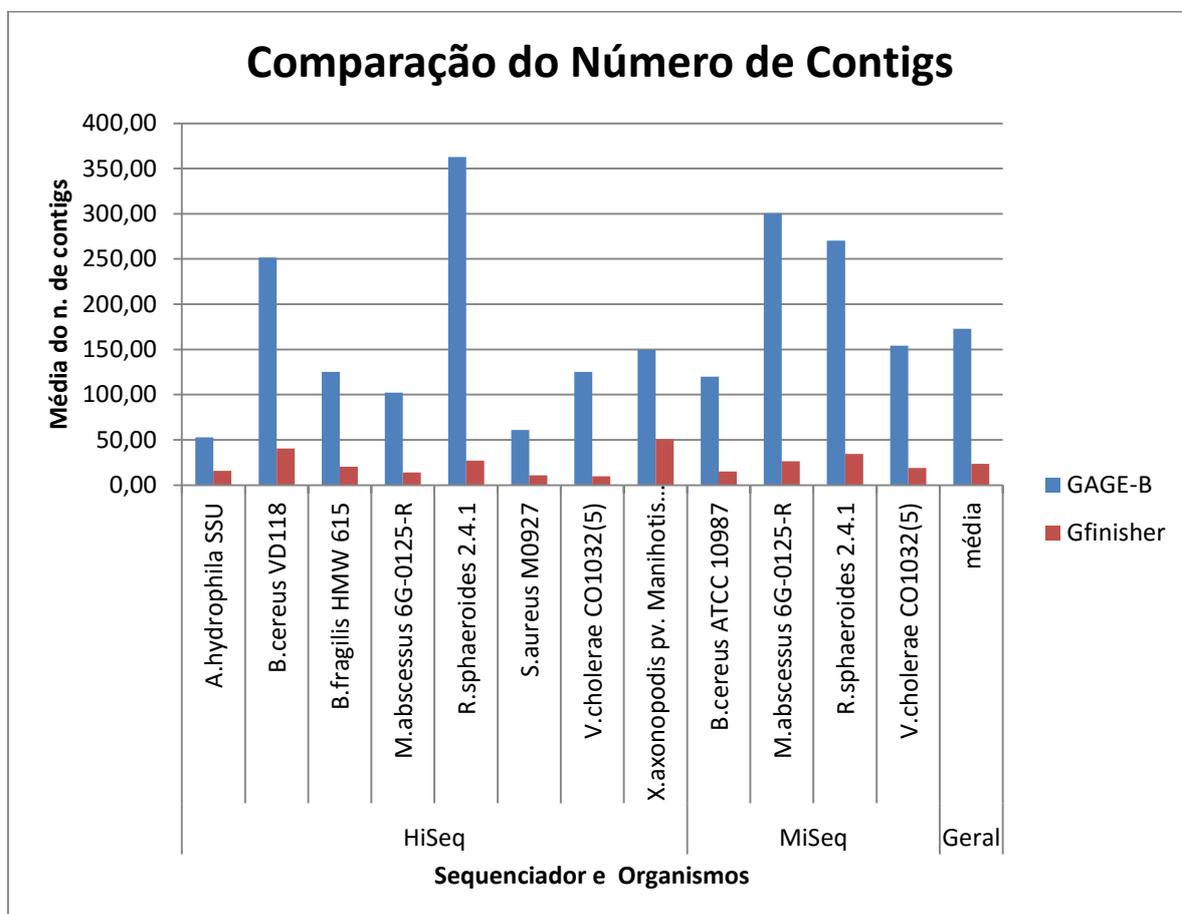


FIGURA 19 - COMPARAÇÃO DAS MÉDIAS DOS NÚMEROS DE *CONTIGS* DAS QUINZE MONTAGENS POR ORGANISMO ENTRE OS RESULTADOS DO GAGE-B (AZUL) E DO G-FINISHER (VERMELHO)

FONTE: O autor

A Tabela 11 apresenta a comparação da média do número de *contigs* por montador com a média do número de *contigs* após o tratado pelo G-Finisher. Na primeira coluna estão relacionados os montadores e as sequências fornecidas pelos mesmos na forma de *contigs* (ctg) ou scaffolds (scf). Na segunda coluna é apresentada a média do número de *contigs*/scaffolds obtidos pelo GAGE-B, seguido por três colunas contendo respectivamente as médias dos números de *contigs* obtidas pelo G-Finisher nas etapas 2 (B), 5 (E) e 6 (F). As colunas identificadas por B2, E2 e F2 representam os percentuais de redução dos valores correspondentes às médias descritas nas colunas B, E e F em relação a média do GAGE-B. A coluna

identificada por “Redução média (E,B)” apresenta a diferença entre as taxas de redução indicadas nas colunas E2 e B2 ou o ganho médio obtidos com a quebra dos *contigs* na etapa 3 do G-Finisher. E na última coluna “Redução total (F,B)” se vê o ganho global obtido pela quebra dos *contigs* e a ampliação da borda considerada no jFGap na etapa 6.

TABELA 11 MÉDIA DO NÚMERO DE *CONTIGS* POR MONTADOR E TAXA DE REDUÇÃO OBTIDAS PELO G-FINISHER.

Montador	Média do n. de <i>contigs</i>				Taxa de redução no n. de			Redução média (E,B)	Redução total (F,B)
	GAGE-B	Montagens alternativas G-Finisher			contigs do G-Finisher em relação ao GAGE (%)				
		B	E	F	B2	E2	F2		
ABySS ctg	176,17	40,33	41,67	27,67	77,11	76,35	84,30	-0,76	7,19
ABySS scf	164,00	42,75	44,67	28,67	73,93	72,76	82,52	-1,17	8,59
CABOG ctg	219,08	26,33	26,00	18,00	87,98	88,13	91,78	0,15	3,80
CABOG scf	187,25	26,42	26,50	18,33	85,89	85,85	90,21	-0,04	4,32
MaSuRCA ctg	102,75	43,33	37,00	15,83	57,83	63,99	84,59	6,16	<u>26,76</u>
MaSuRCA scf	99,50	43,33	37,00	15,83	56,45	62,81	84,09	6,37	<u>27,64</u>
MIRA ctg	215,58	69,83	62,91	40,58	67,61	70,82	81,18	3,21	13,57
SGA ctg	344,92	31,25	36,29	25,33	90,94	89,48	92,66	-1,46	1,72
SGA scf	305,50	26,83	27,83	21,27	91,22	90,89	93,04	-0,33	1,82
SOAPdenovo2 ctg	163,42	32,25	29,64	24,17	80,27	81,86	85,21	1,60	4,95
SOAPdenovo2 scf	130,75	32,25	29,64	24,17	75,33	77,33	81,52	2,00	6,18
SPAdes ctg	91,67	29,08	28,82	20,64	68,27	68,56	77,49	0,29	9,21
SPAdes scf	79,75	30,08	28,67	22,75	62,28	64,05	71,47	1,78	9,20
Velvet ctg	201,00	35,75	35,70	24,08	82,21	82,24	88,02	0,02	5,80
Velvet scf	112,92	35,75	35,70	24,08	68,34	68,38	78,67	0,04	10,33
Average	172,95	36,37	35,20	23,43	78,97	79,65	86,45	0,68	7,48

FONTE: O autor

NOTA: ctg identificam às montagens que forneceram *contigs* e scf as montagens que produziram scaffolds. As colunas B, E e F representam respectivamente a média do número de *contigs* obtidos nas montagens de cada etapas do GFinisher (Figura 15), B2, E2 e F2 são taxas de redução dos números das colunas B, E e F em relação ao número de *contigs* do GAGE-B.

As taxas de 26,76 e 27,64 observadas no MaSuRCA, indicam que o montador tem produzido *contigs* com maior número de situações que causam a variação na

curva do GC SKew e, conseqüentemente, as montagens possuem mais *contigs* com possíveis erros de montagem.

4.7 REDUÇÃO DOS ERROS DE MONTAGEM PELA PRESERVAÇÃO DO CONTEXTO DO GC SKEW

Em *contigs* com erros de montagem existem variações espúrias no GC Skew. Os dados coletados e observados nos experimentos do G-Finisher indicam que se os montadores preservarem o sinal da equação do GC Skew podem reduzir em até 50% os erros de montagem causados por regiões de repetição.

A Figura 20 demonstra o dilema do caminho, descrito por Pevzner (2001), em que duas regiões distintas do genoma compartilham uma região de alta similaridade e, portanto, uma região de repetição (retângulo vermelho). A região repetida (R) apresenta quatro regiões adjacentes, sendo que as regiões A e C antecedem R e as regiões B e D estão localizadas após a região R. Dessa forma, o montador pode optar em montar ARB, ARD, CRB ou CRD. Entre as quatro combinações possíveis, existem 50% de chance de optar pelo caminho correto.

Propomos como solução para o dilema do caminho a análise do contexto GC nos segmentos A, B, C e D e a preservação da relação $G > C$ ou $C > G$ pelo montador no momento da escolha. Dessa forma, o dilema do caminho deixa de existir nessas condições. Ao reduzir o efeito do dilema do caminho, as variações no GC Skew reduzirão, de modo que a frequência de erros nas montagens dos *contigs* reduzirão na mesma proporção.

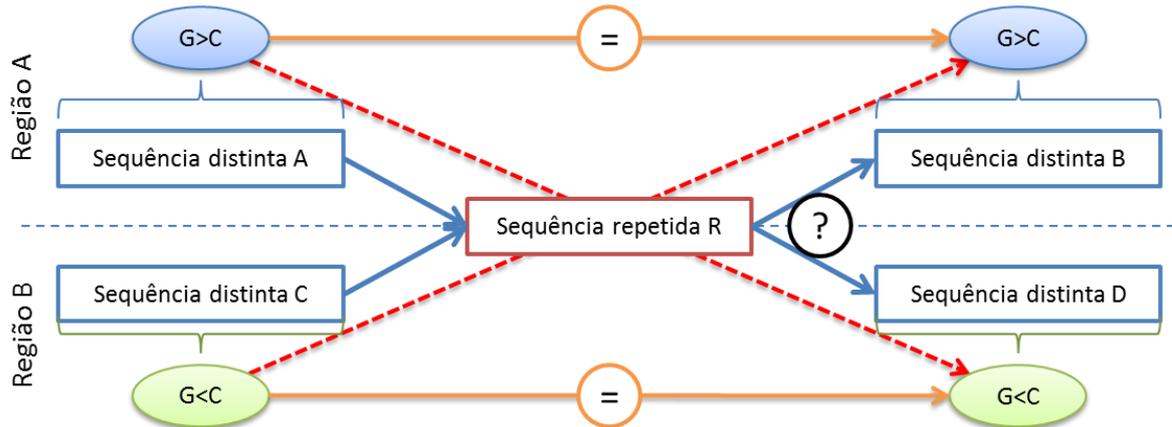


FIGURA 20 - REDUÇÃO DE ERRO NA ESCOLHA PELOS MONTADORES NA OCORRÊNCIA DO DILEMA DO CAMINHO, ATRAVÉS DA PRESERVAÇÃO DO CONTEXTO GC

FONTE: O autor

4.8 ESTUDOS DE CASOS

4.8.1 *Herbaspirillum hiltneri* N3

As leituras obtidas na plataforma SOLiD foram tratadas pelo jTrimmer para garantir qualidade mínima de Q17 por base e tamanho mínimo de 25 pb. As leituras obtidas pelo MiSeq foram tratadas para garantir o pareamento nos arquivos R1 e R2. O efeito do jTrimmer nas leituras SOLiD estão apresentados na Figura 21.

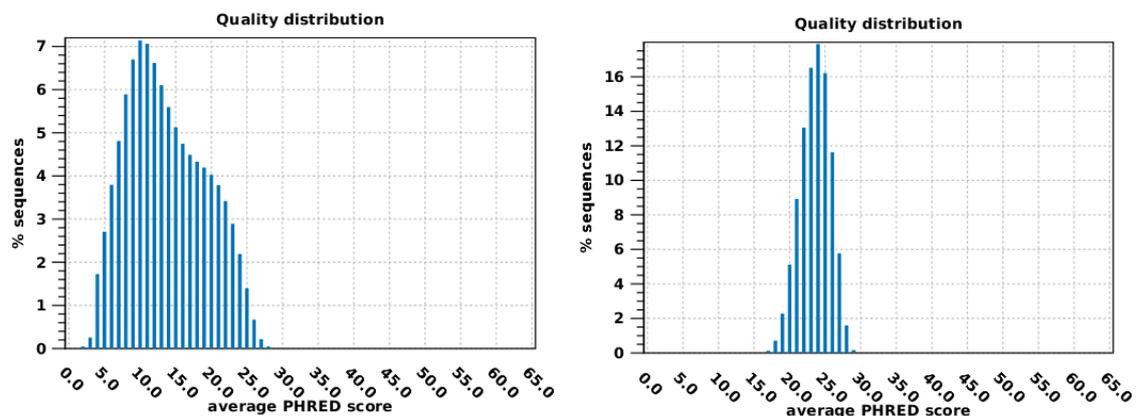


FIGURA 21 - CAPTURA DAS TELAS DO FASTQC DA ANÁLISE DA QUALIDADE DAS LEITURAS SOLiD ORIGINAIS E DEPOIS DO TRATAMENTO

FONTE: O autor

Após os tratamentos, as leituras foram submetidas aos montadores Velvet, MaSuRCA e CLC Genomics Workbench 6.5. O MaSuRCA e o CLC começaram a

ser usados no final de 2013, quase um ano após o início deste trabalho de doutorado e quase um ano após o início dos trabalhos de montagens da *H. hiltneri* N3 e *A. brasilense* FP2. As montagens foram realizadas com todos os conjuntos de dados em separado, por sequenciamento e combinados.

A Tabela 12 apresenta a visão geral das montagens obtidas. Destaca o efeito do QuorUM, que realiza a correção das leituras com base frequência de kmer, com o aumento do número de *contigs*, porém com redução do tamanho total da montagem e aumento no tamanho dos *contigs*.

TABELA 12-MONTAGENS OBTIDAS COM O SEQUENCIAMENTO DA *H. hiltneri* N3

Montador	Sequenciamento	N. <i>contigs</i>	Maior <i>contig</i>	N50	L50	Tam total
CLC Genomics Workbench 6.5	MiSeq	300	462.129	166.587	10	5.041.981
	SOLiD fragmento	14.749	5.194	240	4055	3.333.607
	SOLiD mate-pair	18.613	5.404	239	5723	4.181.447
	Misto	266	2.168.931	1.042.213	3	5.045.406
Velvet	MiSeq – k19	27.328	478	129	11.112	3.619.689
	MiSeq – k31	1347	734	180	25.092	8.328.434
	SOLiD – k31	6323	709	145	2356	938.395
	SOLiD – k19	102	142	107	48	11.097
Velvet+QuorUM	MiSeq – k31	4378	14615	2672	589	5.046.891
MaSuRCA	MiSeq	175	630.166	281.048	6	5.100.906

FONTE: O autor

Quando a *H. hiltneri* N3 estava sendo montada, o G-Finisher ainda não existia, de modo que os dotplots iniciais foram feitos no MUMmer. A busca de organismos semelhantes com base no gene rRNA 16S reportava o *Herbaspirillum lusitanum* P6-12 e *Herbaspirillum seropedicae* SmR1. Destacamos que o QUASt ainda não estava disponível.

Enquanto não tínhamos o QUASt, utilizamos a ferramenta desenvolvida por nós, denominada *jAnalyzer* para levantar os principais indicadores para comparar as montagens. Durante o processo de montagem, para determinar a ordem dos *contigs* e estender os *contigs*, eram necessário extrair as subsequências das pontas dos *contigs*, alinhar as pontas com as leituras e identificar onde o par complementar se

alinhas nas montagens. Essas operações levaram ao desenvolvimento da ferramenta DnaTools. Em fevereiro de 2014, iniciamos as tentativas de quebrar os *contigs* e reorganizá-los com base no GC-Skew. A Figura 22 apresenta o GC Skew da melhor montagem de *H. hiltneri* N3 em 02/04/2014.

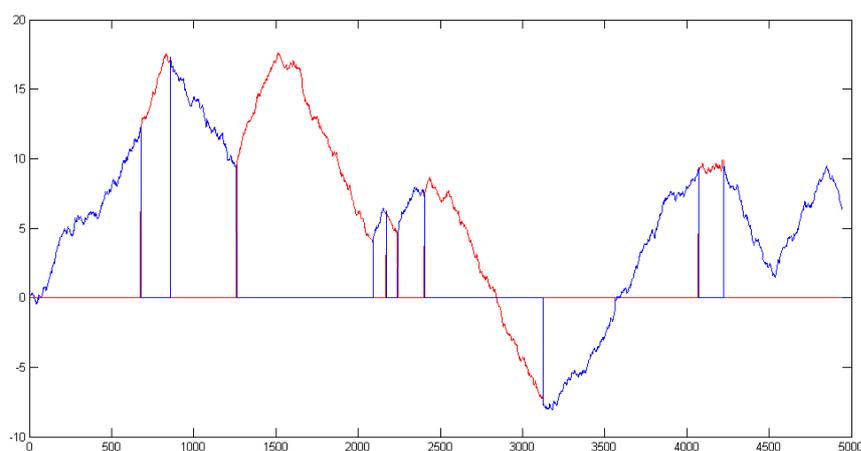


FIGURA 22 - GC SKEW DA MONTAGEM DO GENOMA DA *Herbaspirillum hiltneri* N3 EM 02/04/2014

FONTE: O autor

A Figura 23 mostra a montagem de 03/04/2014, após a quebra dos *contigs* com base no GC Skew, o alinhamento dos *contigs* com a referência *H. lusitanum* P6-12 e o fechamento das lacunas com o FGap.

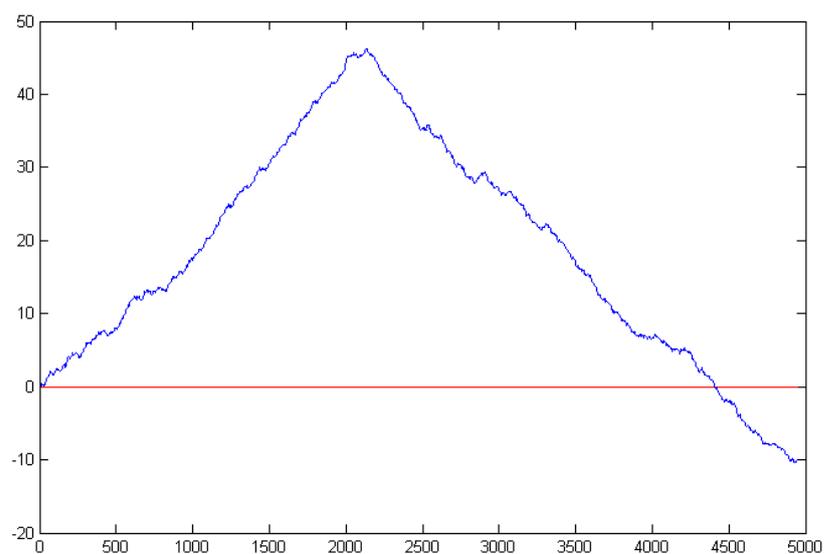


FIGURA 23 - GC SKEW DA MONTAGEM DO GENOMA DA *Herbaspirillum hiltneri* N3 EM 03/04/2014

FONTE: O autor

Em abril de 2014, tivemos a primeira montagem completa da *H. hiltneri* N3. Entretanto, ao alinhar as leituras na montagem e analisar a distância entre os pares de leituras, estes indicavam que mais de 20% dos pares de leituras estavam quebrados. Leituras quebradas ocorrem quando apenas uma das leituras do par é alinhável à montagem ou quando o número de bases entre os alinhamentos é superior ao valor estimado. No MiSeq, esse limite era próximo de 900 pb e no SOLiD de até 10.000 pb.

Ao quebrar a montagem e remontá-la com base no GC Skew, em 16/06/2014 obtivemos uma montagem que os pares de leituras alinhavam em torno de 94%. Mas a anotação automática indicava ausência do tRNA-Trp (triptofano). As leituras foram alinhadas na montagem com alta restrição e novos *contigs* foram montados com as leituras que não alinharam. Nessa nova montagem, não foram obtidos *contigs* maiores que 1.000 pb, mas o tRNAScan localizava a sequência do tRNA-Trp em alguns desses *contigs*.

Ao comparar as anotações dos genes da *H. hiltneri* N3 com a *Herbaspirillum seropedicae* SmR1, identificamos que a SmR1 tinha duas cópias do gene *tufB* (fator de alongamento da tradução EF-Tu 2) enquanto que na montagem da N3 ocorria apenas um. Entre as cópias desse gene, ocorre uma grande quantidade de genes ribossomais que também possui várias cópias no genoma. Dessa forma, o procedimento de alinhar as leituras não revelava erro na montagem. Após a correção dessa região, a análise dos pares de leituras apresentou um índice de 97,9%.

A estratégia desenvolvida para o G-Finisher permitiu a montagem e o fechamento do genoma de *Herbaspirillum hiltneri* N3. Destaca-se que a montagem não avançava após a obtenção de 11 *contigs* e ao quebrá-los nos pontos que correspondiam aos picos críticos, reorganizá-los com base no genoma de referência e remontar com o FGap, obtivemos a montagem do genoma completo.

A sequência completa do genoma de *Herbaspirillum hiltneri* N3 tem um cromossomo circular de 4.965.474 pb, com um percentual de GC de 61,84%. A anotação do genoma usando RAST e nossa plataforma (in-house) SILA previu 4.581 sequências codificadoras (CDS); o tRNAScan previu 49 tRNAs, e NCBI Blast identificou quatro cópias do operons ribossomal 16S-23S-5S.

A sequência do genoma foi depositada no NCBI Genbank e recebeu o número de acesso (identificação) CP011409. Por sua vez, o anúncio do genoma de

Herbaspirillum hiltneri N3 foi publicado na revista *Genome Announcements* em outubro de 2015 (GUIZELINI et al, 2015).

4.8.2 *Azospirillum brasilense* FP2

A montagem da *Azospirillum brasilense* FP2 seguiu o protocolo geral aqui proposto, que consiste em tratar os dados SOLiD, MiSeq e 454 com filtragem de qualidade e filtro de bases para remoção de sequências de bases ambíguas (Ns). As montagens foram realizadas no Velvet, no CLC, no Edena V3 e no MaSuRCA. O conjunto das montagens convergiu para um conjunto de 319 *contigs*.

Nas leituras MiSeq foram tentadas técnicas de correção de base por programas que analisam a frequência de k-mers. Tentamos a abordagem do Quake (KELLEY et al., 2010) e do QuorUM (parte do pipeline MaSuRCA). Enfatizamos que as montagens aumentavam o número de *contigs* e reduziam o N50.

Buscamos separar as moléculas (cromossomo e plasmídeo) com base nos genomas de referência da *A. brasilense* 245. A estratégia para separação foi baseada em alinhamentos realizados com o Blast, considerando apenas os melhores alinhamentos. Alguns *contigs* alinhavam as mesmas partes em diferentes partes do cromossomo e dos plasmídeos. Optamos em duplicar esses *contigs* e colocá-los nos diferentes grupos. Posteriormente, para cada molécula da *A. brasilense* FP2, realizamos alinhamento com a molécula correspondente da *A. brasilense* 245 e com uma pseudosequência formada pelo cromossomo concatenado com as sequências dos plasmídeos da 245.

Após essa separação, utilizando o G-Finisher obtivemos o fechamento e a circularização do cromossomo e dos plasmídeos 1,2 e 6. O cromossomo apresenta 2.617.511 pb, com um conteúdo GC de 68,02%. O maior plasmídeo tem 1.640.799 pb e o conteúdo GC de 68,59%. O segundo maior tem 709.346 pb e o sexto 148.695 pb e o conteúdo de 67,13%. Os três prováveis plasmídeos intermediários estão respectivamente com 547 kpb, 445 kpb e 188 kpb.

Após as etapas de montagens realizadas na *A. brasilense* FP2, foram depositados no NCBI os genomas do *A. brasilense* Az39 e da *A. brasilense* Sp7. A Sp7 foi sequenciada no PacBio, entretanto a estirpe depositada tem um plasmídeo a menos que as demais estirpes descritas e algumas diferenças em relação ao

número de plasmídeos e ao tamanho de cada molécula levantados por eletroforese em campo pulsante (MARTIN-DIDONET et al., 2000).

A Tabela 13 apresenta as diferenças e a relação de correspondência entre as moléculas das diferentes estirpes baseadas na comparação por Dotplot. O cromossomo e os plasmídeos 1, 2 e o menor da *A. brasilense* FP2 correspondem com as respectivas moléculas da 245, Az39 e Sp7. O terceiro plasmídeo da *A. brasilense* FP2 que corresponde ao terceiro plasmídeo da *A. brasilense* 245 tem uma correspondência parcial, com aproximadamente 200 kpb, com a região final do cromossomo da Sp7 e mais uma parcial com o segundo plasmídeo da Sp7. Mas essa região identificável no dotplot feito pelo Gepard não reflete na análise do QFAST, que, ao mapear o terceiro plasmídeo da FP2 ao cromossomo da Sp7, não identifica nenhum gene completo ou parcial. O mesmo ocorre com a região de interseção deste plasmídeo com o segundo plasmídeo da Sp7. O quarto plasmídeo da *A. brasilense* que corresponde ao terceiro plasmídeo da Sp7 se encontra com uma lacuna de aproximadamente 50 kpb e uma provável região de exclusão de 25 kpb. O quinto plasmídeo da *A. brasilense* FP2 apresenta correspondência com 49 genes espalhados no cromossomo da *A. brasilense* Sp7. O sexto plasmídeo da *A. brasilense* FP2 corresponde com o quinto plasmídeo da Sp7.

TABELA 13 - COMPARAÇÃO ENTRE OS TAMANHOS DAS SEQUÊNCIAS NAS MONTAGENS DAS ESTIRPES DE *A. brasilense*.

molécula	Tamanho determinado por PFGE (kpb) ¹			Tamanho das montagens (kpb)			
	FP2	245	Sp7	FP2	245	Sp7	Az39
Cromossomo	2500	2600	2500	2617	3023	3005	3064
Plasmídeo 1	1720	1760	1740	1640	1766	1754	1901
Plasmídeo 2	810	900	810	709	912	819	933
Plasmídeo 3	700	780	700	434	778		686
Plasmídeo 4	630	720	640	572	690	645	641
Plasmídeo 5	170	210	210	185	191	206	
Plasmídeo 6	150	140	200	149	167	156	163

FONTE: O autor

Nota: ¹ Os tamanhos determinados por PFGE foram obtidos de MARTIN-DIDONET et al (2000)

A Figura 24 mostra os dois *contigs* onde a lacuna está delimitada na região entre 350.000 pb e 400.000 pb e a região de exclusão na região entre 575 kpb e 600 kpb (eixo y).

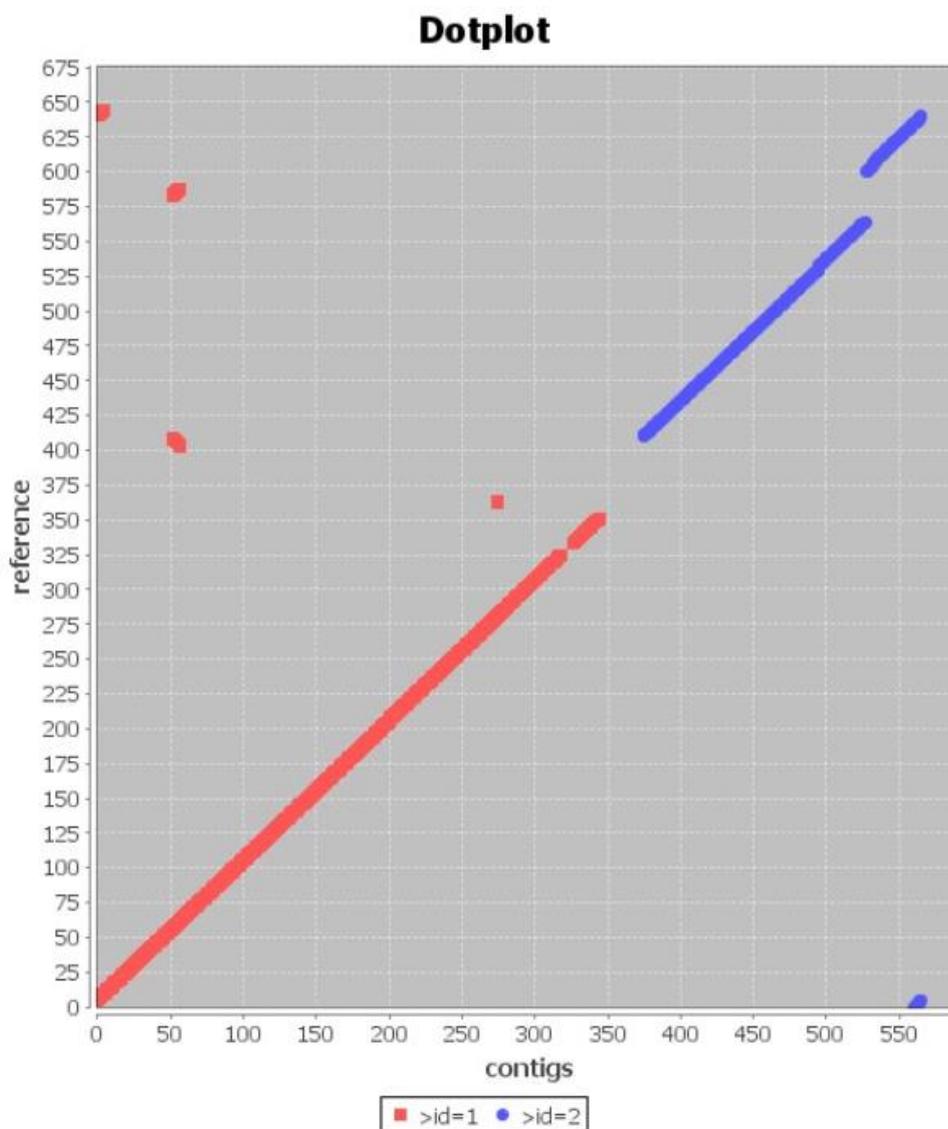


FIGURA 24 - CAPTURA DA IMAGEM PRODUZIDA PELO G-FINISHER DO DOTPLOT DOS DOIS *CONTIGS* DA MONTAGEM DO 4º PLASMÍDEO DA *A. brasilense* FP2 (X) COMPARADOS COM O 3º PLASMÍDEO DA *A. brasilense* Sp7

FONTE: O autor

O QUAST foi utilizado para acompanhar e comparar as montagens. Porém, a comparação pelo QUAST entre as estirpes montadas demonstram significantes diferenças entre as estirpes e entre os arranjos estruturais desses genomas. Essas condições indicam que a *A. brasilense* 245 está mais próxima da estirpe *A.*

brasiliense Az39 e provavelmente a *A. brasiliense* FP2 está mais próxima da estirpe Sp7, sequenciada pelos coreanos no PacBio. Contudo, as montagens ainda divergem em tamanho, em número de moléculas, em arranjos estruturais e na composição nucleotídica dos genes, conforme podemos observar nos dados da Tabela 14. Na coluna “genes”, relacionamos o número de genes identificados com alinhamento completo pelo QUASt, o número de genes com alinhamento parcial, bem como o número de genes nas respectivas moléculas da referência (Sp7).

TABELA 14 – SÍNTESE DAS COMPARAÇÕES REALIZADAS PELO QUASt DA MELHOR MONTAGEM DA *A. brasiliense* FP2 COM A *A. brasiliense* Sp7

Molécula	Tamanho (em pb)		Número de genes:			Rearranjos na FP2
	FP2	Sp7	completo	parcial	referência	
Cromossomo	2.617.511	3.005.726	2164	26	2773	18
Plasmídeo 1	1.640.799	1.754.523	1459	12	1557	11
Plasmídeo 2	709.346	819.020	557	21	651	11
Plasmídeo 3	434.862					
Plasmídeo 4	566.664	645.064	470	13	585	4
Plasmídeo 5	185.710		49	1		
Plasmídeo 6	148.695	156.480	110	1	115	1

FONTE: O autor

NOTA: * O QUASt identificou 49 sequências gênicas no plasmídeo 5 da *A. brasiliense* FP2 que alinham integralmente em regiões do cromossomo da *A. brasiliense* Sp7.

A tabela 15 apresenta o número de genes identificados da *A. brasiliense* FP2 em cada molécula da *A. brasiliense* Sp7. Para essa comparação, empregamos quatro montagens: a) formada por 321 *contigs*, proveniente de diferentes estratégias; b) montagem mista no CLC.; c) montagem realizada no Allpaths-LG; d) montagem atual.

TABELA 15 - MAPEAMENTO DOS GENES DA *A. brasilense* Sp7 NOS *CONTIGS* DA *A. brasilense* FP2

Molécula	N. de genes na referência	Conjunto de montagem			
		312 <i>contigs</i>	Mista no CLC	Allpaths-LG	Atual
Cromossomo	2.773	2628+110	2697+64	2603+126	2.164+26
Plasmídeo 1	1557	1500+45	282+1004	1508+39	1459+12
Plasmídeo 2	651	615+26	99+465	617+24	557+21
Plasmídeo 3	858	548+21	99+369	548+25	470+13
Plasmídeo 4	181	155+19	43+110	170+5	
Plasmídeo 5	115	114-1	37+64	107+4	110+1

FONTE: O autor

NOTA: Os números apresentados na forma X+Y representam os X genes com alinhamento completo e os Y genes com alinhamento parcial segundo a análise realizada pelo QUAST.

A montagem da *A. brasilense* FP2 ainda demanda mais trabalho, mais análise e não descartamos uma eventual necessidade de mais dados para confirmar o número de moléculas, os respectivos tamanhos e a organização dos plasmídeos 3, 4 e 5.

A melhor montagem obtida apresenta o cromossomo e os plasmídeo 1, 2 e 6 completos, tendo confirmado a circularização da sequência. Os demais estão divididos em 27 *contigs*. Os *contigs* montados revelam o conteúdo GC de 68.02%. As principais dificuldades na montagem de *A. brasilense* FP2 se devem ao número de moléculas; à quantidade de regiões repetidas e à diversidade de arranjos observados nas montagens de outras estirpes do gênero.

O processo de separação das moléculas foi realizado por meio da *A. brasilense* 245. Assim, reiniciamos o processo de separação e sua remontagem considerando a *A. brasilense* Sp7 como referência. Essa nova tentativa tende a produzir uma nova organização, especialmente entre os três plasmídeos que ainda apresentam lacunas.

Os dados apresentados indicam que o genoma da bactéria *A. brasilense* FP2 é muito semelhante ao genoma da bactéria *A. brasilense* Sp7; porém, ainda precisamos identificar o arranjo que acomode os genes e ao mesmo tempo

preservem a relação de mate-pair das leituras SOLiD e a relação pair-end das leituras MiSeq.

4.8.3 *Burkholderia contaminans* LTEB

As leituras Illumina MiSeq da *Burkholderia contaminans* LTEB foram importadas no CLC e analisadas pelo FastQC. Com base na análise, foi feitas duas montagens no CLC; uma com as leituras tratadas pelo próprio CLC e outra com as leituras brutas.

Para a montagem no Velvet, as leituras foram podadas pelo jTrimmer, que procedeu a verificação do parelramento das leituras nos dois arquivos. Foi executado o todosMer para obter as montagens Velvet. Por sua vez, o MaSuRCA foi executado como descrito pelo projeto GAGE-B. Todas as montagens foram analisadas pelo QUASt. As sequências dos três cromossomos de *Burkholderia lata* 383 foram juntadas formando uma única sequência. O G-Finisher foi usado e as lacunas, reduzidas. Ainda, foi feita a separação dos *contigs* para obter um conjunto de *contigs* para cada um dos três cromossomos e novamente realizados os alinhamentos. Obtivemos duas versões das montagens, sendo uma com 29 *contigs* e outra com um número um pouco menor. Entretanto, retornamos à versão dos 29 *contigs* em função de ter um número maior de genes preservados, incluindo uma cópia dos genes que codificam para os RNA ribossomais.

As montagens por referência não possibilitaram reduzir o número de *contigs*. Dessa forma, decidimos que o avanço da montagem depende da comparação dos genes que flanqueiam as bordas dos *contigs* e preferimos identificar em outros genomas uma situação que apresente essa combinação. Para isso, a montagem foi submetida ao RAST e ao SILA, que forneceram uma anotação automática. O terceiro cromossomo apresenta maior complexidade, em decorrência de apresentar muito pouca similaridade com o genoma de referência utilizado.

A montagem do genoma de *Burkholderia contaminans* LTEB se encontra incompleta e organizada em 29 *contigs*, com tamanho total de 8.4 mb e o conteúdo GC de 66.6%. A anotação automática do RAST indica 8.002 CDS, 66 tRNAs e uma cópia do operon 16S-23S-5S rRNA.

Dois foram as maiores complexidades na montagem, a saber: o grande número de cópias do operon ribossomal e a ausência de identificação de

similaridade do terceiro cromossomo com outros genomas completos depositados. Estimamos dez cópias do operon com base na média de cobertura, caso essa informação se confirme, um terço das lacunas podem ser fechadas com o devido posicionamento das cópias do operon. Em relação à similaridade dos cromossomos, enquanto os dois primeiros cromossomos apresentam alta semelhança com os cromossomos de *Burkholderia lata* 383, o terceiro continua sem um paralelo identificado.

4.8.4 *Herbaspirillum seropedicae* Z67

A montagem da *Herbaspirillum seropedicae* Z67 foi um caso atípico, de modo que as montagens realizadas no CLC, Velvet e MaSuRCA das corridas Illumina e SOLiD, ao serem submetidas ao G-Finisher, resultaram em 15 *contigs*. Os *contigs* foram alinhados ao genoma de referência - *H. seropedicae* SmR1. As regiões da SmR1 não cobertas pelos *contigs* da Z67 e expandidas em 1.000 pb para cada lado foram utilizadas para atrair as leituras provenientes do sequenciamento da Z67. Essas leituras foram novamente montadas e os novos *contigs* juntamente com os 15 primeiros *contigs* foram tratados pelo G-Finisher. Essa operação resultou em três *contigs*. Novamente, os *contigs* foram comparados com a sequência da SmR1, as leituras foram novamente alinhadas e novos *contigs* foram produzidos e culminando no fechamento do genoma em um tempo de duas semanas.

Enquanto que na montagem do genoma da *H. hiltneri* N3 o G-Finisher estava em desenvolvimento, na montagem do genoma da *Herbaspirillum seropedicae* Z67 o programa G-Finisher foi aplicado sistematicamente.

A sequência completa do genoma de *Herbaspirillum seropedicae* Z67 tem um cromossomo circular de 5,509,723 pb, com um percentual de GC de 63,4%. A anotação do genoma usando RAST e nossa plataforma (in-house) SILA previu 4.697 sequências codificadoras (CDS). A montagem do genoma de *H. seropedicae* Z67 revelou uma semelhança de base de 99.8327% com *H. seropedicae* SmR1.

A comparação das estirpes Z67 e SmR1 revelou que a Z67 tem 4.170 pb a menos que a SmR1 e o relatório do QUASt indica 476 diferenças locais. Sendo 115 bases inseridas, 183 exclusões e 178 modificações de bases. O alinhamento dos 4.796 genes anotadas da SmR1 com a Z67 apresenta 4.776 genes idênticos e 9

parciais. Ocorrendo 11 genes anotados na SmR1 que não foram identificados na Z67 e 1 gene da Z67 que não ocorre na SmR1.

A sequência do genoma foi depositada no NCBI Genbank e recebeu o número de acesso (identificação) CP011930.1.

5 DISCUSSÃO

O processo de montagem de genomas é uma das tarefas mais difíceis da bioinformática. Reconstruir a sequência de um genoma é mais complexo do que encaixar as peças em um quebra-cabeça. No quebra-cabeça, as cores, as linhas e os formatos das peças direcionam a montagem, de maneira que mesmo na ausência da imagem de referência é possível estender e reconstruir. Na montagem da sequência genoma, as cores e linhas não são identificáveis facilmente, mas estão presentes na forma de códons, sentido das fitas e da composição numérica dos nucleotídeos, tais como o medido pelo GC Skew.

A etapa de pré-processamento é fundamental em qualquer projeto, pois a análise da cobertura e da frequência de sementes (ou kmers) pode reduzir significativamente o efeito GIGO (do inglês, *garbage in garbage out*), reduzir os ruídos decorrentes dos erros de sequenciamento e remover eventuais artefatos como adaptadores.

Na etapa de montagem, o uso de diferentes algoritmos permite obter montagens alternativas, particularmente das regiões mais complexas do genoma, como áreas de repetição.

Após a etapa de montagem um dos procedimentos que realizamos com frequência foi a comparação da montagem com o genoma de referência ou entre as montagens utilizando o Dotplot (item 3.6), entretanto ao alinhar as pontas dos *contigs* com as leituras, para obter indicação de continuidade por meio do BLAST, detectávamos que os pares das leituras ou mesmo parte das leituras se alinhavam na parte interna do mesmo contig ou de outros *contigs*. A primeira interpretação é que representavam regiões de repetição ou de alta similaridade. Porém, em muitos casos essas regiões não apresentavam linhas paralelas no dotplot, o que poderia ser justificado, entre outras, pelas seguintes razões: a) a região é muito pequena para ser visualizada no dotplot, onde cada ponto geralmente representa 1.000 pb ou; b) a sensibilidade do algoritmo do NCBI Blast é maior que os algoritmos do MUMmer e do Gepard ou; c) a característica da sequência influencia o montador tais como grampos e espelhos e; d) erros de montagem.

As imagens obtidas nos programas de Dotplot não permitiam identificar facilmente se as ocorrências demonstradas na Figura 10 eram internas aos *contigs* ou entre os *contigs*. Essa dificuldade levou ao desenvolvimento de um novo

programa para desenhar o dotplot atribuindo uma cor para cada contig. Posteriormente, esse programa foi incorporado ao G-Finisher e é um dos relatórios produzidos em cada etapa.

No pós-montagem ou finalização de montagem, o G-Finisher melhora as montagens com uma redução média de 86% do número de *contigs* em relação às 96 montagens obtidas pelo GAGE-B. A estratégia de mapeamento dos *contigs* em genomas de referência é mais eficiente que os programas que se baseiam em pares de leituras para construção de scaffolds e resolução de regiões de repetição. O FGap apresentou melhores resultados que os programas GapCloser, GapFiller e IMAGE na proposta de preenchimento de lacuna. O G-Finisher melhorou os resultados do FGap, ao acrescentar no processo a estratégia de ordenação, construção de scaffolds, com base em genomas de referência e na correção de *contigs* que apresentam erros de montagem.

O G-Finisher demonstrou ser uma boa estratégia para combinar diferentes montagens, contrariando os resultados obtidos pelo GAGE. O GAGE fez combinação de montagens para verificar se diferentes algoritmos ou parâmetros poderiam melhorar a montagem. Na estratégia, eles utilizaram os programas minimus2 (pacote AMOS), *Graph Accordance Assembly* (GAA) e o CD-HIT. Segundo os autores, na maioria das combinações os resultados eram piores que as melhores montagens obtidas. Mas a estratégia de fusão, usando o GAA, apresentou melhora ao utilizar como montagem alvo a montagem do SOAPdenovo e as montagens do ABySS, CABOG ou MaSuRCA como consulta. Em todos os três casos, a correção do N50 melhorou 458 kpb, porém o número de contigs permaneceu inalterado. Ainda, segundo os autores, ao combinar as montagens do CABOG com o ABySS observou um incremento do N50 em 151 kpb (incremento de 90%). Dessa forma, os autores concluem que encontrar um par de algoritmos que complementem uns aos outros de maneira vantajosa, pode exigir uma grande quantidade de tentativas e erros (MAGOC et al., 2013).

No G-Finisher, durante o processo de mapeamento dos *contigs* no genoma de referência (*jContigsort*), o grupo com maior densidade de pontos é escolhido para ancorar o contig. Entretanto, o programa de análise dos resultados do *jContigsort* agora emite alertas quando partes distintas dos *contigs* alinham em diferentes posições e quando as mesmas regiões do contig apresentam um mapeamento em regiões distintas do genoma de referência. Esses indicadores estão sendo

estudados no intuito de permitir, em uma versão futura do G-Finisher, a duplicação com segurança de partes dos *contigs* e, dessa forma, aumentar a cobertura do genoma de referência.

O gráfico do GC Skew apresentado na Figura 23 apresenta “assinaturas” muito semelhantes ao GC SKew obtido do genoma de *H. seropedicae* SmR1. Essas semelhanças permitem especular que o GC Skew captura algumas características conservadas na estrutura do arranjo genômico dentro do gênero *Herbaspirillum*.

No estudo do GAGE-B, o MaSuRCA foi o montador mais eficiente em relação ao número de *contigs* e ao de valores de N50. O MaSuRCA apresentou o menor número de *contigs* em oito dos 12 experimentos e o maior N50 em dez dos 12 genomas estudados. Entretanto, os dados do G-Finisher demonstram que as montagens realizadas pelo MaSuRCA são mais beneficiadas com a quebra dos *contigs*, indicando que o montador comete mais erros de montagens de *contigs* no padrão identificado pela variação do GC Skew.

O G-Finisher e o FGap devem ser utilizados graduando do menor para o maior os limites de inserção e remoção para evitar o efeito de “inchaço” ou “compressão” da montagem. Esses valores podem ser estimados com base nos Dotplots gerados pelo G-Finisher.

6 CONCLUSÃO

Neste trabalho, foi levantada a hipótese de que o uso de informações *a priori* poderia aprimorar os resultados das montagens, de modo que a estratégia obtida melhorou todos os casos aplicados.

Nossos dados corroboram com outros estudos que afirmam que o número de repetições e o tamanho das áreas repetidas são as maiores causas de erros por parte dos montadores e a principal causa da complexidade do processo.

Contudo, o uso do padrão observado na curva GC Skew permite reduzir em 50% os erros nas situações que ocorrem o *dilema do caminho*, bem como identificar erros de montagem que tendem a coincidir com os picos críticos da curva GC Skew. A tendência representada na curva do GC Skew é uma evidência forte a ser observada para resolver problemas de regiões repetidas.

A utilização de genomas de referência, mesmo com maiores distâncias evolutivas, é uma solução para organizar o arranjo estrutural do genoma. As leituras provenientes de bibliotecas *mate-pair* com distâncias maiores que 1500 pb ou leituras maiores que 10 kpb são as únicas alternativas ao uso de genomas de referências.

O G-Finisher é uma aplicação computacional que pode ser usada pelos pesquisadores para melhorar as montagens de genoma com diminuição significativa no número de *contigs* e com o aumento do N50, possibilitando o fechamento de projetos de montagem de genoma bacteriano. Abrimos o código e estabelecemos uma licença que assegura o uso e permite sua integração por outros programas.

A estratégia do G-Finisher contraria os resultados do GAGE-B em relação à combinação de montagens e mostra que a combinação de montagens obtida por diferentes montadores permite melhorar a montagem.

A adoção de padrões biológicos pelos montadores é uma estratégia a ser adotada, a qual melhorará a qualidade das montagens.

Vale destacar, ainda, que o cálculo do Fuzzy GC Skew tem um custo computacional mais elevado que a versão equivalente do GC Skew. Todavia, o modelo fuzzy melhora a precisão do processo de identificação dos pontos críticos.

REFERÊNCIAS

- ALBUQUERQUE, P.; CARIDADE, C. M. R.; MARCAL, A. R. S.; et al. Identification of *Xanthomonas fragariae*, *Xanthomonas axonopodis* pv. *phaseoli*, and *Xanthomonas fuscans* subsp. *fuscans* with novel markers and using a dot blot platform coupled with automatic data Analysis. **Applied and Environmental Microbiology**, v. 77, n. 16, p. 5619–5628, 2011.
- AL-OKAILY, A. A. Open Access HGA: de novo genome assembly method for bacterial genomes using high coverage short sequencing reads. , p. 1–11, 2016.
- ALTSCHUL, S. F.; GISH, W.; MILLER, W.; MYERS, E. W.; LIPMAN, D. J. Basic local alignment search tool. **Journal of Molecular Biology**, v. 215, n. 3, p. 403–410, 1990. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0022283605803602>>. .
- ARONESTY, E. Command-line tools for processing biological sequencing data. .
- BABRAHAM INSTITUTE. FastQC: A quality control tool for high throughput sequence data. , 2016. Disponível em: <<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>>. .
- BANKEVICH, A.; NURK, S.; ANTIPOV, D.; et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. **Journal of Computational Biology**, v. 19, n. 5, p. 455–477, 2012.
- BOETZER, M.; HENKEL, C. V.; JANSEN, H. J.; BUTLER, D.; PIROVANO, W. Scaffolding pre-assembled *contigs* using SSPACE. **Bioinformatics**, v. 27, n. 4, p. 578–579, 2011.
- BOETZER, M.; PIROVANO, W. Toward almost closed genomes with GapFiller. **Genome biology**, v. 13, n. 6, p. R56, 2012. BioMed Central Ltd. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3446322&tool=pmcentrez&rendertype=abstract>>. .
- BRADNAM, K. R.; FASS, J. N.; ALEXANDROV, A.; et al. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. **GigaScience**, v. 2, n. 1, p. 10, 2013. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3844414&tool=pmcentrez&rendertype=abstract>>. .
- CHEVREUX, B.; PFISTERER, T.; DRESCHER, B.; et al. Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs. **Genome Research**, v. 14, n. 6, p. 1147–1159, 2004.
- COCK, P. J. A.; FIELDS, C. J.; GOTO, N.; HEUER, M. L.; RICE, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. **Nucleic Acids Research**, v. 38, n. 6, p. 1767–1771, 2009.
- COIL, D.; JOSPIN, G.; DARLING, A. E. A5-miseq: An updated pipeline to assemble microbial genomes from Illumina MiSeq data. **Bioinformatics**, v. 31, n. 4, p. 587–589, 2015.

COLLYN, F.; ROTEN, C. A. H.; GUY, L. Solving ambiguities in contig assembly of *Idiomarina loihiensis* L2TR chromosome by in silico analyses. **FEMS Microbiology Letters**, v. 271, n. 2, p. 187–192, 2007.

COMPEAU, P. E. C.; PEVZNER, P. A.; TESLER, G. How to apply de Bruijn graphs to genome assembly. **Nature Biotechnology**, v. 29, n. 11, p. 987–991, 2011. Nature Publishing Group. Disponível em: <<http://dx.doi.org/10.1038/nbt.2023>>. .

CONWAY, T. C.; BROMAGE, A. J. Succinct data structures for assembling large genomes. **Bioinformatics**, v. 27, n. 4, p. 479–486, 2011.

DEL FABBRO, C.; SCALABRIN, S.; MORGANTE, M.; GIORGI, F. M. An extensive evaluation of read trimming effects on illumina NGS data analysis. **PLoS ONE**, v. 8, n. 12, p. 1–13, 2013.

FRANK, A. C.; LOBRY, J. R. Oriloc: prediction of replication boundaries in unannotated bacterial chromosomes. **Bioinformatics (Oxford, England)**, v. 16, n. 6, p. 560–561, 2000. Disponível em: <<papers://42181af0-5306-4306-9bb6-bc1c9f05c1fe/Paper/p1393>>. .

GAO, S.; SUNG, W.-K.; NAGARAJAN, N. Opera: Reconstructing Optimal Genomic Scaffolds with High-Throughput Paired-End Sequences. **Journal of Computational Biology**, v. 18, n. 11, p. 1681–1691, 2011. Disponível em: <<http://www.liebertonline.com/doi/abs/10.1089/cmb.2011.0170>>. .

GRIGORIEV, A. Analyzing genomes with cumulative skew diagrams. **Nucleic Acids Research**, v. 26, n. 10, p. 2286–2290, 1998.

GUIZELINI, D.; BAURA, V. A.; MONTEIRO, R. A.; et al. Complete Genome Sequence of *Herbaspirillum hiltneri* N3 (DSM 17495), Isolated from Surface-Sterilized Wheat Roots. , v. 3, n. 5, p. 2–3, 2015.

GUIZELINI, D.; PEDROSA, F. DE O.; TIBÃES, J.; et al. *jContigsort*: a new computer application for *contigs* ordering. 7th International Conference of The Brazilian Association for Bioinformatics and Computacional Biology. **Anais...** , 2011.

GUREVICH, A.; SAVELIEV, V.; VYAHHI, N.; TESLER, G. QUASt: Quality assessment tool for genome assemblies. **Bioinformatics**, v. 29, n. 8, p. 1072–1075, 2013.

HAUER, V. **SEQUENCIAMENTO DO GENE *glnA* DAS ESTIRPES MUTANTES HM14, HM26, HM053 E HM210 DE *Azospirillum brasilense***, 2012. Universidade Federal do Paraná.

HUANG, Y.; LI, Y. Prediction of protein subcellular locations using fuzzy k-NN method. **Bioinformatics**, v. 20, n. 1, p. 21–28, 2004.

ILLUMINA. De Novo Assembly of Bacterial Genomes: Using mate pair technology and the MiSeq System to generate high-quality genome assemblies. Disponível em: <<https://support.illumina.com/content/dam/illumina-marketing/documents/products/appnotes/appnote-nextera-mate-pair-bacteria.pdf>> .

JECK, W. R.; REINHARDT, J. A.; BALTRUS, D. A.; et al. Extending assembly of short DNA sequences to handle error. **Bioinformatics**, v. 23, n. 21, p. 2942–2944, 2007.

KATSY, E.; PETROVA, L. Genome Rearrangements in *Azospirillum brasilense* Sp7 with the Involvement of the Plasmid pRhico and the Prophage phiAb-Cd. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/27055294>>. .

KINGSFORD, C.; SCHATZ, M. C.; POP, M. Assembly complexity of prokaryotic genomes using short reads. **BMC bioinformatics**, v. 11, p. 21, 2010.

KONTUR, W. S.; SCHACKWITZ, W. S.; IVANOVA, N.; et al. Revised sequence and annotation of the *Rhodobacter sphaeroides* 2.4.1 genome. **Journal of Bacteriology**, v. 194, n. 24, p. 7016–7017, 2012.

KRUMSIEK, J.; ARNOLD, R.; RATTEI, T. Gepard: A rapid and sensitive tool for creating dotplots on genome scale. **Bioinformatics**, v. 23, n. 8, p. 1026–1028, 2007.

KURTZ, S.; PHILLIPPY, A.; DELCHER, A. L.; et al. Versatile and open software for comparing large genomes. **Genome biology**, v. 5, n. 2, p. R12, 2004. Disponível em: <<http://genomebiology.com/2004/5/2/R12>\n<http://www.ncbi.nlm.nih.gov/pubmed/14759262>\n<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC395750>>. .

LAND, M.; HAUSER, L.; JUN, S.; et al. Insights from 20 years of bacterial genome sequencing. , p. 141–161, 2015.

LI, Z.; CHEN, Y.; MU, D.; et al. Comparison of the two major classes of assembly algorithms : overlap ^ layout ^ consensus and de-bruijn-graph. , v. 11, n. 1, 2011.

LI, Z.; CHEN, Y.; MU, D.; et al. Comparison of the two major classes of assembly algorithms: Overlap-layout-consensus and de-bruijn-graph. **Briefings in Functional Genomics**, v. 11, n. 1, p. 25–37, 2012.

LIU, L.; LI, Y.; LI, S.; et al. Comparison of next-generation sequencing systems. **Journal of Biomedicine and Biotechnology**, v. 2012, 2012.

LOBRY, J. R. Substitution Patterns in the Two DNA Strands of Bacteria. **Molecular Biology**, , n. 3, p. 660–665, 1996.

LUO, R.; LIU, B.; XIE, Y.; et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. **GigaScience**, v. 1, n. 1, p. 18, 2012. Disponível em: <<http://www.gigasciencejournal.com/content/1/1/18>\n<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3626529&tool=pmcentrez&rendertype=abstract>>. .

M. A. DYMOVA, O. I. ALKHOVIK, L. S. EVDOKIMOVA, A. G. CHEREDNICHENKO, T. I. P. F. Complete Genome Sequence of a Novel Clinical Isolate , *Mycobacterium abscessus* Strain NOV0213. **American society for microbiology**, v. 4, n. 1, p. 5660, 2016.

MAGOC, T.; PABINGER, S.; CANZAR, S.; et al. GAGE-B: An evaluation of genome assemblers for bacterial organisms. **Bioinformatics**, v. 29, n. 14, p. 1718–1725, 2013.

MARÇAIS, G.; YORKE, J. A.; ZIMIN, A. QuorUM: An error corrector for Illumina reads. **PLoS ONE**, v. 10, n. 6, p. 1–13, 2015.

MARÇAIS, G.; KINGSFORD, C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. **Bioinformatics**, v. 27, n. 6, p. 764–770, 2011.

MARTIN-DIDONET, C. C. G.; CHUBATSU, L. S.; SOUZA, M.; et al. Genome Structure of the Genus *Azospirillum*. **Journal of Bacteriology**, v. 182, n. 14, p. 14–18, 2000.

METZKER, M. L. Sequencing technologies - the next generation. **Nature reviews. Genetics**, v. 11, n. 1, p. 31–46, 2010. Nature Publishing Group. Disponível em: <<http://dx.doi.org/10.1038/nrg2626>>. .

MILLER, J. R.; DELCHER, A. L.; KOREN, S.; et al. Aggressive assembly of pyrosequencing reads with mates. **Bioinformatics**, v. 24, n. 24, p. 2818–2824, 2008.

MILLER, J. R.; KOREN, S.; SUTTON, G. Assembly algorithm for next-generation sequencing data. **Genomics**, v. 95, n. 6, p. 315–327, 2010.

MYERS, E. W. Toward Simplifying and Accurately Formulating Fragment Assembly. **Journal of Computational Biology**, v. 2, n. 2, p. 275–290, 1995. Disponível em: <<http://www.liebertonline.com/doi/abs/10.1089/cmb.1995.2.275>>. .

NAGARAJAN, N.; COOK, C.; DI BONAVENTURA, M.; et al. Finishing genomes with limited resources: lessons from an ensemble of microbial genomes. **BMC genomics**, v. 11, p. 242, 2010.

NAGARAJAN, N.; POP, M. Sequence assembly demystified. **Nature reviews. Genetics**, v. 14, n. 3, p. 157–67, 2013. Nature Publishing Group. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/23358380>>. .

NARZISI, G.; MISHRA, B. Comparing De Novo genome assembly: The long and short of it. **PLoS ONE**, v. 6, n. 4, 2011.

NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION. No Title. Disponível em: <<https://www.ncbi.nlm.nih.gov/home/about/mission.shtml>>. Acesso em: 23/5/2016.

NIKITINA, A. S.; KHARLAMPIEVA, D. D.; BABENKO, V. V.; et al. Complete Genome Sequence of an Enterotoxigenic *Bacteroides fragilis*. , v. 3, n. 3, p. 4–5, 2015.

PASZKIEWICZ, K.; STUDHOLME, D. J. De novo assembly of short sequence reads. **Briefings in Bioinformatics**, v. 11, n. 5, p. 457–472, 2010.

PEDROSA, F. O.; MONTEIRO, R. A.; WASSEM, R.; et al. Genome of *herbaspirillum seropedicae* strain SmR1, a specialized diazotrophic endophyte of tropical grasses. **PLoS Genetics**, v. 7, n. 5, 2011.

PENG, Y.; LEUNG, H. C. M.; YIU, S. M.; CHIN, F. Y. L. IDBA-UD: A de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. **Bioinformatics**, v. 28, n. 11, p. 1420–1428, 2012.

PEVZNER, P. A.; TANG, H.; WATERMAN, M. S. An Eulerian path approach to DNA fragment assembly. **Proceedings of the National Academy of Sciences of the United States of America**, v. 98, n. 17, p. 9748–53, 2001. Disponível em: <<http://www.pnas.org/content/98/17/9748.abstract>>. .

PHILLIPPY, A. M.; SCHATZ, M. C.; POP, M. Genome assembly forensics: finding the elusive mis-assembly. **Genome biology**, v. 9, n. 3, p. R55, 2008.

PIRO, V. C.; FAORO, H.; WEISS, V. A.; et al. FGAP: an automated gap closing tool. **BMC research notes**, v. 7, n. 1, p. 371, 2014. Disponível em: <<http://www.biomedcentral.com/1756-0500/7/371>>. .

POP, M. Genome assembly reborn: Recent computational challenges. **Briefings in Bioinformatics**, v. 10, n. 4, p. 354–366, 2009.

POP, M.; PHILLIPPY, A.; DELCHER, A. L.; et al. Comparative genome assembly. **Oxford**, v. 5, n. 3, p. 237–248, 2004. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/19573591>>. .

ROTEN, C. A.; GAMBA, P.; BARBLAN, J. L.; KARAMATA, D. Comparative Genometrics (CG): a database dedicated to biometric comparisons of whole genomes. **Nucleic acids research**, v. 30, n. 1, p. 142–144, 2002. Disponível em: <<papers://42181af0-5306-4306-9bb6-bc1c9f05c1fe/Paper/p1213>>. .

ROTHBALLER, M.; SCHMID, M.; KLEIN, I.; et al. Herbaspirillum hiltneri sp. nov., isolated from surface-sterilized wheat roots. **International Journal of Systematic and Evolutionary Microbiology**, v. 56, n. 6, p. 1341–1348, 2006.

SÁ, P. H. C. G.; MIRANDA, F.; VERAS, A.; et al. GapBlaster—A Graphical Gap Filler for Prokaryote Genomes. **Plos One**, v. 11, n. 5, p. e0155327, 2016. Disponível em: <<http://dx.plos.org/10.1371/journal.pone.0155327>>. .

SALZBERG, S. L.; PHILLIPPY, A. M.; ZIMIN, A.; et al. GAGE: A critical evaluation of genome assemblies and assembly algorithms. **Genome Research**, v. 22, n. 3, p. 557–567, 2012.

SCHATZ, M. Assembly of Large Genomes using Cloud Computing. , p. 1165–1173, 2010. Disponível em: <<http://cbcb.umd.edu/~mschatz/Presentations/2010-07-23.Illumina.pdf>>. .

SESHADRI, R.; JOSEPH, S. W.; CHOPRA, A. K.; et al. Genome sequence of *Aeromonas hydrophila* ATCC 7966T: Jack of all trades. **Journal of Bacteriology**, v. 188, n. 23, p. 8272–8282, 2006.

SILVA, D. M.; DA SILVA, M. P.; VIDIGAL, P. M. P.; et al. Erratum for Silva et al., Draft Genome Sequences of *Staphylococcus aureus* Strains Isolated from Subclinical Bovine Mastitis in Brazil. **Genome announcements**, v. 4, n. 2, p. 15–17, 2016. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/27103732>> <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4841147>>. .

SILVA, E. L.; MENEZES, E. M. Metodologia da Pesquisa e Elaboração de Dissertação - 4a edição. **Portal**, p. 138p, 2005.

SILVEIRA, L. DA. **MONTAGEM E ANOTAÇÃO PARCIAL DA SEQUÊNCIA GENÔMICA DA BACTÉRIA DIAZOTRÓFICA *Azospirillum brasilense* FP2**, 2012. Universidade Federal do Paraná.

SIMPSON, J. T.; DURBIN, R.; ZERBINO, D. R.; et al. Efficient de novo assembly of large genomes using compressed data structures sequence data. **Genome research**, p. 549–556, 2012.

SIMPSON, J. T.; WONG, K.; JACKMAN, S. D.; et al. ABySS : A parallel assembler for short read sequence data ABySS : A parallel assembler for short read sequence data. , p. 1117–1123, 2009.

SONNHAMMER, E. L. L.; DURBIN, R. A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. **Gene**, v. 167, n. 1-2, p. 1–10, 1995.

STENFORS ARNESEN, L. P.; FAGERLUND, A.; GRANUM, P. E. From soil to gut: *Bacillus cereus* and its food poisoning toxins. **FEMS Microbiology Reviews**, v. 32, n. 4, p. 579–606, 2008.

SZAFRANSKI, K.; JAHN, N.; PLATZER, M. tuple_plot: Fast pairwise nucleotide sequence comparison with noise suppression. **Bioinformatics**, v. 22, n. 15, p. 1917–1918, 2006.

TRITT, A.; EISEN, J. A.; FACCIOTTI, M. T.; DARLING, A. E. An Integrated Pipeline for de Novo Assembly of Microbial Genomes. **PLoS ONE**, v. 7, n. 9, 2012.

TSAI, I. J.; OTTO, T. D.; BERRIMAN, M. Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. **Genome biology**, v. 11, n. 4, p. R41, 2010.

WU, C. H.; CHEN, C.; MORALES, C.; KIANG, D. Draft Genome Sequence of an ortho -Nitrophenyl- β - D -Galactoside (ONPG) -Negative Strain of *Vibrio cholerae* , Isolated from Drakes Bay , , v. 4, n. 2, p. 1232–1233, 2016.

ZADEH, L. A. Outline of a new approach to the analysis of complex systems and decision processes. **Systems, Man and Cybernetics, IEEE Transactions on** , , n. 1, p. 28–44, 1973.

ZERBINO, D. R. 2009. **Genome assembly and comparison using de Bruijn graphs**. Tese. Universidade de Cambridge, Hinxton, Reino Unido, 2009. 149f.

ZERBINO, D. R. Genome assembly and comparison using de Bruijn graphs. **Molecular Biology**, p. 149, 2009.

ZERBINO, D. R. e BIRNEY, E. 2008. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. **Genome research**, Vol. 18 No.5, pp 821–9.

ZERBINO, D. R.; BIRNEY, E. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. **Genome Research**, v. 18, n. 5, p. 821–829, 2008.

ZHANG, W.; CHEN, J.; YANG, Y.; et al. A practical comparison of De Novo genome assembly software tools for next-generation sequencing technologies. **PLoS ONE**, v. 6, n. 3, 2011.

ZIMIN, A. V.; MARÇAIS, G.; PUIU, D.; et al. The MaSuRCA genome assembler. **Bioinformatics**, v. 29, n. 21, p. 2669–2677, 2013.

7 ANEXO 1 – LISTA DE MONTADORES E ENDEREÇOS ELETRÔNICOS

ABYSS: <http://www.bcgsc.ca/platform/bioinfo/software/abyss>

CABOG (Celera Assembler with Best Overlap Graph):
<http://www.jcvi.org/cms/research/projects/cabog/overview/>

MIRA (Mimicking Intelligent Read Assembly): <https://sourceforge.net/projects/mira-assembler/> e <http://www.bioinfo.de/isb/gcb99/talks/chevreux/>

MaSuRCA (Maryland Super Read - Celera Assembler)
<http://www.genome.umd.edu/masurca.html>

SGA (String Graph Assembler) <https://github.com/jts/sga> e
<http://genome.cshlp.org/content/22/3/549>

SOAPDenovo2 (Short Oligonucleotide Analysis Package)
<http://soap.genomics.org.cn/soapdenovo.html> sucedido por MEGAHIT (05/2015 -
<http://www.ncbi.nlm.nih.gov/pubmed/25609793>).

SPAdes Genome Assembler <http://bioinf.spbau.ru/en/spades>

8 ANEXO 2 – TABELAS SUPLEMENTARES DOS ORGANISMOS E MONTADORES UTILIZADOS NO GAGE-B

TABELA 16 - MONTADORES ESTUDADOS NO GAGE-B

Montador	Algoritmo	Última versão e atualização	Autores / Grupo	Indicação
ABYSS	De Bruijn	1.9 29/05/2015	Simpson / Canada's Michael Smith Genome Sciences Centre (BC Cancer Agency)	Grandes genomas – Solexa e SOLID
CABOG	OLC	CA 8.3rc2 24/05/2015	Myers e Miller / J. Craig Venter Institute	Samger, 454 e Solexa
Mira	OLC	4.0	Chevreur / University of Heidelberg	Sanger, 454, Illumina e IonTorrent
MaSuRCA	De Bruijn/OLC	3.1.3 26/08/2015	Zimin et al University of Maryland e Johns Hopkins University	Sanger, Illumina, 454
SGA	String Graph	0.10.14 12/01/2016	Simpson, J e Durbin, R Trust Sanger Institute	Illumina, Sanger
SOAPDenovo	De Bruijn	2.0.1 9/10/2015	Li et al	Solexa
SPades	De Bruijn	3.7 24/02/2016	Bankevich University of Russian	Illumina, Solexa, Sanger, 454, Ion Torrent, PacBio e Oxford Nanopore
Velvet	De Bruijn	1.2.10 15/08/2014	Zerbino EMBL-EBI	Sanger, 454, Solexa, SOLiD

TABELA 17 INFORMAÇÕES DOS ORGANISMOS E DOS SEQUENCIAMENTOS

Organismo				Sequenciamento			
Espécie	Tam. Genoma (mb)	%GC	Genoma	Tecnologia de sequenciamento	Tam.da leitura (pb)	Tam.do fragmento	Cobertura
<i>A. hydrophila</i> SSU	4.7	65	1 c	HiSeq	101	180	250x
<i>B. cereus</i> VD118	5.4	35	1 c 1 p	HiSeq	101	180	100-300x
<i>B. cereus</i> ATCC	5.4	35	1 c 1 p	MiSeq	250	600	100x
<i>B. fragilis</i> HMW 615	5.3	43	1 c	HiSeq	101	180	250x
<i>M. abscessus</i> 6G-0125-R	5.1	64	1 c 1 p	HiSeq	100	335	115x
<i>M. abscessus</i> 6G-0125-R	5.1	64	1 c 1 p	MiSeq	250	335	100x
<i>R. sphaeroides</i> 2.4.1	4.6	69	2 c 5 p	HiSeq	101	220	210x
<i>R. sphaeroides</i> 2.4.1	4.6	69	2 c 5 p	MiSeq	251	540	100x
<i>S. aureus</i> M0927	2.9	33	1 c 2 p	HiSeq	101	180	250x
<i>V. cholerae</i> CO1032(5)	4.0	48	2 c	HiSeq	100	335	110x
<i>V. cholerae</i> CO1032(5)	4.0	48	2 c	MiSeq	250	335	100x
<i>X. axonopodis</i> pv. Manihotis UA323	2.9	33	1 c	HiSeq	101	400	250x

Nota: Na coluna “genoma”, a letra “c” indica o número de cromossomos e a letra “p” o número de plasmídeos. Tabela adaptada do artigo do GAGE-B.

9 ANEXO 3 – RESULTADOS INDIVIDUAIS DAS MONTAGENS DO GAGE-B

TABELA 18 REDUÇÃO DO NÚMERO DE *CONTIGS* NAS 96 MONTAGENS FORNECIDAS PELO GAGE-B

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>A. hydrophila</i> SSU - HiSeq	52,93	19,20	195,00	24,00	19,53	15,67	63,10	70,40
Abyss ctg	52	18	197	24	18	15	65,38	71,15
ABYSS scf	43	19	198	26	22	16	48,84	62,79
CABOG ctg	81	17	189	23	19	16	76,54	80,25
CABOG scf	37	17	189	23	19	16	48,65	56,76
MaSuRCA ctg	25	15	187	19	14	11	44,00	56,00
MaSuRCA scf	23	15	187	19	14	11	39,13	52,17
MIRA ctg	58	43	219	46	37	30	36,21	48,28
SGA ctg	132	20	198	27	23	16	82,58	87,88
SGA scf	131	21	198	23	21	16	83,97	87,79
SOAPdenovo2 ctg	36	16	194	21	18	14	50,00	61,11
SOAPdenovo2 scf	28	16	194	21	18	14	35,71	50,00
SPAdes ctg	38	18	190	22	16	14	57,89	63,16
SPAdes scf	31	17	191	22	18	16	41,94	48,39
Velvet ctg	51	18	197	22	18	15	64,71	70,59
Velvet scf	28	18	197	22	18	15	35,71	46,43

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>B. cereus</i> ATCC 10987 - MiSeq	119,73	28,80	51,87	28,47	25,73	15,13	78,51	87,36
Abyss ctg	82	58	81	58	54	33	34,15	59,76
ABySS scf	71	59	82	59	55	33	22,54	53,52
CABOG ctg	77	14	36	15	14	10	81,82	87,01
CABOG scf	33	14	36	15	14	10	57,58	69,70
MaSuRCA ctg	55	36	59	34	29	8	47,27	85,45
MaSuRCA scf	48	37	60	34	30	8	37,50	83,33
MIRA ctg	96	37	62	34	30	20	68,75	79,17
SGA ctg	320	16	38	17	16	13	95,00	95,94
SGA scf	310	18	41	19	16	12	94,84	96,13
SOAPdenovo2 ctg	73	20	42	19	17	13	76,71	82,19
SOAPdenovo2 scf	51	20	42	19	17	13	66,67	74,51
SPAdes ctg	99	16	40	19	16	10	83,84	89,90
SPAdes scf	60	19	43	21	18	10	70,00	83,33
Velvet ctg	351	34	58	32	30	17	91,45	95,16
Velvet scf	70	34	58	32	30	17	57,14	75,71

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>B. cereus</i> VD118 - HiSeq	251,60	67,80	96,53	63,67	58,60	40,33	76,71	83,97
Abyss ctg	325	78	117	75	71	42	78,15	87,08
ABYSS scf	302	87	123	80	74	42	75,50	86,09
CABOG ctg	163	58	89	54	49	32	69,94	80,37
CABOG scf	150	58	89	54	49	32	67,33	78,67
MaSuRCA ctg	180	79	114	77	66	40	63,33	77,78
MaSuRCA scf	178	80	114	75	68	42	61,80	76,40
MIRA ctg	437	73	104	68	63	44	85,58	89,93
SGA ctg	463	57	75	52	49	41	89,42	91,14
SGA scf	251	42	49	35	35	30	86,06	88,05
SOAPdenovo2 ctg	248	72	101	70	65	53	73,79	78,63
SOAPdenovo2 scf	235	72	101	70	65	53	72,34	77,45
SPAdes ctg	179	62	94	58	53	40	70,39	77,65
SPAdes scf	183	67	102	59	52	38	71,58	79,23
Velvet ctg	251	66	88	64	60	38	76,10	84,86
Velvet scf	229	66	88	64	60	38	73,80	83,41

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>B. fragilis</i> HMW 615 - HiSeq	125,33	34,33	193,20	35,80	31,53	20,20	74,41	83,88
Abyss ctg	141	47	241	45	39	19	72,34	86,52
ABYSS scf	128	49	250	52	47	21	63,28	83,59
CABOG ctg	136	32	176	33	29	19	78,68	86,03
CABOG scf	129	33	178	33	29	19	77,52	85,27
MaSuRCA ctg	102	44	210	26	26	11	74,51	89,22
MaSuRCA scf	98	42	212	44	33	10	66,33	89,80
MIRA ctg	109	60	211	74	63	38	42,20	65,14
SGA ctg	233	30	172	33	30	26	87,12	88,84
SGA scf	183	31	174	30	30	23	83,61	87,43
SOAPdenovo2 ctg	118	31	182	33	27	21	77,12	82,20
SOAPdenovo2 scf	106	31	182	33	27	21	74,53	80,19
SPAdes ctg	86	20	177	22	21	19	75,58	77,91
SPAdes scf	83	21	183	27	24	18	71,08	78,31
Velvet ctg	120	22	175	26	24	19	80,00	84,17
Velvet scf	108	22	175	26	24	19	77,78	82,41

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>M. abscessus</i> 6G-0125-R - HiSeq	102,20	22,07	117,00	22,00	17,73	13,87	82,65	86,43
Abyss ctg	87	18	112	19	19	15	78,16	82,76
ABYSS scf	59	25	119	26	26	19	55,93	67,80
CABOG ctg	127	21	113	20	20	16	84,25	87,40
CABOG scf	109	21	113	20	20	16	81,65	85,32
MaSuRCA ctg	57	19	115	19	19	13	66,67	77,19
MaSuRCA scf	50	19	115	19	19	13	62,00	74,00
MIRA ctg	59	19	347	19	18	13	69,49	77,97
SGA ctg	281	15	107	15	15	12	94,66	95,73
SGA scf	281	14	108	15	15	12	94,66	95,73
SOAPdenovo2 ctg	65	18	114	17	16	14	75,38	78,46
SOAPdenovo2 scf	61	18	114	17	16	14	73,77	77,05
SPAdes ctg	60	12	106	13	12	11	80,00	81,67
SPAdes scf	52	15	117	16	15	14	71,15	73,08
Velvet ctg	132	17	111	18	18	13	86,36	90,15
Velvet scf	53	17	111	18	18	13	66,04	75,47

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>M. abscessus</i> 6G-0125-R - MiSeq	300,73	57,60	171,40	57,47	46,67	26,27	84,48	91,26
Abyss ctg	131	20	113	22	22	18	83,21	86,26
ABYSS scf	129	20	113	22	22	18	82,95	86,05
CABOG ctg	856	30	119	28	28	24	96,73	97,20
CABOG scf	847	30	119	28	28	24	96,69	97,17
MaSuRCA ctg	316	166	259	158	128	21	59,49	93,35
MaSuRCA scf	315	166	259	158	127	21	59,68	93,33
MIRA ctg	103	158	543	168	82	76	20,39	26,21
SGA ctg	612	21	108	21	21	19	96,57	96,90
SGA scf	611	21	114	21	21	19	96,56	96,89
SOAPdenovo2 ctg	84	25	120	26	25	23	70,24	72,62
SOAPdenovo2 scf	72	25	120	26	25	23	65,28	68,06
SPAdes ctg	53	37	130	44	39	26	26,42	50,94
SPAdes scf	53	37	130	44	40	28	24,53	47,17
Velvet ctg	189	54	162	48	46	27	75,66	85,71
Velvet scf	140	54	162	48	46	27	67,14	80,71

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>R. sphaeroides</i> 2.4.1 - HiSeq	362,87	34,00	305,40	32,87	31,67	27,00	91,27	92,56
Abyss ctg	464	53	321	51	48	41	89,66	91,16
ABYSS scf	459	57	326	55	52	43	88,67	90,63
CABOG ctg	531	35	294	29	29	23	94,54	95,67
CABOG scf	320	35	294	29	29	23	90,94	92,81
MaSuRCA ctg	89	23	305	25	23	16	74,16	82,02
MaSuRCA scf	84	23	305	25	23	16	72,62	80,95
MIRA ctg	446	26	303	26	26	20	94,17	95,52
SGA ctg	511	42	309	42	40	39	92,17	92,37
SGA scf	452	32	293	28	27	23	94,03	94,91
SOAPdenovo2 ctg	613	28	304	27	27	25	95,60	95,92
SOAPdenovo2 scf	472	28	304	27	27	25	94,28	94,70
SPAdes ctg	134	40	312	42	39	34	70,90	74,63
SPAdes scf	104	40	307	39	37	33	64,42	68,27
Velvet ctg	513	24	302	24	24	22	95,32	95,71
Velvet scf	251	24	302	24	24	22	90,44	91,24

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução		
	A	B	C	D	E	F	1	2	
<i>R. sphaeroides</i> 2.4.1 - MiSeq	270,13	39,80	307,33	39,27		37,13	34,13	86,25	87,27
Abyss ctg	339	49	325	49		46	45	86,43	86,73
ABySS scf	339	49	325	49		46	45	86,43	86,73
CABOG ctg	145	30	266	23		23	21	84,14	85,52
CABOG scf	131	30	266	23		23	21	82,44	83,97
MaSuRCA ctg	53	28	288	28		26	22	50,94	58,49
MaSuRCA scf	47	28	288	28		26	22	44,68	53,19
MIRA ctg	528	82	365	82		78	69	85,23	86,93
SGA ctg	726	31	307	31		31	30	95,73	95,87
SGA scf	656	21	274	22		21	21	96,80	96,80
SOAPdenovo2 ctg	302	31	314	31		31	30	89,74	90,07
SOAPdenovo2 scf	180	31	314	31		31	30	82,78	83,33
SPAdes ctg	100	43	317	50		44	44	56,00	56,00
SPAdes scf	69	44	323	50		45	42	34,78	39,13
Velvet ctg	307	50	319	46		43	35	85,99	88,60
Velvet scf	130	50	319	46		43	35	66,92	73,08

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução		
	A	B	C	D	E	F	1	2	
<i>S. aureus</i> M0927 - MiSeq	60,93	21,53	55,27	20,07		17,80	10,73	70,79	82,39
Abyss ctg	89	30	66	13		12	8	86,52	91,01
ABYSS scf	81	32	68	15		14	8	82,72	90,12
CABOG ctg	56	24	59	24		21	15	62,50	73,21
CABOG scf	53	24	59	24		21	15	60,38	71,70
MaSuRCA ctg	35	16	51	17		15	9	57,14	74,29
MaSuRCA scf	34	16	50	17		15	8	55,88	76,47
MIRA ctg	68	33	62	34		29	19	57,35	72,06
SGA ctg	130	30	59	29		23	18	82,31	86,15
SGA scf	88	13	44	11		11	9	87,50	89,77
SOAPdenovo2 ctg	48	19	51	21		18	10	62,50	79,17
SOAPdenovo2 scf	46	19	51	21		18	10	60,87	78,26
SPAdes ctg	40	16	51	17		17	9	57,50	77,50
SPAdes scf	40	15	52	16		15	7	62,50	82,50
Velvet ctg	57	18	53	21		19	8	66,67	85,96
Velvet scf	49	18	53	21		19	8	61,22	83,67

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução		
	A	B	C	D	E	F	1	2	
<i>V. cholerae</i> CO1032(5) - HiSeq	125,13	19,13	71,07	17,73		13,87	9,73	88,92	92,22
Abyss ctg	121	20	61	20		16	13	86,78	89,26
ABYSS scf	87	20	63	22		19	14	78,16	83,91
CABOG ctg	127	12	51	11		11	9	91,34	92,91
CABOG scf	108	12	51	11		11	9	89,81	91,67
MaSuRCA ctg	80	18	57	19		11	7	86,25	91,25
MaSuRCA scf	77	18	57	19		11	7	85,71	90,91
MIRA ctg	115	8	230	9		5	5	95,65	95,65
SGA ctg	285	25	64	14		14	10	95,09	96,49
SGA scf	274	20	59	19		14	10	94,89	96,35
SOAPdenovo2 ctg	81	27	67	21		17	8	79,01	90,12
SOAPdenovo2 scf	58	27	67	21		17	8	70,69	86,21
SPAdes ctg	121	12	52	10		9	7	92,56	94,21
SPAdes scf	95	12	53	14		11	9	88,42	90,53
Velvet ctg	183	28	67	28		21	15	88,52	91,80
Velvet scf	65	28	67	28		21	15	67,69	76,92

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>V. cholerae</i> CO1032(5) - MiSeq	154,13	33,27	71,27	32,00	28,93	18,80	81,23	87,80
Abyss ctg	149	18	58	20	20	16	86,58	89,26
ABYSS scf	149	18	58	20	20	16	86,58	89,26
CABOG ctg	241	14	53	14	14	11	94,19	95,44
CABOG scf	241	14	53	14	14	11	94,19	95,44
MaSuRCA ctg	142	49	84	43	36	14	74,65	90,14
MaSuRCA scf	142	50	85	44	37	14	73,94	90,14
MIRA ctg	96	110	155	111	100	58	-4,17	39,58
SGA ctg	247	14	50	12	12	10	95,14	95,95
SGA scf	240	20	57	18	18	13	92,50	94,58
SOAPdenovo2 ctg	147	32	70	30	27	18	81,63	87,76
SOAPdenovo2 scf	136	32	70	30	27	18	80,15	86,76
SPAdes ctg	73	36	75	36	31	27	57,53	63,01
SPAdes scf	72	36	75	36	30	26	58,33	63,89
Velvet ctg	124	28	63	26	24	15	80,65	87,90
Velvet scf	113	28	63	26	24	15	78,76	86,73

Espécie / montador	GAGE	jFGap	FGCSkew	jFGap	Result-1	Result-2	% de redução	
	A	B	C	D	E	F	1	2
<i>X. axonopodis</i> pv. <i>Manihotis</i> UA323 - HiSeq	149,67	63,13	250,67	66,20	61,20	50,80	59,11	19,54
Abyss ctg	134	75	256	75	71	67	47,01	50,00
ABYSS scf	121	78	260	78	73	69	39,67	42,98
CABOG ctg	89	29	223	29	24	20	73,03	77,53
CABOG scf	89	29	224	31	27	24	69,66	73,03
MaSuRCA ctg	99	27	224	25	22	18	77,78	81,82
MaSuRCA scf	98	26	222	25	23	18	76,53	81,63
MIRA ctg	472	189	385	220	187	95	60,38	79,87
SGA ctg	199	74	258	76	73	70	63,32	64,82
SGA scf	189	69	249	69	68	67	64,02	64,55
SOAPdenovo2 ctg	146	68	251	67	65	61	55,48	58,22
SOAPdenovo2 scf	124	68	251	67	65	61	47,58	50,81
SPAdes ctg	117	37	226	40	37	30	68,38	74,36
SPAdes scf	115	38	227	45	39	32	66,09	72,17
Velvet ctg	134	70	252	73	72	65	46,27	51,49
Velvet scf	119	70	252	73	72	65	39,50	45,38
Média geral	172,95	36,37	158,09	36,3	32,53	23,56	81,19	86,38

TABELA 19 COMPARAÇÃO DOS RESULTADOS PELAS MEDIDAS DO N50, N75, L50 E L75.

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>A. hydrophila</i> SSU - HiSeq	330086,40	486043,67	155194,87	348002,47	8,53	3,87	16,73	6,80
ABySS ctg	237457	494270	121603	349770	7	4	15	7
ABySS scf	268392	493890	137136	283363	6	4	13	8
CABOG ctg	278382	495425	133236	338708	6	4	12	7
CABOG scf	278382	495425	189722	338708	6	4	11	7
MaSuRCA ctg	828647	593831	235264	491514	3	3	7	5
MaSuRCA scf	862315	593831	245339	491514	3	3	6	5
MIRA ctg	244066	301518	137773	159472	10	6	16	12
SGA ctg	67129	522070	38026	295072	25	3	50	6
SGA scf	67129	558543	39052	490694	25	3	49	5
SOAPdenovo2 ctg	243851	388749	136903	337492	8	4	15	7
SOAPdenovo2 scf	334435	388749	230185	337492	5	4	9	7
SPAdes ctg	379681	490966	129901	388900	5	4	12	6
SPAdes scf	379681	490966	230049	388900	5	4	9	6
Velvet ctg	180361	491211	93861	264219	9	4	18	7
Velvet scf	301388	491211	229873	264219	5	4	9	7

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>B. cereus</i> ATCC 10987 - MiSeq	193160,47	2368047,07	92515,73	734182,33	19,80	1,93	41,40	4,00
ABySS ctg	130570	321407	69698	156642	12	4	26	11
ABySS scf	135613	253241	74995	190193	11	5	24	12
CABOG ctg	155352	2036756	76333	2036756	13	2	26	2
CABOG scf	431479	2036756	211009	2036756	4	2	9	2
MaSuRCA ctg	246697	3990675	170659	675048	6	1	13	2
MaSuRCA scf	337861	3980432	209762	463928	4	1	10	2
MIRA ctg	116480	578335	52231	465985	14	4	32	6
SGA ctg	25767	3822494	16197	577461	64	1	129	2
SGA scf	25876	3824409	16762	577660	63	1	126	2
SOAPdenovo2 ctg	246346	3365030	78462	458355	6	1	16	3
SOAPdenovo2 scf	456635	3365030	128609	458355	4	1	10	3
SPAdes ctg	103691	2726563	60216	1359438	17	1	34	2
SPAdes scf	212506	2803866	120480	579122	8	1	16	3
Velvet ctg	24786	1207856	13917	488518	65	2	136	4
Velvet scf	247748	1207856	88406	488518	6	2	14	4

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>B. cereus</i> VD118 - HiSeq	62416,33	277112,27	24661,00	156352,93	39,67	4,93	62,20	10,20
ABySS ctg	42572	261213	19054	162298	38	5	92	12
ABySS scf	48439	226524	20272	180476	32	6	81	12
CABOG ctg	79405	351059	31477	187850	17	4	43	9
CABOG scf	80931	351059	31510	187850	16	4	40	9
MaSuRCA ctg	97165	321434	34913	147012	16	5	40	11
MaSuRCA scf	103595	316758	34913	156484	15	5	39	12
MIRA ctg	43373	232562	15003	126713	28	5	85	12
SGA ctg	23397	257307	10868	110731	69	5	155	11
SGA scf	22086	220913	8897	100752	215		0	0
SOAPdenovo2 ctg	57780	251748	21257	123854	30	6	74	13
SOAPdenovo2 scf	58377	251748	22983	123854	27	6	67	13
SPAdes ctg	94312	302932	37419	215692	15	4	38	9
SPAdes scf	94312	307943	37419	225388	15	4	38	8
Velvet ctg	44126	251742	20909	148170	33	5	74	11
Velvet scf	46375	251742	23021	148170	29	5	67	11

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>B. fragilis</i> HMW 615 - HiSeq	121265,13	567196,33	61223,67	301485,07	16,80	3,93	34,67	7,53
ABYSS ctg	106769	589718	56017	313284	19	4	42	7
ABYSS scf	126082	587877	58781	269349	19	4	38	8
CABOG ctg	95840	550210	42294	323543	18	4	38	7
CABOG scf	95840	550211	47426	323543	18	4	37	7
MaSuRCA ctg	158716	663436	74628	550573	11	3	23	5
MaSuRCA scf	158716	771485	74628	663433	11	3	23	4
MIRA ctg	134263	254432	84346	129390	12	8	24	15
SGA ctg	45047	370803	26391	263506	38	5	76	9
SGA scf	51632	561525	30518	200815	34	4	66	8
SOAPdenovo2 ctg	125627	707242	58562	202611	14	3	30	7
SOAPdenovo2 scf	132255	707242	60952	202611	13	3	28	7
SPAdes ctg	157732	520951	86371	254171	10	4	22	8
SPAdes scf	157732	549841	87246	286941	10	4	21	7
Velvet ctg	125214	561486	59460	269253	14	3	29	7
Velvet scf	147512	561486	70735	269253	11	3	23	7

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>M. abscessus</i> 6G-0125-R - HiSeq	141046,60	687615,40	78565,33	370133,07	18,60	2,93	36,93	5,87
ABySS ctg	119446	473197	68977	294233	15	4	29	7
ABySS scf	147937	473234	113145	278360	8	4	18	9
CABOG ctg	81416	587078	42966	342307	21	3	43	6
CABOG scf	94359	587078	50959	342307	19	3	37	6
MaSuRCA ctg	246830	551132	110872	322228	9	3	17	6
MaSuRCA scf	246830	551132	122899	322228	9	3	17	6
MIRA ctg	147272	587763	109542	492304	12	3	22	5
SGA ctg	28734	1199465	17153	448434	55	2	110	5
SGA scf	28734	1198999	17153	446848	55	2	110	5
SOAPdenovo2 ctg	148639	588528	88169	369809	11	3	22	6
SOAPdenovo2 scf	150256	588528	95162	369809	10	3	20	6
SPAdes ctg	150258	1199284	94999	492081	10	2	20	5
SPAdes scf	215724	551219	114286	334580	9	3	18	6
Velvet ctg	60955	588797	37038	348234	28	3	54	5
Velvet scf	248309	588797	95160	348234	8	3	17	5

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>M. abscessus</i> 6G-0125-R - MiSeq	79155,00	389556,33	42751,93	249374,33	57,67	5,80	120,07	10,73
ABySS ctg	70424	520421	39013	334866	24	4	48	7
ABySS scf	73179	520421	39013	334866	23	4	47	7
CABOG ctg	8655	421052	4755	218317	176	5	378	9
CABOG scf	9070	421052	4805	218317	171	5	370	9
MaSuRCA ctg	36211	437292	18405	332798	52	5	104	8
MaSuRCA scf	36211	435889	18481	334201	52	5	104	8
MIRA ctg	89612	126512	46469	78572	15	18	36	34
SGA ctg	12890	519353	7614	333361	126	4	255	7
SGA scf	12890	520553	7614	335186	126	4	255	7
SOAPdenovo2 ctg	131561	347630	68567	218504	14	4	29	9
SOAPdenovo2 scf	147990	347630	79670	218504	13	4	24	9
SPAdes ctg	220161	302660	122266	192025	8	6	16	12
SPAdes scf	220161	252408	122266	191266	8	7	16	13
Velvet ctg	47327	335236	25959	199916	35	6	71	11
Velvet scf	70983	335236	36382	199916	22	6	48	11

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>R. sphaeroides</i> 2.4.1 - HiSeq	50721,87	467189,27	28355,00	215958,67	67,53	3,33	141,47	6,93
ABySS ctg	13523	284948	8662	131220	98	5	205	11
ABySS scf	13523	266342	8674	139066	97	6	204	11
CABOG ctg	12421	493289	7179	222671	111	3	222	6
CABOG scf	23585	493289	12169	222671	57	3	117	6
MaSuRCA ctg	176783	698991	102004	284085	10	3	18	5
MaSuRCA scf	196511	698991	102004	284085	9	3	17	5
MIRA ctg	18568	493885	9191	234259	69	3	154	6
SGA ctg	12872	333449	6960	174004	96	3	204	8
SGA scf	13674	333349	7216	222097	89	3	185	7
SOAPdenovo2 ctg	11128	493881	5955	222587	121	3	259	6
SOAPdenovo2 scf	15950	493881	8188	222587	90	3	187	6
SPAdes ctg	74486	493881	47310	212293	19	3	38	7
SPAdes scf	127911	493881	73118	222631	10	3	22	6
Velvet ctg	13807	467891	7500	222562	98	3	207	7
Velvet scf	36086	467891	19195	222562	39	3	83	7

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>R. sphaeroides</i> 2.4.1 - MiSeq	62356,73	226651,46	36860,60	143970,62	51,07	7,08	105,87	13,62
ABySS ctg	22050	169949	11736	129944	67	10	137	18
ABySS scf	22050	169949	11736	129944	67	10	137	18
CABOG ctg	41794	270924	22363	170790	30	5	64	10
CABOG scf	45887	270924	25501	170790	26	5	55	10
MaSuRCA ctg	142742	270321	102371	174203	12	6	21	10
MaSuRCA scf	165131	270321	104435	174203	11	6	19	10
MIRA ctg	15792	164072	8236	72855	90	8	191	17
SGA ctg	9224	250749	4941	142423	150	6	321	13
SGA scf	9086		4930		136		291	
SOAPdenovo2 ctg	33829	251087	16889	142633	44	6	88	13
SOAPdenovo2 scf	45133	251087	27903	142633	32	6	63	13
SPAdes ctg	118093		53389		15		30	
SPAdes scf	151794	202404	104015	142764	11	8	20	15
Velvet ctg	24347	202341	14804	139218	58	8	115	15
Velvet scf	88399	202341	39660	139218	17	8	36	15

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>S. aureus</i> M0927 - HiSeq	135066,40	555324,73	73181,33	359672,07	8,73	2,33	17,87	4,33
ABySS ctg	64963	669711	38351	437698	13	2	29	4
ABySS scf	73887	669711	38856	438067	12	2	27	4
CABOG ctg	111165	435989	56580	330564	7	2	16	4
CABOG scf	153356	435989	62375	330564	6	2	13	4
MaSuRCA ctg	221821	453254	123428	396538	5	2	9	4
MaSuRCA scf	221821	447955	123428	401885	5	2	9	4
MIRA ctg	132448	275293	50006	232112	7	4	16	7
SGA ctg	39954	333949	28545	205590	22	3	43	6
SGA scf	51537	500001	34718	334718	15	2	31	4
SOAPdenovo2 ctg	146332	612876	92436	430362	7	2	13	4
SOAPdenovo2 scf	146332	612876	92436	430362	7	2	13	4
SPAdes ctg	187080	1005964	105739	349706	6	2	11	3
SPAdes scf	187080	1050091	105739	349983	6	2	11	3
Velvet ctg	122507	413106	52580	363466	7	3	15	5
Velvet scf	165713	413106	92503	363466	6	3	12	5

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>V. cholerae</i> CO1032(5) - HiSeq	119957,40	1045511,60	55909,93	397999,53	17,27	2,20	36,47	4,20
ABySS ctg	94508	479770	58395	275156	13	3	27	5
ABySS scf	217596	992852	80568	275408	6	2	15	5
CABOG ctg	61249	740634	29636	355373	19	2	43	4
CABOG scf	67078	740634	34758	355373	16	2	36	4
MaSuRCA ctg	241604	996806	85155	472010	6	2	13	4
MaSuRCA scf	246505	996806	85155	472009	6	2	13	4
MIRA ctg	92000	1039874	56477	741987	14	2	27	3
SGA ctg	23808	564738	12382	358919	49	2	103	4
SGA scf	24152	564509	12815	263928	48	3	102	5
SOAPdenovo2 ctg	135118	2513108	65780	532404	10	1	21	2
SOAPdenovo2 scf	200529	2513108	102918	532404	6	1	13	2
SPAdes ctg	83518	2264500	46841	475931	16	1	32	3
SPAdes scf	98274	564423	52916	359323	13	2	26	4
Velvet ctg	40877	355456	22754	249884	30	4	62	7
Velvet scf	172545	355456	92099	249884	7	4	14	7

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>V. cholerae</i> CO1032 - MiSeq	93313,93	455143,47	47503,67	272481,93	20,33	4,40	43,47	7,87
ABYSS ctg	60973	470868	27970	422454	21	3	44	5
ABYSS scf	60973	470868	27970	422454	21	3	44	5
CABOG ctg	33710	569339	17245	296993	34	3	74	5
CABOG scf	33710	569339	17245	296993	34	3	74	5
MaSuRCA ctg	76131	577012	45736	216064	16	3	33	6
MaSuRCA scf	76131	576880	45736	216064	16	3	33	6
MIRA ctg	112926	152804	59711	96266	9	14	20	26
SGA ctg	27573	893192	15449	441050	45	2	94	4
SGA scf	27931	373230	15534	270891	44	4	92	7
SOAPdenovo2 ctg	71357	442168	37218	246179	15	4	34	6
SOAPdenovo2 scf	91942	442168	39820	246179	14	4	32	6
SPAdes ctg	262160	236237	136216	166536	5	6	11	12
SPAdes scf	262160	250329	136216	153646	5	6	11	11
Velvet ctg	92036	401359	43442	297730	15	4	31	7
Velvet scf	109996	401359	47047	297730	11	4	25	7

Espécies / montadores	Média de N50		Média de N75		Média de L50		Média de L75	
	GAGE	GF	GAGE	GF	GAGE	GF	GAGE	GF
<i>X. axonopodis</i> pv. Manihotis UA323 - HiSeq	94463,20	269437,07	47683,80	119931,20	20,13	10,07	40,47	19,93
ABySS ctg	89864	120035	47805	66728	21	14	40	26
ABySS scf	94163	120035	55842	66728	19	14	36	26
CABOG ctg	107676	610232	55276	267770	16	4	32	7
CABOG scf	105803	610232	58882	213806	17	4	32	8
MaSuRCA ctg	117883	610409	61875	227363	15	3	30	7
MaSuRCA scf	117883	610409	61875	227362	15	3	30	7
MIRA ctg	91268	91219	21435	55400	21	20	53	41
SGA ctg	49072	113537	23614	63700	35	15	70	28
SGA scf	49072	119999	24538	66701	35	13	69	26
SOAPdenovo2 ctg	79834	131760	41329	66790	21	11	44	23
SOAPdenovo2 scf	91216	131760	43762	66790	19	11	38	23
SPAdes ctg	117467	265902	63036	138922	16	6	30	12
SPAdes scf	117467	265901	63036	138922	16	7	30	13
Velvet ctg	83025	120063	42656	65993	20	13	40	26
Velvet scf	105255	120063	50296	65993	16	13	33	26
Média Geral	123584,12	654320,94	62033,91	307613,61	28,84	4,37	58,13	8,44

FONTE: O Autor

10 ANEXO 4 – CERTIFICADO DE REGISTRO DO JTRIMMER



REPÚBLICA FEDERATIVA DO BRASIL
 MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA E COMÉRCIO EXTERIOR
 INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL

**CERTIFICADO DE REGISTRO
 DE PROGRAMA DE COMPUTADOR**

Processo: BR 51 2014 001121-6

O INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL expede o presente Certificado de Registro de Programa de Computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de criação indicada, em conformidade com o art. 3º da Lei Nº 9.609, de 19 de Fevereiro de 1998, e arts. 1º e 2º do Decreto 2.566 de 20 de Abril de 1998.

Título: **J TRIMMER**

Criação: 18 de agosto de 2010

Titulares: UNIVERSIDADE FEDERAL DO PARANÁ (75.095.879/0001-49)

Autores: ALEXANDRE QUADROS LEJAMBRE (034.684.219-88)
 BRUNO THIAGO DE LIMA NICHIO (947.570.462-00)
 DIEVAL GUILIELMI (003.434.629-08)
 FÁBIO DE OLIVEIRA PEDROSA (076.894.099-87)
 JERONZANUNES MARCHAUKOSKI (078.020.159-87)
 MARIA BERÊNICE REYNAUD STEFFENS (519.158.399-34)
 ROBERTO TADEU RAITZ (724.350.309-10)

Linguagem: JAVA

Aplicação: BL-02, BL-04, BL-07

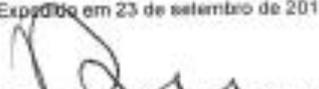
Tipo Prog.: TC-01, UT-01

DOCUMENTAÇÃO TÉCNICA EM DEPOSITO SOB SIGILO ATÉ 30/09/2024.

Os Direitos Patrimoniais relativos ao programa de computador objeto do presente Registro foram cedidos aos Criadores para o Titular, na data de 15 de setembro de 2010, conforme documentação.

A exclusividade de comercialização deste programa de computador não tem a abrangência relativa à exclusividade de fornecimento estabelecida pelo art. 25, I, da Lei nº 9.609, de 21 de Junho de 1998, para fins de inviabilidade de aplicação para compras pelo poder público.

Expedido em 23 de setembro de 2015


Bruno Belto de Almeida Neves
 Diretor de Contratos, Indicações Geográficas e Registros

