Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz
Instituto Oswaldo Cruz

INSTITUTO OSWALDO CRUZ
Pós-Graduação em Biologia Computacional e Sistemas

ANTONIO CLÁUDIO BELLO RIBEIRO

LASZLO @ GALAXY - Um protótipo de serviço de montagem de genomas a partir de dados de sequenciamento de próxima geração (NGS)

Dissertação apresentada ao Instituto Oswaldo Cruz como parte dos requisitos para obtenção do título de Mestre em Biologia Computacional e Sistemas

Orientador (es): Prof. Dr. André Nóbrega Pitaluga
Prof. Dr. Alberto Mártin Rivera Dávila

RIO DE JANEIRO
2012

Ficha catalográfica elaborada pela
Biblioteca de Ciências Biomédicas / ICICT / FIOCRUZ - RJ

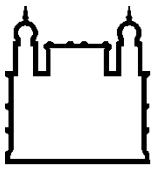
R484 Ribeiro, Antonio Cláudio Bello

Laszlo @ Galaxy: um protótipo de serviço de montagem de genomas a partir de dados de sequenciamento de próxima geração (NGS) / Antonio Cláudio Bello Ribeiro. – Rio de Janeiro, 2012.
xxx,245 f. : il. ; 30 cm.

Dissertação (Mestrado) – Instituto Oswaldo Cruz, Pós-Graduação em Biologia Computacional e Sistemas, 2012.
Bibliografia: f. 138-156.

1. Biologia computacional 2. Genoma 3. Genômica. 4. Sequenciamento. 5. NGS. 6. Montagem. 7. Galaxy. I. Título.

CDD 572.8628



Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz
Instituto Oswaldo Cruz

INSTITUTO OSWALDO CRUZ
Pós-Graduação em Biologia Computacional e Sistemas

ANTONIO CLÁUDIO BELLO RIBEIRO

LASZLO @ GALAXY - Um protótipo de serviço de montagem de genomas a partir de dados de sequenciamento de próxima geração (NGS)

Dissertação apresentada ao Instituto Oswaldo Cruz como parte dos requisitos para obtenção do título de Mestre em Biologia Computacional e Sistemas

Orientador (es): Prof. Dr. André Nóbrega Pitaluga
Prof. Dr. Alberto Mártin Rivera Dávila

Aprovada em: 31/08/2012

EXAMINADORES:

Prof. Dr. Marcos Paulo Catanho de Souza (IOC/FIOCRUZ/RJ) **- Presidente**
Prof. Dr. Leonardo Barbosa Koerich (UFRJ/FIOCRUZ/RJ)
Prof. Dr. Rafael Dias Mesquita (UFRJ/RJ)
Prof. Dr. Fabio Faria da Mota (IOC/FIOCRUZ/RJ)
Prof. Dr. Oswaldo Gonçalves Cruz (PROCC/FIOCRUZ/RJ)

Rio de Janeiro, 31 de agosto de 2012.

DEDICATÓRIA

À minha mãe, Vera Lúcia Bello Ribeiro (*in memorium*), que presenciou o início deste trabalho mas, infelizmente, não pôde assistir à sua conclusão.

Saudades, mãe, muitas saudades!

AGRADECIMENTOS

À Gabriela, minha esposa, por todo seu amor, desde 1996, e pelo apoio à minha decisão de largar tudo e passar do sonho de estudar Bioinformática à ação, mesmo sabendo que, para isso, nosso patrimônio teria de ser investido na empreitada. Você é a pessoa mais corajosa que eu conheço! Te amo muito!

À Vitória e ao Laszlo, meus filhos, por terem compreendido os momentos em que tive de me ausentar para estudar ou me dedicar a este trabalho. Vocês são as razões pelas quais, a cada dia, eu tento ser uma pessoa melhor do que aquela do dia anterior.

Aos meus pais Antonio Abel e Vera Lúcia (*in memorium*). Em primeiro lugar, por sempre terem me incentivado a estudar. E, mais recentemente e enquanto isto foi possível, por toda a ajuda prestada com as crianças. Como a Gabriela sempre disse, temos certeza de que os pequenos são mais felizes porque sempre puderam contar com o amor incondicional e generoso de vocês.

À minha sogra Marilene Lima, minha cunhada Carla Castelo Branco, minhas sobrinhas Fernanda Castelo Branco e Juliana Dreyer, meu irmão Marcelo Bello Ribeiro, meu cunhado Ivson Alves, meu amigo Sérgio Dreyer e à nossa querida "Cíntia", também pela ajuda prestada com as crianças e por todo o carinho com elas. É muito bom tê-los por perto e saber que podemos contar com vocês!

Aos demais familiares dos "clãs" Ribeiro, Alvim Lima e Castelo Branco, por todo o apoio de sempre.

Aos meus orientadores André Pitaluga e Alberto Dávila, por todas as ideias, dicas e sugestões recebidas durante o meu período de mestrado. Ao André, especialmente, obrigado pela confiança do tipo "voo cego" em meu trabalho, apesar de todas as atribulações pelas quais passei.

À Dra. Yara Maria Traub-Cseko, pesquisadora-chefe do Laboratório de Biologia Molecular de Parasitos e Vetores do IOC/FIOCRUZ, por ceder um espaço de bancada de seu laboratório de Biologia Molecular para que um engenheiro pudesse fazer experimentos de Bioinformática e, também, por compartilhar sua experiência e conhecimento nos seminários do laboratório.

Ao pesquisador Antonio Tempone, do LBMPV, por sua sabedoria, dicas e conversas.

Aos colegas Rodrigo Jardim e Michel Abanto, doutorandos da PGBCS, pelos conselhos e "orientações" recebidas durante a execução deste projeto. Respectivamente, obrigado pelas dicas de Informática e de Biologia!

À colega Adriana Degrossoli, pelo trabalho conjunto que culminou na tentativa de desenvolvimento da ferramenta *Extract region tool* do protótipo.

Aos colegas Rodrigo Jardim (novamente), Raphael Tavares e Vanessa Emmel, pela grande oportunidade de aprendizado, quando das mudanças no sistema STINGRAY.

A todos os colegas do Laboratório de Biologia Molecular de Parasitos e Vetores e do Laboratório de Biologia Computacional e Sistemas, do IOC/FIOCRUZ, com os quais tive o prazer de conviver, desde meados de 2010. Desculpem-me, mas não vou citar nomes aqui, pois a lista é longa e corro risco de cometer injustiças... :o)

À coordenação da Pós-Graduação de Biologia Computacional e Sistemas e às secretárias Márcia Verônica e Alessandra Portugal, por todas as ajudas concedidas no período.

A todos os colegas das turmas de mestrado e doutorado da PGBCS, especialmente aos colegas da minha turma de mestrado de 2010, Amanda Sutter, Bruno Gabriel e Raphael Tavares, pela troca de informações e compartilhamento das ansiedades, dúvidas, dicas, etc. durante o período do curso.

Ao Dr. Marcos Paulo Catanho de Souza, que além de participar da banca, aceitou ser o revisor deste trabalho. Muito obrigado por todas as dicas na "reta final" de formatação!

Ao Dr. Leonardo Barbosa Koerich, Dr. Rafael Dias Mesquita, Dr. Fabio Faria da Mota e Dr. Oswaldo Gonçalves Cruz, por terem aceitado participar desta banca e pelas sugestões propostas.

À CAPES, pelo apoio financeiro durante o período do trabalho.

MUITO OBRIGADO!

*"**Viva como se fosse morrer amanhã e aprenda***

***como se fosse viver para sempre.**"*

Mahatma Gandhi

Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz
Instituto Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

LASZLO @ GALAXY - Um protótipo de serviço de montagem de genomas a partir de dados de sequenciamento de próxima geração (NGS)

RESUMO

DISSERTAÇÃO DE MESTRADO

Antonio Cláudio Bello Ribeiro

As tecnologias NGS (*Next-Generation Sequencing*), desenvolvidas para reduzir o custo e o tempo do processo de sequenciamento, geram uma grande massa de dados, a um custo relativamente baixo e com grande acurácia. No entanto, as leituras curtas, por elas produzidas, dificultam sobremaneira o processo de montagem de genomas, originando novos problemas computacionais. Para tentar suplantar esses desafios, várias ferramentas de software estão disponíveis e continuam a ser desenvolvidas. Cada um desses pacotes possui vantagens e desvantagens e, na maioria das vezes, se apresenta como uma solução individual, não estando integrado a outros. Além disso, tipicamente é exigido um conhecimento mais avançado de informática para a sua correta instalação, configuração e operação; o que, nem sempre, é a realidade do usuário final. Neste contexto, o projeto nomeado ***LASZLO*** (***L****inkage of* ***A****ssembly* ***S****cripts* ***Z****ero-costed and with* ***L****icense* ***O****pened*) *@ GALAXY* propõe combinar diferentes ferramentas de tratamento de dados de NGS de uso livre, na forma de um protótipo básico de serviço de montagem de genomas, buscando facilitar o trabalho do usuário através da disponibilização de uma interface *Web*, sugestões de parametrização e de fluxos de trabalho para esse tipo de análise. Tomando por base o *framework* Galaxy, foram agregados fluxos de trabalho para montagens de dados de sequenciamento reais de diferentes organismos e provenientes das tecnologias Illumina, SOLiD™ e 454. O caráter aplicado do projeto originou soluções pontuais para atender a necessidades específicas, as quais foram reunidas sob o módulo *NGS: LASZLO's Sandbox*, uma "caixa de ferramentas" especialmente designada às abordagens de montagem do tipo *de novo* e com auxílio de genoma de referência. Durante a pesquisa, o protótipo *LASZLO @ GALAXY* processou, por exemplo, dados de sequenciamento de *Leishmania amazonensis*, contribuindo para um primeiro processo de avaliação do genoma do referido organismo. Atualmente, observa-se que a produção de dados não é o mais o "gargalo" em projetos de sequenciamento, mas sim o fluxo de análise subsequente sobre o material obtido. Muitas vezes, tais dados não se traduzem imediatamente em expansão do conhecimento biológico, devido às dificuldades encontradas pelo biólogo experimental em lidar, não somente com a miríade de ferramentas disponíveis, mas também com fatores como a inerente necessidade de integração entre elas e a implementação de infra-estrutura adequada para a sua operação. Os resultados obtidos no projeto indicam que o sistema proposto, vislumbrado como um eventual serviço institucional ou mesmo de menor âmbito, pode se tornar um aliado do usuário final quanto à manipulação dos dados de NGS.

Palavras-chave
Biologia computacional; Genoma; Genômica; Sequenciamento; NGS; Montagem; Galaxy.

Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz
Instituto Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

LASZLO @ GALAXY - A genome assembly service prototype using Next-Generation Sequencing (NGS) data

ABSTRACT

MSc Dissertation

Antonio Cláudio Bello Ribeiro

The NGS (*Next-Generation Sequencing*) technologies, designed to reduce sequencing process costs and time, generate a huge amount of data, at a relatively low cost and with great accuracy. However, the produced short reads strongly difficult the genome assembly process, originating new computational issues. To overcome those challenges, there are several software tools available and continuously being developed. Each of these tools presents advantages and disadvantages and most of them are isolated, not integrated solutions. Moreover, typically it is required a higher level of computer-literacy for their proper installation, configuration and usage, which, not always, is the end-user reality. In this context, the project named ***LASZLO*** (***L****inkage of* ***A****ssembly* ***S****cripts* ***Z****ero-costed and with* ***L****icense* ***O****pened*) *@ GALAXY* suggests to combine different open source tools for NGS data handling, as a basic prototype service for genome assembly, aiming at simplifying the end-user task by providing a Web interface, suggestions of parametrization and workflows for this kind of analysis. Based on the Galaxy framework, some workflows for the assembly of real sequencing data from different organisms and produced by the Illumina, SOLiD™ and 454 technologies were aggregated. Also, due to the applied characteristic of the project, a few punctual solutions were generated to address specific needs. Those solutions were encapsulated in the *NGS: LASZLO's Sandbox* module, a "toolbox" especially tailored for the *de novo* and reference-guided assembly approaches. During the research, the *LASZLO @ GALAXY* prototype processed, for instance, sequencing data of the *Leishmania amazonensis* organism, contributing for a first evaluating process of its genome. Presently, it's noticed that the data generation is no longer the "bottleneck" of the sequencing projects, but the downstream data analysis. Frequently, the acquired data is not immediately translated into biological knowledge expansion, due to the obstacles met by the experimental biologist when dealing, not only with the myriad of available tools, but also with factors like the inherent need of their integration and the deployment of the adequate infrastructure for their operation. The results achieved during project execution indicate that the proposed system, glimpsed as an eventual institutional service or even as one of smaller scope, might become an end-user's ally in the NGS data manipulation.

Keywords
Computational Biology; Genome; Genomics; Sequencing; NGS; Assembly; Galaxy.

Lista de figuras

Lista de tabelas

Lista de quadros

Lista de abreviaturas

A	Adenina
ABySS	*Assembly By Short Sequences* (Montagem por sequências curtas)
AMD	*Advanced Micro Devices*
AMOS	*A Modular Open-Source consortium* (Consórcio modular e de código aberto)
APS	Adenosina 5'-fosfosulfato
ASCII	*American Standard Code for Information Interchange* (Código padrão americano para intercâmbio de informações)
ASID	*Assembly Assistant for SOLiD™* (Assistente de montagem para *SOLiD™*)
ATP	Adenosina trifosfato
avg	*average* (média)
BAC	*Bacterial Artificial Chromosome* (Cromossomo artificial bacteriano)
BAM	*Binary Alignment/Map* (Alinhamento/Mapa binário)
Bash	*Bourne Again Shell* (*Shell Bourne* novamente)
BFAST	*Blat-like Fast Accurate Search Tool* (Ferramenta de busca acurada e rápida tipo BLAT)
BLAST	*Basic local alignment search tool* (Ferramenta básica de busca de alinhamento local)
BLAT	*the BLAST-like alignment tool* (Ferramenta de alinhamento tipo BLAST)
bp	*base pairs* (pares de bases)
BWA	*Burrows-Wheeler Aligner* (Alinhador *Burrows-Wheeler*)
C	Citosina
CABOG	*Celera Assembler with Best Overlap Graph* (Montador Celera com melhor grafo de sobreposição)
CAP3	*Contig Assembly Program 3* (Programa de montagem de *contigs* 3)
CBI	*Center for Bioinformatics* (Centro para Bioinformática)
CCD	*Charged-coupled device* (Dispositivo de carga acoplada)
CE	*Capillary Electrophoresis* (Eletroforese capilar)
CLI	*Command Line Interface* (Interface de linha de comando)
CPU	*Central Processing Unit* (Unidade Central de Processamento)
dATP	desoxiadenosina trifosfato

dCTP	desoxicitosina trifosfato
dGTP	desoxiguanina trifosfato
dTTP	desoxitimina trifosfato
ddATP	dideoxiadenosina trifosfato
ddCTP	dideoxicitosina trifosfato
ddGTP	dideoxiguanina trifosfato
ddNTP	didesoxinucleotídeo
DDR3	*Double-Data-Rate 3* (Taxa de dados duplicada 3)
ddTTP	dideoxitimina trifosfato
DNA	*Deoxyribonucleic acid* (Ácido desoxirribonucléico)
dNTP	desoxinucleotídeo
EBI	*European Bioinformatics Institute* (Instituto Europeu de Bioinformática)
Edena	*Exact De Novo Assemble* (Montagem *De Novo* exata)
ELAND	*Efficient Large-Scale Alignment of Nucleotide Databases* (Alinhamento em larga escala eficiente de bases de dados de nucleotídeos)
emPCR	*Emulsion PCR* (PCR em emulsão)
EST	*Expressed Sequence Tags* (Etiquetas de sequências expressas)
FM	Ferragina-Manzini
FSF	*Free Software Foundation* (Fundação de Software Livre)
FTP	*File Transfer Protocol* (Protocolo de transferência de arquivo)
G	Guanina
G	bilhão
Gb	Gigabases
GB	Gigabyte
gDNA	DNA genômico
GHz	Gigahertz
GMT	*Greenwich Mean Time* (Hora média de Greenwich)
GNU	*GNU is Not Unix* (GNU não é Unix)
HTML	*Hypertext Marlup Language* (Linguagem de Marcação de Hipertexto)
Ibis	*Improved base identification system* (Sistema de identificação de base aprimorado)
iCORN	*Iterative correction of reference nucleotides* (Correção iterativa de nucleotídeos de referência)
IMAGE	*Integrative Mapping and Assembly for Gap Elimination* (Mapeamento e montagem integrativos para eliminação de lacunas)
InDels (ou *Indels*)	*Insertions and Deletions* (Inserções e deleções)
inGAP	*Integrative Next-Generation Genome Analysis Pipeline* (*Pipeline* integrativo de análise de genoma de nova geração)

LASZLO	*Linkage of Assembly Scripts Zero-costed and with License Opened* (Ligação de *scripts* de montagem de custo zero e licença aberta)
LTS	*Long Term Support* (Suporte de longo prazo)
kb	kilobases
kbp	*kilo base pairs* (kilo pares de bases)
kpb	kilo pares de bases
M	milhão
MAQ	*Mapping and Assembly with Qualities* (Mapeamento e montagem com qualidades)
Mb	Megabases
MB	Megabyte
MHz	Megahertz
MIP	*Mixed Integer Programming* (Programação com mistura de inteiros)
MIRA	*Mimicking Intelligent Read Assembly* (Simulação de montagem de leituras inteligente)
Mpb	Milhão de pares de bases
MPI	*Message Passing Interface* (Interface de passagem de mensagem)
MRTG	*Multi Router Traffic Grapher* (Plotador gráfico de tráfego em múltiplos roteadores)
NCBI	*National Center for Biotechnology Information* (Centro Nacional de Informações em Biotecnologia)
NGS	*Next-Generation Sequencing* (Sequenciamento de próxima geração)
NNGS	*Next-next-generation sequencing* (Sequenciamento de próxima-próxima geração)
NP	*Non-Deterministic polynomial time* (Tempo polinomial não-determinístico)
N50	Valor N50
OBF	*Open Bioinformatics Foundation* (Fundação de Bioinformática Aberta)
OLC	*Overlap-layout-consensus* (Sobreposição-Arranjo-Consenso)
OSI	*Open Source Initiative* (Iniciativa pelo código aberto)
OSLay	*Optimal Syntenic Layout of Unfinished Assemblies* (Leiaute sintênico ótimo de montagens não-finalizadas)
pb	pares de bases
PCAP	*Parallel Contig Assembly Program* (Programa de montagem de *contigs* paralelizado)
PCR	*Polymerase Chain Reaction* (Reação em cadeia da polimerase)
PE	*Paired-Ended*
Perl	*Practical extraction and report language* (Linguagem prática de extração e relatórios)
PeRM	*Periodic Seed Mapping* (Alinhamento de sementes periódicas)

Phrap	*PHRagment Assembly Program* (Programa de montagem de fragmentos) ou *PHil's Revised Assembly Program* (Programa de montagem revisado pelo Phil)
Phred	*PHil's Read EDitor* (Editor de leituras do Phil)
PIQA	*Pipeline for Illumina G1 Genome Analyzer Data Quality Assessment* (*Pipeline* para avaliação da qualidade dos dados de *Illumina G1 Genome Analyzer*)
PPi	pirofosfato inorgânico
PRINSEQ	*PReprocessing and INformation of SEQuences* (Pré-processamento e informação de sequências)
PTP	PicoTiterPlate™
QA	*Quality Assessment* (Avaliação de qualidade)
QSRA	*Quality-value guided Short Read Assembler* (Montador de leituras curtas guiado por valor de qualidade)
RAM	*Random Access Memory* (Memória de acesso aleatório)
RNA	*Ribonucleic acid* (Ácido ribonucléico)
RNA-Seq	Sequenciamento de RNA
SAET	*SOLiD™ Accuracy Enhancement Tool* (Ferramenta de aprimoramento de acurácia *SOLiD™*)
SAM	*Sequence Alignment/Map* (Alinhamento/Mapa de sequências)
SATA II	*Serial Advanced Technology Attachment II* (Tecnologia de conexão serial avançada II)
SE	*Single-Ended*
SFF	*Standard Flowgram Format* (Formato padrão de pirograma)
SGA	*String Graph Assembler* (Montador de grafo de *strings*)
SGS	*Second-generation sequencing* (Sequenciamento de segunda geração)
SHARCGS	*SHort read Assembler based on Robust Contig extension for Genome Sequencing* (Montador de leituras curtas baseado em extensão robusta de *contigs* para sequenciamento de genomas)
SHRiMP	*SHort Read Mapping Package* (Pacote de mapeamento de leituras curtas)
SMRT	*Single Molecule Real Time* (Molécula única em tempo real)
SNA	*Single nucleotide addition* (Adição de nucleotídeo único)
SNMP	*Simple Network Management Protocol* (Protocolo simples de gerência de rede)
SNP	*Single Nucleotide Polymorphism* (Polimorfismo de nucleotídeo único)
SOAP	*Short Oligonucleotide Analysis Package* (Pacote de análise de oligonucleotídeos curtos)
SOAP2	*Short Oligonucleotide Analysis Package 2* (Pacote de análise de oligonucleotídeos curtos 2)
SOAPdenovo	*Short Oligonucleotide Analysis Package de novo* (Pacote de análise de oligonucleotídeos curtos *de novo*)
SOCS	*Short Oligonucleotide Color Space* (Oligonucleotídeos curtos em espaço de cores)

SOLiD	*Sequencing by Oligonucleotide Ligation and Detection* (Sequenciamento por ligação e detecção de oligonucleotídeo)
SOPRA	*Statistical Optimization of Paired Read Assembly* (Montagem de leituras pareadas por otimização estatística)
SRA	*Sequence Read Archive* (Arquivo de leituras de sequências) ou *Short read Archive* (Arquivo de leituras curtas)
SSAHA2	*Sequence Search and Alignment by Hashing Algorithm* (Alinhamento e busca de sequências por algoritmo de *hashing* 2)
SSAKE	*Short Sequence Assembly by K-mer search and 3' read Extension* (Montagem de sequências curtas por busca de *k-mer* e extensão de leitura na extremidade 3')
SSPACE	*SSAKE-based Scaffolding of Pre-Assembled Contigs after Extension* (*Scaffolding* baseado em SSAKE de *contigs* pré-montados após extensão)
sstDNA	DNA de fita simples
STINGRAY	*System for Integrate Genomic Resources and Analyses* (Sistema para análise e recursos genômicos integrados)
T	Timina
TAE	*Tris-Acetate-EDTA* (Tris-Acetato-EDTA)
TB	Terabyte
TGS	*Third-generation sequencing* (Sequenciamento de terceira geração)
TIGR	*The Institute for Genomic Research* (O Instituto de Pesquisa Genômica)
UCSC	*University of California, Santa Cruz* (Universidade da Califórnia, Santa Cruz)
URL	*Unified Resource Locator* (Localizador unificado de recursos)
US$	dólar
VCAKE	*Verified Consensus Assembly by K-mer Extension* (Montagem de consenso por extensão de *K-mer* verificada)
WHO	*World Health Organization* (Organização Mundial da Saúde)
XML	*eXtensible Markup Language* (Linguagem de marcação extensível)
XP	*eXtreme Programming* (Programação extrema)
ZOOM	*Zillions Of Oligos Mapped* (Zilhões de oligos mapeados)

Nota: Tradução nossa quanto aos termos colocados entre parênteses.

Sumário

1. Introdução

Diversos aspectos da vida de um organismo se devem à sequência de bases de DNA ou RNA que compõe o seu genoma. A possibilidade de se obter a sequência nucleotídica é considerada, então, como de suma importância no estudo, entendimento e manipulação dos processos biológicos (Walker; Rapley, 1997; Pop et al., 2002). De fato, conhecer o conteúdo de um genoma representa a possibilidade de determinar a função dos genes que o constituem, sendo um tema de bastante interesse, uma vez que o entendimento de sua estrutura, organização e finalidade, pode, por exemplo, revelar quais os genes que codificam RNAs e proteínas (Watson; Berry, 2005), além de abrir outros campos de estudo como a identificação de genes causadores de doenças (diagnóstico médico) (Meldrum, 2000a, 2000b; Mardis, 2008a, 2009, 2011; Davies, 2011; Zhang J et al., 2011), a comparação de genomas de diferentes espécies (estudos evolutivos) (Catanho, 2005; Weiss, 2010), a comparação de genomas e variações de genes específicos entre populações diferentes de uma mesma espécie (genética populacional) (Santos; Bonatto, 2004), os estudos de genotipagem (Tschoeke, 2010), a identificação de alvos terapêuticos (desenvolvimento de possíveis produtos para uso farmacêutico) (Luscombe et al., 2001; Oliva, 2004), o melhoramento genético de plantas economicamente importantes e preservação ecológica e ambiental (Arruda, 2004; Carvalho; Silva, 2010; Milne et al., 2010; Bayer et al., 2011; Thudi et al., 2012), etc. Além disso, segundo Hall (2007), o sequenciamento de genomas impulsionou uma revolução nas ciências biológicas, ao possibilitar, também, o estudo de processos moleculares sob a ótica de sistemas celulares completos, introduzindo, dessa forma, o conceito de *biologia de sistemas*. Esse estudo dos genomas dos organismos, portanto, é o objeto de análise da área conhecida como *Genômica*, a qual "é definida como o estudo sistemático, em escala de genomas completos, para a identificação de contribuições genéticas às condições humanas" (Zhang J et al., 2011).

Em linhas gerais, projetos para estudo de genomas partem de uma fase de sequenciamento, na qual são gerados dados brutos de computador, ou seja, "sequências de DNA" (também chamadas de *leituras* ou *fragmentos*) sem significado biológico (Lemos, 2004; Flicek; Birney, 2009). O objetivo dessa fase é "ler", na ordem correta, as bases

nucleotídicas — adenina (A), guanina (G), citosina (C) e timina (T) — de uma molécula de DNA e convertê-las para cadeias (ou *strings*) de nucleotídeos, que possam ser analisadas posteriormente por uma pessoa ou um programa de computador (Olson, 2009). O sequenciamento de DNA é, então, o processo que fica na fronteira entre a biologia experimental e a biologia computacional, ou seja, é aquele que consegue fazer a identificação da molécula de DNA sob a forma de dados passíveis de serem analisados posteriormente em computador.

Durante aproximadamente 30 anos após a sua publicação em 1977 (Sanger et al., 1977), o método de Sanger de terminação de cadeia por didesoxinucleotídeos foi o padrão utilizado. A despeito dos contínuos melhoramentos da técnica ao longo do tempo, como a introdução dos sistemas de eletroforese capilar e a redução dos custos envolvidos, tal método ainda se mostrou proibitivamente caro e demorado para ser empregado em projetos de sequenciamento de larga escala rotineiros. A demanda, então, por baixo custo e maior rapidez, impulsionou o desenvolvimento das chamadas tecnologias de "próxima geração" ou de "segunda geração"[1] ou, simplesmente, NGS (do inglês *Next-Generation Sequencing*) (Droege; Hill, 2008; Metzker, 2010; Klassen; Currie, 2012). O gráfico da Figura 1.1 demonstra a evolução da produção de pares de bases por "rodada" de experimento ou "corrida" no instrumento sequenciador, conforme a entrada, no mercado, das principais plataformas NGS, na última década (Mardis, 2011).

[1] O método de Sanger automatizado é considerado o de "primeira geração" (Metzker, 2010).

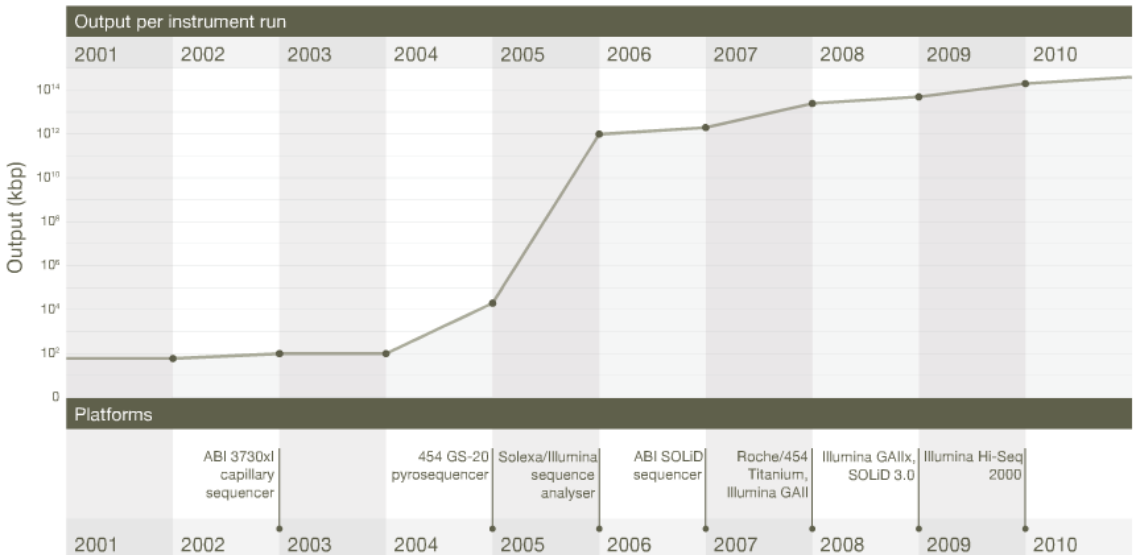


Figura 1.1 - Evolução da capacidade dos sequenciadores na década de 2000. Parte superior do gráfico: escala logarítmica crescente da produção de dados por experimento. Parte inferior: linha do tempo representando os principais marcos de introdução, no mercado, das plataformas de sequenciamento de alto desempenho e respectivas revisões de versão dos equipamentos.
Fonte: Modificado de Mardis, 2011, p.199.

Independentemente da tecnologia, sofisticação dos equipamentos ou das técnicas utilizadas, antes que qualquer análise de cunho biológico possa ser feita sobre os dados gerados, as leituras obtidas devem ser combinadas, por meio de uma etapa conhecida como "montagem", para que os produtos desse processo possam se aproximar o máximo possível da sequência completa de DNA, tal como ela se apresenta originalmente na célula. Essa tarefa de construir sequências maiores, a partir de outras menores, é um dos problemas mais fundamentais em Genômica (Olson, 2009). Isso porque, na prática, nenhuma tecnologia disponível é capaz de "ler" toda a extensão de uma molécula de DNA; somente conseguindo fazê-lo através de vários segmentos menores. A montagem de genomas é, portanto, um elemento crucial para o avanço desse campo do conhecimento. E, com o advento das tecnologias NGS, tal afirmação aparenta ser cada vez mais veemente. Isso procura ser demonstrado através do gráfico da Figura 1.2, o qual detalha a evolução do crescimento das sequências nucleotídicas depositadas no clássico banco de dados GenBank[2] desde a sua criação, em 1982, em relação aos dados de sequenciamento depositados no banco de dados Sequence Read Archive (originalmente conhecido como Short Read Archive; ambos abreviados pela sigla SRA) (Thompson; Milos, 2011). A página *Web*[3] deste último traz a seguinte definição a respeito de sua finalidade[4]: "O Sequence Read Archive (SRA) armazena dados brutos de sequenciamento originados em plataformas de sequenciamento NGS,

[2] http://www.ncbi.nlm.nih.gov/genbank.
[3] http://www.ncbi.nlm.nih.gov/sra.
[4] Tradução nossa.

incluindo Roche 454 GS System®, Illumina Genome Analyzer®, Applied Biosystems SOLiD® System, Helicos Heliscope®, Complete Genomics® e Pacific Biosciences SMRT®". Pois bem, menos de um ano após seu lançamento, o SRA havia ultrapassado o GenBank em quantidade de dados e, pouco tempo depois, já respondia por mais de 95% de todos os depósitos de sequências nucleotídicas. E, tudo isso, sem levar em conta a possibilidade dos dados estarem sub-representados, em virtude de limitações relacionadas à falta de prática dos usuários com os novos formatos produzidos e dificuldades de transferência de grandes volumes de informação, os quais são inerentes às tecnologias NGS (Thompson; Milos, 2011).

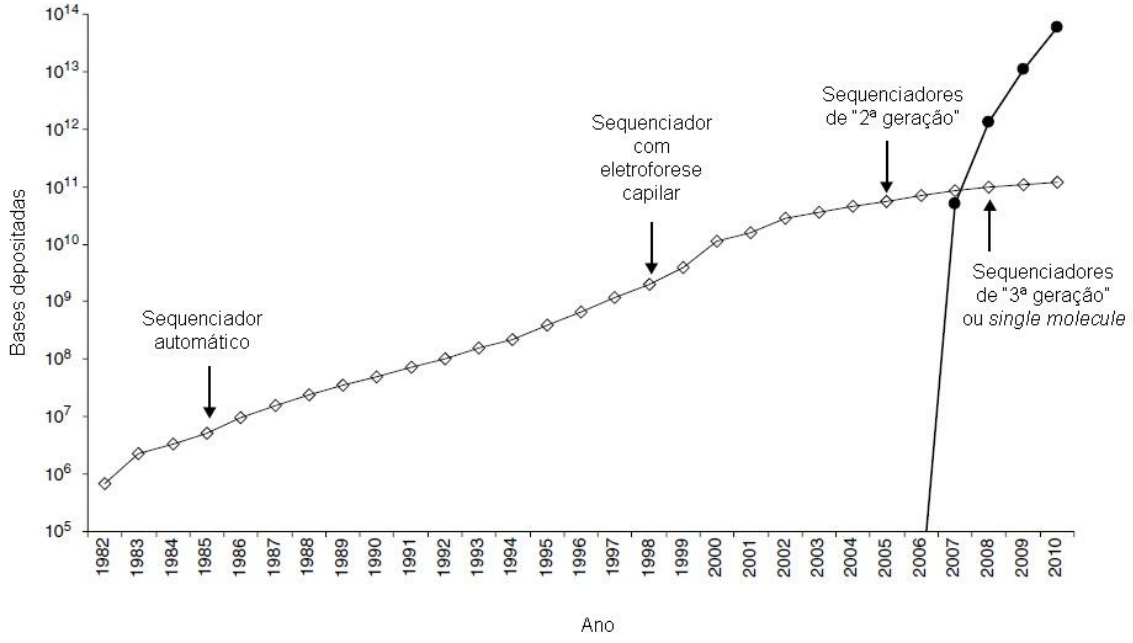


Figura 1.2 - Submissões de sequências aos bancos de dados GenBank e SRA entre 1982 e 2010. As quantidades de "nucleotídeos" submetidos à versão clássica do GenBank (linha fina com "losangos") e ao Sequence Read Archive (linha grossa com "pontos") são mostradas em função do tempo. As "flechas" indicam os marcos de avanços tecnológicos relacionados ao sequenciamento.
Fonte: Modificado de Thompson; Milos, 2011, p.3.

Nas tecnologias NGS, os dados tipicamente são gerados na quantidade de centenas de milhares a dezenas de milhões de leituras muito menores (Thudi et al., 2012) do que as do método Sanger tradicional, as chamadas *short reads* (leituras curtas), e, por isso, algoritmos específicos são necessários para realizar a montagem desses pequenos fragmentos. Apesar da alta produtividade, acurácia e grande cobertura de sequenciamento[5] proporcionadas por tais plataformas, as leituras curtas dificultam de maneira significativa o processo de montagem,

[5] Do inglês *coverage depth*. Uma vez que as plataformas NGS produzem leituras curtas, a cobertura de sequenciamento é uma questão importante. Ela pode ser definida como o número de leituras curtas que se sobrepõem, umas às outras, em uma determinada região genômica. Por exemplo, uma cobertura de 30 vezes (30x ou 30-*fold*) para um determinado gene, significa afirmar que cada um dos nucleotídeos dessa região está representado em, pelo menos, 30 leituras curtas distintas e sobrepostas (Zhang J et al., 2011).

originando novos problemas computacionais. Identificar corretamente leituras idênticas que podem pertencer a regiões repetitivas do genoma de um dado organismo, lidar com erros inerentes à própria química das reações de sequenciamento e manipular o grande volume de milhões de leituras curtas; tudo isso simultaneamente, são algumas das dificuldades encontradas (Sasson, 2010). Para tentar suplantar esses desafios, várias ferramentas de software estão disponíveis e continuam a ser desenvolvidas. As abordagens de solução normalmente se dividem entre alinhadores das leituras contra um genoma de referência (processo também conhecido como *mapeamento*) ou montadores *de novo* (ou *ab initio*), e, por generalização, tais programas são designados simplesmente como "montadores".

Mas a variedade de fatores envolvidos é ainda maior. Se levarmos em conta apenas as soluções que fazem parte desse grupo (montadores), temos que, basicamente por causa do tamanho dos genomas envolvidos, algumas ferramentas são mais direcionadas à montagem de genomas procariotos, porém não de genomas eucariotos (Zhang W et al., 2011). Muitos dos programas são específicos por plataforma de sequenciamento, sendo compatíveis com uma e/ou outra plataforma, mas não o sendo com todas ao mesmo tempo (Li H et al., 2008). Há, ainda, aqueles que são pacotes proprietários e, portanto, só disponibilizados para os usuários que possuem um determinado equipamento sequenciador e, consequentemente, o suporte da empresa fornecedora (Chaisson; Pevzner, 2007) ou, então, por aquisição da respectiva licença de uso (CLC bio, Aarhus, Dinamarca). Os programas, na grande maioria das vezes, se apresentam apenas como ferramentas individuais, não estando integrados com outras soluções que poderiam auxiliar o próprio processo de montagem da sequência final ou nas devidas análises subsequentes.

Além dos pacotes "montadores" propriamente ditos, muitas outras ferramentas/algoritmos, de distintas funções, estão disponíveis para o tratamento dos dados produzidos nas diversas etapas do processo. Com isso, vários são, também, os formatos de arquivos existentes, dificultando as tentativas de integração entre os programas, as quais devem ser implementadas com a utilização de pacotes conversores de formatos (Li H et al., 2009).

Do ponto de vista do usuário final, muitas dessas soluções exigem um conhecimento bem mais avançado de informática para a sua correta instalação, configuração e operação e, nem sempre, esta última é de caráter amigável, envolvendo a emissão de comandos e parâmetros por interação direta com o sistema operacional do computador, através do uso de interface de linha de comando (CLI). E, se por conta própria, for vislumbrada a possibilidade

de integração de algumas das ferramentas para abordar um dado problema, isso também exige, novamente, um nível de conhecimento de informática que, nem sempre, é a realidade do usuário final (Paszkiewicz; Studholme, 2012). Por exemplo, ao se aventurar em tal tipo de empreitada, há de se ter tempo, habilidade e conhecimento para lidar com problemas como a falta de documentação adequada do software, *bugs*[6], dimensionamento da capacidade adequada do hardware, resolução de dependências entre diferentes programas, etc. Paszkiewicz e Studholme (2012) ainda reforçam que, no cenário atual, poucos são os indivíduos realmente proficientes nas técnicas de ambas as biologias experimental e computacional. Dessa forma, podem ser considerados como exemplos de usuários finais de tais ferramentas, os biólogos experimentais que possuem conhecimento para lidar com seus próprios dados computacionais ou aqueles que, por causa da falta desse tipo de competência ou menor experiência em informática, são obrigados a recorrer a algum tipo de suporte advindo de um especialista ou laboratório de bioinformática. Isso, portanto, pode contribuir para o cenário — virtualmente corroborado pelos gráficos ilustrados anteriormente — apresentado por Blankenberg et al. (2011) e, também, compartilhado pelos já mencionados Paszkiewicz e Studholme (2012): nos dias de hoje, a produção de dados não é mais o "gargalo" em projetos de sequenciamento, mas sim o fluxo de análise subsequente que deve ser executado sobre o material que é obtido das plataformas. O surgimento e rápida proliferação das novas tecnologias, apesar de proporcionar uma relativa facilidade quanto ao sequenciamento de genomas, muitas vezes não se traduz imediatamente em expansão do conhecimento biológico.

Do exposto, denota-se muito difícil encontrar uma ferramenta ou sistema disponível que, além de abordar o problema da montagem de genomas, o faça para ambas as estratégias (alinhamento contra um genoma de referência e montagem *de novo*), seja de uso livre, compatível com as principais tecnologias de sequenciamento NGS e consiga integrar as vantagens e funcionalidades de outras ferramentas disponíveis, sem deixar de proporcionar um uso minimamente amigável. Além disso, almejando, como público-alvo, o biólogo experimental que já lide ou tenha adquirido competência para lidar com os seus próprios dados computacionais (Blankenberg et al., 2011; Paszkiewicz; Studholme, 2012) e/ou o bioinformata/equipe de suporte em bioinformática que deseje disponibilizar uma ferramenta

[6] (a) Acepção: substantivo masculino; rubrica: informática — defeito, falha ou erro no código de um programa que provoca seu mau funcionamento (Dicionário eletrônico Houaiss da língua portuguesa - versão 1.0, 2001). (b) Troca de mensagens com Peter Cock, autor do código do *wrapper* do programa MIRA para a plataforma GALAXY, alertando-o sobre a existência de um *bug* no referido *wrapper*, culminou com a confirmação do problema pelo autor do código. Cock P (James Hutton Institute). RES: mira.xml (Galaxy wrapper). [mensagem pessoal]. Mensagem recebida por acbellorib@gmail.com em 09 de maio de 2012.

mais acessível aos seus usuários finais menos experientes em informática, este trabalho foi idealizado. Por meio de uma proposta básica de solução, vislumbrada como um eventual serviço institucional ou mesmo de menor âmbito (por exemplo, para atender a um único laboratório ou grupo de laboratórios), espera-se que o usuário final ganhe mais um aliado na manipulação do grande volume de dados gerados pelos sequenciamentos NGS. Mais detalhadamente, a proposta da solução é combinar e integrar diferentes ferramentas/algoritmos de tratamento de dados de NGS de uso livre, já disponíveis atualmente, na forma de um protótipo básico de serviço de montagem de genomas, buscando facilitar o trabalho do usuário final, por meio da disponibilização de uma interface *Web*, sugestões de parametrização e de fluxos de trabalho para esse tipo de análise. Além disso, ela pretende ser flexível para poder acomodar futuras alterações e refinamentos, tais como, por exemplo, a ampliação da gama de parâmetros disponíveis ao usuário, no que diz respeito a uma determinada ferramenta, e/ou a incorporação de outras ferramentas e alternativas de soluções, sendo estas já existentes ou que constantemente surgem no campo das tecnologias NGS. Uma abordagem desse tipo, além de poder contribuir para a questão prática da montagem de genomas, pode colocar outras funcionalidades ao alcance do usuário, não o deixando refém de apenas um único programa ou solução.

Esta dissertação trata, portanto, da estratégia de implementação desse protótipo básico de serviço de montagem de genomas. Tal protótipo compreende, também, alguns outros programas próprios, desenvolvidos especificamente para facilitar a integração de ferramentas ou para solucionar determinadas questões para as quais não foi encontrada uma solução adequada na literatura. Visando demonstrar a versatilidade da solução, dados de sequenciamento de diferentes organismos e plataformas NGS são empregados e têm seus resultados apresentados. Mais especificamente, o capítulo 2 do texto provê a fundamentação teórica para o trabalho e pode ser entendido como um desmembramento mais aprofundado desta seção introdutória. O capítulo 3 lista os principais objetivos do trabalho. O capítulo 4 contempla os materiais e métodos. O capítulo 5 fornece mais informações sobre os *workflows* (fluxos de trabalho) produzidos na versão básica do protótipo, apresenta os resultados obtidos e os discute. No capítulo 6 são apresentadas as considerações finais sobre esta etapa do projeto e o que são vislumbrados como possíveis trabalhos futuros, visando municiar a solução com mais recursos e funcionalidades. Os Apêndices, além de algumas informações complementares, trazem as representações esquemáticas dos fluxos de trabalho básicos usados nos experimentos e os códigos dos programas *wrappers* das ferramentas implementadas. Tais códigos foram elaborados de maneira relativamente simples e podem, eventualmente, servir

para fins didáticos em empreitadas similares à aqui descrita. Além disso, no Apêndice A, pode ser encontrado um detalhamento maior a respeito de um dos fluxos de trabalho produzidos; detalhamento este que pode servir como exemplo de roteiro ou sugestão de formato de "manual" para a devida utilização do serviço da plataforma pelo usuário final. Por fim, os Anexos trazem algumas informações adicionais pertinentes ao trabalho, tais como algumas listas de ferramentas NGS disponíveis e, de forma resumida, as principais características e estratégias de funcionamento de programas "montadores" de genoma típicos.

2. Fundamentação teórica

Este capítulo procura fornecer uma visão geral a respeito do universo que envolve a questão da montagem de genomas, especialmente no que diz respeito à sua execução a partir de dados obtidos em equipamentos de próxima geração de sequenciamento (NGS). Além disso, a contextualização biológica associada à empreitada é fornecida e uma visão geral sobre as principais tecnologias de sequenciamento é apresentada, bem como aspectos relacionados às ferramentas computacionais tipicamente utilizadas.

2.1. Contextualização biológica

Na célula, o DNA está compactado na forma de cromossomos e o conjunto destes (mais alguns outros elementos extracromossômicos, como, por exemplo, mitocôndrias e/ou cloroplastos) forma seu genoma[7] (Gomes; 2010). O número de cromossomos em um genoma é uma característica da espécie. Organismos procariotos — representados por dois grupos de micro-organismos, bactérias e Archaea, cujas células não possuem núcleo, deixando o DNA livre em seu interior — geralmente possuem apenas um cromossomo, o qual, muitas vezes, é circular. Já, em organismos eucariotos — representados por todas as outras formas de vida, incluindo plantas e animais, e cujas células possuem núcleo, o qual, por sua vez, acomoda e separa o DNA do resto da célula —, os cromossomos se apresentam em pares (e, por esse motivo, as células que os carregam são chamadas de *diplóides*[8]), sendo, cada membro do par, herdado de cada um dos pais.

[7] Excepcionando-se alguns vírus, nos quais o genoma está sob a forma de RNA, o DNA serve como material genético de todos os organismos existentes na Terra (Klug et al., 2006).

[8] Células usadas na reprodução sexual, por exemplo, carregam apenas um membro de cada par de cromossomos e, por isso, são chamadas de *haplóides* (Setubal; Meidanis, 1997). Muitos eucariotos, tais como os fungos, são haplóides também (Griffiths et al., 2012).

A Figura 2.1 exemplifica a hierarquia da vida para um determinado organismo (no caso, o ser humano), desde sua organização como um indivíduo completo até apenas uma de suas moléculas de DNA. E as Figuras 2.2 e 2.3 ilustram as diferenças estruturais existentes entre organismos procariotos, vírus e eucariotos.

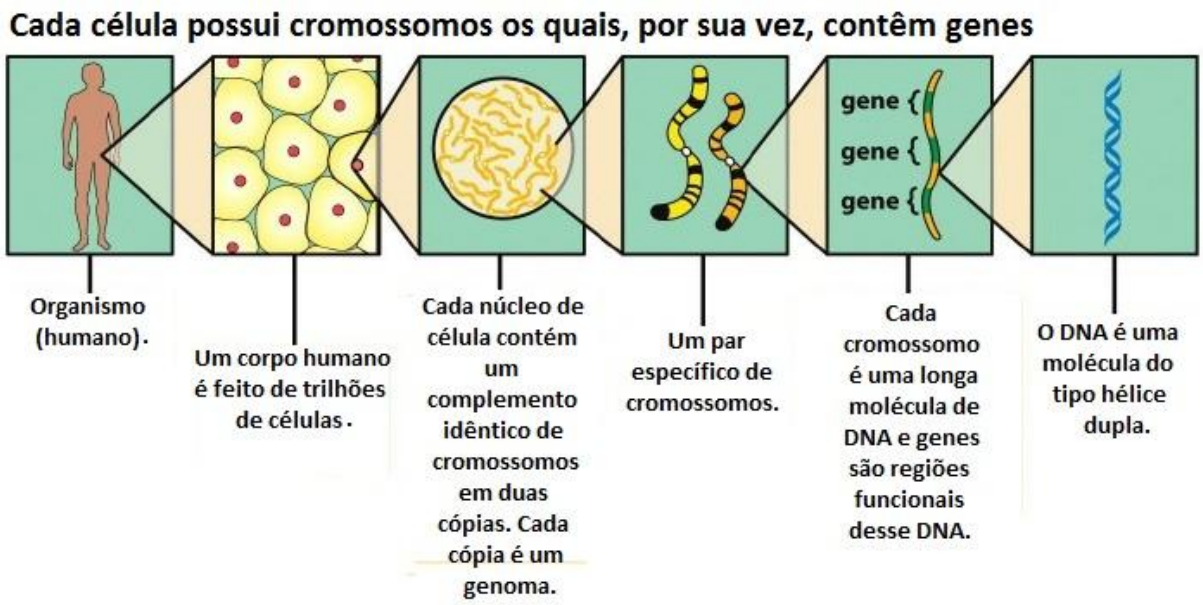


Figura 2.1 - Do organismo, como um todo, ao seu DNA.
Fonte: Modificado de Griffiths et al., 2012, p.3.

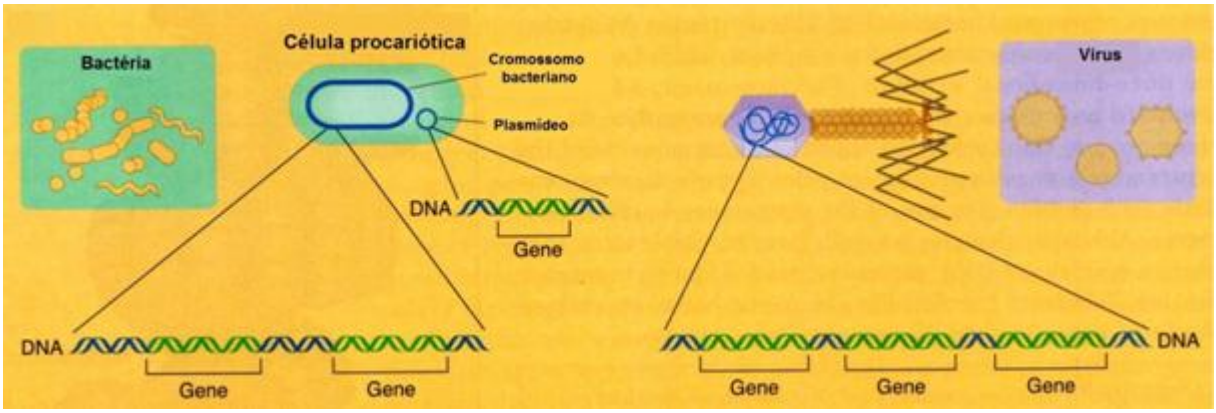


Figura 2.2 - Comparação estrutural dos genomas de procariotos, vírus e eucariotos (estes na Figura 2.3, abaixo). Todos contêm cromossomos onde residem os genes, mas existem algumas diferenças estruturais entre os genomas. Nos procariotos, por exemplo, cromossomos são circulares, enquanto nos vírus e nos cromossomos nucleares eucarióticos eles são lineares.
Fonte: Modificado de Griffiths et al., 2012, p.8.

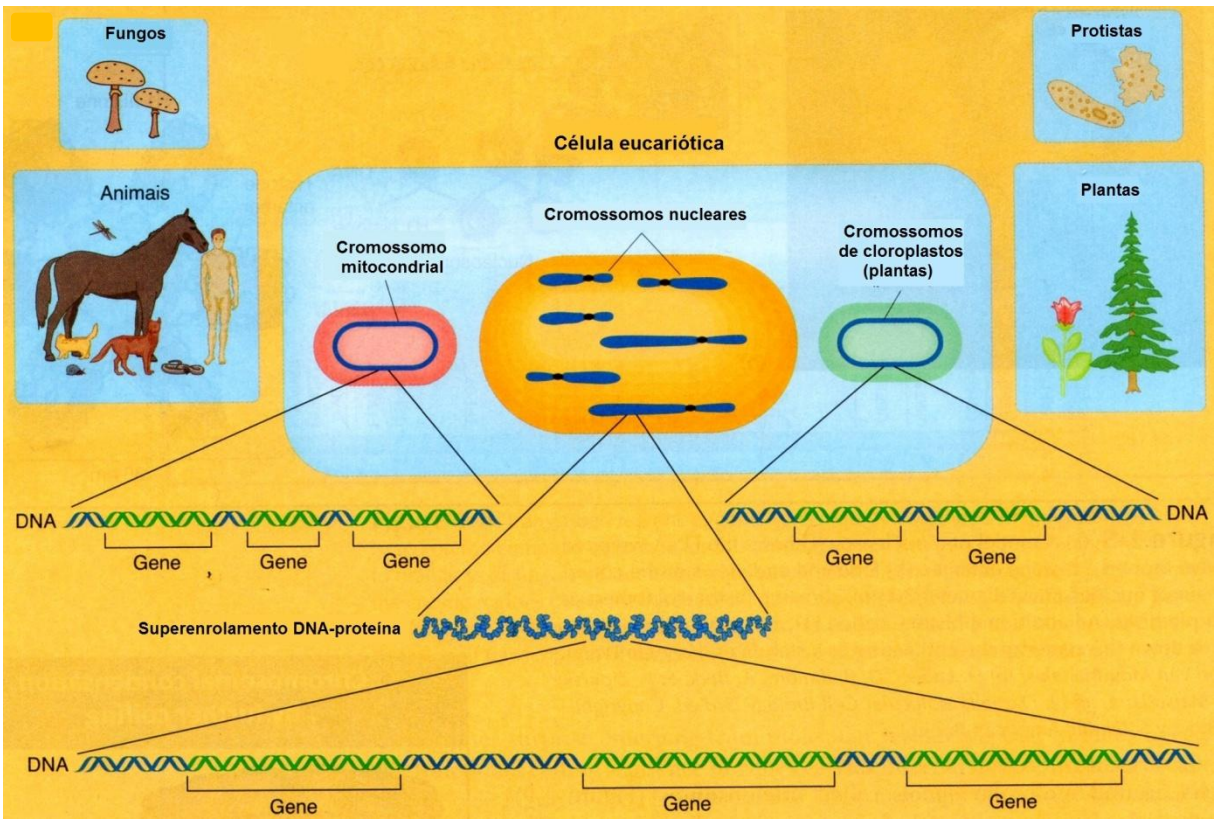


Figura 2.3 - Os cromossomos nucleares eucarióticos são lineares. Duas organelas eucarióticas — a mitocôndria e os cloroplastos (estes presentes só em plantas) — contêm cromossomos circulares.
Fonte: Modificado de Griffiths et al., 2012, p.8.

Cada molécula de DNA cromossômico contém, portanto, regiões funcionais denominadas genes, os quais podem ser considerados como trechos limitados distribuídos ao longo de sua extensão. Conforme dito anteriormente, tais trechos correspondem à informação para a construção de RNAs que posteriormente podem dar origem a proteínas. A molécula de DNA (Figura 2.4) consiste de duas longas cadeias polinucleotídicas dispostas em uma estrutura de dupla hélice, cadeias essas constituídas de quatro tipos de subunidades de nucleotídeos. Estes, por sua vez, são compostos de um açúcar, ao qual estão ligados um ou mais grupos fosfatos e uma base contendo nitrogênio. A base pode ser *adenina* (A), *guanina* (G), *citosina* (C) ou *timina* (T).

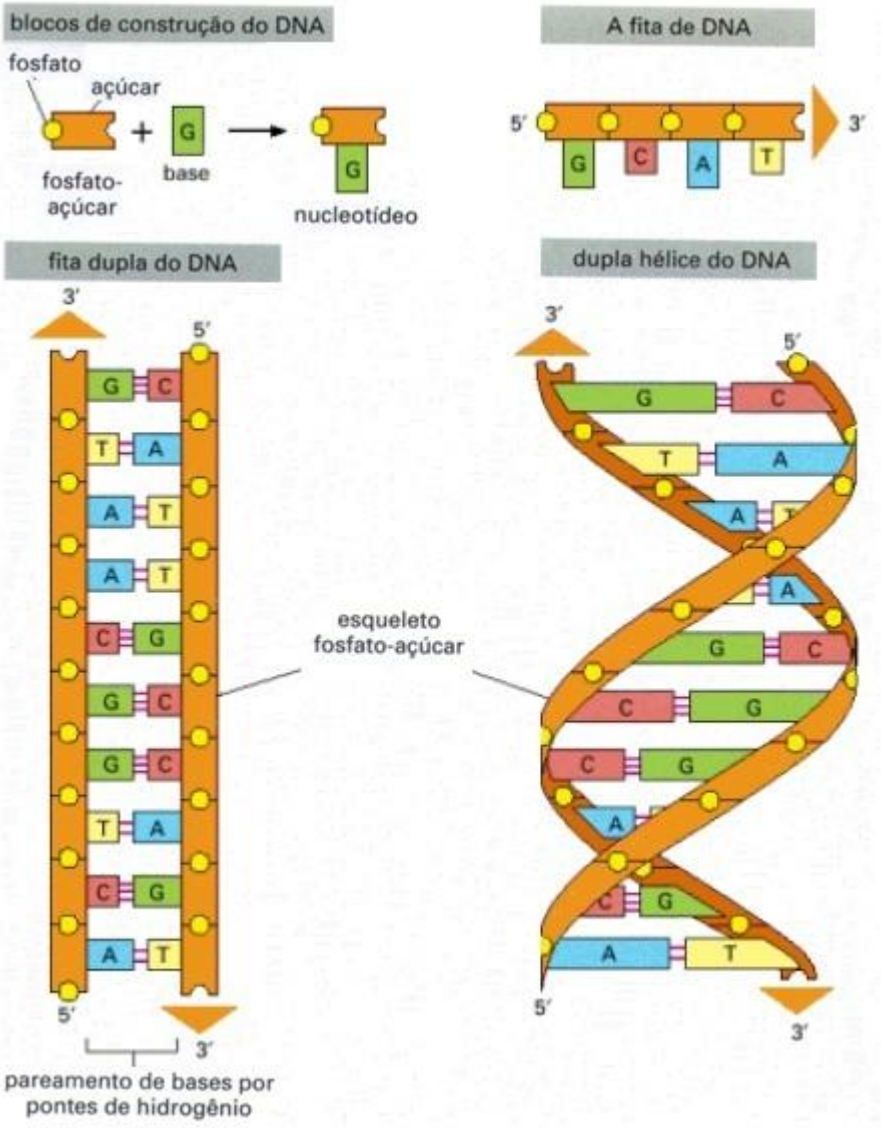


Figura 2.4 - Exemplo ilustrativo da estrutura de uma molécula de DNA e de seus "blocos" construtores. Quatro tipos de nucleotídeos covalentemente ligados formam as cadeias polinucleotídicas, com um esqueleto fosfato-açúcar a partir do qual as bases (A - adenina, C - citosina, T - timina e G - guanina) se estendem. As duas cadeias (ou *fitas*) polinucleotídicas são mantidas unidas por pontes de hidrogênio entre as bases pareadas. As "setas", no desenho, indicam as polaridades das fitas, as quais correm antiparalelas, uma em relação à outra. Na parte inferior esquerda do diagrama, para fins ilustrativos, o DNA está mostrado de forma plana. Na realidade, porém, ele se apresenta como uma estrutura em dupla hélice, tal como é mostrado à direita.
Fonte: Modificado de Alberts et al., 1999, p.188.

2.2. Visão geral da tecnologia convencional de sequenciamento de DNA

Conforme menção feita no Capítulo 1, o sequenciamento de DNA é o processo cuja finalidade é determinar a sequência de nucleotídeos de uma dada molécula de DNA. Esse pode ser considerado como o primeiro passo em busca de seu significado biológico que, como se procurou demonstrar na seção anterior, é importante para o maior entendimento sobre, por

exemplo, a organização dos genes e as funções dos produtos que eles codificam, como proteínas e RNAs.

O sequenciamento de DNA convencional, como já dito, se baseia no princípio da terminação de cadeia (didesóxi) e, mesmo com o surgimento das tecnologias NGS, é, até hoje, utilizado, principalmente para regiões mais restritas do genoma (Griffiths et al., 2012). A seguir, é apresentada uma visão geral sobre ele e sua evolução ao longo do tempo.

2.2.1. O sequenciamento de DNA baseado no método de Sanger e sua evolução ao longo do tempo

Basicamente, o princípio se baseia na extensão enzimática da fita de DNA e sua inibição pela inserção de um nucleotídeo análogo ao desoxinucleotídeo (dNTP, o tipo que seria naturalmente incorporado à fita), só que deficiente do grupo 3'-hidroxila (3'OH), o didesoxinucleotídeo (ddNTP) (Guimarães, 2010) (Figura 2.5).

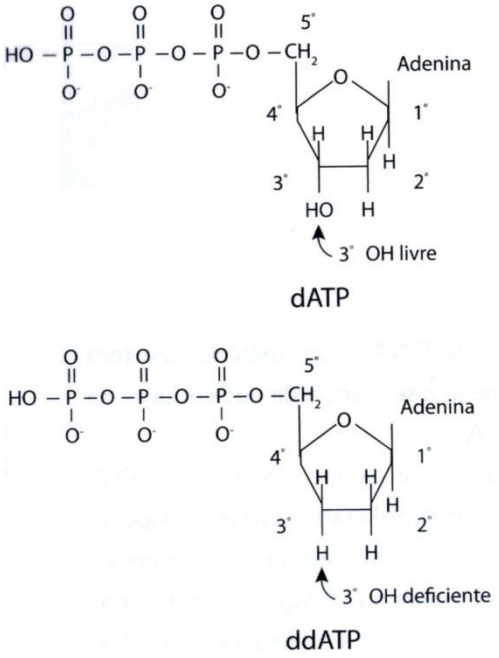


Figura 2.5 - Diferença entre as estruturas dos desoxinucleotídeos e dos didesoxinucleotídeos. A ausência do grupamento 3'OH impede a formação de uma ligação fosfodiéster com o eventual próximo dNTP da reação de sequenciamento.
Fonte: Guimarães, 2010, p.134.

Para que a síntese possa ocorrer, a enzima DNA polimerase tem de catalizar a reação entre o grupo 3'-hidroxila do nucleotídeo anterior e o grupo 5'-fosfato do próximo nucleotídeo a ser adicionado na cadeia. A ausência do grupo 3'-hidroxila, no didesoxinucleotídeo, impede que tal reação aconteça e, desta forma, a síntese é interrompida no ponto em que o

didesoxinucleotídeo é incorporado. Esse nucleotídeo modificado é, portanto, o elemento-chave da técnica de Sanger, por causa de sua capacidade de interromper a reação usando dNTPs marcados com fluoróforos específicos para todas as bases (Griffiths et al., 2012).

Inicialmente, para a detecção dos fragmentos gerados, o método usava marcação radioativa com os isótopos 32^{P} ou 32^{S}, marcação essa que podia ser implementada em diferentes pontos: desoxinucleotídeos (usualmente citosinas), o *primer* ou os didesoxinucleotídeos terminadores. Esse tipo de marcação, no entanto, em virtude dos avanços tecnológicos associados às tentativas de automação do processo, foi substituída pela marcação com compostos fluorescentes (chamados *fluorocromos* ou *fluoróforos*) (Figura 2.6) no *primer* (Hutchison III, 2007 apud Smith et al., 1986) ou nos didesoxinucleotídeos terminadores (Ewing et al., 1998 apud Prober et al., 1987). Tipicamente, era usado um corante diferente para cada uma das quatro reações, de maneira que seus resultados pudessem ser combinados e separados em uma única raia de gel (além disso, para o caso específico da marcação feita diretamente nos didesoxinucleotídeos terminadores, as quatro reações também podiam ser conduzidas em conjunto em um único tubo) (Ewing et al., 1998; Hutchison III, 2007, Guimarães, 2010; Weiss, 2010; Rodríguez-Ezpeleta et al., 2012).

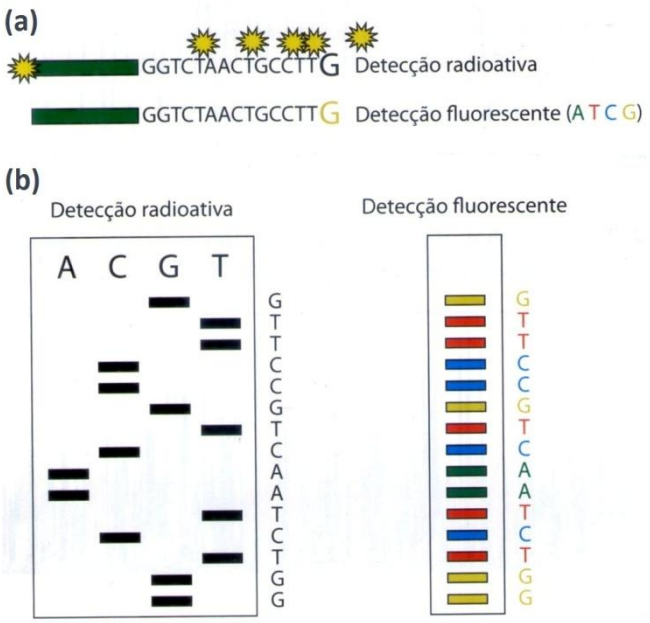


Figura 2.6 - Detecção radioativa *versus* detecção fluorescente. (a) Diferenças na forma de marcação entre os dois métodos. Para a detecção radioativa, os pontos possíveis são: o *primer*, os dNTPs ou os ddNTPs terminadores. Na detecção fluorescente, a marcação pode ser feita no *primer* ou no ddNTP terminador. (b) Diferença entre a eletroforese em gel com material marcado radioativamente, a qual necessita de quatro raias para cada reação de sequenciamento (uma para cada ddNTP terminador) e a detecção fluorescente, na qual apenas uma raia é necessária, uma vez que cada ddNTP possui uma cor diferente.
Fonte: Modificado de Guimarães, 2010, p.277.

Nos sequenciadores que empregam esses tipos de marcação, diferentes amostras são analisadas simultaneamente em raias separadas de um mesmo gel. Um laser varre continuamente o gel e excita os fluoróforos dos fragmentos à medida que estes últimos avançam, produzindo uma imagem na qual cada raia apresenta um padrão de bandas com quatro cores distintas, sendo, cada banda, correspondente a um fragmento de tamanho específico. Sensores detectam as diferentes intensidades de sinal para os respectivos quatro tipos de comprimentos de onda (de cada base associada) e, por meio de análise computacional, a imagem do gel é interpretada de maneira que possa ser inferida a sequência de bases para cada amostra original (o que também é chamado de *leitura*) (Ewing et al., 1998).

No fim da década de 1990, o sequenciamento se tornou verdadeiramente automático, a partir da introdução de equipamentos dotados de eletroforese capilar (Swerdlow et al., 1990; Swerdlow; Gesteland, 1990; Shendure; Ji, 2008 apud Hunkapiller et al., 1991). Com isso, a trabalhosa preparação manual de géis deu lugar à recarga automática dos capilares (onde a

eletroforese passou a ser realizada). A substituição dos géis pelos capilares também simplificou o processo de separação dos fragmentos e permitiu o aumento do tamanho das leituras (Madabhushi, 1998). Não era mais necessária a parada da corrida, no gel, para a verificação dos resultados e, consequentemente, a interrupção do processo de separação. A máquina, agora, poderia simplesmente usar o conjunto laser-sensores para continuamente detectar a passagem dos fragmentos por um ponto fixo (Dale et al., 2012). Instrumentos desenvolvidos com o recurso passaram a rodar um número maior de amostras, aumentando, assim, o rendimento e diminuindo os custos associados. Em suma, a mudança no meio físico da eletroforese trouxe mais confiabilidade, rapidez e precisão ao processo. Em 2001, a tecnologia usada para sequenciar o genoma humano era baseada em eletroforese capilar de amostras individuais detectadas por fluorescência. Cada instrumento podia detectar a passagem de 500 a 600 bases, de cada uma das 96 reações de sequenciamento simultâneas, em cerca de 10 horas e culminando com uma produção de 115 kpb (mil pares de bases) por dia. Atualmente, dispõe-se de equipamentos capazes de realizar de 96 a 384 sequenciamentos paralelos e produzindo leituras da ordem de 700 bases a 1000 bases (Petterson et al., 2009; Guimarães, 2010; Mardis, 2011; Dale et al., 2012). O AB3730xl, equipamento fabricado pela Applied Biosystems, por exemplo, pode gerar 2,88 Mb (milhões de pares de bases) por dia, com tamanho de leitura típico de 900 bases (Liu et al., 2012).

Conforme mencionado, o conjunto de sinais captados pelo sequenciador durante a eletroforese das amostras é interpretado por algoritmos de computador específicos e transformado no que é chamado de *leitura* ou *eletroferograma* do sequenciamento de DNA, tornando a identificação de bases fácil e rápida (Guimarães, 2010). Um exemplo de eletroferograma é mostrado na Figura 2.7.

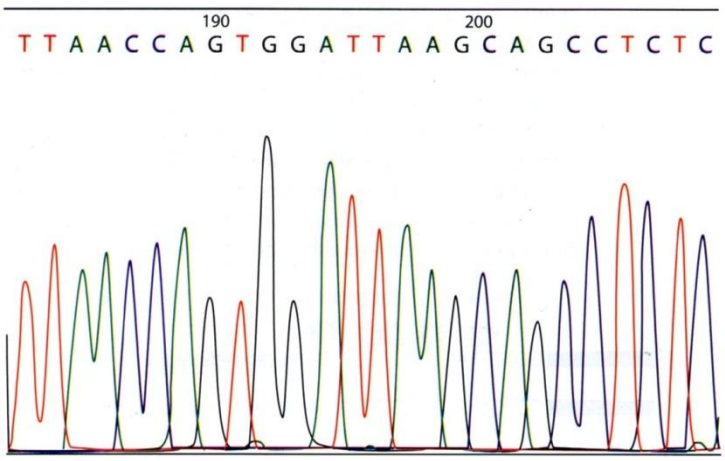


Figura 2.7 - Exemplo de resultado do sequenciamento liberado para o usuário final, após o tratamento computacional dos dados, sob a forma de *eletroferograma* ou *leitura*.
Fonte: Modificado de Guimarães, 2010, p.278.

A Figura 2.8 apresenta a visão geral do processo no sequenciador automático.

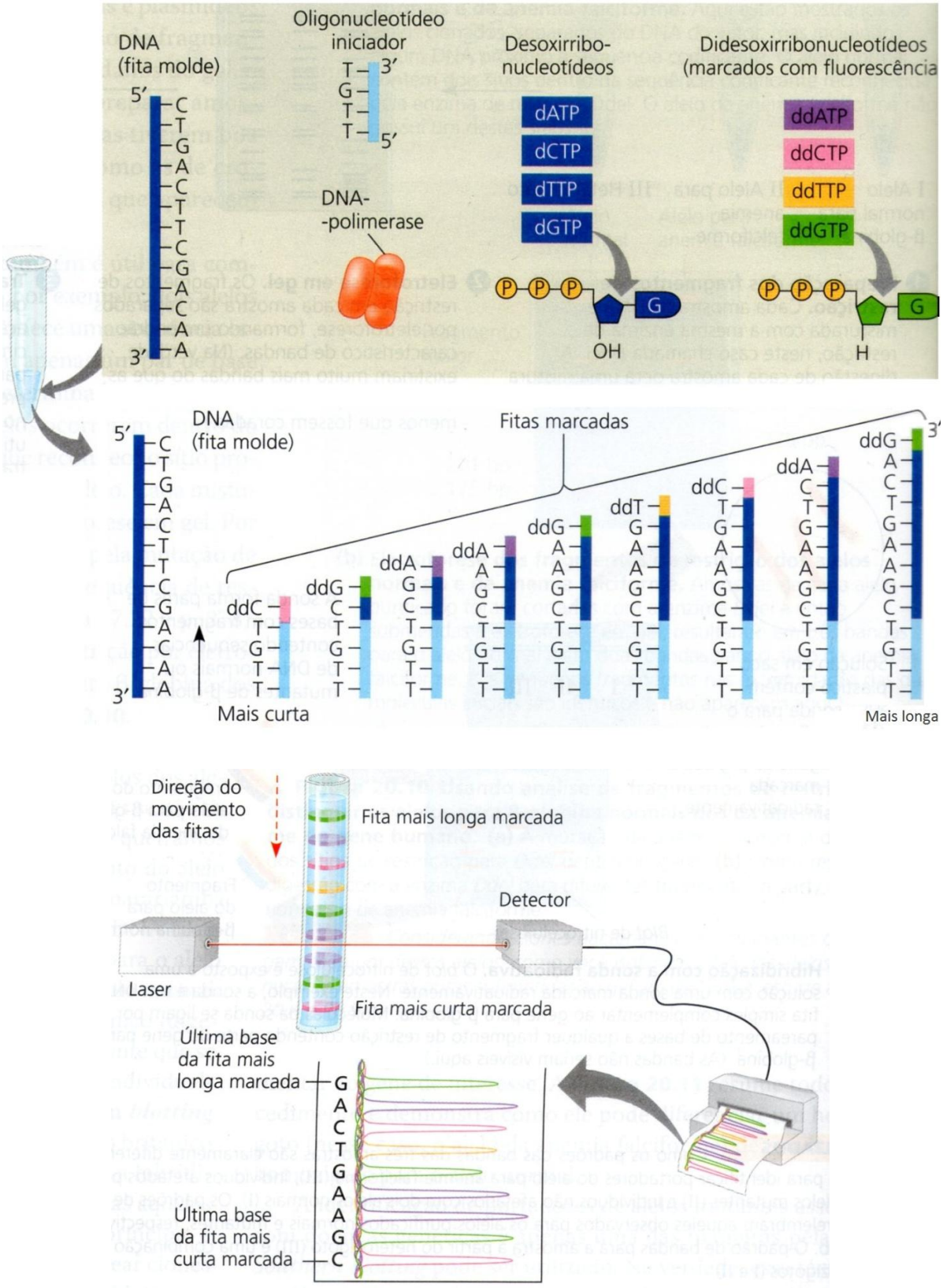


Figura 2.8 - O processo do sequenciamento pelo método de Sanger no sequenciador automático, desde o "coquetel" até o eletroferograma.
Fonte: Modificado de Campbell et al., 2010, p.408.

2.3. Visão geral das principais tecnologias NGS para sequenciamento de DNA

Como visto, o sequenciamento de DNA baseado no método de Sanger obteve grande sucesso e esteve envolvido virtualmente em todos os projetos desse tipo durante aproximadamente três décadas, desde a sua aparição em 1977. No entanto, apesar de sua robustez, o método apresenta desvantagens, dentre as quais podem ser citadas a necessidade de uso de bibliotecas de clones para a preparação das amostras, processo este caro e demorado, e a baixa velocidade de produção alcançada com esse tipo de tecnologia, a qual é, em muitas ordens de magnitude, inferior àquela desejável em projetos de genômica. Assim sendo, com o intuito de superar as deficiências inerentes ao método de Sanger, novas técnicas alternativas surgiram (Ronaghi et al., 1996; Ronaghi et al., 1998; Preparata; Upfal, 2000; Böcker, 2004; Braslavsky et al., 2003; Mitra et al., 2003; Chaisson et al., 2004 apud Drmanac et al., 1989; Ansorge, 2009 apud Ansorge, 1991). Tais técnicas visavam um processo de sequenciamento livre de clonagem baseada em vetores e a redução significativa da duração e custo do processo como um todo (Chaisson et al., 2004). De fato, mesmo tendo atingindo a escala "industrial", a qual fora viabilizada pelas melhorias realizadas em prol da automação e a partir da condução do processo em grandes "fábricas" especializadas para esse fim, o sequenciamento de DNA convencional ainda envolvia cifras da ordem de 10 milhões de dólares e cerca de 10 anos para ser concluído, considerando-se a sua aplicação na obtenção de um rascunho do genoma humano, por exemplo (Janitz, 2008). Dentre as alternativas criadas em busca de menor custo e maior rapidez para o processo, as tecnologias NGS foram as primeiras a se consolidarem como soluções realmente eficientes, em resposta à limitada produtividade dos métodos de "1ª geração", pelo poder de gerarem informação muitas vezes maior e com grande economia de tempo e custo por base sequenciada (Carvalho; Silva, 2010). Mais precisamente, três representantes dessas novas tecnologias se sobressaíram logo de início. Em outubro de 2005, a empresa 454 Life Sciences, foi a primeira a apresentar ao mercado uma dessas soluções: a tecnologia de sequenciamento 454[9] (Droege; Hill, 2008), baseada no método do pirossequenciamento (Nyrén; Lundin, 1985; Hyman, 1988; Ronaghi et al., 1996; Ronaghi et al., 1998; Margulies et al., 2005; Rothberg; Leamon, 2008). No rastro da 454, entre 2006 e 2007, outros competidores, como Solexa[10] e Agencourt[11], também se lançaram no mercado. Todas as tecnologias das três empresas fundadoras foram, em seguida,

[9] Com o sistema GS 20 (Pareek et al.; 2011) (http://www.454.com/).
[10] Com o sistema Genome Analyzer (Bennett et al., 2005; Bentley et al., 2008).
[11] Com a tecnologia que seria a base do sistema SOLiD™-*Sequencing by Oligonucleotide Ligation and Detection*.

compradas por outras corporações: ainda em 2006, a Agencourt passou ao controle da Applied Biosystems/Life Technologies[12], e, em 2007, a 454 foi comprada pela Roche Applied Science[13], enquanto a Illumina[14] adquiriu a Solexa (Liu et al., 2012). Essas empresas, prevendo grandes oportunidades de negócios no mercado de sequenciamento, buscavam consolidar as suas respectivas posições, reforçando ainda mais a visão do "genoma humano por mil dólares" (Bennett et al., 2005; Janitz, 2008; Mardis, 2008a; Wold; Myers, 2008; Davies, 2011). Em pouco tempo, tais tecnologias se tornaram as plataformas NGS mais usadas no mundo (Karow, 2009)[15].

As tecnologias NGS, também conhecidas como de alta vazão ou de alto fluxo (*high-throughput*), são capazes de gerar dados na ordem de milhões a bilhões de pares de bases por experimento (ou *corrida*) (Petterson et al., 2009; Carvalho; Silva, 2010; Mardis, 2011; Pareek et al., 2011), contra as relativamente poucas milhares de bases proporcionadas pelo método de Sanger automatizado. Diferem deste último, também, por não fazerem uso da clonagem convencional baseada em vetores, porém do sequenciamento direto e altamente paralelizado de diversos fragmentos de DNA independentes, bem como por gerarem produtos do sequenciamento (leituras) de pequena extensão — tipicamente da ordem de 30 a 400 pares de bases — em contraposição aos comprimentos de 700 a 1000 típicos da tecnologia baseada em Sanger. O tamanho da leitura, nas plataformas NGS, é sacrificado em prol de uma vazão maior, permitindo que elas gerem uma grande cobertura de sequenciamento (de 30 vezes ou mais) a um baixo custo relativo, se comparada com a cobertura típica obtida com as plataformas de eletroforese capilar (da ordem de 10 vezes). Assim, com relação a esta última, conseguem promover um sequenciamento muito mais rápido, produtivo e bem menos oneroso (Chaisson et al., 2004). A Tabela 2.1, apenas de caráter informativo, traz características básicas de uma plataforma de sequenciamento Sanger e as de plataformas NGS típicas.

[12] http://www.appliedbiosystems.com/.
[13] http://www.roche-applied-science.com/.
[14] http://www.illumina.com/.
[15] Outros equipamentos NGS que surgiram na mesma época foram o *Polonator* (http://www.polonator.org) e o *HeliScope* (Helicos BioSciences; http://www.helicosbio.com). Entretanto, sua participação no mercado foi bem menor do que a das outras três tecnologias pioneiras.

Tabela 2.1 - Comparação de características básicas de plataformas NGS e Sanger típicas.

Sequenciador	454 GS FLX	HiSeq 2000	SOLiD™ v4	Sanger 3730xl
Processo de sequenciamento	Pirossequenciamento	Sequenciamento por síntese	Ligação e codificação em duas bases	Terminação de cadeia por didesoxinucleotídeos
Tamanho da leitura	700 pb	50SE, 50PE, 101PE	50 + 35 pb ou 50 + 50 pb	400 ~ 900 pb
Acurácia (*)	99,9%	98% (100PE)	99,94% (dados brutos)	99,999%
Leituras	1M	3G	1200 ~ 1400M	-
Pares de bases por corrida	0,7 Gb	600 Gb	120 Gb	1,9 ~ 84 kb
Tempo por corrida	24 horas	3 ~ 10 dias	7 dias para SE 14 dias para PE	20 minutos ~ 3 horas
Vantagens	Tamanho da leitura, rapidez	Alta produção	Acurácia	Alta qualidade, grande tamanho de leitura
Desvantagens	Taxa de erros com homopolímeros (mais de 6 bases sucessivas), alto custo, baixa produção	Pequeno tamanho de leitura	Pequeno tamanho de leitura	Alto custo, baixa produção
Preço do equipamento	Instrumento: US$500.000,00; Corrida: US$7.000	Instrumento: US$690.000,00; Corrida: US$6.000/genoma humano com cobertura de 30x	Instrumento: US$495.000,00; Corrida: US$15.000 / 100 Gb	Instrumento: US$95.000,00; Corrida: ~ US$4 por reação com 800 pb
CPU	2 x Intel Xeon X5675	2 x Intel Xeon X5560	8 processadores de 2,0 GHz	Pentium IV de 3,0 GHz
Memória	48 GB	48 GB	16 GB	1 GB
Disco rígido	1,1 TB	3 TB	10 TB	280 GB
Automação na preparação das bibliotecas	Sim	Sim	Sim	Não
Outros dispositivos requeridos	REM e system	cBot system	EZ bead system	Não
Custo por milhão de bases	US$10	US$0,07	US$0,13	US$2.400,00

(*) Conforme avaliação de Liu et al (2012), após filtragem dos dados. Como informação adicional, Henson et al. (2012) apontam as seguintes taxas de erro de base para as plataformas NGS citadas na tabela: 0,5% para 454 GS FLX Titanium XL+ (tipo de erro principal: Indels provocados pela limitação típica da plataforma ao lidar com regiões homopoliméricas de bases idênticas); ~ 1-2% em 100 pb para HiSeq 2000 (tipo de erro principal: substituições de bases); 0,06% para SOLiD™ v4 (tipo de erro principal: viés A-T).

Fonte: Modificado de Liu et al., 2012, pp.6-7 e Henson et al., 2012, p.903.

Embora cada equipamento NGS apresente vários detalhes técnicos que os tornam distintos entre si, principalmente no que diz respeito àqueles relacionados às reações de sequenciamento propriamente ditas, esses instrumentos altamente paralelizados compartilham de certos atributos comuns que caracterizam a sua maior eficiência. Primeiramente, a etapa inicial de preparação das amostras, que envolve menos e muito mais simples passos do que o sequenciamento do tipo Sanger, parte da produção de bibliotecas de fragmentos formadas pela ligação de pequenas moléculas de DNA sintéticas — adaptadores de oligonucleotídeos (Mardis, 2008a), os quais são específicos para cada plataforma — às extremidades de cada um dos fragmentos de DNA da coleção a ser sequenciada. Em segundo lugar, todas as plataformas realizam a amplificação dessas bibliotecas em suportes sólidos (por exemplo, uma placa de vidro ou uma nanoesfera), através de uma reação — mediada por uma enzima do tipo polimerase — que produz diversas cópias de cada uma delas. Tal amplificação se faz necessária para que as reações de sequenciamento subsequentes consigam produzir um sinal suficientemente forte para a devida detecção pelo sistema ótico/de aquisição de imagem do instrumento[16]. Aliás, aqui surge outra característica comum: combinando diversas inovações na química do sequenciamento, este é obtido, de forma altamente paralelizada, por meio da captura de imagem microscópica e em tempo real das emissões de luz que ocorrem a partir da síntese da fita complementar no DNA sequenciado (Thompson; Milos, 2011; Rodríguez-Ezpeleta et al., 2012). Outro ponto comum é que as reações de sequenciamento acontecem como uma orquestrada série de ciclos repetidos, sendo executadas e automaticamente detectadas. Independentemente das especificidades de cada plataforma, a ideia principal é que, no sequenciamento altamente paralelizado, as reações são caracterizadas por esse ritmo "passo a passo", ou seja, de nucleotídeo em nucleotídeo, e não através da separação e detecção distintas (de 96 ou 384 a cada vez, por exemplo) dos produtos já sequenciados, tal como acontece com a química Sanger em um sistema com tubos capilares. Do exposto neste parágrafo, portanto, grande parte da maior eficiência advém do uso da clonagem *in vitro* e de sistemas de suporte sólido para as unidades de sequenciamento, evitando-se, assim, o intensivo trabalho laboratorial de produção de clones bacterianos, da montagem das placas de sequenciamento e da separação dos fragmentos em géis (Carvalho; Silva, 2010; Mardis, 2011). A Figura 2.9 ilustra o método "passo a passo" das tecnologias de segunda geração, comparado ao método tradicional de sequenciamento.

[16] Cabe ressaltar que, devido a essa etapa, um tipo de erro de sequenciamento é eventualmente originado aqui e acaba se perpetuando por toda a cadeia posterior de análise dos dados, uma vez que as enzimas polimerases nunca são 100% precisas (Mardis, 2011).

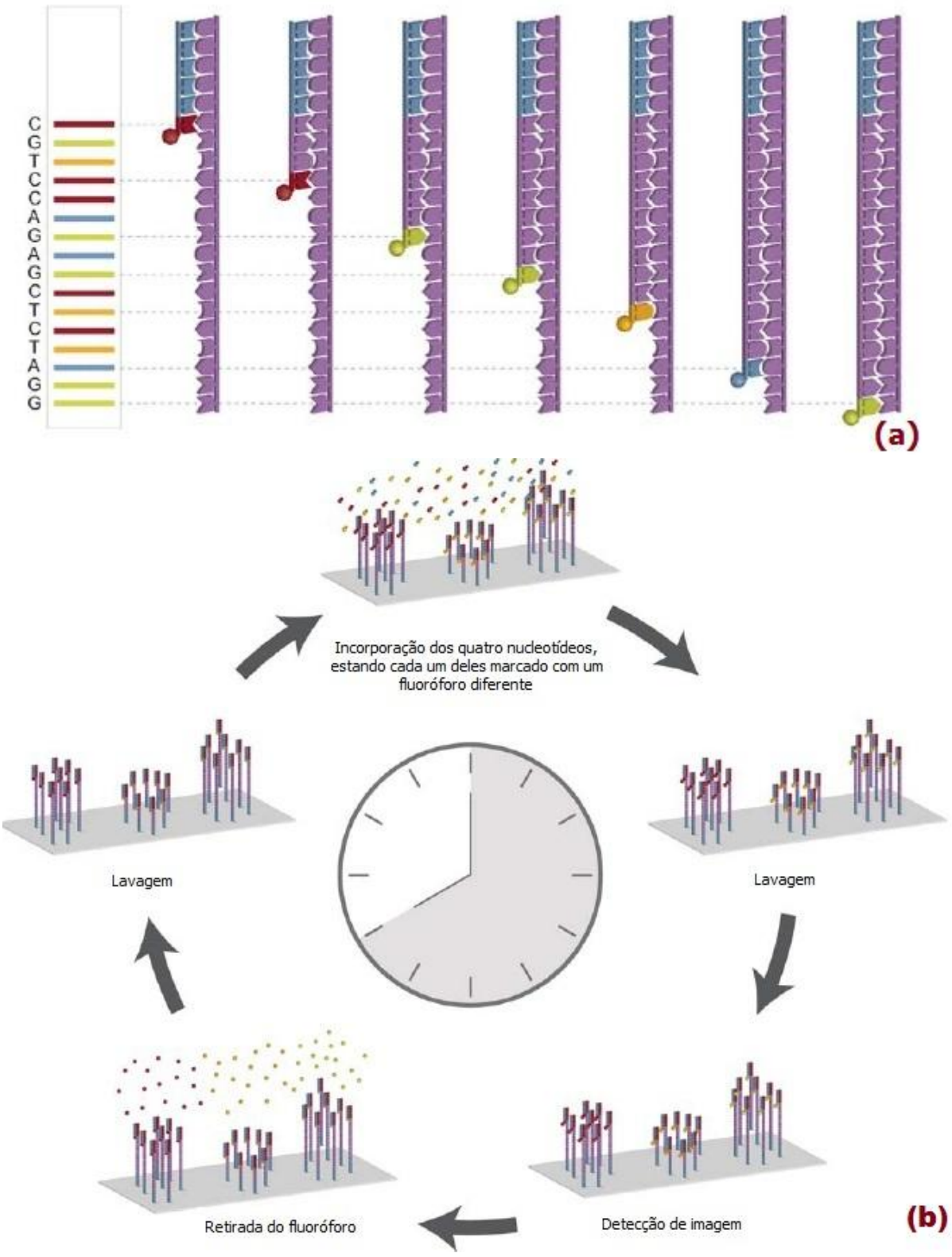


Figura 2.9 - Exemplos de tecnologias de sequenciamento Sanger e NGS. (a) Uma implementação moderna do sequenciamento Sanger: uso da química de terminação de cadeia por nucleotídeos especiais marcados com fluoróforos, seguida da separação por tamanho do fragmento para a resolução da sequência. (b) O processo de sequenciamento Illumina ilustra o paradigma "passo a passo" de lavagem-captura de imagem comumente utilizado nas tecnologias de sequenciamento de DNA de segunda geração.
Fonte: Modificado de Schadt et al., 2010, p.R229.

É necessário frisar que, em linhas gerais, todas as três soluções pioneiras — 454, Solexa/Illumina e ABI SOLiD™ (Valouev et al., 2008) — podem ser caracterizadas pelas informações descritas nos três últimos parágrafos. No entanto, isso não significa dizer que qualquer outro instrumento NGS irá seguir à risca a todo esse "pacote" de características. De fato, além das três soluções, as quais continuam sendo constantemente aprimoradas, outras também despontaram como bastante promissoras. Por causa de outras inovações tecnológicas atreladas a algumas das novas soluções, como, por exemplo, a possibilidade de realizar o sequenciamento a partir de uma molécula de DNA única (para o caso específico das plataformas do tipo *single molecule*) e, com o surgimento dessas novas máquinas logo em seguida ao lançamento do próprio conceito de NGS, suas respectivas estratégias de *marketing* passaram a designá-las como sendo pertencentes à classe das tecnologias de "terceira geração" (Pacific Biosciences; 2010). Cabe aqui, no entanto, uma ressalva sobre a questão terminológica associada à essa "mais nova geração" de tecnologias NGS: conforme Thompson e Milos (2011), o termo lógico para uma nova rodada de avanços na tecnologia de sequenciamento seria o "sequenciamento de terceira geração" e, de fato, isso tem sido frequentemente usado para designar o sequenciamento de molécula única. Entretanto, eles reforçam que o termo também tem sido aplicado para designar outras inovações tecnológicas, como a possibilidade de sequenciamento em tempo real ou de sequenciamento em estado sólido, presentes em algumas das novas plataformas. Com isso, deve-se ter em mente que o termo "terceira geração" é bastante abrangente e não apenas específico para tecnologias de molécula única. Thudi et al. (2012) ressaltam que o termo SGS (do inglês *Second-generation sequencing*, ou "sequenciamento de segunda geração") tem sido usado para designar as tecnologias NGS pioneiras, Roche/454, Illumina, ABI SOLiD™ e outras como, por exemplo, o Polonator G.007. Já as tecnologias mais recentes, como o Ion Personal Genome Machine (PGM™)[17], ou de molécula única, como HeliScope™ Single Molecule Sequencer ou o Single-Molecule Real-Time (SMRT™) Sequencer PacBio*RS*[18], têm sido designadas sob o termo TGS (do inglês *Third-generation sequencing*, ou "sequenciamento de terceira geração") ou, também, por NNGS (do inglês *Next-next-generation sequencing*, algo como "sequenciamento de próxima-próxima geração", em português). Essa últimas tecnologias dispõem de recursos técnicos diferentes daqueles empregados pelos primeiros dispositivos NGS.

[17] http://www.iontorrent.com/technology.
[18] http://www.pacificbiosciences.com/.

Independentemente da geração à qual fazem parte, o fato é que todas elas têm evoluído muito rapidamente. Shendure e Ji (2008), por exemplo, já enfatizavam que o campo das tecnologias de sequenciamento de DNA havia se tornado um "alvo móvel" veloz. Impressão também compartilhada por Zhang J et al. (2011), ao afirmarem que as tecnologias NGS, diariamente, apresentam novas mudanças e melhorias. Glenn (2011), inclusive, foi feliz ao mencionar que as tecnologias e plataformas de sequenciamento têm sido atualizadas (e renovadas) tão frequentemente, que quaisquer trabalhos de revisão sobre elas se assemelham ao "trabalho de Sísifo". No mercado de NGS, entretanto, parece ainda prevalecer a dominância das plataformas baseadas nas tecnologias pioneiras (Zhang J et al., 2011), conforme também relatado no trabalho de Klassen e Currie (2012), o qual aponta uma pesquisa conduzida pela empresa multinacional de serviços financeiros e consultoria J.P.Morgan (Peterson et al., 2010). Tal estudo, em que foram ouvidos 30 diretores de centros de sequenciamento espalhados pelos Estados Unidos e Europa, possuidores de base instalada (ou prevista para um futuro próximo) de um ou mais dispositivos de segunda ou terceira geração, buscou entender e projetar as principais tendências de uso desses equipamentos para um período de 3 anos, a partir de meados de 2010. Foi prevista, então, conforme o levantamento, a aquisição ou *upgrade* (atualização de versão) de cerca de 136 sistemas NGS, até 2013, e um aumento da participação dos estudos envolvendo NGS, nas atividades de pesquisa, para mais de 50%, pelo menos até 2012. Por outro lado, foi observada queda do emprego dos equipamentos do tipo Sanger nas atividades de sequenciamento e previsto o seu contínuo declínio de utilização — de aproximadamente 28%, em 2009, até cerca de 14,6%, em 2012. As Figuras 2.10 e 2.11 exibem, respectivamente, a penetração das plataformas de sequenciamento mais usuais (já adquirida ou para a qual há expectativa de aquisição ou realização de *upgrade* até 2013) nos centros participantes da pesquisa e a previsão de tendência mencionada para as plataformas de primeira geração (na Figura 2.10, rotuladas como CE, do inglês *Capillary Electrophoresis*).

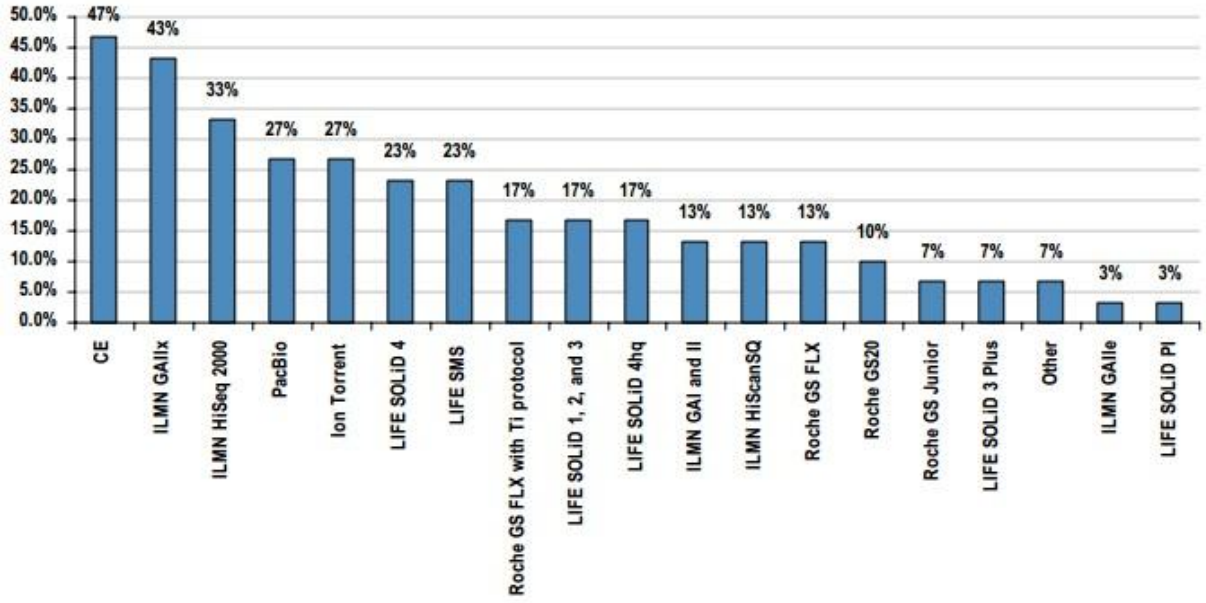


Figura 2.10 - Sequenciadores atualmente adquiridos ou para os quais existe previsão de aquisição ou realização de *upgrade*, até 2013, pelos centros participantes da pesquisa da J.P.Morgan. As plataformas de NGS da Illumina lideram em tais termos. Entre os entrevistados, 43%, por exemplo, possuem ou esperam adquirir ou realizar *upgrade* em uma plataforma Illumina GAIIx, porcentagem essa um pouco abaixo dos 47% que já possuem ou esperam adquirir um equipamento de eletroforese capilar. Para as versões ainda mais novas dos equipamentos de sequenciamento, 33% dos centros entrevistados possuem ou esperam adquirir um Illumina HiSeq 2000, 23% possuem ou esperam adquirir um SOLiD™ 4 e 17% esperam adquirir o *upgrade* para o SOLiD™ 4hq. Para equipamentos de terceira geração, ambos PacBio e Ion Torrent são opções de aquisição para 27% dos centros entrevistados. Segundo o estudo, o equipamento Oxford Nanopore também foi mencionado como candidato a uma eventual aquisição, considerando a classe de plataformas de terceira geração.
Fonte: Modificado de Peterson et al., 2010, p.13.

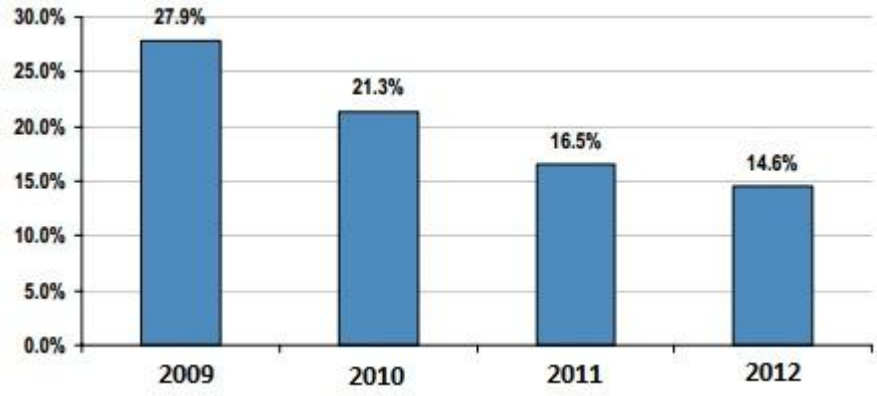


Figura 2.11 - Previsão de tendência (2009 a 2012) de participação das plataformas do tipo Sanger nas atividades de sequenciamento.
Fonte: Modificado de Peterson et al., 2010, p.24.

Também foi ventilado, pelos entrevistados do estudo, uma eventual participação das plataformas de terceira geração em 50% das atividades de sequenciamento, até 2013. O gráfico da Figura 2.12 ilustra isso.

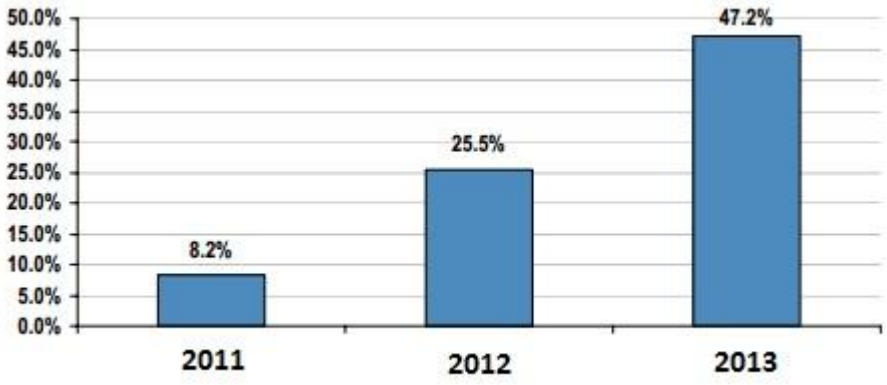


Figura 2.12 - Previsão de tendência (2011 a 2013) de participação das plataformas de terceira geração nas atividades de sequenciamento.
Fonte: Modificado de Peterson et al., 2010, p.30.

Um outro panorama que podemos obter a respeito do uso das principais plataformas de sequenciamento NGS pelo mundo foi idealizado por Hadfield (2009, 2011) e, posteriormente, construído em parceria com Loman (Hadfield; Loman, 2009; Macmillan Publishers Limited, 2010; SEQanswers, 2010), no qual um mapa — mantido atualizado pela própria comunidade científica, sob o regime de "melhor esforço", e disponível em http://omicsmaps.com — ilustra a distribuição aproximada dessas plataformas pelo globo (Figura 2.13).

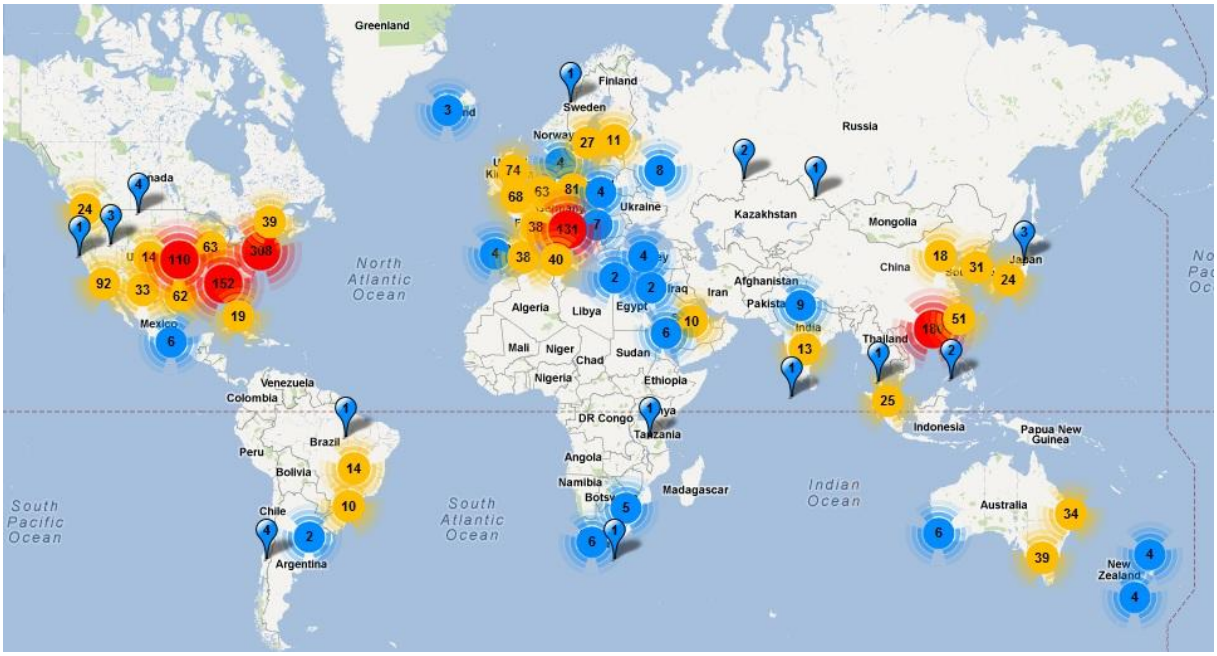


Figura 2.13 - Distribuição aproximada das principais plataformas de sequenciamento NGS pelo mundo.
Fonte: Modificado de http://omicsmaps.com.

Pode-se ter uma ideia, portanto, da relevância da abordagem NGS e de como, em menos de uma década, seu uso se difundiu pelo mundo.

A Tabela 2.2 "traduz" o mapa anterior em números mais precisos quanto ao uso de cada plataforma principal. Conforme informação mais atual, existem 1981 máquinas espalhadas por 706 centros ou laboratórios de sequenciamento pelo mundo, perfazendo uma média de 2,8 máquinas por centro.

Tabela 2.2 - Número de máquinas por plataforma de sequenciamento NGS.

Plataforma	Número de máquinas
Illumina Genome Analyzer IIx	555
Illumina HiSeq 2000	554
ABI SOLiD™	344
Roche 454	323
Ion Torrent	133
Illumina MiSeq	36
Pacific Biosciences	27
Polonator	9

Fonte: Modificado de http://omicsmaps.com/stats.

A seguir, são fornecidos detalhes de funcionamento a respeito das três principais tecnologias NGS abordadas no trabalho.

2.3.1. 454

Com o lançamento do sistema GS 20 (Genome Sequencer 20), a tecnologia 454 debutou no mercado de sequenciamento, em 2005, como a primeira de próxima geração a ser comercializada. Tal tecnologia realiza o *sequenciamento por síntese* (ou, mais precisamente, *sequenciamento por adição de nucleotídeo único* — SNA - *single nucleotide addition* (Metzker, 2010)), obtendo a leitura da sequência por meio do princípio do pirossequenciamento, no qual, basicamente, uma combinação de reações enzimáticas produz um sinal de luz que é capturado por uma câmera CCD (*charged-coupled device*) (Carvalho; Silva, 2010).

A preparação das bibliotecas de sequenciamento ocorre por meio de um método conhecido como PCR em emulsão (emPCR, do inglês *emulsion PCR*) (Dressman et al., 2003; Metzker, 2010), no qual o DNA genômico é isolado, fragmentado mecanicamente (e aleatoriamente) (Figura 2.14(a)), e as extremidades 3' e 5' dos fragmentos são ligadas, *in vitro*, a adaptadores contendo sítios de iniciação (*priming*) universais específicos da química pertinente à tecnologia 454 (Figura 2.14(b)). Os fragmentos são, então, separados em moléculas de fita simples (Figura 2.14(c)).

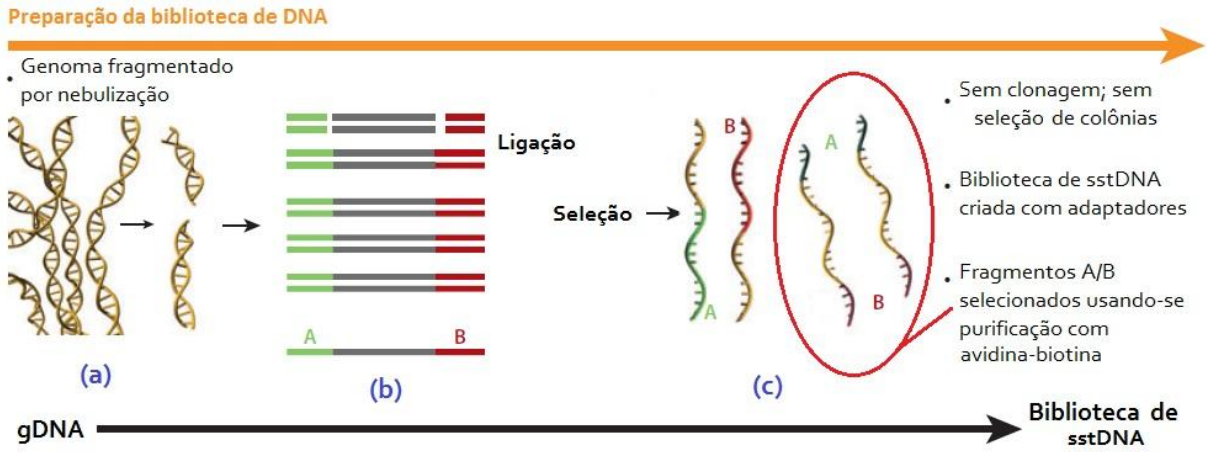


Figura 2.14 - Tecnologia 454. (a) Fragmentação do DNA genômico (gDNA). (b) Ligação dos adaptadores universais da tecnologia 454. (c) Seleção dos fragmentos de sstDNA (DNA de fita simples), com adaptadores nas duas extremidades, para uso nas etapas posteriores do processo.
Fonte: Modificado de Mardis, 2008b, p.390.

Uma solução de água em óleo, também contemplando esses fragmentos, nanoesferas de agarose (Mardis, 2008b) revestidas de adaptadores complementares aos dos fragmentos e reagentes da PCR, é agitada vigorosamente (Mardis, 2008a) para formar micelas que favorecem a ocorrência de uma relação de 1:1, ou seja, de uma molécula de DNA para cada esfera (Metzker, 2010) e os devidos reagentes (Figura 2.15).

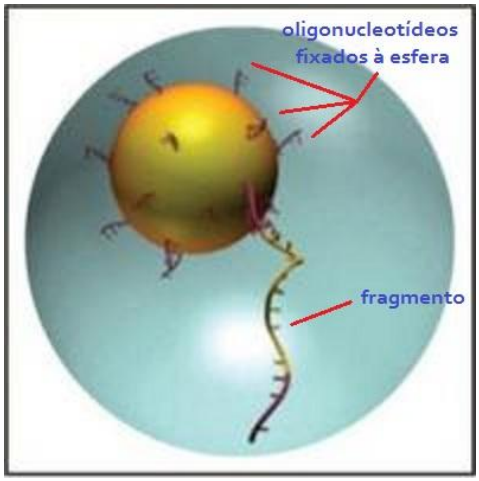


Figura 2.15 - Microrreator (gota de água) isolando uma nanoesfera e seu respectivo fragmento de DNA já anelado, antes da etapa de amplificação.
Fonte: Modificado de https://www.uppnex.uu.se/system/files/imagecache/for_lightbox/fig3_1.jpg.

Assim, cada micela funciona como um microrreator (Figura 2.16(a)), produzindo, quando da etapa de amplificação, várias milhares de cópias idênticas de um mesmo fragmento de DNA (Ansorge, 2009; Petterson et al. 2009; Carvalho; Silva, 2010; Griffiths et al., 2012).

As novas cópias de fragmentos geradas irão popular a nanoesfera, a esta se ligando por meio dos oligonucleotídeos adaptadores eventualmente livres em sua superfície (Figura 2.16(b)).

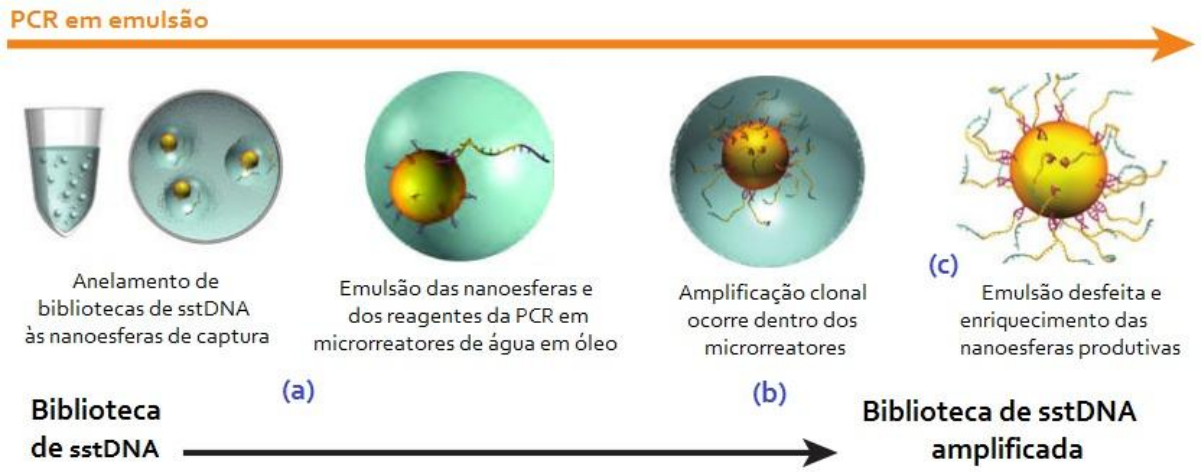


Figura 2.16 - Tecnologia 454 (continuação - parte I). (a) Micelas de água em óleo para formar os microrreatores contendo as respectivas esferas e os reagentes da PCR. (b) Amplificação, dentro do microrreator, produz várias cópias de um mesmo fragmento de sstDNA (DNA de fita simples). (c) A emulsão é desfeita e as nanoesferas produtivas são enriquecidas e aproveitadas para a etapa posterior do sequenciamento.
Fonte: Modificado de Mardis, 2008b, p.390.

Após a amplificação, a emulsão é desfeita e as nanoesferas improdutivas, ou seja, aquelas deficientes de DNA amplificado, são removidas. As nanoesferas produtivas são tratadas com um desnaturante, de maneira que moléculas de fitas simples que tenham ficado soltas sejam removidas. Tal processo de tratamento seletivo das esferas é denominado enriquecimento (Margulies et al., 2005; Shendure et al., 2005; Shendure; Ji, 2008; Petterson et al., 2009) (Figura 2.16(c)).

Após a PCR em emulsão e o devido enriquecimento das nanoesferas, estas são, então, depositadas em poços distintos, entalhados em uma placa denominada PicoTiterPlate™ (Leamon et al., 2003). Tal placa é confeccionada de maneira que cada poço tenha capacidade para abrigar somente uma esfera. Assim, evita-se a competição por reagentes, quando das reações de sequenciamento, entre os fragmentos da biblioteca (Figura 2.17).

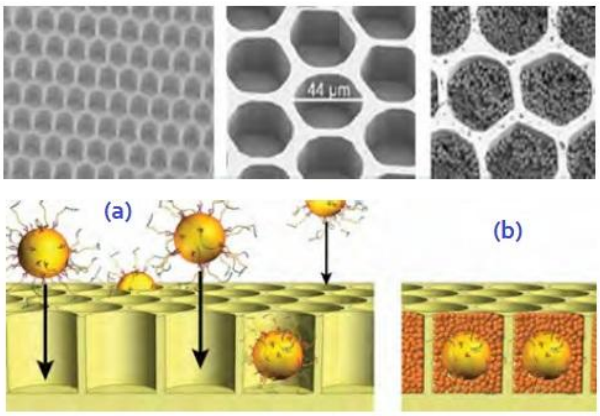


Figura 2.17 - Tecnologia 454 (continuação - parte II). (a) Nanoesferas sendo depositadas na placa PicoTiterPlate™ e, em seguida, (b) recebendo o despejo das esferas que carregam as enzimas necessárias às reações do pirossequenciamento.
Fonte: Modificado de Mardis, 2008b, p.390.

A placa é montada em uma câmara de fluxo, para receber a aplicação e remoção de reagentes (Figura 2.18(b)), durante os ciclos do sequenciamento. Ela também possui um conjunto de fibras óticas que conduzem a eventual emissão de luz até o sistema de aquisição de imagem (a câmera CCD) (Figura 2.18(c)), possibilitando a detecção posicional dessa emissão (Mardis, 2008a; Shendure; Ji, 2008; Ansorge, 2009; Carvalho; Silva, 2010; Metzker, 2010).

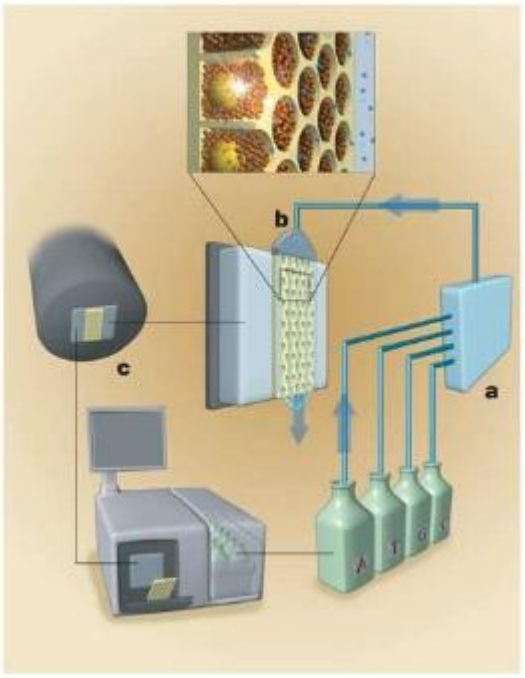


Figura 2.18 - Tecnologia 454 (continuação - parte III). O equipamento sequenciador consiste dos seguintes subsistemas principais: (a) um conjunto bombeador dos reagentes; (b) uma câmara de fluxo que inclui a placa com o arranjo poços-fibras óticas; (c) um sistema de aquisição de imagem baseado em câmera CCD (*charged-coupled device*), o qual detecta as emissões de luz conduzidas pelas fibras óticas e um computador que serve como interface para o usuário e para controlar a operação do instrumento.
Fonte: Extraído de Margulies et al., 2005, p.377 (tradução nossa).

Após a distribuição das nanoesferas pela placa, outros grânulos[19] ainda menores, carregando enzimas necessárias ao princípio do pirossequenciamento (ATP sulfurilase e luciferase) também são adicionados aos poços (Shendure; Ji, 2008; Metzker, 2010) (Figura 2.19(b)), ficando dispostos completamente ao redor das nanoesferas maiores. Os outros reagentes, tais como a DNA polimerase e o *primer* de sequenciamento, também são adicionados por toda a placa, para a obtenção do sequenciamento simultâneo de um a dois milhões de poços (Mardis, 2008a; Petterson et al., 2009; Carvalho; Silva, 2010; Metzker, 2010).

No pirossequenciamento, cada evento de incorporação bem sucedida de um nucleotídeo é detectado pela emissão de luz (Petterson et al., 2009). Para isso, a técnica se baseia na propriedade da bioluminescência (Metzker, 2010) e mede a liberação de um

[19] Feitos de látex e com propriedades magnéticas (Mardis, 2008a).

pirofosfato inorgânico (PPi), oriundo da adição de um desoxinucleotídeo, pela DNA polimerase, à cadeia sendo sequenciada. Esse pirofosfato é, então, convertido, pela enzima ATP sulfurilase, para ATP (adenosina trifosfato), que, por sua vez, é usado pela luciferase para oxidar a luciferina. A reação produz um sinal de luz que é capturado pela câmera CCD (Carvalho; Silva, 2010) (Figura 2.19).

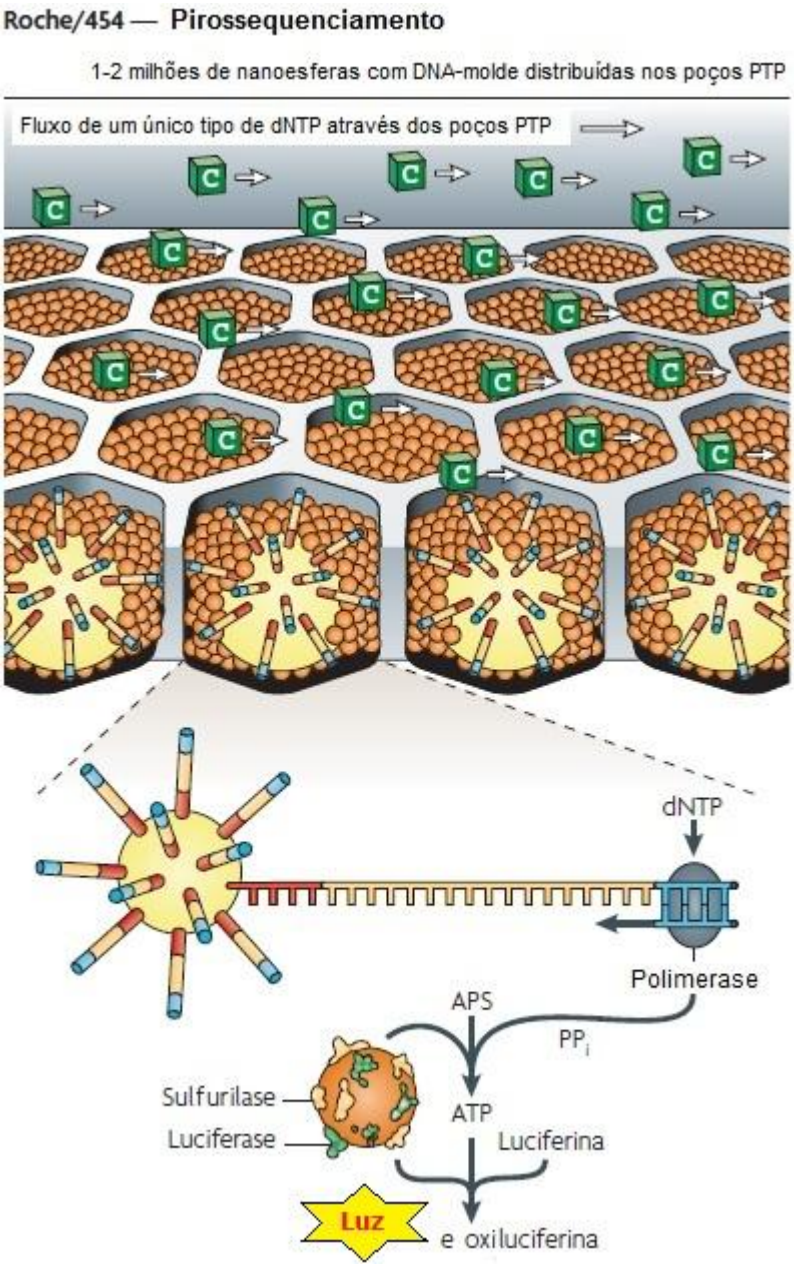


Figura 2.19 - Pirossequenciamento na plataforma Roche/454 *Titanium*. Após a distribuição das nanoesferas com DNA-molde amplificado pelos poços PicoTiterPlate™ (PTP) individuais, grânulos ainda menores, carregados com sulfurilase e luciferase, são adicionadas ao processo. Neste exemplo, é mostrado o fluxo de um único tipo de 2'-desoxirribonucleosídeo trifosfato (dNTP) — no caso, citosina —, através dos poços PTP. A placa com o arranjo poços-fibras óticas é montada em uma câmara de fluxo, possibilitando a aplicação dos reagentes aos poços preenchidos com as esferas. A parte inferior da placa (com as fibras óticas) é acoplada diretamente a uma câmera CCD de alta resolução, permitindo a detecção de luz gerada em cada poço PTP no qual esteja ocorrendo a reação de pirossequenciamento. PPi - pirofosfato inorgânico; APS - adenosina 5'-fosfosulfato; ATP - adenosina trifosfato.
Fonte: Modificado de Metzker, 2010, p.38 (tradução nossa).

O processo é realizado em ciclos e, a cada um deles, um dos quatro tipos de nucleotídeos, seguindo uma ordem pré-definida, é adicionado à reação e forçado a fluir pelos poços. Se o nucleotídeo adicionado for incorporado à sequência em síntese, um sinal de luz

visível será gerado, sendo, sua intensidade, um reflexo da quantidade de nucleotídeos sucessivos, desse tipo específico, que foram incorporados à molécula. Como o nucleotídeo aplicado ao processo é conhecido de antemão, a emissão de luz (Figura 2.19) denuncia a identidade da base incorporada à sequência da fita de DNA em crescimento (Ronaghi, 2001; Ansorge, 2009; Carvalho; Silva, 2010; Griffiths et al., 2012). Ao final de cada rodada, uma outra enzima, apirase, é usada para a degradação de desoxinucleotídeos que não foram incorporados, bem como para interromper a reação, ao degradar ATP (Ronaghi et al., 1998; Petterson et al., 2009).

A reação é repetida por diversas vezes e os sinais obtidos em cada poço, referentes a todos os ciclos, são integrados de forma a gerarem a sua respectiva leitura de sequência. A leitura é apresentada sob a forma de um *pirograma*, o qual exibe, além da ordem de nucleotídeos incorporados, a intensidade do sinal de quimioluminescência proporcional ao total de moléculas de pirofosfato liberadas (Weiss, 2010; Griffiths et al., 2012) (Figura 2.20).

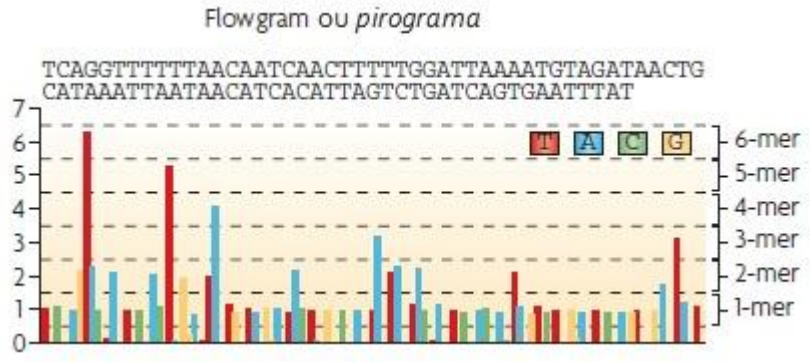


Figura 2.20 - A luz visível, gerada pelas reações enzimáticas em cascata, é gravada como uma série de picos, registro este denominado *flowgram* ou *pirograma*.
Fonte: Modificado de Metzker, 2010, p.38 (tradução nossa).

2.3.2. Solexa/Illumina

A tecnologia Solexa/Illumina surgiu em 2006, baseada no princípio químico do *sequenciamento por síntese*, empregando quatro tipos de nucleotídeos proprietários, dotados da capacidade de terminação reversível e marcados com diferentes fluoróforos e, também, uma enzima DNA polimerase especialmente habilitada para incorporá-los (Illumina, Inc., 2007; Ansorge, 2009).

Nessa tecnologia, o DNA é fragmentado aleatoriamente e, após a fragmentação, dois tipos diferentes de adaptadores são ligados às extremidades dos fragmentos. Em seguida, esses fragmentos são desnaturados, distribuídos e imobilizados (por uma das extremidades), também aleatoriamente, em uma superfície sólida proprietária da tecnologia, denominada *Flow cell*, a qual é revestida por uma camada densa de oligonucleotídeos complementares aos dois tipos de adaptadores dos fragmentos (Illumina, Inc., 2007; Ansorge, 2009). Por meio de um processo conhecido como PCR de fase sólida (Carvalho; Silva, 2010; Metzker, 2010), os fragmentos são multiplicados usando-se uma técnica de amplificação em "ponte", a qual forma pequenos aglomerados (*clusters*) de fragmentos do tipo fita simples. Tais *clusters* são formados espontaneamente, devido ao fato das novas cópias produzidas se ligarem à placa muito próximas ao fragmento original. Ao fim da fase de amplificação, depois de vários ciclos de PCR, a placa fica populada por diversos desses *clusters*, os quais contêm por volta de 1000 cópias de um mesmo fragmento de fita simples (Ansorge, 2009; Carvalho; Silva, 2010; Dahlö, 2010). A Figura 2.21 ilustra essas primeiras etapas descritas.

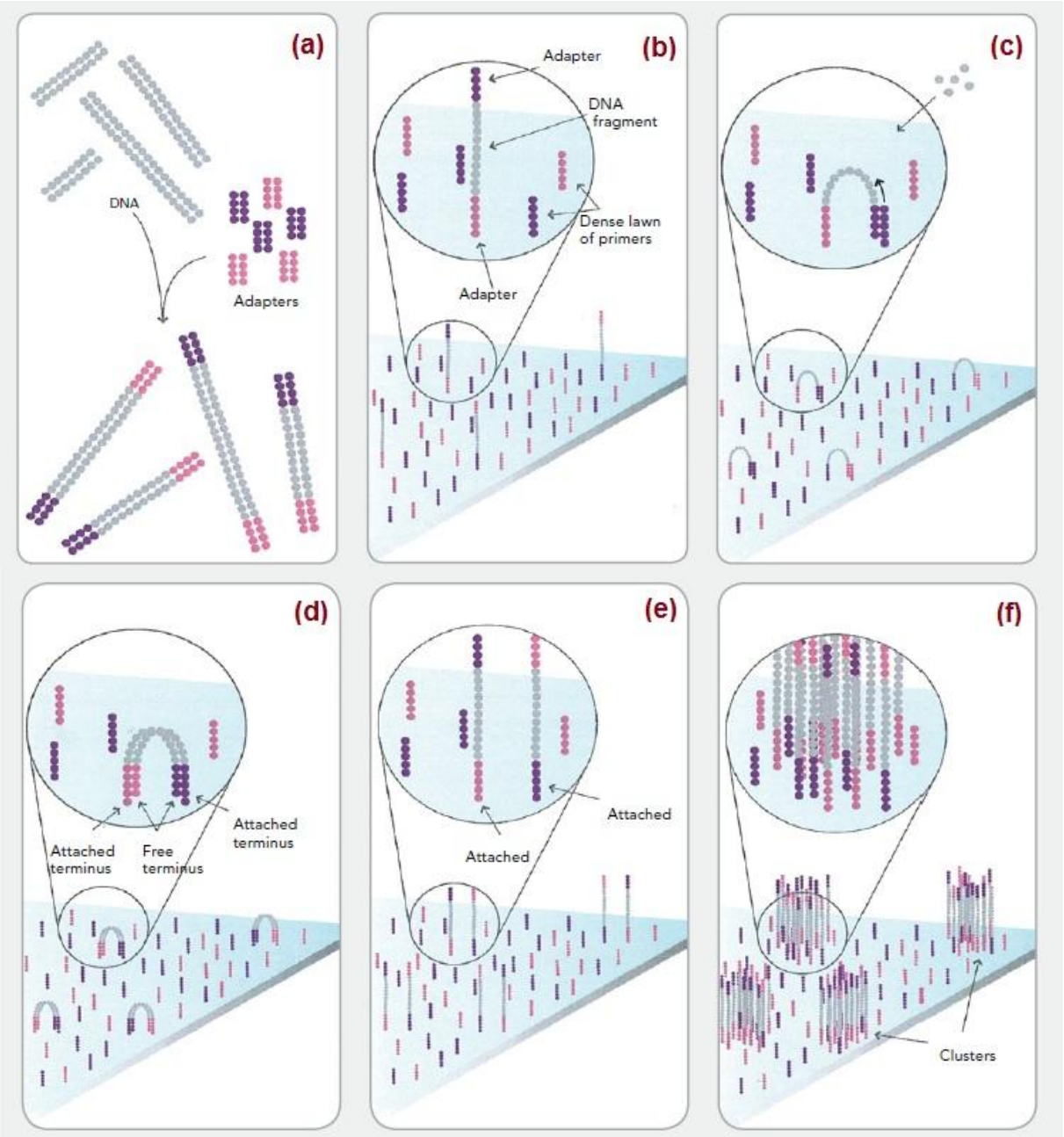


Figura 2.21 - Tecnologia Solexa/Illumina. (a) Preparo da amostra de DNA genômico: o DNA genômico é aleatoriamente fragmentado e cada fragmento recebe a aplicação de adaptadores em ambas as suas extremidades; (b) Fixação do DNA à superfície: fragmentos de fita simples se ligam aleatoriamente à superfície dos canais da *Flow cell*; (c) Amplificação em "ponte": nucleotídeos comuns, não marcados, e a enzima polimerase são fornecidos para iniciar a amplificação de fase sólida em "ponte"; (d) Fragmentos se tornam de fita dupla: a polimerase incorpora nucleotídeos para formar "pontes" de fita dupla no substrato de fase sólida; (e) Desnaturação das moléculas de fita dupla: um processo de desnaturação deixa os moldes de fita simples ancorados ao substrato; (f) Término da fase de amplificação: milhões de *clusters* de fragmentos de DNA fita simples são gerados em cada canal da *Flow cell*.
Fonte: Modificado de Illumina, Inc., 2007. p.2 (tradução nossa).
Nota:
Os termos da figura foram mantidos em inglês, tal como no original.

Uma vez formados os *clusters*, o processo de *sequenciamento por síntese* é iniciado. Uma mistura para as reações de sequenciamento é aplicada à placa, contemplando os quatro nucleotídeos terminadores reversíveis marcados, iniciadores e a enzima DNA polimerase. Após a incorporação do nucleotídeo especial, a replicação é interrompida, os nucleotídeos não

usados são lavados e um laser varre a superfície, excitando o fluoróforo do nucleotídeo terminador e permitindo a sua identificação, bem como a de sua posição no suporte sólido, por uma câmera CCD. Tal câmera, na verdade, registra a cor da luz sendo emitida por cada *cluster* de fragmentos e é dessa forma que o sequenciamento, de fato, acontece. Ao final do ciclo, o grupo bloqueador da extremidade 3' da base e o fluoróforo são removidos e o ciclo de síntese se repete (Ansorge, 2009; Dahlö, 2010). É importante ressaltar que as sequências de milhões de *clusters* espalhados pela placa podem ser determinadas simultaneamente, resultando na vasta produção de dados de sequenciamento (Ansorge, 2009). A Figura 2.22 demonstra essa parte do processo.

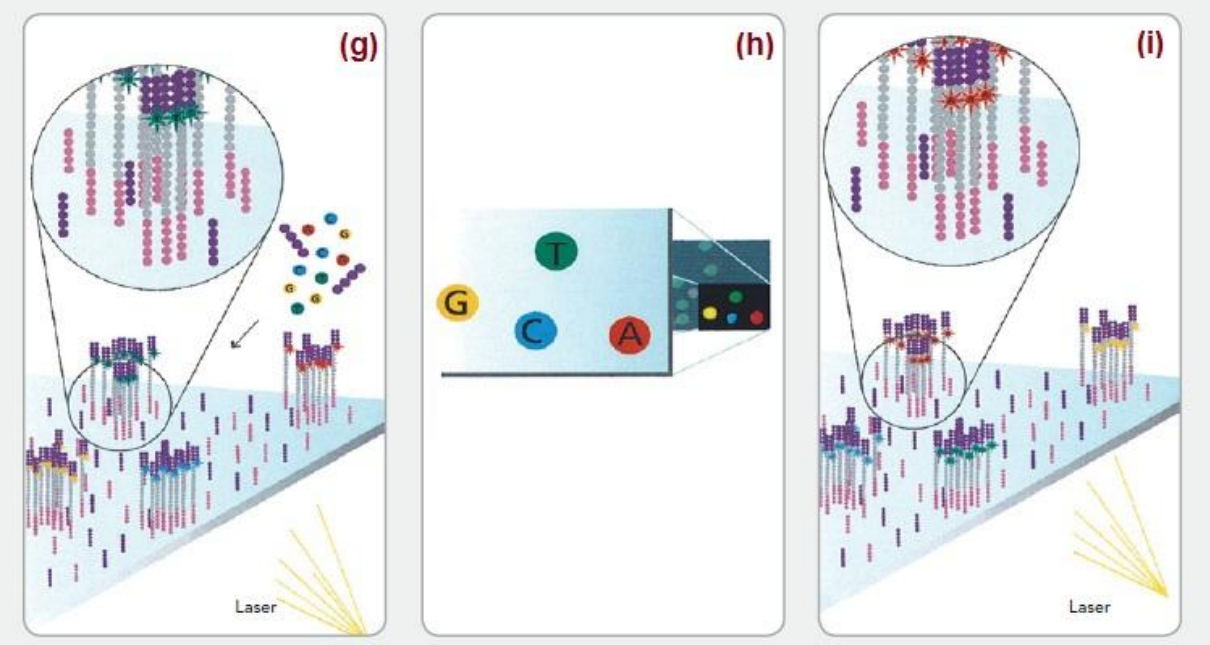


Figura 2.22 - Tecnologia Solexa/Illumina (continuação - parte I). (g) Identificação da primeira base: para iniciar o primeiro ciclo de sequenciamento, os quatro nucleotídeos terminadores reversíveis marcados, *primers* e a enzima DNA polimerase são adicionados à *Flow cell*; (h) Detecção da primeira base: após a excitação pelo laser, a imagem da fluorescência emitida é capturada de cada *cluster* na *Flow cell*. A identidade da primeira base é registrada para cada um dos *clusters*; (i) Identificação da segunda base: para iniciar o segundo ciclo de sequenciamento, os nucleotídeos terminadores reversíveis marcados e a enzima DNA polimerase são aplicados à placa.
Fonte: Modificado de Illumina, Inc., 2007. p.3 (tradução nossa).

A leitura das bases é realizada através da análise sequencial de cada uma das imagens obtidas em cada ciclo de sequenciamento, para cada posição de *cluster* (Carvalho; Silva, 2010; Dahlö, 2010). A Figura 2.23 ilustra a sequência de imagens capturadas e a consequente determinação da leitura de sequenciamento.

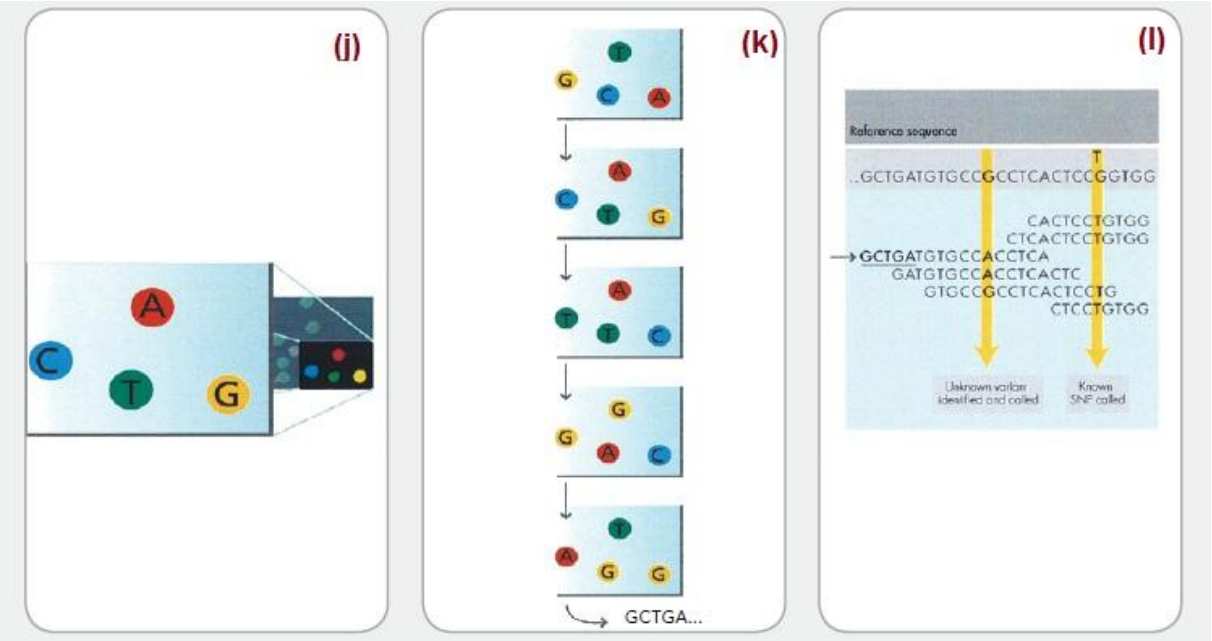


Figura 2.23 - Tecnologia Solexa/Illumina (continuação - parte II). (j) Captura de imagem do segundo ciclo da química de sequenciamento: após a excitação pelo laser, a imagem é capturada tal como da vez anterior. Com isso, é feito o registro da segunda base de cada *cluster*; (k) Leituras de sequências ao longo de vários ciclos da química de sequenciamento: a repetição dos ciclos de sequenciamento determina a sequência de bases para um dado fragmento, uma base de cada vez; (l) Alinhamento dos dados: as leituras obtidas podem ser alinhadas contra um genoma de referência para a identificação de ocorrências de variações entre as sequências.
Fonte: Modificado de Illumina, Inc., 2007. p.3 (tradução nossa).
Nota:
Os termos da figura foram mantidos em inglês, tal como no original.

2.3.3. ABI SOLiD™

O sistema ABI SOLiD™, introduzido no mercado em 2007, ao contrário dos sistemas anteriores, usa uma abordagem de *sequenciamento por ligação*, na qual a enzima DNA polimerase é substituída pela enzima DNA ligase (Ansorge, 2009; Metzker, 2010).

Nesta técnica, o DNA é fragmentado e os segmentos resultantes recebem adaptadores universais (P1 e P2) em suas extremidades. Um dos adaptadores (P1) serve ao anelamento do *primer* da PCR em emulsão e, de maneira similar ao que ocorre na tecnologia 454, os fragmentos são ligados a nanoesferas, as quais são capturadas em micelas que irão atuar como microrreatores na etapa de amplificação. Ao final dessa etapa, em vez de utilizar uma placa com poços para o depósito das nanoesferas carregadas de moléculas de DNA, a tecnologia emprega uma superfície sólida de vidro, na qual as esferas são ligadas quimicamente (Ansorge, 2009; Carvalho; Silva, 2010) (Figura 2.24).

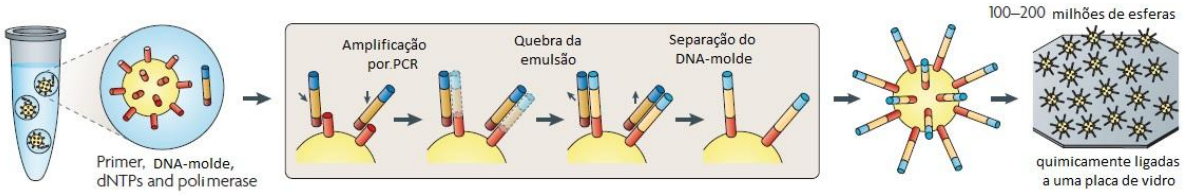


Figura 2.24 - PCR em emulsão do sistema SOLiD™: nanoesferas quimicamente ligadas à uma superfície sólida de vidro.
Fonte: Modificado de Metzker (2010). p.33 (tradução nossa).

No processo de sequenciamento SOLiD™, um *primer universal* é hibridizado ao adaptador P1 ligado à nanoesfera. Cinco etapas distintas compõem o processo, as quais são diferenciadas pelos tamanhos dos *primers* universais utilizados. Na primeira etapa, o *primer* tem *n* bases e se anela exatamente na extremidade do adaptador P1. Na segunda etapa, o *primer* tem *n* - 1 bases, e assim por diante. Além dos *primers*, também participam, da reação, a enzima ligase e sondas curtas (de oito bases cada), marcadas com uma entre quatro cores de fluoróforos possíveis, dependendo do tipo de dinucleotídeo existente em sua extremidade 3' (Carvalho; Silva, 2010). Isso significa dizer que somente as duas primeiras bases apresentam função seletiva de determinação das correspondentes bases na sequência-alvo. As três bases seguintes no octâmero são degeneradas para qualquer sonda, ao passo que as três últimas são universais, sendo responsáveis por carregar o fluoróforo marcador (Carvalho; Silva, 2010; Metzker, 2010).

Ao ser hibridizada a primeira sonda à sequência da fita-molde, a enzima ligase realiza sua ligação à extremidade do *primer* universal. A marcação fluorescente da sonda ligada é detectada e, em seguida, o fluoróforo é clivado, deixando um grupo 5'-fosfato livre na quinta posição da sonda. Uma nova rodada acontece e, da mesma forma, sondas e ligase são adicionadas para a leitura do próximo dinucleotídeo seletivo. As rodadas de ligação de sondas se repetem até que a sequência-alvo tenha sido totalmente coberta (Carvalho; Silva, 2010).

Na segunda etapa do processo de sequenciamento, o sistema é reiniciado como um todo, a partir da desnaturação da fita dupla e retirada do primeiro fragmento que estendeu o *primer* complementar à fita-molde. Um novo *primer* se liga ao adaptador P1, só que, desta vez, com um tamanho de *n* - 1 bases, deixando exposta a última base do adaptador e, consequentemente, deslocando de uma posição para a esquerda todo o mecanismo de determinação de bases (Metzker, 2010).

Ao final das cinco etapas do processo, todas as bases da sequência-alvo terão sido "visitadas" por algum dinucleotídeo das sondas. Assim, cada base é, na verdade, lida duas vezes, o que reduz a taxa de erros (Carvalho; Silva, 2010). Outro detalhe da tecnologia é que as leituras obtidas são de dinucleotídeos (e não de bases), os quais, por sua vez, são representados por um código de cores. O sistema SOLiD™, portanto, gera o resultado do sequenciamento no espaço de cores e não no espaço de bases, sendo necessária uma decodificação dos sinais das leituras pela combinação dos dados (Carvalho; Silva, 2010; Metzker 2010). Como as bases do adaptador P1 são conhecidas, é possível fazer a identificação da primeira base do fragmento, quando da segunda etapa do processo de sequenciamento. Daí em diante, os demais sinais são especificados pela única combinação possível de cores, levando-se em consideração a base conhecida (Carvalho; Silva, 2010). As Figuras 2.25, 2.26 e 2.27 oferecem mais detalhes a respeito do processo de sequenciamento SOLiD™ e de decodificação mencionados.

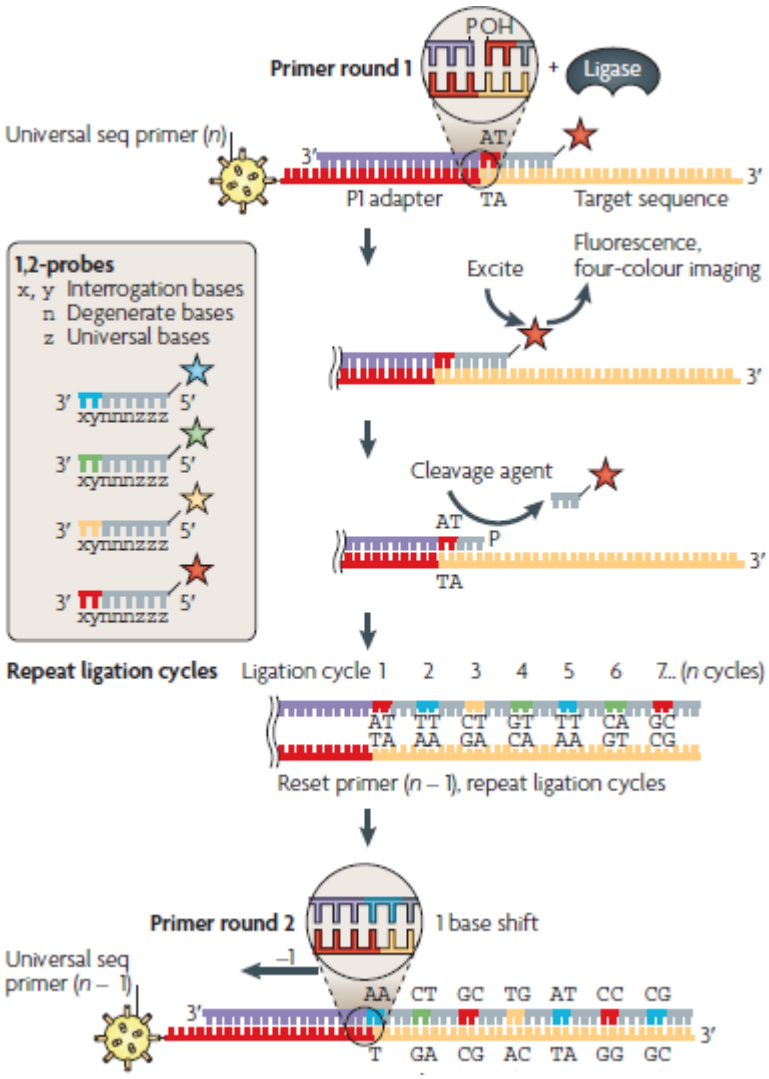


Figura 2.25 - O sequenciamento no sistema SOLiD™. A partir do anelamento de um *primer* universal (com *n* bases de tamanho) ao adaptador P1, o qual se liga especificamente a uma esfera, uma biblioteca de sondas de dinucleotídeos é adicionada. Condições apropriadas permitem a hibridização seletiva de uma das sondas do *pool* nas devidas posições complementares da sequência-alvo. Na primeira etapa do processo, a enzima ligase une a primeira base da sonda com a última base do *primer*. O sinal de fluorescência é lido e a sonda é clivada para a remoção das três últimas bases que carregam o fluoróforo. Isso deixa livre um grupo 5'-fosfato na quinta base da sonda, permitindo que uma nova sonda possa ser usada para interrogar as duas bases seguintes e, assim por diante, até que a sequência-alvo tenha sido coberta. O ciclo SOLiD™ é repetido mais cinco vezes para que todas as bases da sequência-alvo possam ser determinadas. Em todo início de etapa, o fragmento que estendeu o *primer* é removido após a desnaturação da fita dupla. Em seguida, um novo *primer* (com *n* - 1 bases de tamanho em relação ao anteriormente usado) é adicionado ao adaptador P1 e novas rodadas de ligação de sondas acontecem, até que a sequência-alvo seja totalmente varrida novamente. Ao serem concluídas as cinco etapas, todas as bases da sequência-alvo terão sido lidas duas vezes.

Fonte: Modificado de Metzker (2010). p.38 (tradução nossa).

Nota:

Os termos da figura foram mantidos em inglês, tal como no original. Informações compiladas de Carvalho e Silva (2010) e Metzker (2010).

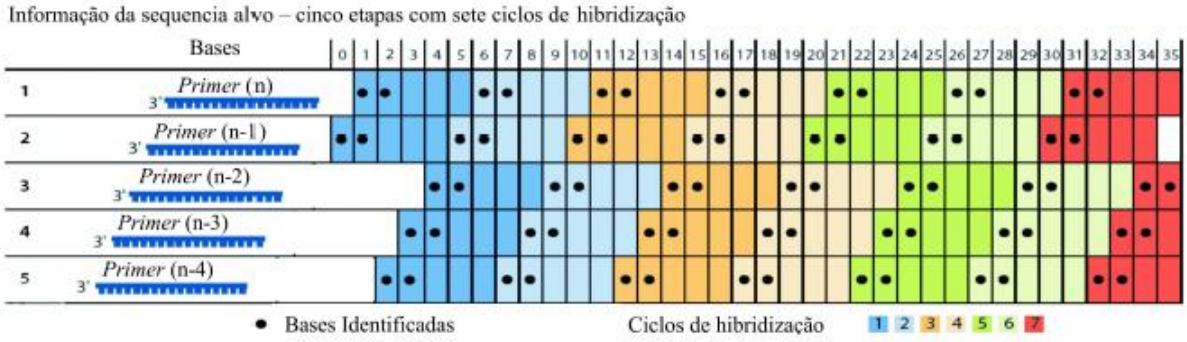


Figura 2.26 - As cinco etapas do processo e a ordem na qual as bases da sequência-alvo são determinadas por dupla leitura por etapas diferentes.
Fonte: Extraído de Carvalho e Silva (2010). p.740.

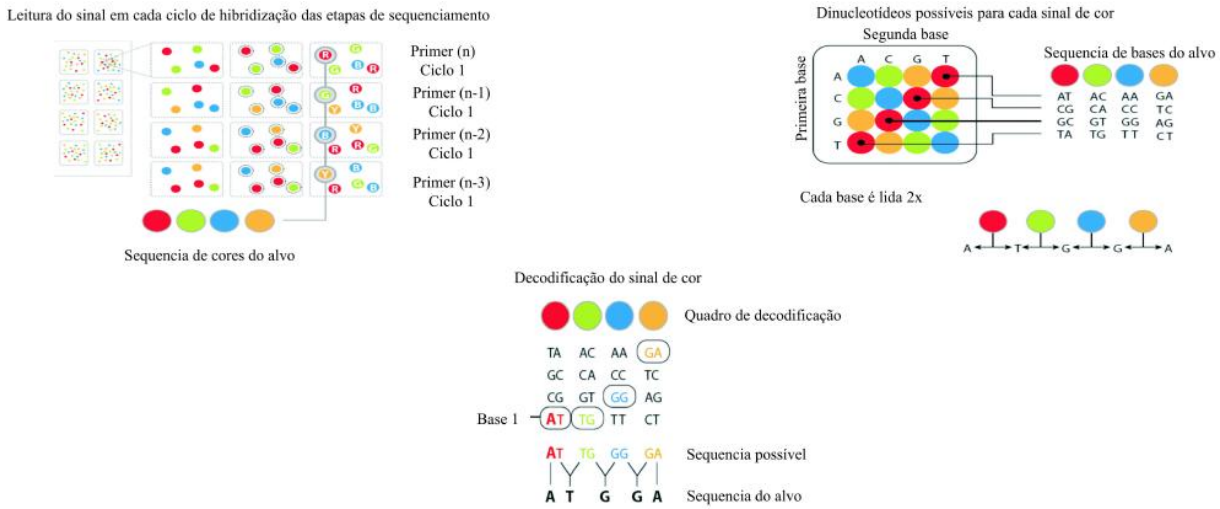


Figura 2.27 - Esquema de codificação em duas bases: quatro sequências de dinucleotídeos são associadas a uma cor de fluoróforo (por exemplo, AA, CC, GG e TT são codificadas pela cor azul). Consequentemente, todas as (dezesseis) combinações possíveis de dinucleotídeos são codificadas por apenas quatro fluoróforos. Cada base da sequência-alvo é interrogada duas vezes e compilada em uma *string* de bits de dados em espaço de cores. As duas leituras de cada base são necessárias para que a sequência do dinucleotídeo da sonda possa ser resolvida. O processo de resolução se inicia a partir da identificação da primeira base da sequência-alvo, na segunda etapa de sequenciamento (*primer n* - 1), quando ocorre a liberação de uma base previamente conhecida para a hibridização com a sonda, no caso, a última base do adaptador P1.
Fonte: Extraído de Carvalho e Silva (2010). p.740.
Nota:
Informações compiladas de Carvalho e Silva (2010) e Metzker (2010).

2.4. A montagem de genomas

Conforme visto na seção introdutória deste trabalho, a montagem de genomas é um elemento crucial para o desenvolvimento da Genômica, o estudo dos genomas em sua completude. A sequência de DNA completa de um genoma pode ser considerada como seu mapa de mais alta resolução (Griffiths et al., 2012). Para proceder à análise biológica mais realista a seu respeito, o ideal seria obter um número de sequências completas, que fossem correspondentes aos cromossomos existentes em uma determinada célula. Na realidade, entretanto, a tarefa não é tão simples. Como visto, máquinas de sequenciamento só

conseguem "ler" a sequência de DNA em pedaços cujos tamanhos, atualmente, variam tipicamente da ordem de 30 a 1000 pares de bases (Olson, 2009). Por outro lado, o número de nucleotídeos presentes nos genomas dos organismos é bem superior. Genomas bacterianos, por exemplo, alcançam a ordem de milhões de pares de bases. E esse número aumenta para bilhões de pares de bases no homem, bem como na maioria dos animais e plantas (Pop et al., 2002).

Assim, para a obtenção da sequência completa de um genoma, usualmente é necessária uma abordagem do tipo "dividir para conquistar", na qual, relembrando o que já foi abordado nas seções anteriores, o DNA é quebrado (por técnicas de laboratório) em diversos fragmentos os quais, por sua vez, são sequenciados, gerando trechos de sequências, também chamados de leituras (ou *reads*), que, posteriormente, são "montados" por programas de computador (Adams et al., 2000; Myers et al., 2000; Waterston et al., 2002). Para isso, em abordagens de montagem clássicas (Figura 2.28), tais fragmentos são sobrepostos de maneira que as regiões de identidade entre eles sejam alinhadas, ao mesmo tempo em que a justaposição das regiões sem identidade vai sendo revelada, por meio de uma abordagem tipicamente conhecida como *Overlap-Layout-Consensus* (algo como "Sobreposição-Arranjo-Consenso", em português).

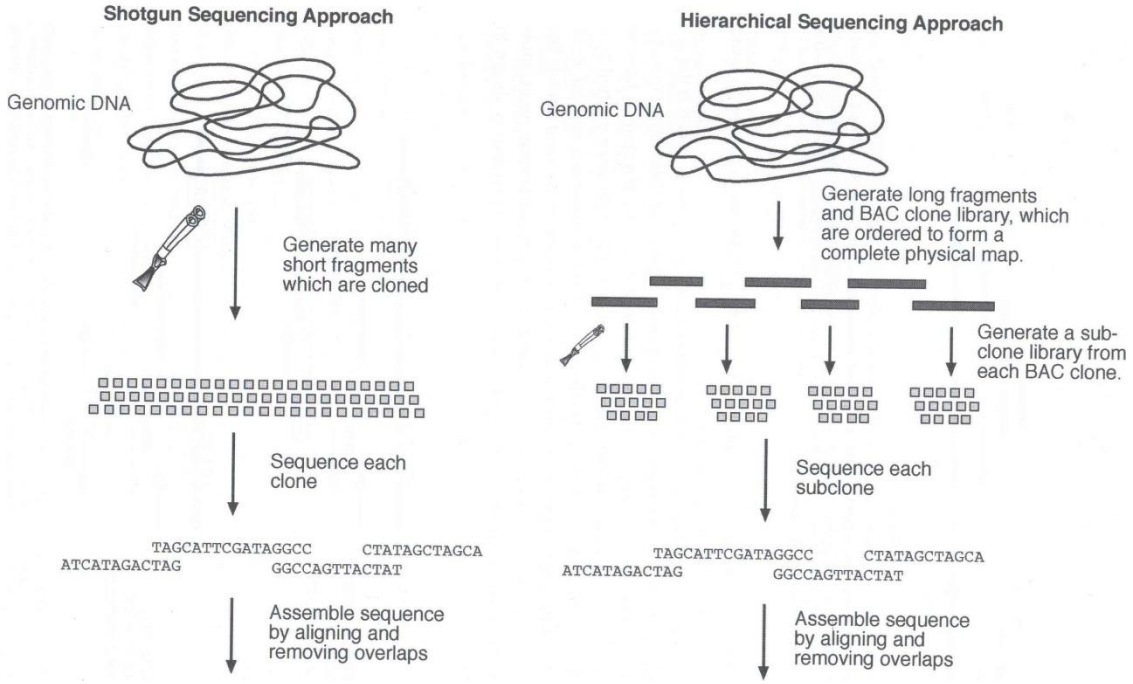


Figura 2.28 - Esquema comparativo de duas estratégias de sequenciamento de genoma completo clássicas. *Shotgun*: O DNA é fragmentado em segmentos (~2 kpb - 50 kpb), os quais são clonados em vetores pequenos (por exemplo, plasmídeos, cosmídeos, fosmídeos) e sequenciados individualmente. Os fragmentos sequenciados são remontados em uma sequência final. *Hierárquica*: O DNA é fragmentado em segmentos intermediários (~150 kbp) que são clonados em BACs, por exemplo. Um mapa físico é elaborado usando-se os BACs como referência. Cada clone BAC é submetido ao sequenciamento e, em seguida, uma sequência final é remontada.
Fonte: Modificado de Xiong, 2006. p.247.
Nota:
Os termos da figura foram mantidos em inglês, tal como no original.

Primeiramente, os fragmentos curtos são unidos para formar fragmentos maiores (depois da remoção das sobreposições); os *contigs* (etapa *Overlap*). Estes são, então, sobrepostos e, em seguida, unidos para formar *scaffolds* (ou *supercontigs*) que, por sua vez, são orientados em um único sentido (etapa *Layout*). Ao final, espera-se encontrar uma sequência contínua e única derivada a partir da união dos *scaffolds* (Xiong, 2006; Olson, 2009). A essa etapa realizada no computador, dá-se o nome de "montagem" de genoma e é legítimo deduzir que, quanto maior o genoma em questão, mais difícil se torna juntar todos os segmentos adquiridos no processo de sequenciamento. Além disso, problemas como erros de sequenciamento (por exemplo, bases incorretamente identificadas), contaminação por vetores ou adaptadores, e a ocorrência de polimorfismos e/ou regiões repetitivas no genoma fazem com que a tarefa seja ainda mais desafiadora do ponto de vista computacional. Nesses casos, muito provavelmente, as sequências resultantes irão possuir *gaps* (lacunas) entre os *contigs*, impedindo a formação de uma sequência contínua e única.

Ainda com relação à montagem de genomas, do ponto de vista relacionado à análise dos dados, vale também ressaltar que ela pode ser de dois tipos: o primeiro, às vezes designado como *ressequenciamento*, utiliza um genoma de referência, contra o qual as

leituras são alinhadas por similaridade. Normalmente, esse genoma de referência é escolhido levando-se em consideração a sua proximidade filogenética em relação ao genoma sequenciado e, em projetos deste tipo, a cobertura de sequenciamento necessária é menor (da ordem de 8 a 12 vezes) (Schuster, 2008). Por causa do processo de alinhamento das leituras, tal abordagem de trabalho é, frequentemente, designada como *mapeamento* (Shendure e Ji, 2008; Horner et al., 2009; Bao et al., 2011) ou *alinhamento* (Paszkiewicz; Studholme, 2012) simplesmente. O segundo tipo é conhecido como montagem *de novo* ou *ab initio* ou sequenciamento de genomas desconhecidos. Nele, a montagem é executada usando-se as próprias leituras, ou seja, não há um genoma de referência para auxiliar o processo. Nesse caso, a cobertura de sequenciamento ideal é maior (da ordem de 25 a 70 vezes) (Schuster, 2008). Há situações, também, em que ambas as abordagens podem ser utilizadas em um mesmo projeto (Pop, 2009; Paszkiewicz; Studholme, 2012).

Portanto, de forma resumida, montagem de genomas pode ser definida como a reconstrução da sequência de um genoma, a partir das várias leituras obtidas na etapa de sequenciamento.

2.5. Bioinformática para NGS

Apesar das tecnologias NGS serem relativamente recentes, já existe uma grande variedade de pacotes de software disponíveis para a análise dos dados produzidos, com funcionalidades que se enquadram em diversas categorias, as quais incluem: (i) alinhamento das leituras de sequenciamento contra um genoma de referência; (ii) *base-calling* (atribuição ou identificação de bases) e/ou detecção de polimorfismos; (iii) montagem *de novo*, a partir de leituras pareadas ou não-pareadas e (iv) visualização e anotação do genoma (Shendure; Ji, 2008). Especialmente quanto à classe de programas "montadores", ou seja, aqueles pertencentes às primeira e terceira categorias, um dos motivos para a alta disponibilidade de soluções é que, de maneira geral, os programas que funcionam adequadamente para leituras longas, provenientes do sequenciamento tradicional, não o fazem tão bem quando lidam com a massa de leituras curtas das plataformas NGS (Pop; Salzberg, 2008; Shendure; Ji, 2008; Flicek; Birney, 2009; Horner et al., 2009; Paszkiewicz; Studholme, 2010; Bao et al., 2011; Earl et al., 2011).

No que diz respeito especificamente à montagem de genomas, programas do tipo filtro de qualidade ou de *base-calling* são usados para, previamente, preparar ou melhorar os dados para as etapas seguintes. Esses programas analisam a qualidade das leituras (Ewing et al., 1998; Cock et al., 2010) provenientes do sequenciador, removendo ou corrigindo aquelas de baixa qualidade. O objetivo é, então, facilitar a etapa seguinte do alinhamento/montagem das sequências (Smith et al., 2008; Li et al., 2010; Ramos et al., 2011). Programas como Ibis (Kircher et al., 2009), PRINSEQ (Schmieder; Edwards, 2011), PyroBayes (Quinlan et al., 2008), QA (Ramos et al., 2011), SAET - *SOLiD™ Accuracy Enhancement Tool* (http://solidsoftwaretools.com/gf/project/denovo/) e a ferramenta de manipulação de arquivos FASTQ do *pipeline* GALAXY (Blankenberg et al., 2010a; Giardine et al., 2005), podem ser considerados como exemplos desse tipo. Já programas como o PIQA (Martínez-Alcántara et al., 2009) permitem a análise da qualidade e a visualização de possíveis problemas técnicos, antes das leituras serem encaminhadas adiante, facilitando a tomada de decisão pelo usuário final.

Existem, também, programas alinhadores/montadores que já levam em conta a qualidade das leituras durante o processo de montagem. São exemplos desse tipo os pacotes: MAQ (Li H et al., 2008) e RMAP (Smith et al., 2008), para alinhamento contra um genoma de referência e QRSA (Bryant et al., 2009), para montagem do tipo *de novo*.

Para a categoria de programas alinhadores contra um genoma de referência, tais como o MAQ e RMAP já citados, são ainda exemplos, dentre outros: AMOScmp (Pop et al., 2004a), BFAST (Homer et al., 2009), Bowtie (Langmead et al., 2009), BWA (Li; Durbin, 2009), ELAND (Cox, 2007), MapNext (Bao et al., 2009), Mosaik (The MarthLab, 2009 - http://bioinformatics.bc.edu/marthlab/Mosaik), PerM (Chen et al., 2009), SHRiMP (Rumble et al., 2009), SeqMap (Jiang; Wong, 2008), a série SOAP/SOAP2 (Li R et al., 2008; Li R et al., 2009), SOCS (Ondov et al., 2008), SSAHA2 (Ning et al., 2001), STAMPY (Lunter; Goodson, 2010) e ZOOM (Lin et al., 2008). Outros programas, como Corona Lite (http://solidsoftwaretools.com/gf/project/corona/), MUMmer2 (Delcher et al., 1999; Delcher et al., 2002) e SHORE (Ossowski et al., 2008), contemplam, dentre outras funcionalidades, a possibilidade de realizar alinhamentos e, por isso, podem ser usados para auxiliar a estratégia de montagem usando genoma de referência.

Para a categoria de montagem do tipo *de novo*, além do já mencionado QRSA, podem ser citados, dentre outros: ABySS (Simpson et al., 2009), a série ALLPATHS/ALLPATHS2 (Butler et al., 2008; MacCallum et al., 2009), CABOG (Miller et al., 2008), Edena (Hernandez et al., 2008), Euler-SR (Chaisson; Pevzner, 2007), Meraculous (Chapman et al., 2011), MIRA (Chevreux et al., 1999; Chevreux et al., 2004; Chevreux, 2010), Newbler (Margulies et al., 2005), Phusion/Phusion2 (Mullikin; Ning, 2002; ftp://ftp.sanger.ac.uk/pub/users/zn1/phusion2/), SGA (Simpson; Durbin, 2012), SHARCGS (Dohm et al., 2007), SOAPdenovo (Li et al., 2010), SSAKE (Warren et al., 2007), VCAKE (Jeck et al., 2007) e Velvet (Zerbino; Birney, 2008).

Além dos pacotes citados, existem outros, com diferentes funcionalidades, que também podem ser usados para auxiliar, direta ou indiretamente, o processo de montagem como um todo. O software iCORN (Otto et al., 2010), por exemplo, usa dados de sequenciamento NGS para avaliar a acurácia e corrigir genomas de referência. Posteriormente, esses genomas de referência "refinados" podem ser usados em uma nova tentativa de alinhamento.

Já a solução IMAGE (Tsai et al., 2010) usa uma abordagem de montagem "local" de leituras para corrigir *gaps*. A estratégia padrão para o fechamento de *gaps*, por exemplo, envolve o desenho de *primers* de oligonucleotídeos para que um novo sequenciamento, específico para as extremidades de *contigs*, possa ser realizado e, após a obtenção das novas leituras, estas possam ser manualmente alinhadas para resolver as regiões duvidosas. Isso envolve um intenso trabalho laboratorial. IMAGE procura evitar esse trabalho, aumentando a

qualidade dos "rascunhos" de montagem (ou seja, aqueles que ainda não foram finalizados), através da montagem local de leituras nas regiões com *gaps*, sem intervenção manual.

A ferramenta inGAP (Qi et al., 2010) é outra que pode ser usada para auxiliar o fechamento de *gaps*, além de incorporar também softwares alinhadores como BWA, BLAT (Kent, 2002) e BLAST (Altschul et al., 1990) e, também, outras funcionalidades como a detecção de SNPs (*Single Nucleotide Polymorphisms*) e *InDels*, os quais podem ser visualizados em sua interface gráfica.

Para a validação da sequência final ou dos *scaffolds* obtidos após a montagem, OSLay (Richter et al., 2007) pode ser uma opção, pois usa sintenia gênica entre a sequência-alvo e uma outra de referência. Se necessário, OSLay pode ser usado também para reorganizar *contigs*. Seus resultados podem ser visualizados em sua interface gráfica ou podem ser transferidos para outras ferramentas de maneira a auxiliar, por exemplo, o desenho de *primers* para fechamento de *gaps*.

ZORRO (http://www.lge.ibi.unicamp.br/zorro/) é uma outra ferramenta, sob a forma de *pipeline*, que combina alguns pacotes já existentes (é baseada, por exemplo, no *pipeline* MINIMUS2 (http://sourceforge.net/apps/mediawiki/amos/index.php?title=Minimus2), além de usar AMOS (http://sourceforge.net/apps/mediawiki/amos/index.php?title=AMOS) e mais algumas ferramentas já mencionadas, como MUMmer e Bowtie) para servir como montador de leituras provenientes de tecnologias de sequenciamento NGS diferentes (no caso, 454 e Illumina). A idéia básica de ZORRO é usar dois conjuntos de *contigs* já pré-montados para tentar obter uma montagem ainda melhor e mais consistente.

Do exposto, mais uma vez reitera-se a grande variedade de programas existentes para o tratamento dos dados de NGS. No Anexo A, podem ser encontradas, como exemplo, tabelas com listagens de programas típicos para NGS, baseadas nos respectivos trabalhos de Shendure e Ji (2008), Horner et al. (2009), Bao et al. (2011), Zhang J et al. (2011) e Thudi et al. (2012). É possível observar a variedade de soluções tornadas disponíveis ao longo do tempo. Uma outra listagem "clássica" também pode ser obtida no sítio da *Web* SeqAnswers (http://seqanswers.com/wiki/Software/list), onde cerca de 400 ferramentas de NGS são apresentadas (Paszkiewicz; Studholme, 2012).

Ainda, especificamente quanto à categoria de programas "montadores" (alinhadores contra um genoma de referência ou montadores *de novo*), o Anexo B traz um compilado de informações, baseado nos trabalhos de Flicek e Birney (2009), Pop (2009), Miller et al.

(2010), Paszkiewicz e Studholme (2010), Sasson (2010), Bao et al. (2011), Ruffalo et al. (2011), Zhang J et al. (2011), Henson et al. (2012) e Paszkiewicz e Studholme (2012), a respeito das principais características e estratégias de funcionamento de seus algoritmos.

2.6. Aspectos teóricos adicionais relacionados ao projeto

Devido à multidisciplinaridade inerente ao trabalho, esta seção visa prover detalhes adicionais a ele relacionados quanto a essa particularidade.

2.6.1. Organismos candidatos às montagens básicas de dados de sequenciamento

Os seguintes organismos, de importância médica, veterinária ou biológica, foram considerados para o estudo.

2.6.1.1 Leishmania amazonensis

Agente etiológico da leishmaniose, o parasito protozoário unicelular do gênero *Leishmania* (Figura 2.29), família *Trypanosomatidae*, ordem *Kinetoplastidae*, tem grande importância médica e econômica. A infecção, transmitida através da picada de fêmeas de mosquitos da subfamília *Phlebotominae*, apresenta diferentes formas clínicas (visceral, muco-cutânea, cutânea difusa e cutânea), dependendo da espécie de leishmania infectante e do estado imunológico do hospedeiro. Caracterizada por alto grau de morbidade e considerável taxa de mortalidade, especialmente no caso do homem, tipicamente mantém seus portadores à margem da sociedade, devido às deformidades, mutilações e cicatrizes que provoca. Epidemiologicamente falando, a doença pode ser classificada como zoonose, quando inclui animais hospedeiros como reservatórios no ciclo de transmissão, e como antroponose, na qual o homem é a única fonte de infecção para o mosquito vetor (Wagner, 2006; Pitaluga, 2007; Gomes, 2010; Tschoeke, 2010; Alonso, 2011).

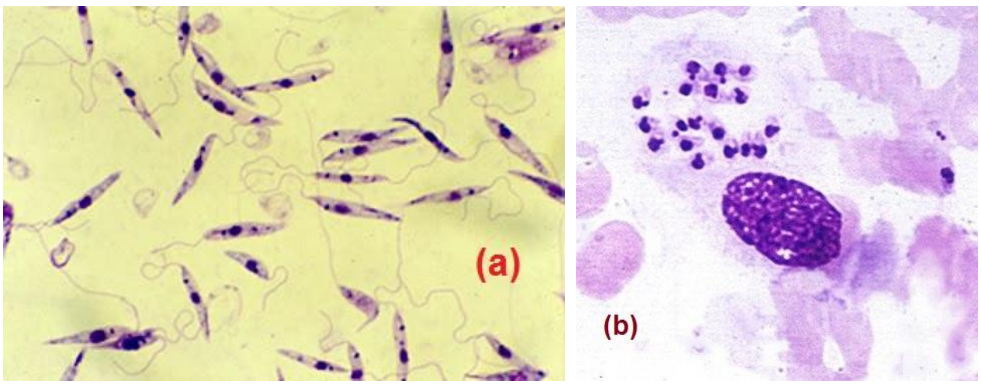


Figura 2.29 - *Leishmania* sp. (a) Formas promastigotas de *Leishmania* sp. Nesta fase de seu ciclo de vida, o parasito é extracelular, flagelado e móvel, sendo encontrado no intestino do inseto vetor. (b) Formas amastigotas de *Leishmania* sp. No outro estágio de seu ciclo de vida, o parasito é intracelular e com flagelo interno, sendo encontrado nos macrófagos do hospedeiro mamífero (Pitaluga, 2007).
Fonte: (a) Modificado de http://www.uni-tuebingen.de/modeling/Mod_Leish_Intro_de.html. (b) Modificado de http://enfermagem-sae.blogspot.com.br/2009/04/leishmaniose-visceral-ou-calazar.html.

A *Leishmania amazonensis*, pertencente ao complexo *Leishmania mexicana*, subgênero e gênero *Leishmania*, é patogênica ao homem, causando leishmaniose cutânea difusa e sendo transmitida, principalmente, pelo flebotomíneo *Lutzomyia flaviscutellata*. É encontrada na região da floresta amazônica, tendo sido isolada em outros estados brasileiros também. Produzir informações sobre seu genoma, portanto, pode auxiliar o maior entendimento da biologia do parasito e, consequentemente, de seu papel na doença (Tschoeke et al., 2011).

2.6.1.2 Escherichia coli DH10B

A cepa DH10B de *E. coli* é amplamente usada em estudos de biologia molecular e possui genoma conhecido (Durfee et al., 2008). Tipicamente, empresas fabricantes de sequenciadores utilizam tal cepa para a aferição e avaliação de seus instrumentos (Applied Biosystems, 2010; Illumina, Inc., 2010b).

2.6.1.3 Phlebotomus papatasi

A leishmaniose cutânea é uma das principais doenças de pele em regiões tropicais e, no Velho Mundo, tal forma clínica é geralmente causada pelo parasito protozoário *Leishmania major*. Nessa região do globo, o flebotomíneo invertebrado *Phlebotomus papatasi* (Figura 2.30), atua como seu principal vetor (Gomes, 2010; The Genome Institute, 2012). Os parasitos são transmitidos ao hospedeiro vertebrado quando a fêmea de flebotomíneo faz o repasto sanguíneo, regurgitando-os na pele do hospedeiro.

Figura 2.30 - O flebotomíneo *Phlebotomus papatasi.*
Fonte: Extraído de The Genome Institute, 2012.

Conforme o Genome Institute (2012), informações e dados de experimentos de montagem do genoma desse organismo estão disponíveis e podem ser usados como referência[20]. De maneira análoga à mencionada no item 2.6.1.1, entender mais sobre a biologia do inseto vetor pode refletir na melhor compreensão das relações parasito/vetor e, consequentemente, contribuir no combate à doença.

2.6.2. Um paradigma de desenvolvimento de software para sustentar o projeto

"Análise de sistemas é a atividade que tem como finalidade a realização de estudos de processos a fim de encontrar o melhor caminho racional para que a informação possa ser processada. Os analistas de sistemas estudam os diversos sistemas existentes entre hardwares, softwares e o usuário final" (Wikipédia, 2012).

A Bioinformática é definida por BISTI (2000, tradução nossa) como "a pesquisa, desenvolvimento, ou aplicação de ferramentas e abordagens computacionais, no sentido de expandir o uso de dados biológicos, médicos, comportamentais ou de saúde, incluindo aquelas voltadas à captura, armazenamento, organização, arquivo, análise, ou visualização de tais dados". Na visão de Luscombe et al. (2001), parte da definição de Bioinformática envolve tarefas como as de compreensão e organização, em larga escala, das informações biológicas.

Ao visar desenvolver ou aplicar ferramentas computacionais com a finalidade de expandir e tornar mais úteis e acessíveis os dados biológicos — especialmente aqueles provenientes de plataformas NGS —, sob a proposta de utilizar, combinar e integrar ferramentas de uso livre já disponíveis, o trabalho procura manter-se alinhado a essas

[20] http://genome.wustl.edu/data.cgi;
http://genome.wustl.edu/pub/organism/Invertebrates/Phlebotomus_papatasi/assembly/Phlebotomus_papatasi-2.0/ASSEMBLY;
http://genome.wustl.edu/pub/organism/Invertebrates/Phlebotomus_papatasi/assembly/Phlebotomus_papatasi-2.0/README.

definições. Antevendo, assim, a necessidade de execução de alguma atividade de desenvolvimento de software, ainda que em nível básico e de maneira autônoma[21] (Santos, 2010), procura adotar, dentro do possível e quando aplicáveis, o modelo *Ágil*[22] de desenvolvimento, mais precisamente por meio de alguns métodos e práticas presentes na metodologia da *eXtreme Programming*[23] — *XP*, ou *Programação extrema*, em português — (Teles, 2006; Hemrajani, 2007; Sommerville, 2007; Pressman, 2011) (Figura 2.31), para guiar minimamente tal atividade[24], sem ousar ser considerado um projeto de engenharia de software.

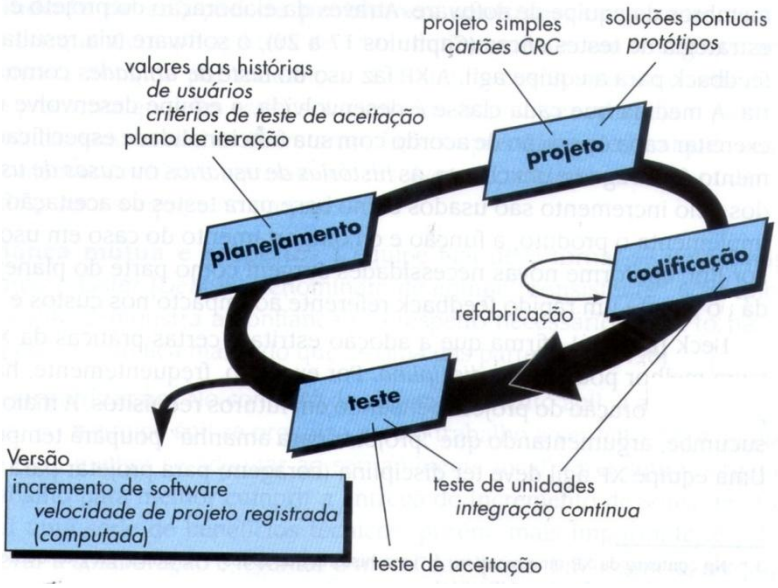


Figura 2.31 - O fluxo de desenvolvimento da *eXtreme Programming* adaptado, na medida do possível, à execução do projeto. A prática de "Programação em Par", por exemplo, não foi considerada, uma vez que não havia um par de programadores no projeto.
Fonte: Modificado de Pressman, 2011, p.88.

2.6.3. Considerações sobre software de uso livre

O termo "*Zero-costed and Licensed Open*" (algo como "de custo zero e licença aberta", em português), associado ao projeto, advém da proposta de empregar, na maioria das funcionalidades implementadas, soluções de terceiros disponibilizadas como de "uso livre" (sem custos para fins acadêmicos) e, em alguns casos, também de "código aberto". Isso significa dizer que o trabalho, tão somente, permeia as definições formais presentes na

[21] http://programmers.stackexchange.com/questions/59713/best-development-methodology-for-one-person.
[22] http://agilemodeling.com.
[23] http://extremeprogramming.org.
[24] No Apêndice D podem ser encontradas as características do trabalho que foram julgadas como adequadas a alguns valores e práticas da metodologia *XP*.

literatura para "software livre"[25], *freeware*[26] e "código aberto"[27], não estando, entretanto, estritamente vinculado a quaisquer umas dessas categorias.

2.6.4. O núcleo do protótipo de montagem de genomas: a plataforma GALAXY

Por estar disponível publicamente[28], ser oferecida sem custos (sob licença para fins acadêmicos), prover código aberto[29], já embutir ferramentas com as mais variadas finalidades para a análise de dados biológicos, permitir a integração e adição de outras ferramentas, além de incorporar um mecanismo que, simultaneamente, gerencia o acesso e contas de usuários — o que garante privacidade e segurança —, armazenando seus dados e configurações de parâmetros (sob a forma de "históricos" dos experimentos conduzidos — garantindo, assim, reprodutibilidade e proveniência dos dados) e facilitando a colaboração entre eles (através da possibilidade de "publicação" e compartilhamento dos fluxos de trabalho criados e utilizados), a plataforma GALAXY (Giardine et al., 2005; Blankenberg et al., 2007; Miller et al., 2007; Taylor et al., 2007; Lazarus et al., 2008; Koskovsky et al., 2009; Blankenberg et al., 2010a; Goecks et al., 2010; Schatz, 2010; Afgan et al., 2011; Blankenberg et al., 2011; Hillman-Jackson et al., 2012) foi escolhida para compor o núcleo do sistema. Além das propriedades citadas, ela apresenta a grande vantagem de ter sido concebida especificamente para a comunidade de profissionais e pesquisadores das áreas de ciências da vida e de bioinformática, com o intuito de proporcionar um ambiente "amigável" para as suas respectivas realidades. É, também, um "sistema de fluxos de trabalho (*pipeline*) completamente personalizável" (Blankenberg et al., 2010b). De fato, abordagens semelhantes, baseadas em tal plataforma, têm sido disponibilizadas para diversas outras finalidades, sejam como serviços públicos oferecidos à comunidade científica ou como personalizações para uso

[25] "Software livre" é um conceito (ou filosofia) que prevê que todo software desta categoria será distribuído com seu código-fonte, podendo ser alterado e, até mesmo, redistribuído depois de alterado. Se refere, portanto, à liberdade de poder executar, modificar e redistribuir versões, originais ou modificadas, de um determinado programa, o qual, não necessariamente, precisa ser gratuito (Mota Filho, 2007). A Free Software Foundation (FSF, http://www.fsf.org), fundada por Richard Stallman, determina as condições que devem ser observadas no sentido de classificar um software como "livre". Boas observações a respeito do tema também podem ser obtidas no artigo de Barr (2001).

[26] *freeware* é o termo mais apropriado para indicar que um software é gratuito (Mota Filho, 2007).

[27] O conceito "open source" é mais voltado para uma metodologia de desenvolvimento, não tornando o software necessariamente "livre". Apesar de manter o princípio da redistribuição livre, o conceito não garante absolutamente nada sobre a forma de distribuição, modificação e comercialização do software (Gugik, 2009). A Open Source Initiative (OSI, http://opensource.org) provê definições a respeito de software de "código aberto".

[28] http://galaxyproject.org (URL do projeto como um todo) ou http://usegalaxy.org (servidor público disponível para a realização de algumas análises).

[29] http://getgalaxy.org (URL de onde é possível fazer o *download* do pacote Galaxy e obter instruções de instalação e configuração de uma instância local da plataforma Galaxy). Respeitando-se algumas poucas restrições, devido à licença livre para fins acadêmicos, é permitido aos desenvolvedores modificar e redistribuir a plataforma e suas respectivas aplicações.

interno por determinadas instituições de pesquisa[30]. O próprio projeto GALAXY sugere a criação de instâncias locais, quando se deseja: (1) desenvolver algo além do que está disponível na versão pública do serviço; (2) adicionar outras ferramentas às normalmente existentes; (3) atrelar novas fontes de dados e (4) rodar um servidor de produção local específico para a sua instituição, por motivo de privacidade de dados ou por ser vislumbrada a necessidade de tratamento de grandes volumes de dados ou de demandas de processamento para os quais o servidor público não seja a melhor opção.

Considerando, a princípio, os itens (1) e (2) e vislumbrando os demais itens para o futuro, foi idealizada a instalação de uma instância local básica sobre a qual se pudesse, gradativamente, agregar novas funcionalidades. Tal agregação tomaria forma por meio da implementação do módulo *NGS: LASZLO's Sandbox* (ou, simplesmente, **LASZLO - *L**inkage of **A**ssembly **S**cripts **Z**ero-costed and with **L**icense **O**pened*), o qual funcionaria como um "organizador" de algumas ferramentas NGS originalmente não disponíveis na plataforma GALAXY básica, podendo ser posteriormente alterado e refinado para incorporar outras ferramentas e funcionalidades ou para incrementar os recursos inicialmente disponibilizados, com possibilidade de expansão virtualmente ilimitada. A arquitetura da plataforma GALAXY, representando o núcleo do sistema no qual o módulo *NGS: LASZLO's Sandbox* (algo como "NGS: Caixa de areia do LASZLO", em português), objeto deste trabalho, estaria baseado, é mostrada na Figura 2.32.

[30] http://wiki.g2.bx.psu.edu/PublicGalaxyServers.

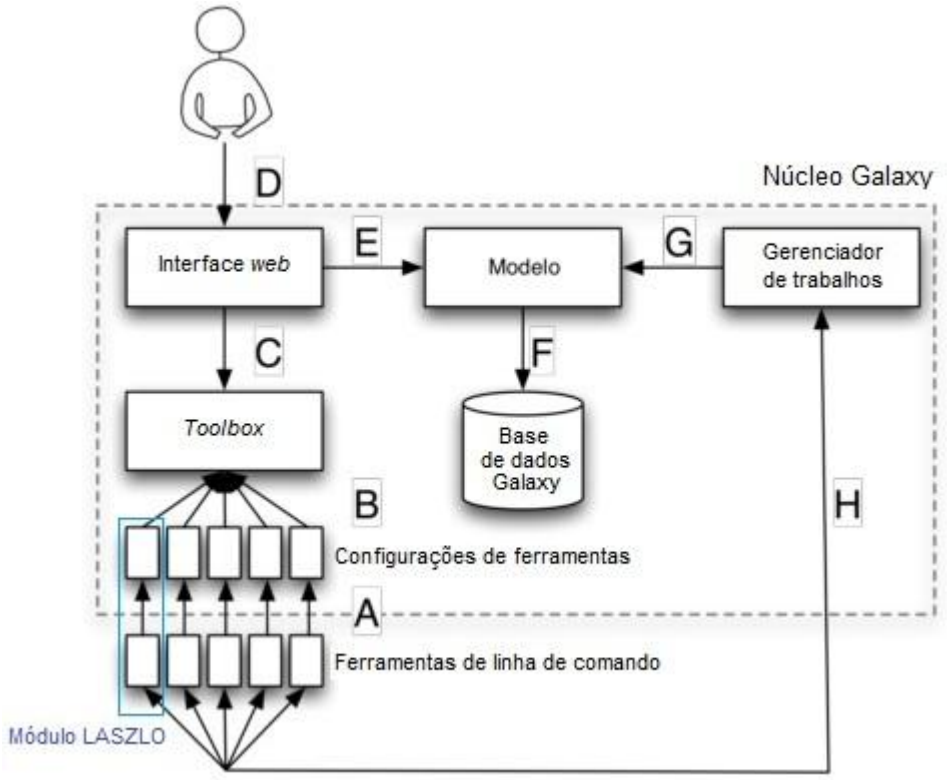


Figura 2.32 - Visão geral dos componentes do *framework* GALAXY e posição ilustrativa do módulo *NGS: LASZLO's Sandbox*. (A) Ferramentas de linha de comando são descritas em arquivos de configuração específicos de ferramentas e (B) validadas pelo componente *Toolbox* ("caixa de ferramentas"). (C) A interface *Web* provê o acesso às ferramentas. (D) Conforme os trabalhos são emitidos pelo usuário, controladores *Web* interagem com o componente Modelo (E) para armazenar as informações relevantes na base de dados da plataforma GALAXY (F). Em seguida, o componente Gerenciador de trabalhos acessa a base de dados (G), prepara a execução dos trabalhos e os envia aos recursos apropriados (H). Quando os trabalhos são finalizados, o Gerenciador os trata e importa para o *framework* novamente.
Fonte: Modificado de Afgan et al., 2011, p.13.

O Anexo C traz uma tabela comparativa, baseada no trabalho de Dinov et al. (2011), na qual o ambiente de *workflow* GALAXY aparece com algumas de suas características listadas em relação a outros ambientes similares. Como outro fator importante para a escolha desse ambiente como núcleo do protótipo, além das vantagens já descritas nesta subseção, pode ser mencionado o modelo "cliente-servidor" de funcionamento da plataforma — modelo este de uso comum, atualmente, entre a maioria dos usuários de computadores. Além disso, conta com um bom nível de suporte técnico[31], característica comumente encontrada em soluções do tipo *open source*. De fato, suas listas de correspondência[32], destinadas às comunidades de desenvolvedores e usuários, são bastante ativas, representando uma boa fonte de consulta.

[31] http://wiki.g2.bx.psu.edu/Support.
[32] http://wiki.g2.bx.psu.edu/Mailing%20Lists.

2.6.4.1 O conceito de wrappers e seu uso na plataforma

A plataforma GALAXY pode ser considerada, antes de mais nada, como um *framework* de integração de ferramentas de software. Ela abstrai as ferramentas individuais por meio de uma interface *Web* fácil de ser utilizada, proporcionando, assim, capacidade de análise ao usuário final, bem como a reprodutibilidade de seus experimentos e proveniência dos dados, sem exigir dele uma experiência mais avançada em informática. Por outro lado, sendo um *framework*, a solução também permite que desenvolvedores de outras ferramentas de software possam agregá-las ao conjunto principal, "personalizando" a plataforma para diversos fins. Em linhas gerais, qualquer programa que seja executado por meio de linha de comando (CLI) pode ser integrado à solução, bastando, para tal, que detalhes a respeito de como o referido programa funciona (por exemplo, seus parâmetros, os tipos de dados de entrada e de saída, etc.) sejam fornecidos sob a forma de uma interface abstrata. De fato, apesar da plataforma ser implementada em linguagem Python, ela é "agnóstica" em relação à maneira como uma determinada ferramenta é implementada. Desde que exista uma forma de acionar essa ferramenta via linha de comando, ela poderá ser escrita em qualquer linguagem do tipo interpretada (por exemplo, Perl, Python, R) ou compilada (por exemplo, C ou Fortran) (Afgan et al., 2011). Para isso, o *framework* se utiliza do conceito de *wrappers*, os quais são módulos de software que provêm uma interface para os programas executáveis (Hoon et al., 2003). Uma vez implementada a interface de uma determinada ferramenta, ela pode fazer referência a um *script* escrito em Perl, Bash, Python ou, ainda, a uma combinação desses *scripts*, por exemplo, para prover a funcionalidade idealizada.

3. Objetivos

3.1. Objetivo geral

Elaborar um protótipo básico de sistema de serviço de montagem de genomas a partir de dados de NGS.

3.2. Objetivos específicos

1. Construir fluxos de trabalho básicos, usando ferramentas existentes para tratamento de dados de NGS, para as estratégias de montagem: (i) usando genoma de referência; (ii) *de novo*; e que sejam compatíveis com os dados de sequenciamento das plataformas Illumina, ABI SOLiD™ e 454.

2. Acomodar tais fluxos de trabalho em uma única solução (protótipo) que contemple interface *Web*, sugestões de parametrização para o usuário final, armazenamento dos dados dos experimentos e que possa, eventualmente, servir como "módulo de montagem de genomas" para outros sistemas.

3. Testar a ferramenta com dados de sequenciamento das plataformas Illumina, ABI SOLiD™ e 454.

4. Testar a alimentação do banco de dados do sistema STINGRAY com os resultados de montagens produzidas pelo protótipo, os quais, posteriormente, poderão ser usados como fonte para anotação e/ou realização de outras análises permitidas por esse sistema.

5. Produzir esboços do genoma do organismo *Leishmania amazonensis*, os quais possam eventualmente servir como base ou auxílio em atividades de anotação ou análise detalhada de seu conteúdo.

.

4. Material e métodos

4.1. Dados de sequenciamento utilizados nos fluxos de trabalho básicos dos experimentos de montagem

4.1.1. Dados da plataforma Solexa/Illumina para montagem com auxílio de genoma de referência

Foram usados dados de sequenciamento do tipo NGS (de Solexa/Illumina), do organismo *Leishmania amazonensis*, gentilmente cedidos pelo Laboratório de Biologia Computacional e Sistemas, do Instituto Oswaldo Cruz, e pelo Dr. Jeremy Mottram, da Universidade de Glasgow. Tais dados se referem à cepa MHOM/BR/1973/M2269, isolada de pacientes com lesões simples e sequenciada em equipamento Solexa/Illumina, disponíveis sob o formato FASTQ[33] (típico dessa plataforma de sequenciamento), contendo leituras pareadas de 51 pb cada, separadas em dois arquivos designados como "lane_2_1.txt" e "lane_2_2.txt" (Figura 4.1). Informações adicionais obtidas sobre os dados denotavam uma cobertura de sequenciamento de 70 vezes e tamanho de inserto de 200 pb (desvio de 40 pb).

[33] No Apêndice C deste trabalho podem ser encontradas mais informações a respeito do formato FASTQ. Explicações mais detalhadas sobre o formato também podem ser encontradas nos endereços *Web* http://en.wikipedia.org/wiki/FASTQ_format e http://maq.sourceforge.net/fastq.shtml.

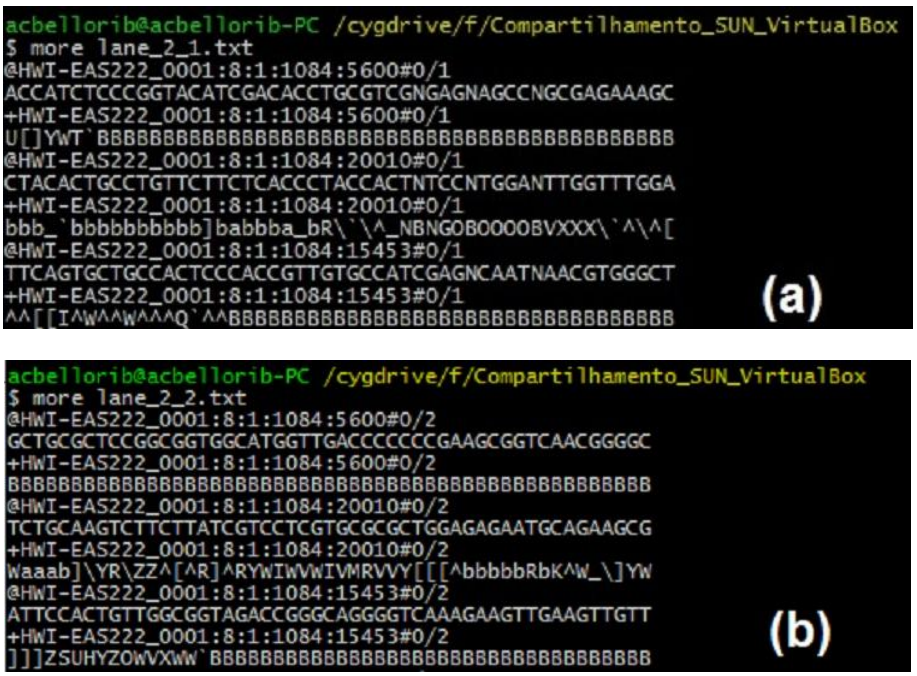


Figura 4.1 - Exemplo de leituras pareadas, de Solexa/Illumina no formato FASTQ, utilizadas neste trabalho. (a) Leituras pareadas #0/1 do arquivo "lane_2_1.txt"; (b) Leituras pareadas #0/2 do arquivo "lane_2_2.txt".

4.1.1.1 Genoma de referência para Leishmania amazonensis

O genoma de *Leishmania mexicana* em formato FASTA, nas versões "LmexicanaGenomic_TriTrypDB-3.0.fasta" e "LmexicanaGenomic_TriTrypDB-4.0.fasta"[34], foi o utilizado nos experimentos de montagem de dados de Solexa/Illumina auxiliados por genoma de referência. A classificação taxonômica do organismo, relatada no trabalho de Bañuls et al. (2007 apud WHO, 1990) e ilustrada na Figura 4.2, foi usada como embasamento para a escolha desses arquivos.

[34] Ambas as versões obtidas em http://tritrypdb.org/tritrypdb.

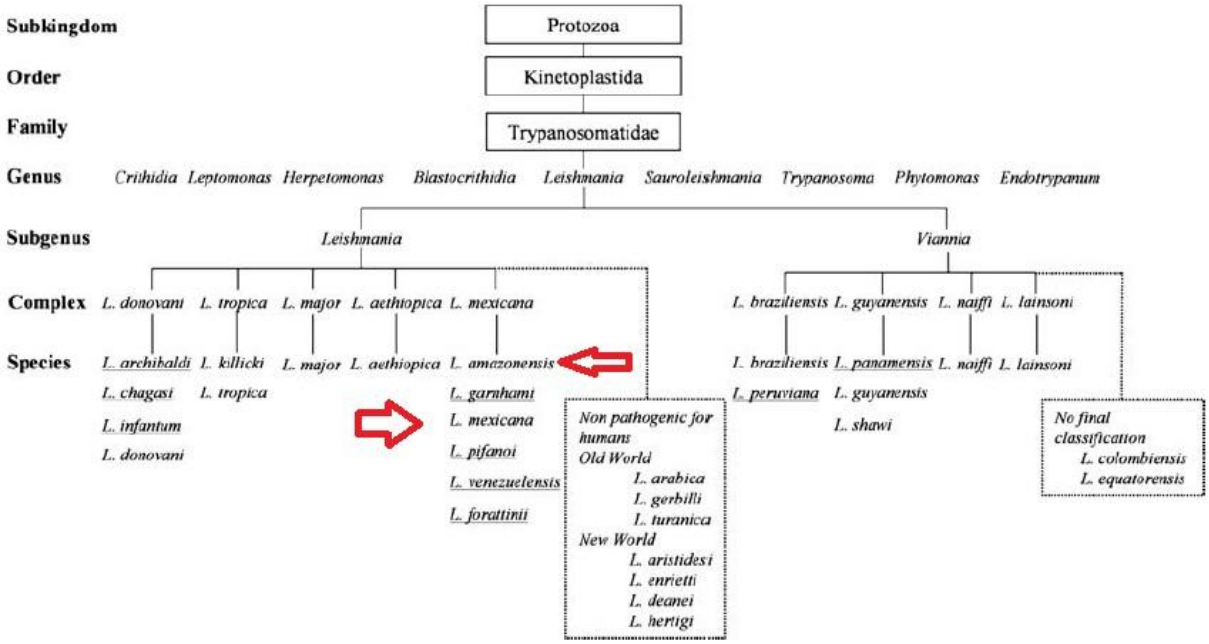


Figura 4.2 - Detalhe da classificação taxonômica de *Leishmania amazonensis*, a qual orientou a escolha do genoma de referência (*Leishmania mexicana*).
Fonte: Modificado de Bañuls et al. (2007 apud WHO, 1990), p.7.

4.1.2. Dados da plataforma Solexa/Illumina para montagem *de novo*

Neste cenário, foram usados os mesmos dados de sequenciamento descritos acima, sem a utilização de genoma de referência para auxiliar a montagem.

4.1.3. Dados da plataforma ABI SOLiD™ para montagem *de novo*

Foram empregados os mesmos subconjuntos de dados usados pela Applied Biosystems (2010), quando do teste, ajuste fino e divulgação de sua ferramenta *de novo accessory tools 2.0*. Conforme documentado e disponibilizado pela empresa[35], tais subconjuntos eram compostos pelos produtos resultantes da corrida de sequenciamento, em instrumento ABI SOLiD™, de uma biblioteca *mate-pair* (2 x 50 pb) da cepa de *E. coli* DH10B, com cobertura de 600 vezes e tamanho de inserto de 1,2 kpb (desvio de 300 pb), sob a forma dos arquivos de leituras diretas "ecoli_600x_F3.csfasta" e seu correspondente com os valores de qualidade "ecoli_600x_F3.qual", bem como dos arquivos de leituras reversas "ecoli_600x_R3.csfasta" e seu correspondente "ecoli_600x_R3.qual". A Figura 4.3 exibe as características dos arquivos de leituras produzidos no sistema ABI SOLiD™.

[35] Dados obtidos em http://www.solidsoftwaretools.com/gf/project/ecoli2x50/, em dezembro de 2010.

f3.csfasta

```
>469_29_17_F3
T203303103012313303232311310133211223331321213103 20
>469_29_1434_F3
T132113.212313112122223110213122111211222 0..2123221
```

r3.csfasta

```
>469_29_17_R3
G203231123031.212031013120130300012233123311200320
>469_29_1434_R3
G100233132323220.002321100102122111211222 0..2123221
```

f3.qual

```
>469_29_17_F3
30 31 24 22 25 17 20 21 17 29 22 30 15 2 31 15 21 4 3 28 10 24 26 18 22
17 24 4 8 12 10 14 5 21 15 5 23 12 13 7 6 15 14 17 6 18 21 12 11 13
>469_29_1434_F3
31 24 29 31 30 27 2 31 30 27 31 27 22 30 28 29 32 21 31 31 23 22 31 30 23
31 16 17 22 13 8 21 31 17 7 31 8 29 23 13 8 22 2 1 14 8 27 20 10 17
```

Figura 4.3 - Exemplo de dados pareados em formato .csfasta e arquivo com valores de qualidade em formato .qual gerados pela plataforma SOLiD™.
Fonte: Modificado de Applied Biosystems, 2010.

4.1.4. Dados da plataforma 454 para montagem *de novo*

Foram utilizados os dados de sequenciamento do organismo *Phlebotomus papatasi* (Díptera:Psychodidae) disponíveis publicamente no banco SRA[36], sob o número de acesso SRR066482, experimento SRX027131, estudo SRP003608, resultantes de corrida em sequenciador 454 GS FLX Titanium. Os dados do referido organismo, na forma do arquivo SRR066482, já haviam sido utilizados no "teste-piloto" do sistema STINGRAY (a ser detalhado mais adiante) e, por esse motivo, foram reutilizados neste trabalho apenas para fins de teste de montagem. A captura desses dados foi realizada conforme as instruções contidas no *SRA Handbook* (NCBI, 2010). A mesma documentação orienta quanto à utilização de uma ferramenta própria do NCBI — *SRA Toolkit* — para a devida conversão do formato padrão SRA (do banco) para o formato específico de uma dada tecnologia de sequenciamento. Para a tecnologia 454, o formato padrão é o SFF (*Standard Flowgram Format*), o qual foi desenvolvido, em colaboração entre a empresa 454 e o NCBI, para padronizar e submeter apropriadamente os pirogramas (*flowgrams*) de 454 para armazenamento no *Trace Archive* do próprio NCBI, contendo, basicamente, um cabeçalho com informações sobre a corrida de sequenciamento e os valores referentes às intensidades dos sinais de cada base.

[36] ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR066/SRR066482/SRR066482.sra.

4.2. Recursos de informática e bioinformática

4.2.1. Metodologia de desenvolvimento de software

Conforme menção feita na subseção 2.6.2, métodos e práticas de desenvolvimento da *eXtreme Programming* foram usados como referência no planejamento das tarefas de adaptação e desenvolvimento do protótipo.

4.2.2. Núcleo do protótipo

Em virtude das vantagens citadas na subseção 2.6.4, foi usado como núcleo de funcionamento e desenvolvimento do protótipo, uma instância local do ambiente de *workflow* gráfico e, também, *framework* de integração de ferramentas de software GALAXY. A *build*[37] (ou "última revisão") utilizada foi a 6799:40f1816d6857, de 07 de março de 2012 (13:35:34 GMT -5).

Para a instalação e configuração da instância local GALAXY, foram seguidas as instruções contidas no URL http://getgalaxy.org e na seção 3.1 (*The installation process*) do trabalho de Afgan et al. (2011). A seção 3.2 (*Adding Tools to Galaxy*) desse mesmo trabalho serviu como referência para a adição de ferramentas à plataforma, quando necessário, no sentido de personalizá-la, por exemplo, com a adição do módulo *NGS: LASZLO's Sandbox*. A combinação da referida instância local com tal módulo perfazem o que, neste documento, se convencionou denominar "instância local *LASZLO @ GALAXY*" ou "protótipo *LASZLO @ GALAXY*".

4.2.3. Linguagens de programação, *script* e marcação[38]

Nesta versão do protótipo, foram utilizadas as linguagens de programação Perl e Python, a linguagem Bash *script* do shell Bourne-Again shell (Bash) e as linguagens de marcação XML e HTML.

4.2.4. Sistema operacional

A plataforma GALAXY, a princípio, só é oferecida para os sistemas operacionais UNIX/Linux e Mac OS X. Desta forma, optou-se pela utilização do sistema operacional

[37] A plataforma não possui "versões", tal como o conceito é tipicamente usado para produtos de software comerciais. Em vez disso, ela incorpora o conceito de *builds* ou *changeset revisions* (algo como "revisões de conjuntos de mudanças", em português) as quais, em uma instância local (tal como a deste trabalho), podem ser determinadas com o comando "hg heads", emitido a partir do diretório de instalação. Fonte: http://user.list.galaxyproject.org/How-to-determine-version-of-Galaxy-from-main-page-td4223309.html.

[38] No Apêndice B deste trabalho, podem ser encontradas informações adicionais a respeito das linguagens mencionadas nesta subseção.

Linux Ubuntu, versão 12.04 LTS[39] de 64 bits, na versão básica do protótipo. A instalação foi realizada seguindo-se as instruções e opções de configuração fornecidas com o próprio programa de instalação da referida distribuição Linux.

4.2.5. Hardware utilizado para a elaboração do protótipo

Foi utilizado um servidor do fabricante OMTX, modelo VRILLX 2000, comportando dois processadores AMD Opteron 6212 (2,6 GHz) de oito núcleos, arquitetura de 64 bits, memória RAM DDR3 de 128 GB (1333 MHz) e dois discos SATA II Enterprise de 2 TB cada.

4.2.6. Monitoração básica do hardware durante os experimentos de montagem

A título de uma melhor supervisão quanto ao uso do servidor, foi utilizado o pacote de uso livre MRTG (*Multi Router Traffic Grapher*)[40] para monitorar a utilização de seus recursos, principalmente no que diz respeito ao consumo de memória durante as execuções dos experimentos de montagem. Usando as instruções de configuração publicadas por Andersson (2009) e Danen (2006a, 2006b), o pacote MRTG foi instalado e configurado no equipamento servidor, assim como o serviço SNMP. O *script* básico proposto por Danen (2006b) foi, em seguida, implementado para monitorar o consumo de memória principal e de *swap*[41].

[39] Obtido no URL http://www.ubuntu-br.org/download.

[40] http://oss.oetiker.ch/mrtg/. MRTG é um aplicativo (de uso livre) que cria gráficos a partir de dois valores fornecidos por um *script* ou programa (Shell, Perl, C, etc.). Para isso, podem ser utilizados dados locais ou colhidos pelo protocolo SNMP (*Simple Network Management Protocol*), o qual permite a coleta de informações via rede de computadores, para fins de gerenciamento dos recursos (Mota Filho, 2007).

[41] *Swap* ou memória virtual é uma técnica que consiste em reservar parte da memória secundária (por exemplo, o disco rígido) para que esta funcione como uma extensão da memória RAM. Quando houver necessidade de esvaziamento de parte da RAM, alguns dos processos (programas em execução) existentes que estiverem aguardando para continuar a execução serão transferidos para o disco rígido (área de *swap*). Para que o sistema GNU/Linux possa utilizar o recurso denominado memória virtual, uma partição em disco, conhecida como partição de *swap*, deve ser criada durante a sua instalação (Mota Filho, 2007).

4.2.7. Ferramentas e programas utilizados nos fluxos de trabalho básicos dos experimentos de montagem

Abaixo, são listados os programas e ferramentas inicialmente empregados no projeto para a composição dos fluxos de trabalho.

4.2.7.1 "Teste-piloto" de fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina

No referido fluxo de trabalho, foram usados os seguintes pacotes:

- MAQ (Li H et al., 2008)[42], na versão 0.7.1;
- SAMtools (Li H et al., 2009)[43], na versão 0.1.7a;
- ARTEMIS (Rutherford et al., 2000)[44], na versão 12.0;
- pacote R[45], na versão 2.9.2;
- *scripts* Perl disponíveis publicamente[46]: "Fastq_quality_plot.pl", "shuffleSequences_fastq.pl", "count_fastq_bases.pl" e "fastq_qualitytrim_window.pl".

4.2.7.2 Fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina na instância local LASZLO @ GALAXY

As ferramentas/programas utilizados neste fluxo de trabalho, bem como suas respectivas funções, são apresentados no Quadro 4.1 abaixo. Neste cenário, como exemplos de ferramentas especificamente criadas para fins de personalização da instância local GALAXY, integrantes do módulo *NGS: LASZLO's Sandbox*, podem ser mencionadas a ferramenta conversora de formatos *SAMTOOLS pileup-to-fastq converter* (ou "Conversor SAMTOOLS *pileup*-para-fastq") e a de auxílio ao usuário *Extract Region Tool* ("Ferramenta para extração de região").

[42] http://maq.sourceforge.net/; disponível para *download* em http://sourceforge.net/projects/maq/files/.
[43] http://samtools.sourceforge.net; disponível para *download* em http://sourceforge.net/projects/samtools/files/.
[44] Disponível em http://www.sanger.ac.uk/resources/software/artemis/.
[45] Disponível em http://www.r-project.org.
[46] Disponíveis em http://xyala.cap.ed.ac.uk/GenePool/.

Quadro 4.1 - Ferramentas/programas utilizados no fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina na instância local *LASZLO @ GALAXY*.

(continua)

Ferramenta / Programa	Guia de acesso na instância local	Procedência	Função
Upload File (ou "Transferir arquivo")	*Get Data* (ou "Capturar dados")	Disponível e ativa, por padrão, na instância local.	Importação dos dados de entrada.
FASTQ Groomer (ou "Preparador FASTQ") (Blankenberg et al., 2010b)	*NGS: QC and manipulation* (ou "NGS: controle de qualidade e manipulação"), bloco de ferramentas *ILLUMINA FASTQ*	Disponível e ativa, por padrão, na instância local.	Tratamento do cenário de tripla codificação existente para o formato FASTQ (Cock et al., 2010). Independentemente da codificação do arquivo FASTQ de entrada, gera um arquivo FASTQ Sanger na saída; condição básica para a utilização das demais ferramentas da plataforma GALAXY.
FASTQ Summary Statistics (ou "Estatísticas sumarizadas do arquivo FASTQ ") (Blankenberg et al., 2010b)	*NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ*	Disponível e ativa, por padrão, na instância local.	Avaliação e visualização de um sumário de estatísticas a respeito dos valores de qualidade e distribuição de nucleotídeos.
Bloxplot (ou "Gráfico de caixa") (Blankenberg et al., 2010b)	*Graph/Display Data* (ou "Gráfico/Visualização de dados")	Disponível e ativa, por padrão, na instância local.	Plotagem de gráfico de caixa a partir de arquivo de entrada em formato tabular.
FASTQ joiner (ou "Agregador FASTQ") (Blankenberg et al., 2010b)	*NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ*	Disponível e ativa, por padrão, na instância local.	Transformação de arquivos de dados pareados (com identificadores #0/1 e #0/2, por exemplo) em um único arquivo (com identificador #0), para evitar perda de sincronismo entre as leituras pareadas, quando do eventual momento de filtragem ou poda de bases por razões de baixa qualidade (vide item abaixo).
FASTQ Trimmer (ou "Podador FASTQ") (Blankenberg et al., 2010b)	*NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ*	Disponível e ativa, por padrão, na instância local.	Remoção de bases com baixa qualidade.
FASTQ splitter (ou "Separador FASTQ") (Blankenberg et al., 2010b)	*NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ*	Disponível e ativa, por padrão, na instância local.	Operação inversa à da ferramenta *FASTQ joiner*.
Map with BWA for Illumina (ou "Mapear usando BWA para Illumina") (Li; Durbin, 2009)	*NGS: Mapping* (ou "NGS: Mapeamento")	*Wrapper* disponível, por padrão, na instância local, mas necessária instalação do programa.[47]	Mapeamento de leituras de Illumina contra um genoma de referência.

[47] A página http://wiki.g2.bxpsu.edu/Admin/Tools/Tool%20Dependencies traz informações sobre as dependências de software de algumas ferramentas que são embutidas na plataforma GALAXY. Especialmente no caso de configuração de instância local, muitos pacotes possuem o *wrapper* incluído na plataforma, porém não estão instalados por padrão. Em outras palavras, existe a "interface gráfica" para uma determinada ferramenta, mas ela apresenta erro de execução por não estar devidamente instalada.

Quadro 4.1 - Ferramentas/programas utilizados no fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina na instância local *LASZLO @ GALAXY*.
(conclusão)

Ferramenta / Programa	Guia de acesso na instância local	Procedência	Função
Filter SAM (ou "Filtrar SAM") (Li H et al., 2009)	*NGS: SAM Tools*	*Wrapper* disponível, por padrão, na instância local, mas necessária instalação do programa. (*)	Aplicação de "filtros" sobre arquivo em formato SAM, para a captura de informações específicas.
SAM-to-BAM (ou "SAM para BAM") (Li H et al., 2009)	*NGS: SAM Tools*	*Wrapper* disponível, por padrão, na instância local, mas necessária instalação do programa. (*)	Conversão de arquivo em formato SAM para o formato BAM.
Generate pileup (ou "Gerar *pileup*") (Li H et al., 2009)	*NGS: SAM Tools*	*Wrapper* disponível, por padrão, na instância local, mas necessária instalação do programa. (*)	Geração de arquivo em formato *pileup*, para verificação das posições de mapeamento das leituras no genoma de referência.
SAMTOOLS pileup-to-fastq converter	*NGS: LASZLO's Sandbox*, bloco de ferramentas *MORE SAMTOOLS TOYS* (ou "Mais brinquedos SAMTOOLS")	*Wrapper* especificamente criado para fins de personalização da instância local.	Conversão de arquivo em formato *pileup* para FASTQ.
FASTQ to FASTA (ou "FASTQ para FASTA") (Blankenberg et al., 2010b)	*Convert formats* (ou "Converter formatos")	Disponível e ativa, por padrão, na instância local.	Conversão de arquivo em formato FASTQ para FASTA.
SAM Tools flagstat (Li H et al., 2009)	*NGS: SAM Tools*	Disponível e ativa, por padrão, na instância local.	Levantamento de resultados estatísticos a partir de arquivo BAM.
NCBI BLAST+ blastn (Zhang et al., 2000)	*NCBI BLAST+*	Disponível e ativa, por padrão, na instância local.	Comparação de sequências de nucleotídeos entre uma dada sequência fornecida pelo usuário e os dados de montagem.
Extract region tool	*NGS: LASZLO's Sandbox*, bloco de ferramentas *OTHER TOYS* (ou "Outros brinquedos")	*Wrapper* especificamente criado para fins de personalização da instância local.	Captura de região de interesse nos dados de montagem.

(*) A mesma nota de rodapé associada à ferramenta *Map with BWA for Illumina*, neste quadro listada, também se aplica.

4.2.7.3 "Teste-piloto" com ferramentas para montagem de novo usando o sistema STINGRAY

Um "teste-piloto" de integração de ferramentas de montagem *de novo* ao sistema STINGRAY (Wagner et al., 2007) foi realizado durante o projeto e contemplou os seguintes programas:

- MIRA[48], na versão 3.2.1;
- ABI SOLiD™ *de novo accessory tools 2.0* (Applied Biosystems, 2010) já incluindo,

[48] http://mira-assembler.sourceforge.net/.

de maneira integrada, um diretório especialmente preparado para a instalação do pacote Velvet (Zerbino; Birney, 2008)[49], na versão 0.7.55;

- Sistema STINGRAY (http://stingray.biowebdb.org), na versão 1.0 beta.

4.2.7.4 Fluxo de trabalho para montagem de novo a partir de dados de Solexa/Illumina na instância local LASZLO @ GALAXY

A sequência de ferramentas já descritas, *Upload File*, *FASTQ Groomer*, *FASTQ Summary Statistics*, *Boxplot*, *FASTQ joiner*, *FASTQ Trimmer* e *FASTQ splitter*, foi usada para compor a parte inicial deste fluxo de trabalho. O Quadro 4.2, a seguir, lista as ferramentas/programas complementares utilizados. Neste cenário, como exemplo de ferramenta especificamente criada para fins de personalização da instância local GALAXY, também integrante do módulo *NGS: LASZLO's Sandbox*, pode ser citada a *Velvet shuffling tool* (ou "Ferramenta de intercalação para Velvet"), responsável pela preparação apropriada dos arquivos para uso pelo programa Velvet.

Quadro 4.2 - Ferramentas/programas complementares utilizados no fluxo de trabalho para montagem *de novo* a partir de dados de Solexa/Illumina na instância local *LASZLO @ GALAXY*.

Ferramenta / Programa	Guia de acesso na instância local	Procedência	Função
Velvet shuffling tool	*NGS: LASZLO's Sandbox*, bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS* (ou "NGS: Brinquedos para montagem *de novo*")	*Wrapper* especificamente criado para fins de personalização da instância local.	Preparação dos arquivos pareados FASTQ para uso pelo programa Velvet.
velveth	*NGS: LASZLO's Sandbox*, bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*[50]	*Wrapper* disponível, por padrão, na instância local, mas necessária instalação do programa. (*)	Preparação dos dados para uso pelo programa *velvetg*, baseada em um valor de *k*-mer escolhido previamente.
velvetg	*NGS: LASZLO's Sandbox*, bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*	*Wrapper* disponível, por padrão, na instância local, mas necessária instalação do programa. (*)	Construção do grafo de Bruijn e realização da montagem *de novo*.
assemblystats	*NGS: LASZLO's Sandbox*, bloco de ferramentas *OTHER TOYS*	*Wrapper* disponível na comunidade colaborativa de desenvolvimento GALAXY.[51]	Captura, avaliação e visualização de informações estatísticas sobre a montagem realizada.

(*) A mesma nota de rodapé associada à ferramenta *Map with BWA for Illumina*, listada no Quadro 4.1, também se aplica.

[49] http://www.ebi.ac.uk/~zerbino/velvet/.
[50] Apesar dos *wrappers* dos programas *velveth* e *velvetg* fazerem parte, por padrão, do pacote da instância local, suas guias de acesso foram movidas, neste trabalho, para o bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*, parte integrante do módulo *NGS: LASZLO's Sandbox*, para fins de melhor organização.
[51] http://toolshed.g2.bx.psu.edu/.

4.2.7.5 Fluxo de trabalho para montagem de novo a partir de dados de ABI SOLiD™ na instância local LASZLO @ GALAXY

Além da ferramenta *Upload File*, típica para a carga de arquivos na plataforma, a seguinte lista (Quadro 4.3) foi utilizada no fluxo de trabalho aqui sugerido. As ferramentas *SOLiD(TM) denovo tool for FRAGMENT library* (algo como "Ferramenta *SOLiD™ denovo* para biblioteca de fragmentos", em português), *SOLiD(TM) denovo tool for PAIRED-END library* (ou "Ferramenta *SOLiD™ denovo* para biblioteca *PAIRED-END*") e *SOLiD(TM) denovo tool for MATE-PAIRED library* (ou "Ferramenta *SOLiD™ denovo* para biblioteca *MATE-PAIRED*"), responsáveis por implementar as respectivas funcionalidades presentes no pacote de linha de comando *SOLiD™ accessory tools 2.0*, são, também, exemplos das que foram especificamente criadas para fins de personalização da instância local GALAXY, como parte do módulo *NGS: LASZLO's Sandbox*.

Quadro 4.3 - Ferramentas/programas complementares utilizados no fluxo de trabalho para montagem *de novo* a partir de dados de ABI SOLiD™ na instância local *LASZLO @ GALAXY*.

Ferramenta / Programa	Guia de acesso na instância local	Procedência	Função
Compute quality statistics (ou "Computar estatísticas de qualidade")	*NGS: QC and manipulation*, bloco de ferramentas *AB-SOLID DATA* (ou "Dados AB-SOLID")	Disponível e ativa, por padrão, na instância local.	Avaliação das estatísticas de qualidade das leituras.
Draw quality score boxplot (ou "Plotar gráfico de caixa dos valores de qualidade")	*NGS: QC and manipulation*, bloco de ferramentas *AB-SOLID DATA*	Disponível e ativa, por padrão, na instância local.	Plotagem de gráfico de caixa para as estatísticas de qualidade das leituras.
SOLiD(TM) denovo tool for FRAGMENT library	*NGS: LASZLO's Sandbox*, bloco de ferramentas *SOLID DE NOVO ACCESSORY 2.0 TOYS* (ou "Brinquedos SOLID DE NOVO ACCESSORY 2.0")	*Wrapper* especificamente criado para fins de personalização da instância local.	Execução do *pipeline* SOLiD™ *accessory tools 2.0* especificamente para o caso de bibliotecas de fragmentos únicos.
SOLiD(TM) denovo tool for PAIRED-END library	*NGS: LASZLO's Sandbox*, bloco de ferramentas *SOLID DE NOVO ACCESSORY 2.0 TOYS*	*Wrapper* especificamente criado para fins de personalização da instância local.	Execução do *pipeline* SOLiD™ *accessory tools 2.0* especificamente para o caso de bibliotecas *PAIRED-END*.
SOLiD(TM) denovo tool for MATE-PAIRED library	*NGS: LASZLO's Sandbox*, bloco de ferramentas *SOLID DE NOVO ACCESSORY 2.0 TOYS*	*Wrapper* especificamente criado para fins de personalização da instância local.	Execução do *pipeline* SOLiD™ *accessory tools 2.0* especificamente para o caso de bibliotecas *MATE-PAIRED*.
assemblystats	*NGS: LASZLO's Sandbox*, bloco de ferramentas *OTHER TOYS*	*Wrapper* disponível na comunidade colaborativa de desenvolvimento GALAXY.	Captura, avaliação e visualização de informações estatísticas sobre a montagem realizada.

4.2.7.6 Fluxo de trabalho para montagem de novo a partir de dados de 454 na instância local LASZLO @ GALAXY

Além da ferramenta *Upload File*, a seguinte lista (Quadro 4.4) foi utilizada no fluxo de trabalho aqui sugerido.

Quadro 4.4 - Ferramentas/programas complementares utilizados no fluxo de trabalho para montagem *de novo* a partir de dados de 454 na instância local *LASZLO @ GALAXY*.

Ferramenta / Programa	Guia de acesso na instância local	Procedência	Função
SFF converter (ou "Conversor SFF")	*Convert Formats*	Disponível e ativa, por padrão, na instância local.	Conversão do formato SFF de 454 para os formatos FASTA, QUAL e XML ou, se desejado pelo usuário, para os formatos FASTQ e XML.
Build base quality distribution (ou "Compor distribuição de qualidades das bases")	*NGS: QC and manipulation*, bloco de ferramentas *ROCHE-454 DATA* (ou "Dados ROCHE-454")	Disponível e ativa, por padrão, na instância local.	Avaliação de qualidade das leituras de 454.
Assemble with MIRA (ou "Montar com MIRA")	*NGS: LASZLO's Sandbox*, bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*	*Wrapper* disponível na comunidade colaborativa de desenvolvimento GALAXY.	Montagem *de novo* de leituras provenientes das tecnologias Sanger, 454, Illumina e Ion Torrent ou montagem *de novo* híbridas, combinando diferentes tecnologias.
assemblystats	*NGS: LASZLO's Sandbox*, bloco de ferramentas *OTHER TOYS*	*Wrapper* disponível na comunidade colaborativa de desenvolvimento GALAXY.	Captura, avaliação e visualização de informações estatísticas sobre a montagem realizada.

5. Resultados e discussão

Neste projeto, vislumbrou-se a possibilidade de combinar ferramentas de montagem para dados de NGS em uma única solução que pudesse contribuir, de alguma forma, na abordagem do problema de montagem de fragmentos de DNA, e que, ao mesmo tempo, fosse mais acessível ao usuário final, sendo este o biólogo experimental com alguma proficiência em informática e capaz de lidar com seus próprios dados ou, ainda, aquele com menor experiência em informática, porém com acesso à solução eventualmente disponibilizada por intermédio de um especialista ou laboratório de bioinformática. Neste último cenário, por exemplo, por meio da integração de múltiplas fontes, formatos e ferramentas — priorizando a agilidade na obtenção dos resultados e procurando retirar, do usuário final, o ônus inerente à elaboração de suas próprias soluções e integrações de ferramentas, bem como à respectiva implementação de infra-estrutura adequada — a solução proposta neste trabalho foi idealizada para servir como um mecanismo auxiliar da entidade de suporte técnico em bioinformática (ou da própria instituição de pesquisa), no sentido de liberar seus usuários finais diretamente para as devidas tarefas de manuseio e análise dos típicos grandes volumes de dados de NGS.

Para uma primeira fase de desenvolvimento, coberta neste trabalho, o foco era o de implementar alguns fluxos de trabalho envolvendo abordagens de montagem com auxílio de genoma de referência e do tipo *de novo*, compatíveis com dados de sequenciamento das tecnologias Illumina, ABI SOLiD™ e 454, em um protótipo de serviço de montagem de genomas que contemplasse, basicamente, interface *Web*, sugestões de parametrização para o usuário final e armazenamento dos dados dos experimentos.

Este capítulo, portanto, aborda os resultados obtidos e os discute. São fornecidas, também, explicações sobre alguns aspectos técnicos e correspondentes decisões que serviram para nortear o desenvolvimento do projeto.

5.1. Alguns aspectos técnicos e respectivas decisões de projeto

5.1.1. Tecnologias de sequenciamento NGS abordadas

A Seção 2.3 deste trabalho procurou demonstrar a grande variedade de tecnologias NGS existentes atualmente. Para uma primeira delimitação do escopo quanto às que seriam abordadas nesta etapa do projeto, o já citado panorama de Hadfield e Loman (2009), em conjunto com as informações do levantamento realizado pela J.P.Morgan (Peterson et al., 2010), também anteriormente mencionado, balizaram a decisão de abordar, em um primeiro momento, as tecnologias Illumina, ABI SOLiD™ e 454. Tecnologias como Ion Torrent e PacBio, por exemplo, foram deixadas para uma segunda etapa.

5.1.2. Dados de sequenciamento empregados e seus respectivos formatos

O trabalho, então, previa a utilização de dados de sequenciamento das plataformas Solexa/Illumina, ABI SOLiD™ e 454 em seus formatos originais. Procurando demonstrar a versatilidade da solução, dados de sequenciamento de diferentes organismos, disponíveis nos respectivos formatos de cada uma das tecnologias acima citadas, tal como detalhado no Capítulo 4, foram empregados.

5.1.3. A transformação de instância local original GALAXY em instância local *LASZLO @ GALAXY*

Conforme adiantado na subseção 2.6.4, a agregação de ferramentas de análise de dados NGS tomaria forma por meio da implementação do módulo *NGS: LASZLO's Sandbox*. A adição dessas ferramentas, portanto, seguiu o conceito da implementação de *wrappers* da plataforma GALAXY. Tipicamente, as interfaces gráficas das ferramentas são implementadas em linguagem XML. Além disso, arquivos de configuração da plataforma, tal como o "tool_conf.xml" (Figura 5.1), usado para habilitar (ou desabilitar) o funcionamento das ferramentas, também são escritos nessa linguagem.

```
<!-- Bloco criado para inserção de ferramentas do módulo LASZLO.
#########################################################################################################################-->
<section name="NGS: LASZLO's Sandbox" id="ngs-laszlo">
  <label text="UNIX Toys" id="unix_tools" />
  <tool file="unix_tools/remove_ending.xml" />
  <tool file="unix_tools/find_and_replace.xml" />
  <tool file="unix_tools/word_list_grep.xml" />
  <!--<tool file="unix_tools/awk_tool.xml" />-->
  <!--<tool file="unix_tools/sed_tool.xml" />-->
  <tool file="unix_tools/grep_tool.xml" />
  <tool file="unix_tools/sort_tool.xml" />
  <tool file="unix_tools/uniq_tool.xml" />
  <tool file="unix_tools/cut_tool.xml" />
  <tool file="unix_tools/join_tool.xml" />

  <label text="more SAMTOOLS Toys" id="samtools_laszlo" />
  <tool file="ngs_laszlo/ngs_sam_pileup2fq.xml" />

  <label text="EMBOSS Toys" id="EMBOSSLite" />
  <tool file="emboss_5/emboss_cutseq.xml" />

  <label text="NGS: De novo Assembly Toys" id="ngs_assembly" />
  <label text="Velvet" id="velvet" />
  <tool file="sr_assembly/velvetg.xml" />
  <tool file="sr_assembly/velveth.xml" />
  <tool file="ngs_laszlo/antonio_velvetShufflerFastq.xml" />
  <label text="Mira" id="mira" />
  <tool file="sr_assembly/mira.xml" />

  <label text="SOLiD de novo accessory 2.0 toys" id="solid_denovo" />
```

Figura 5.1 - Porção de código em linguagem XML referente ao arquivo "tool_conf.xml" do protótipo. A *tag* <section>, por exemplo, indica o início do bloco de ferramentas denominado *NGS: LASZLO's Sandbox*. As *tags* do tipo <tool> indicam as localizações, nos caminhos de diretórios, de alguns *wrappers* das ferramentas, também escritos em linguagem XML.

Nesta fase do trabalho, portanto, para a implementação dos *wrappers* das ferramentas que foram criadas ou adaptadas, visando a integração do módulo *NGS: LASZLO's Sandbox* com a instância local da plataforma GALAXY, além das interfaces gráficas escritas em linguagem XML, as demais linguagens citadas no item 4.2.3 foram utilizadas. E, para alterar a página inicial da instância local padrão (Figura 5.2), de maneira a "personalizá-la", foi empregada a linguagem HTML.

Figura 5.2 - Parte da tela inicial do protótipo *LASZLO @ GALAXY*. Gravura modificada (para fins de prototipação somente) de produto comercializado pelo *Smithsonian National Air and Space Museum* (EUA).

5.1.4. Critérios para a escolha de ferramentas e programas de bioinformática utilizados nos fluxos de trabalho básicos dos experimentos de montagem

Sendo várias as tecnologias de sequenciamento NGS, vários os formatos de arquivos e vários os programas de bioinformática disponíveis para a realização das etapas pertencentes ao processo de montagem, consequentemente culminando em possíveis variações nas estratégias de combinação desses programas, ficou clara a necessidade de se realizar algum tipo de triagem para proporcionar um "ponto de partida" para os fluxos de trabalho iniciais do projeto. Apoiando-se, então, na filosofia da prática do *Projeto simples* da metodologia *XP*[52], foram estabelecidos os seguintes critérios (não mutuamente exclusivos) para auxiliar a escolha dos programas e composição dos primeiros fluxos de trabalho:

(1) sugestões de encadeamento de programas de uso livre disponíveis publicamente;

(2) sugestões de encadeamento de programas de uso livre recomendadas pelos fabricantes de sequenciadores;

[52] Mais detalhes a respeito da prática de *Projeto simples* podem ser encontrados no Quadro D.2 do Apêndice D.

(3) sugestões de encadeamento de programas de uso livre já disponíveis na própria plataforma GALAXY[53];

(4) ainda que não embutido por padrão na plataforma GALAXY, existência de programa do tipo *wrapper*, para uma determinada ferramenta de uso livre, na página da comunidade colaborativa de desenvolvedores para o *framework* GALAXY[54];

(5) independentemente da existência de *wrapper*, abrangência e versatilidade de um determinado programa de uso livre para a realização de uma dada tarefa pretendida. Por exemplo, um programa montador capaz de trabalhar com diversas tecnologias de sequenciamento, ou um programa de avaliação de qualidade das *short reads* compatível com diversos formatos de entrada, etc.

5.2. Fluxos de trabalho produzidos

De certa forma, os fluxos de trabalho implementados podem ser considerados como parte dos resultados do projeto, uma vez que atendem ao que foi proposto, inicialmente, como metas para o trabalho. Tais fluxos e demais resultados obtidos com a versão básica do protótipo são detalhados e discutidos nesta seção. No Apêndice E, podem ser encontradas as representações esquemáticas dos fluxos de trabalho elaborados.

5.2.1. Sequência de etapas do "teste-piloto" de fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina

Inicialmente, com o objetivo de verificar a viabilidade de possíveis combinações de ferramentas, foi elaborado um esboço de *pipeline* (encadeamento de programas) usando os pacotes MAQ — programa para a produção de montagens baseadas em alinhamento contra um genoma de referência, especialmente concebido para as leituras curtas geradas pelo sequenciador Solexa/Illumina 1G Genetic Analyzer e o primeiro a ser capaz de levar em consideração, no processo de mapeamento dessas leituras, os valores de qualidade das bases atribuídas (Li H et al., 2008; Paszkiewicz; Studholme, 2012) —, SAMtools — pacote com diversos utilitários para a manipulação de resultados de alinhamentos armazenados no formato SAM; aqui empregado, basicamente, para a conversão de formatos — e ARTEMIS — programa concebido para visualização de genomas e tarefas de anotação; aqui usado para a visualização inicial dos resultados. Além desses programas, para as tarefas específicas de

[53] http://wiki.g2.bx.psu.edu/Learn.
[54] http://toolshed.g2.bx.psu.edu/.

análise e ajuste da qualidade das leituras, foi considerada a utilização do pacote R e de alguns *scripts* escritos em linguagem Perl e disponíveis publicamente — "Fastq_quality_plot.pl", "shuffleSequences_fastq.pl", "count_fastq_bases.pl" e "fastq_qualitytrim_window.pl".

Tal esboço de *pipeline* se baseou no equivalente proposto na própria documentação do programa MAQ (Figura 5.3) e em sugestões recebidas, por meio de comunicação pessoal, dos pesquisadores Thomas Dan Otto e Richard Durbin (ambos do Wellcome Trust Sanger Institute, sendo, este último, um dos próprios autores do já citado software MAQ). Em seguida, a combinação foi testada com os dados de sequenciamento de *Leishmania amazonensis* mencionados anteriormente. O fluxo de trabalho produzido e os resultados obtidos são detalhados mais abaixo.

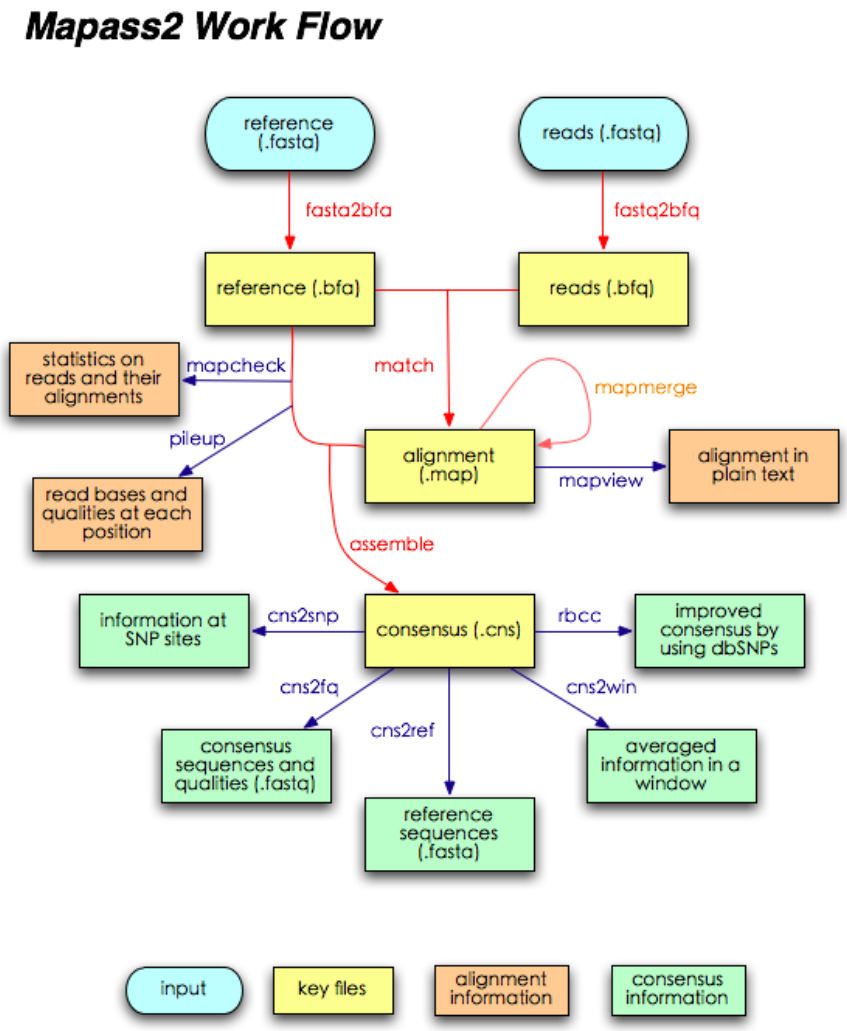


Figura 5.3 - Fluxo de trabalho do pacote MAQ (antigo *Mapass2*).
Fonte: Modificado de http://maq.sourceforge.net/maq-man.shtml.

Basicamente, a sequência inicial de etapas era esta:

(1) O software MAQ usado (versão 0.7.1) trabalha apenas com o formato FASTQ Sanger. Desta forma, dados provenientes dos sequenciadores Solexa/Illumina devem passar por uma conversão para serem corretamente tratados pelo programa MAQ. Para dados provenientes de qualquer versão de sequenciador Solexa ou Illumina com versão inferior à 1.3, o conversor de formatos *sol2sanger*, incluído no próprio pacote MAQ, deve ser usado. No caso de dados originados em sequenciador Illumina com versão igual ou superior à 1.3, um *patch* deve ser aplicado ao programa MAQ[55], para que este último passe a incluir o conversor *ill2sanger*. Outra alternativa seria a utilização de conversores disponíveis em projetos da Open Bioinformatics Foundation (OBF)[56], como BioPerl (Stajich et al., 2002), BioJava (Holland et al., 2008), BioRuby[57], BioPython (Cock et al, 2009) e EMBOSS (Rice et al., 2000). Uma vez que os dados de *Leishmania amazonensis* estavam na versão de codificação Illumina 1.5, o conversor *ill2sanger* do programa MAQ foi, portanto, utilizado;

(2) Após a conversão dos dois arquivos "lane_2_1.txt" e "lane_2_2.txt" para o formato FASTQ Sanger, a qualidade das leituras obtidas no sequenciamento deveria ser analisada. Para isso, foi usado o *script*, em linguagem Perl, "Fastq_quality_plot.pl", o qual gerava arquivos de texto que podiam, em seguida, ser plotados por ferramentas como, por exemplo, o pacote R. Foi observada (Figura 5.4) uma boa qualidade — valores entre 20 e 35 para todos os ciclos de leitura — para os dados do conjunto de leituras pareadas. Assim, optou-se por não realizar nenhuma operação de *trimming* (ou "poda de bases") nesses dados. Se isso fosse necessário, o encadeamento de outros *scripts*, como "shuffleSequences_fastq.pl", "count_fastq_bases.pl" e "fastq_qualitytrim_window.pl", poderia ser utilizado;

[55] http://sourceforge.net/tracker/?func=detail&aid=2824334&group_id=191815&atid=938893.
[56] http://www.open-bio.org/wiki/Main_Page.
[57] http://www.bioruby.org.

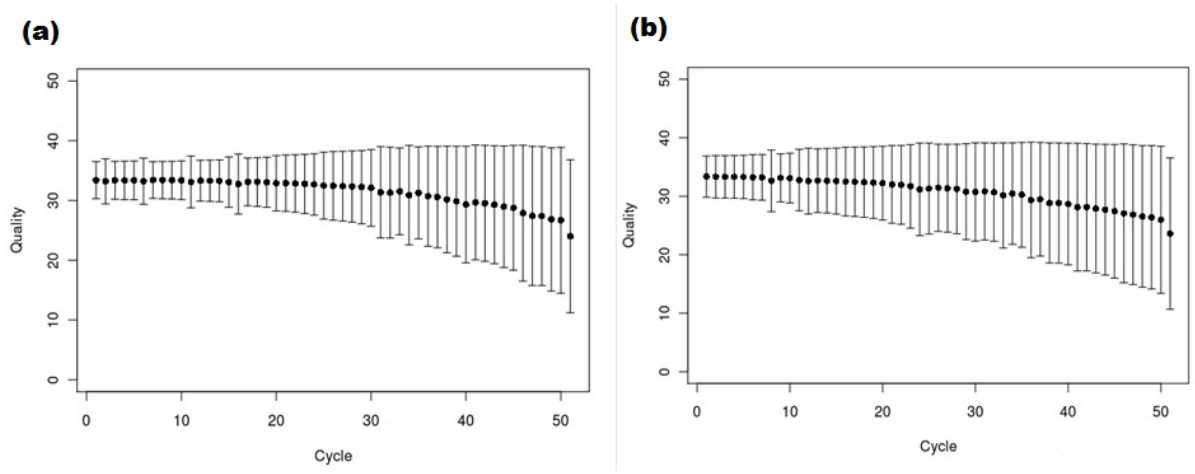


Figura 5.4 - Qualidade das bases do conjunto de dados pareados (a) "lane_2_1.txt" e (b) "lane_2_2.txt".

(3) A seguir, a rotina *easyrun*, do programa MAQ (com a adição do parâmetro `-p`, para indicar leituras pareadas), deveria ser executada. Foram fornecidos, então, como dados de entrada para essa rotina, os dois arquivos FASTQ Sanger, mais o arquivo do genoma de referência em formato FASTA — no caso, "LmexicanaGenomic_TriTrypDB-3.0.fasta". Assim, seria gerado, como um dos arquivos resultantes ao final da rotina (já comprimido em formato binário), o arquivo *all.map* com o resultado do mapeamento das leituras;

(4) Visando tornar esse resultado do alinhamento acessível à análise por outras ferramentas subsequentes (tal como ARTEMIS, por exemplo), o arquivo *all.map* haveria de passar por uma série de etapas de conversão de formatos para algum que fosse comum a essas ferramentas. O formato SAM (*Sequence Alignment/Map*), de tipo texto e delimitado por tabulação, é um que apresenta essa proposta de servir como um padrão comum entre as ferramentas de alinhamento e as posteriores soluções para análise *downstream* (cadeia de etapas subsequentes de análise) dos dados. Ele foi concebido para armazenar grandes alinhamentos de sequências, sendo genérico o suficiente para ser compatível com diferentes programas alinhadores e demais ferramentas. Além disso, procura armazenar o máximo de informações relativas aos alinhamentos, sendo, simultaneamente, o mais compacto possível para minimizar o consumo de memória. O formato SAM é acompanhado por um pacote de utilitários denominado SAMtools, o qual é capaz de realizar várias manipulações em dados de alinhamentos que estejam nesse formato. Desta forma, para a obtenção do arquivo SAM, tendo sido o arquivo .map gerado a partir de versão 0.7.x do programa MAQ (onde "x" indica qualquer algarismo), o conversor *maq2sam-long*, do pacote SAMtools, foi usado[58]. A título de

[58] A página http://bioinf.scri.ac.uk/tablet/assembly-conversion.html traz bons exemplos de conversões de formatos realizadas via linha de comando.

informação, quando o arquivo .map é gerado por um programa MAQ versão 0.6.x, o conversor usado deve ser o *maq2sam-short*;

(5)[59] De posse do arquivo SAM (o qual pode ser lido como um arquivo de texto), o próximo passo seria obter seu respectivo arquivo compactado binário em formato BAM (*Binary Alignment/Map*). Esse arquivo contém as mesmas informações que o arquivo SAM, porém consome menos memória e é mais fácil de ser tratado em operações de busca. Como passo intermediário para sua obtenção, um arquivo de índice (.fai) do genoma de referência deveria ser construído, aplicando-se o comando `samtools faidx` sobre o arquivo "LmexicanaGenomic_TriTrypDB-3.0.fasta". Então, alimentado desse arquivo de índice e do arquivo do alinhamento convertido para formato SAM, o comando `samtools import` geraria, como saída, o resultado do alinhamento no pretendido formato binário BAM;

(6) Para que as leituras do alinhamento pudessem ser facilmente acessadas em operações de busca realizadas por ferramentas de análise subsequentes como, por exemplo, o software ARTEMIS, um respectivo arquivo de índice dessas leituras deveria ser criado. Para tal, duas outras etapas de conversão com SAMtools foram necessárias: `samtools sort` aplicado sobre o arquivo BAM, para gerar o respectivo arquivo *sorted* BAM, e, sobre este último, `samtools index`;

(7) Por fim, alimentando-se o software ARTEMIS com o arquivo *sorted* BAM[60], os resultados puderam ser, então, visualizados (Figura 5.5).

[59] A página http://seqanswers.com/wiki/How-to/RNASeq_analysis traz boas explicações sobre conversões especificamente realizadas com o pacote SAMtools e foi usada como referência para este item (5) e para o item (6) seguinte.

[60] Para que a visualização possa ser corretamente obtida pelo software ARTEMIS, o respectivo arquivo de índice (gerado a partir do arquivo *sorted* BAM) deve ser mantido no mesmo diretório.

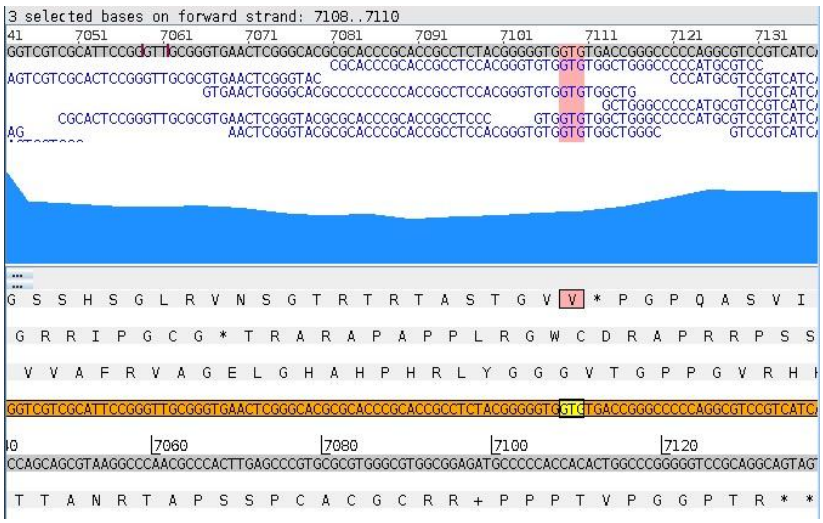


Figura 5.5 - Detalhe das *short reads* sobrepostas (em relação à referência "em cinza") exibidas no software ARTEMIS.

O fluxograma, a seguir (Figura 5.6), ilustra a sequência de passos mencionada mais acima. Para facilitar o entendimento, os números dos passos descritos acima foram usados para rotular as respectivas etapas na figura.

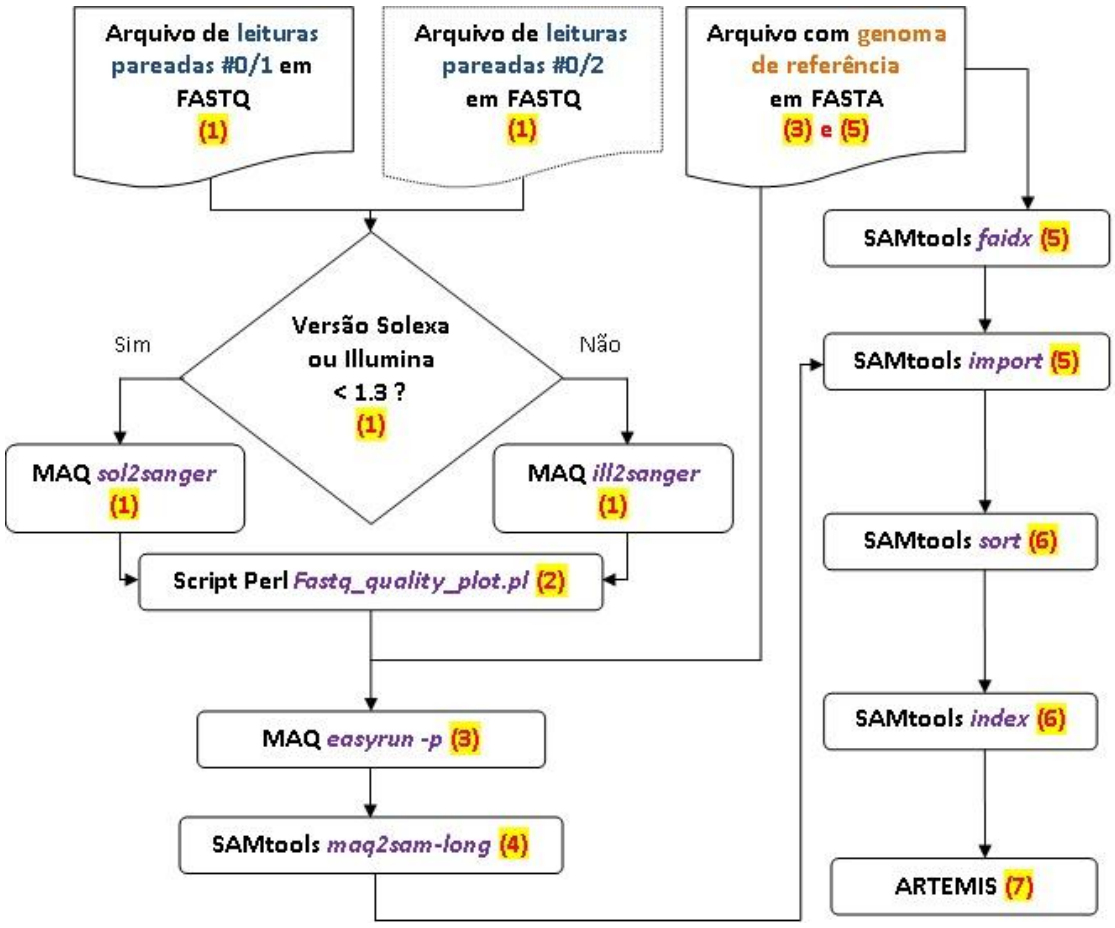


Figura 5.6 - Fluxograma do primeiro esboço de encadeamento de programas para alinhamento do genoma de *Leishmania amazonensis* no programa MAQ e visualização do resultado no programa ARTEMIS.

É importante ressaltar que, no cenário exposto no "teste-piloto", nenhuma das ferramentas estava integrada. Cada pacote de software havia de ser instalado, ter suas dependências de instalação de programas resolvidas e ser corretamente configurado, dentro do ambiente Linux/Unix, para que pudesse rodar a contento. Da mesma forma, a maior parte das ações tinha de ser executada via linha de comando, com os parâmetros corretos tendo, também, de ser escolhidos e fornecidos conforme a orientação da documentação de cada programa. E, cada resultado obtido, apropriadamente repassado para uso pela ferramenta ou comando seguinte.

Para usuários com conhecimentos de informática mais avançados, existem vantagens em se trabalhar via linha de comando, como agilidade, possibilidade de encadear programas diretamente com o uso do comando *pipe* ("|") do Unix e de fazer uso de controles mais finos durante a execução de determinada ferramenta ou solução, etc. Ainda assim, isso não oferece garantia de reprodutibilidade, pois a ordem e os dados utilizados em eventuais experimentos ficarão sujeitos aos critérios de organização de cada usuário específico. Além disso, na prática, tais vantagens mencionadas só representam a verdade para uma gama bem menor de usuários. No que diz respeito àqueles com conhecimentos mais limitados de informática, é necessário um longo período de aprendizado até que estes últimos possam criar uma infra-estrutura razoável que lhes possibilite dar continuidade às suas análises. E, isso, sem levar em consideração os requisitos de hardware, os quais, para determinados tipos de montagens, se tornam inviáveis de serem satisfeitos com máquinas tipicamente encontradas no mercado. Daí, portanto, adveio o interesse em tornar tais dificuldades "transparentes" para esse último tipo de usuário. Consequentemente, a ideia de trabalhar com fluxos de trabalho específicos para NGS já pré-determinados, mas que oferecessem, também, flexibilidade para serem refinados ao longo do tempo e que pudessem ser armazenados e mantidos de alguma forma, pareceu ser uma boa opção. Então, tal como dito anteriormente, pelas vantagens citadas na subseção 2.6.4, a plataforma GALAXY foi vislumbrada como o possível ambiente ideal para o desenvolvimento da ideia, ficando como sugestão inicial para o dimensionamento apropriado dos requisitos de hardware, por exemplo, o número de usuários que deverão ser servidos pela instância local modificada e os requisitos de hardware especificados pelas próprias ferramentas escolhidas para emprego nos diferentes fluxos de trabalho.

5.2.2. A configuração e início do serviço da instância local GALAXY

Conforme menção anterior, foi instalada e configurada uma instância local para possibilitar o uso de suas ferramentas pré-instaladas, bem como do ambiente de

desenvolvimento apropriado para a criação, adaptação e imediata integração de outras. Após a devida instalação e configuração da plataforma, sempre que era desejado iniciar seu serviço, o comando `sh run.sh --reload` era emitido a partir do diretório `/galaxy-dist` da instância local. A Figura 5.7 exibe o resultado do serviço iniciado com sucesso e ativo na porta 8080 do servidor da instância local.

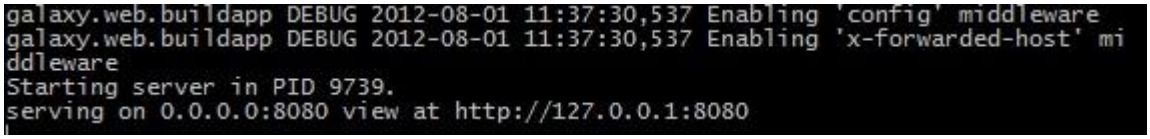


Figura 5.7 - Serviço da instância local GALAXY iniciado com sucesso no servidor.

5.2.3. A carga de arquivos na plataforma GALAXY: uma ferramenta comum aos diversos fluxos de trabalho

Na plataforma GALAXY, qualquer análise tipicamente se inicia através da importação dos dados de entrada. Para isso, ela provê diversos métodos possíveis por intermédio da guia *Get Data* no painel de ferramentas mais à esquerda da tela. Podem ser obtidos dados, por exemplo, diretamente de diferentes bases de dados, como UCSC Genome Browser, FlyMine, RatMine, EBI SRA, CBI-Rice Mart, etc. Neste projeto, no entanto, foi usada apenas a importação direta de arquivos, por meio da ferramenta *Upload File*, uma outra opção, portanto, da guia *Get Data* citada. Um detalhe deve ser ressaltado neste ponto, tal como alertado na própria ferramenta de carga de arquivo: devido a limitações inerentes ao uso do navegador *Web*, a transferência de arquivo de tamanho superior a 2 GB sempre apresenta erros. Recomenda-se, então, para essa situação específica de grandes arquivos, o uso de servidor FTP. Uma vez que os dados de NGS usualmente atingem tamanhos da ordem de vários GB, tal recomendação foi seguida neste projeto, ou seja, os arquivos de entrada eram colocados, de antemão, em uma máquina com servidor FTP ativo e, por meio do uso da ferramenta *Upload File*, o caminho para o arquivo na rede era fornecido no campo *URL/Text* (ou "URL/Texto") da ferramenta[61]. Mais detalhes são fornecidos no roteiro encontrado no Apêndice A deste trabalho.

[61] O *Live Quickie* de número 17 - Using FTP (http://screencast.g2.bx.psu.edu/quickie_17_ftp_upload/flow.html) também foi usado como referência para esse fim.

5.2.4. Sequência de etapas do fluxo de trabalho básico para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina no protótipo *LASZLO @ GALAXY*

Para a elaboração deste fluxo de trabalho e determinação das ferramentas e programas integrantes, tomou-se por base o "teste-piloto" anterior e os protocolos recomendados nas apresentações *Live Quickies*[62] (algo como "Breves apresentações", em português) de números 10 a 14[63] (da lista de tutoriais da plataforma) e no minicurso de Skrabanek (2012).

Para fins de maior concisão do trabalho, durante a exposição, a seguir, dos diferentes fluxos de trabalho produzidos, procurar-se-á descrevê-los o mais textualmente possível, ilustrando-se apenas as ferramentas que foram especificamente adaptadas ou criadas para o projeto. Conforme adiantado, o Apêndice A traz um maior detalhamento dos passos seguidos no fluxo de trabalho descrito nesta subseção 5.2.4, com o objetivo de exemplificar um possível formato de manual do usuário para a devida utilização do serviço da plataforma.

5.2.4.1 A carga dos arquivos "lane_2_1.txt" e "lane_2_2.txt"

Cada arquivo desses continha 4,1 GB de tamanho, contemplando cerca de 24,4 milhões de leituras pareadas. Foi usado, portanto, o método *URL/Text* da ferramenta *Upload File*, para cada um dos dois arquivos.

5.2.4.2 Manipulação dos dados em formato FASTQ

Assim como o programa MAQ, citado no caso do "teste-piloto", as ferramentas disponíveis na plataforma GALAXY só trabalham com o formato FASTQ Sanger. Para lidar com o cenário de tripla codificação existente para o formato FASTQ (Cock et al., 2010), foi usada, então, a ferramenta *FASTQ Groomer* para obter o arquivo FASTQ Sanger na saída; ferramenta essa que "permite aos usuários sem experiência de programação, a fácil manipulação de dados de sequenciamento, usando uma interface de apontar e clicar" (Blankenberg et al., 2010b). Assim, a ferramenta *Groomer* sempre foi aplicada nos passos iniciais dos fluxos de trabalho que envolviam a utilização de um arquivo do tipo FASTQ.

---

[62] http://wiki.g2.bx.psu.edu/Learn/Screencasts.

[63] 10 - *Mapping Against a Custom Reference Genome* (http://screencast.g2.bx.psu.edu/quickie10_custom_genome/flow.html); 11 - *Illumina Single Ends* (http://screencast.g2.bx.psu.edu/quickie_11_illumina_se/flow.html); 12 - *Illumina Paired Ends* (http://screencast.g2.bx.psu.edu/quickie12_illumina_pe/flow.html); 13 - *Basic FASTQ Manipulation* (http://screencast.g2.bx.psu.edu/quickie_13_fastq_basic/flow.html); 14 - *Advanced FASTQ Manipulation* (http://screencast.g2.bx.psu.edu/quickie_14_fastq_adv/flow.html).

5.2.4.3 Análise e refinamento dos valores de qualidade das leituras dos arquivos de entrada

Os primeiros passos após a aquisição dos dados sequenciados produzidos envolvem a sua preparação e verificação de qualidade, por meio do seguinte fluxo de trabalho básico: (i) parseamento[64] da saída produzida pelo sequenciador (e preparação, tal como foi realizada com o programa *Groomer* do item anterior); (ii) avaliação e (iii) visualização de um sumário de estatísticas a respeito dos valores de qualidade e distribuição de nucleotídeos; (iv) poda de bases das leituras, se necessário, e (v) filtragem (remoção) de leituras por valor de qualidade e demais manipulações. Uma vez que os valores de qualidade podem variar ao longo das sequências de bases das leituras, a determinação de como realizar a poda de bases ou remoção de leituras com baixa qualidade passa pela avaliação dos resultados dos itens (ii) e (iii) citados. GALAXY traz embutida a ferramenta *FASTQ Summary Statistics*, disponível na guia *NGS: QC and manipulation*, bloco *ILLUMINA FASTQ*, como uma das opções para auxiliar a realização de tal tarefa. O formato tabular do resultado produzido pela ferramenta, apesar de englobar várias informações estatísticas a respeito de cada posição de base das leituras do arquivo FASTQ, não é muito agradável de ser lido. Entretanto, ele pode ser imediatamente plotado em gráfico por meio da ferramenta *Bloxplot*, disponível na guia *Graph/Display Data* do painel esquerdo de ferramentas, tornando a sua interpretação muito mais fácil. A Figura 5.8 exibe os gráficos de caixa obtidos para os dois conjuntos trabalhados.

[64] Nas áreas de ciência da computação e linguística, parseamento ou análise sintática é o processo de analisar uma sequência de entrada para determinar (e validar) sua estrutura (gramatical) segundo um conjunto de regras formais (gramática) (http://pt.wikipedia.org/wiki/Parsing). *Parsear*, *analisar*, *analisar sintaticamente*, *dividir* significa subdividir a entrada de modo que um programa possa atuar sobre suas informações (Microsoft PRESS®, 1998).

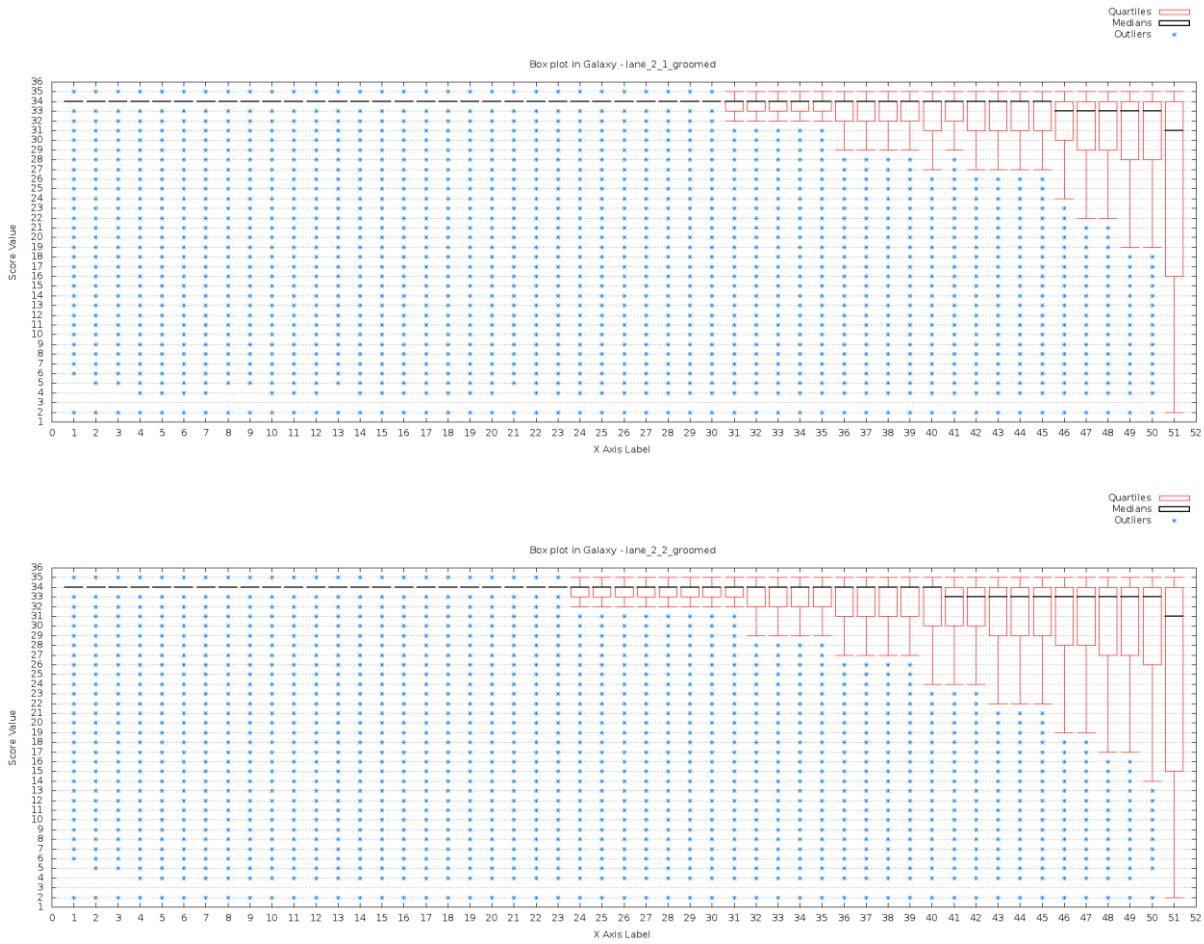


Figura 5.8 - Gráficos de caixa obtidos para os conjuntos "lane_2_1.txt" e "lane_2_2.txt", após a passagem pela ferramenta *FASTQ Groomer* e avaliação com auxílio da ferramenta *FASTQ Summary Statistics*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Uma vez que, a partir do gráfico, foram observados alguns valores extremos povoando o nível mais abaixo do valor de qualidade 20 (escala PHRED) na extremidade 3' das leituras, principalmente para o segundo conjunto de dados ("lane_2_2.txt"), optou-se por realizar a poda de bases nas posições de 49 a 51 de todas as leituras, antes que os dados fossem passados adiante no devido fluxo de análise. No entanto, seguindo a recomendação de que não se deve realizar operações de filtragem/remoção de leituras ou de poda de bases em dados de leituras pareadas que estejam separadas em dois arquivos, tal como era o caso, sob pena de se perder o sincronismo entre as leituras dos dois conjuntos[65], comprometendo o processo de mapeamento posterior, foi usada a ferramenta *FASTQ joiner* para "unir" os dois arquivos, antes da pretendida operação de poda de bases. A ferramenta basicamente transforma os arquivos de dados pareados (por exemplo, com identificadores #0/1 e #0/2), os quais possuem, cada um, leituras de tamanho "x", em um único arquivo (com identificador #0, por exemplo), porém com leituras de tamanho "2x". Assim, quando do eventual momento de

[65] 13 - *Basic FASTQ Manipulation* (http://screencast.g2.bx.psu.edu/quickie_13_fastq_basic/flow.html); 14 - *Advanced FASTQ Manipulation* (http://screencast.g2.bx.psu.edu/quickie_14_fastq_adv/flow.html).

filtragem de leituras ou poda de bases por razões de qualidade, garantir-se-á a não ocorrência de perda de sincronismo entre as leituras pareadas.

Com a união dos dois arquivos de leituras pareadas, foi executada a poda de bases planejada, por meio da ferramenta *FASTQ Trimmer*, do mesmo conjunto de ferramentas de manipulação de arquivos FASTQ da plataforma GALAXY. Essa ferramenta é "inteligente" o suficiente para trabalhar com dados pareados, ou seja, mesmo estando as leituras pareadas concatenadas em uma única linha (efeito obtido com a ferramenta *joiner*), caso o arquivo seja novamente dividido em dois conjuntos, estes últimos irão incorporar quaisquer modificações introduzidas pelo usuário (por exemplo, remoção de leituras ou de bases com baixa qualidade), sem ocorrer perda de sincronismo.

Programas mapeadores (alinhadores) típicos, tais como BWA (Li; Durbin, 2009) ou Bowtie (Langmead et al., 2009), para os quais a plataforma GALAXY já oferece *wrappers* prontos, somente lidam com leituras pareadas, caso elas estejam dispostas em dois conjuntos de dados (tal como estavam, originalmente, os dois arquivos "lane_2_1.txt" e "lane_2_2.txt", antes da aplicação da ferramenta *joiner*). Tendo sido realizada a poda de bases no passo anterior, o arquivo único, agora, poderia ser retornado à condição de organização em dois conjuntos. Para isso, uma ferramenta com função antagônica à da ferramenta *joiner*, *FASTQ splitter*, foi usada, em seguida, gerando dois arquivos na saída de sua execução.

Neste ponto, conforme os protocolos seguidos, os arquivos de leituras já poderiam ser usados pelo programa alinhador. No entanto, antes da execução deste último, o arquivo com o genoma de referência deveria, também, ser inserido no processo.

5.2.4.4 Carga do arquivo com o genoma de referência

Para este fluxo de trabalho, uma nova versão do arquivo com o genoma de referência de *Leishmania mexicana*, em formato FASTA, foi obtida — no caso, "LmexicanaGenomic_TriTrypDB-4.0.fasta", com tamanho de 31,1 MB. Para a carga desse arquivo, a mesma ferramenta utilizada para a inclusão dos arquivos de leituras foi empregada. No entanto, como ele possuía um tamanho bem menor, se comparado aos tamanhos dos arquivos de leituras, o método de seleção e carregamento do arquivo a partir do próprio computador do usuário (sem necessidade de uso de FTP), pôde ser aplicado.

5.2.4.5 Mapeamento das leituras de Illumina com o software BWA

No "teste-piloto" descrito na subseção 5.2.1, realizado para o mesmo conjunto de dados de NGS de entrada, o programa de alinhamento utilizado foi o MAQ. Conforme dito, esse foi o primeiro programa de alinhamento de leituras curtas capaz de levar em consideração os valores de qualidade das bases atribuídas, o que o tornou praticamente uma referência nesse tipo de abordagem de montagem (Paszkiewicz; Studholme, 2012). No entanto, o programa parou de receber atualizações e melhorias de desenvolvimento, por parte de seus desenvolvedores, sendo substituído pelo programa BWA (Koboldt, 2009; Paszkiewicz; Studholme, 2012). Em virtude disso e também por satisfazer à maioria dos critérios anteriormente determinados para a escolha de programas dos primeiros fluxos de trabalho — por exemplo, seu uso era sugerido nos tutoriais da plataforma e no trabalho de Skrabanek (2012) e ele já possuía *wrappers* disponíveis na instância local (apesar de ainda ser necessária a instalação do programa, propriamente dita), sendo capaz de trabalhar com leituras geradas nas tecnologias NGS Solexa/Illumina e SOLiD™ — o pacote BWA, na versão 0.6.1, foi o escolhido para substituir o programa MAQ no fluxo de trabalho idealizado. Além de ater-se aos critérios do projeto, o programa ainda trazia algumas outras vantagens, se comparado ao próprio MAQ, tais como: maior rapidez de execução, geração de saída diretamente em formato SAM, capacidade de realizar alinhamento com lacunas (*gapped alignment*), tanto para leituras únicas (*single-end*), quanto pareadas (*paired-end*) (ao passo que MAQ só o faz para o segundo caso), maior tamanho de leitura curta tratável (200 pb contra 63 do MAQ), dentre outras (Koboldt, 2009). A instância GALAXY também possui o *wrapper* do programa Bowtie disponível, como outra possibilidade de mapeador de leituras para dados de Solexa/Illumina e ABI SOLiD™, porém, na versão suportada, o programa não permite alinhamento com lacunas[66]. Nesses termos, BWA é mais capacitado para a detecção de *Indels* e SNPs e tende a ser mais utilizado em projetos de genomas, enquanto a versão vigente de Bowtie, na plataforma, apesar de mais rápida e de consumir menos memória, tende a ser mais usada em projetos de RNA-Seq (a título de exemplo, Bowtie é usado em conjunto com o programa TopHat (Trapnell et al., 2009), um outro mapeador de leituras especialmente concebido para experimentos de RNA-Seq) (Skrabanek, 2012).

[66] Somente recentemente foi lançada uma versão 2.0 beta para Bowtie, a qual faz alinhamento com lacunas (Langmead; Salzberg, 2012; Koboldt, 2012).

Após a opção por BWA, o pacote foi instalado seguindo-se as instruções de administração e configuração de instância local para NGS[67] e, em seguida, as instruções fornecidas na documentação do próprio pacote[68].

A saída gerada pela ferramenta é apresentada no formato SAM, para o qual alguns detalhes adicionais são fornecidos no Apêndice C.

5.2.4.6 A filtragem de dados usando o pacote SAMtools

Tendo sido concluído o mapeamento das leituras pelo programa BWA e estando todas as informações referentes ao alinhamento reunidas no arquivo de saída em formato SAM, filtros deveriam ser aplicados a este último, mais precisamente sobre o campo FLAG, com o intuito de obter somente as leituras pareadas que eventualmente teriam mapeado "apropriadamente" em relação à referência. Tais filtros faziam parte da ferramenta *Filter SAM*, disponível na seção de ferramentas *NGS: SAM Tools*, da plataforma GALAXY. No entanto, tal como o caso da ferramenta BWA na instância local, os programas binários referentes ao pacote SAMtools também não estavam instalados, só estando disponíveis os respectivos *wrappers*. Assim, procedeu-se com a instalação do software, na versão 0.1.9, conforme as instruções de administração e configuração de instância local para NGS e, em seguida, foram observadas, também, as instruções fornecidas na documentação de SAMtools[69].

5.2.4.7 Conversão do arquivo em formato SAM para o formato BAM usando SAMtools

O formato BAM, concebido para fins de melhoria de desempenho, é a versão binária compacta do equivalente SAM, mantendo as mesmas informações deste último. Ambos os formatos podem ter os alinhamentos classificados por coordenadas para agilizar o processamento dos dados e evitar a carga desnecessária de outros alinhamentos na memória. Quando classificado por coordenada, um arquivo BAM pode, também, ser indexado, tornando-se, assim, uma representação compacta e mais acessível de alinhamentos de sequências de nucleotídeos. Para algumas aplicações de análise de dados de NGS *downstream*, sua principal vantagem é o fato de que, quando indexado, ele possibilita a recuperação rápida de alinhamentos sobrepondo uma dada região específica, sem que, para isso, seja necessário varrer todo o resultado do alinhamento no arquivo[70] (Li H et al., 2009;

[67] Disponíveis em http://wiki.g2.bx.psu.edu/Admin/Data%20Integration; http://wiki.g2.bx.psu.edu/Admin/NGS%20Local%20Setup.
[68] Disponíveis em http://bio-bwa.sourceforge.net/.
[69] http://samtools.sourceforge.net.
[70] http://genome.ucsc.edu/FAQ/FAQformat.html#format5.1.

The SAM Format Specification Working Group, 2011). De posse, portanto, do conjunto filtrado de resultados no formato SAM, o próximo passo seria convertê-lo para o formato BAM já indexado. Para isso, foi usada a ferramenta conversora de formatos *SAM-to-BAM*.

5.2.4.8 Verificação da posição de mapeamento das leituras na referência usando SAMtools

Os formatos SAM e BAM são centrados nas leituras, ou seja, cada uma de suas linhas de arquivo consiste, apenas, de informações sobre cada uma delas. Para a verificação das suas posições de mapeamento no genoma de referência, é necessária a conversão para outro tipo de formato conhecido como *pileup* (algo como "empilhamento" ou "acúmulo"), no qual cada linha representa uma posição genômica (Skrabanek, 2012) (Figura 5.9).

Figura 5.9 - Saída simplificada do tipo *pileup*. Cada linha consiste do nome da sequência de referência, a coordenada classificada, a base para aquela posição específica da referência, o número de leituras que cobriram aquela posição e as bases encontradas nas leituras. No quinto campo, um ponto ou vírgula significam uma base idêntica à da referência, um ponto ou uma letra maiúscula significam uma base de uma leitura que mapeou na fita direta, enquanto que uma vírgula ou uma letra minúscula significam o mapeamento na fita reversa (Li H et al., 2009; tradução nossa).
Fonte: Extraído de Li H et al., 2009, p.2079.

A obtenção, portanto, da informação em formato *pileup* foi realizada por meio da ferramenta *Generate pileup*.

5.2.4.9 Conversão de formato pileup para FASTQ: um primeiro exemplo de uso do módulo NGS: LASZLO's Sandbox

Um dos requisitos deste projeto era o de que, ao final dos fluxos de trabalho, fosse possível a obtenção dos resultados, em formato FASTA, dos esboços de montagem gerados. A ferramenta GALAXY pública[71] não possuía um conversor nativo desse tipo (*pileup* para FASTQ), tampouco a versão da instância local instalada. No entanto, ambas possuem um conversor de formato FASTQ para FASTA embutido por padrão. Desta forma, tomando por base o documento de sintaxe de configuração de ferramentas para a plataforma GALAXY[72], foi escrito um conjunto de pequenos programas *wrappers*, de maneira a capacitar a instância local com a possibilidade de conversão do formato *pileup* para FASTQ e isso servir de

[71] http://main.g2.bx.psu.edu/.
[72] http://wiki.g2.bx.psu.edu/Admin/Tools/Tool%20Config%20Syntax.

"ponte" para a obtenção do formato final em FASTA. Basicamente, o trabalho de codificação envolveu a criação de uma interface gráfica para a ferramenta (Figura 5.10) e a elaboração de um pequeno *script* para que o conversor do pacote SAMTools *pileup2fq* (Li H et al., 2009) pudesse ser executado de forma básica, quando do acionamento de seu botão *Execute*.

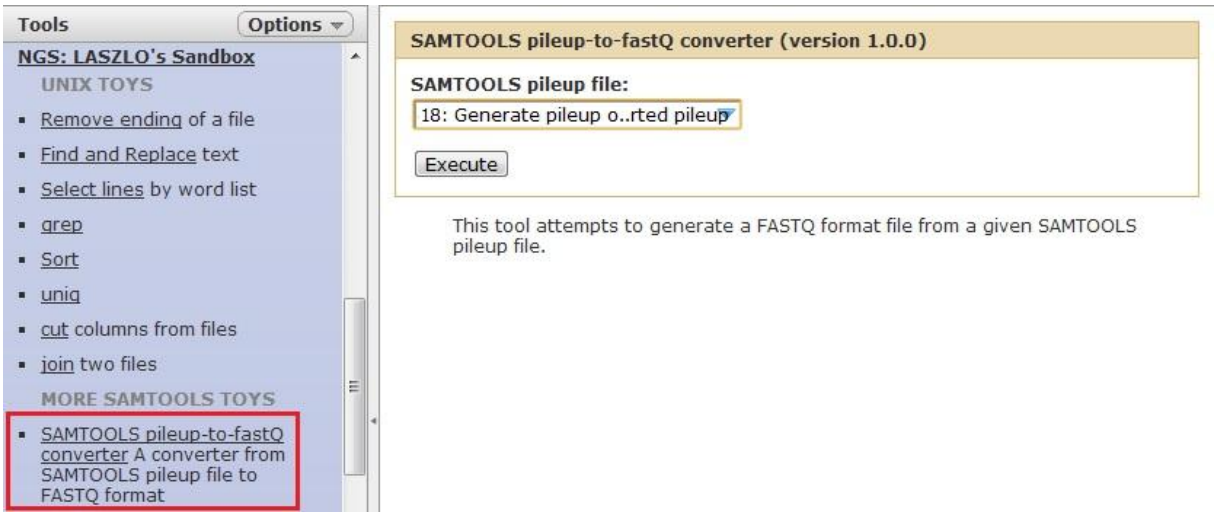


Figura 5.10 - Ferramenta criada para a instância local da plataforma GALAXY: *SAMTOOLS pileup-to-fastq converter* e detalhe da sua guia de acesso no bloco de ferramentas integrante do módulo *NGS: LASZLO's Sandbox*.

Um "bloco" reservado para as ferramentas do módulo *NGS: LASZLO's Sandbox* também foi criado no arquivo de configuração "tool_conf.xml" da instância local da plataforma, tal como mostrado na Figura 5.11, a seguir.

Figura 5.11 - Criação do "bloco" do módulo *NGS: LASZLO's Sandbox* no arquivo de configuração "tool_conf.xml" da instância local da plataforma GALAXY e destaque para a programação da guia da ferramenta personalizada *SAMTOOLS pileup-to-fastq converter*.

Ao ser executada, portanto, a ferramenta sobre o arquivo de *pileup* gerado na etapa anterior ("**18: Generate pileup on data 17 and data 14: converted pileup**"), obteve-se o respectivo resultado em formato FASTQ. A Figura 5.12 exibe o resultado da execução da ferramenta *SAMTOOLS pileup-to-fastQ converter* no painel de histórico dos experimentos do usuário.

Figura 5.12 - Registro, no painel de histórico do usuário, da aplicação da ferramenta *SAMTOOLS pileup-to-fastq converter* sobre o arquivo em formato *pileup* obtido na etapa anterior do fluxo de trabalho.

5.2.4.10 Conversão do formato FASTQ para FASTA

Como passo final para a entrega do resultado em formato FASTA, foi realizada uma última conversão de formato com a ferramenta de conversão *FASTQ to FASTA*, anteriormente mencionada.

5.2.4.11 Captura de informações estatísticas sobre os resultados da montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina no protótipo LASZLO @ GALAXY

A Tabela 5.1 traz as informações estatísticas para a montagem utilizando genoma de referência, colhidas com a ferramenta *SAM Tools flagstat*, presente no bloco de ferramentas *NGS: SAM Tools*[73].

Tabela 5.1 - Informações do relatório produzido pela ferramenta *SAMTools flagstat* em relação aos dados de montagem de *Leishmania amazonensis* com auxílio de genoma de referência.

In total	48849346
QC failure	0
Duplicates	0
Mapped	42459048 (86.92%)
Paired in sequencing	48849346
Read1	24424673
Read2	24424673
Properly paired	40530034 (82.97%)
With itself and mate mapped	40555382
Singletons	1903666 (3.90%)
With mate mapped to a different chr	110481
With mate mapped to a different chr (mapQ>=5)	38661

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Uma vez que a maioria das leituras foi categorizada como tendo sido mapeada de maneira apropriada, ou seja, com ambos os representantes de um par sendo mapeados em um mesmo cromossomo, se mantendo orientados um em relação ao outro e apresentando tamanho de inserto perceptível ao software alinhador; podem ser considerados como bons os resultados obtidos com o fluxo de trabalho básico utilizado — percentual da ordem de 83% para leituras mapeadas apropriadamente.

[73] Resultado obtido empregando-se o conversor *SAM-to-BAM*, disponível no bloco de ferramentas *NGS: SAM Tools*, diretamente sobre o arquivo SAM resultante do mapeamento com o software BWA e o arquivo do genoma de referência, o que possibilita a obtenção de um arquivo BAM com todas as informações a respeito do alinhamento implícitas e sem qualquer tipo de filtragem de dados. Em seguida, a aplicação da ferramenta *flagstat*, presente no mesmo bloco *SAM Tools*, sobre esse arquivo BAM bruto obtido, provê informações estatísticas sobre o alinhamento baseadas no campo FLAG do formato SAM.

5.2.4.12 Monitoração do consumo de memória durante a montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina no protótipo LASZLO @ GALAXY

Para a montagem com auxílio de genoma de referência a partir dos dados de Solexa/Illumina utilizados, o consumo de memória, capturado pela ferramenta MRTG, não ultrapassou a marca dos 10 GB, conforme mostrado na Figura 5.13.

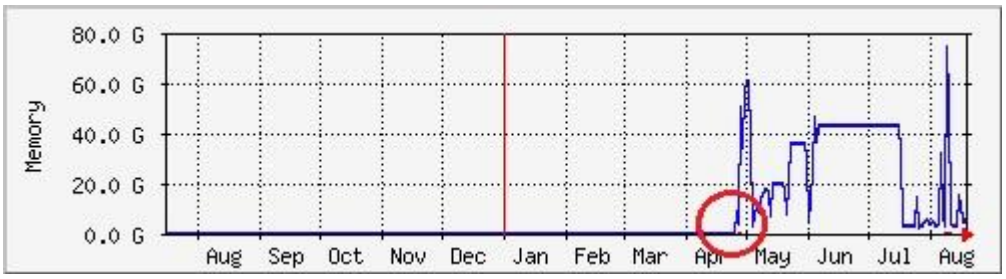


Figura 5.13 - Consumo de memória do servidor durante a montagem utilizando genoma de referência a partir de dados de Solexa/Illumina para *L. amazonensis*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.4.13 Requisito de usuária da área de ciências da vida: uma ferramenta para a busca de regiões específicas nos resultados do mapeamento das leituras de Leishmania amazonensis

Qualquer eventual rascunho de montagem produzido a respeito do genoma de *Leishmania amazonensis* poderia ser, também, de valia para as pesquisas da usuária Adriana Degrossoli, do Laboratório de Bioquímica de Proteínas e Peptídeos, do Instituto Oswaldo Cruz. A ideia básica era, a partir de genes identificados em outras espécies de leishmania, buscar regiões equivalentes no resultado de qualquer montagem obtida. E, uma vez encontrada uma região similar, que também fosse possível "retirar" da montagem, não somente a sequência análoga, mas, também, regiões *upstream* e *downstream* adjacentes. Tais regiões, posteriormente, poderiam ser utilizadas para o desenho de *primers* e determinação de áreas de atuação de enzimas de restrição.

A instância local GALAXY já possuía embutida uma seção de recursos derivados do programa BLAST (Altshchul et al., 1990), denominada *NCBI BLAST+*. Assim, foi idealizada a utilização da ferramenta *NCBI BLAST+ blastn*, para comparar, de forma rudimentar, as sequências de nucleotídeos dos genes já identificados de outras espécies de leishmania com as sequências das montagens produzidas. A tela dessa ferramenta (Figura 5.14) é auto-explicativa e, basicamente, transfere para uma interface amigável as informações usuais que devem ser fornecidas para viabilizar a execução de uma consulta pelo programa BLAST, ao

mesmo tempo permitindo as diferentes opções de saída de resultado típicas para o usuário (por exemplo, formato tabular, em HTML, etc.).

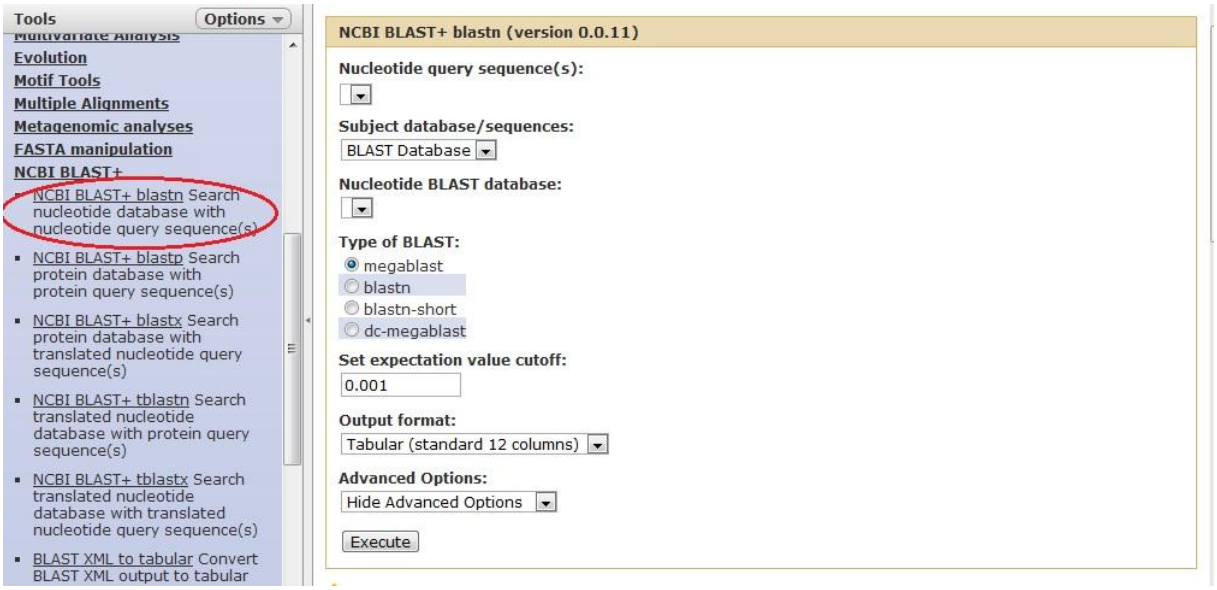


Figura 5.14 - Tela inicial da ferramenta *NCBI BLAST+ blastn* e detalhe de sua respectiva guia de acesso.

Para a extração da região de interesse, foi criado um *wrapper* e, consequentemente, uma ferramenta adicional para o módulo *NGS: LASZLO's Sandbox*, denominada *Extract region tool*, baseada no *script* em linguagem Perl de Smeds (2011). Com isso, a partir do nome do *contig* e das coordenadas da região de interesse, informações estas que deveriam ser planejadas[74] e fornecidas pelo usuário, com base nos dados obtidos com a ferramenta *NCBI BLAST+ blastn* antecessora, a sequência-alvo poderia ser capturada. Basicamente, portanto, a ferramenta funciona a partir dos seguintes passos e parâmetros:

> (1) Guia *NGS: LASZLO's Sandbox*, bloco de ferramentas *OTHER TOYS* → Ferramenta *Extract region tool*;
> (2) No menu suspenso *FASTA file from which the desired region will be extracted* (ou "Arquivo FASTA de onde a região de interesse será extraída"), deve ser selecionada a opção referente ao arquivo que contém a região de interesse a ser extraída (por exemplo, o arquivo com os *contigs* referentes aos resultados ou esboços de montagem);

[74] Na versão inicial da ferramenta, duas "notas auxiliares" foram incluídas para informar ao usuário que, se o propósito for a recuperação de um número adicional de bases ANTES e/ou DEPOIS das posições inicial e/ou final da região de interesse (por exemplo, para o desenho de *primers*), a quantidade de bases desejada deverá ser planejada com um valor (por exemplo, de 1000 bases) que não ultrapasse os limites do *contig* ou cromossomo em questão.

(3) No campo *The sequence header of the contig or chromosome which has the desired region* (ou "O cabeçalho da sequência do *contig* ou cromossomo que contém a região de interesse"), deve ser preenchido, pelo usuário, o nome do cabeçalho do *contig* ou cromossomo que contém a região de interesse;
(4) No campo *The START position of the desired region to be retrieved* (ou "A posição INICIAL da região de interesse a ser capturada"), o usuário deve preencher o número da posição inicial da região de interesse;
(5) E, da mesma forma, no campo *The FINAL position of the desired region to be retrieved* (ou "A posição FINAL da região de interesse a ser capturada"), o usuário deve preencher o número da posição final da região de interesse;
(6) O botão *Execute* (ou "Executar") deve ser, em seguida, pressionado.

A Figura 5.15 exibe (a) a tela da ferramenta *Extract region tool* e (b) o registro de sua execução no painel de histórico dos experimentos do usuário.

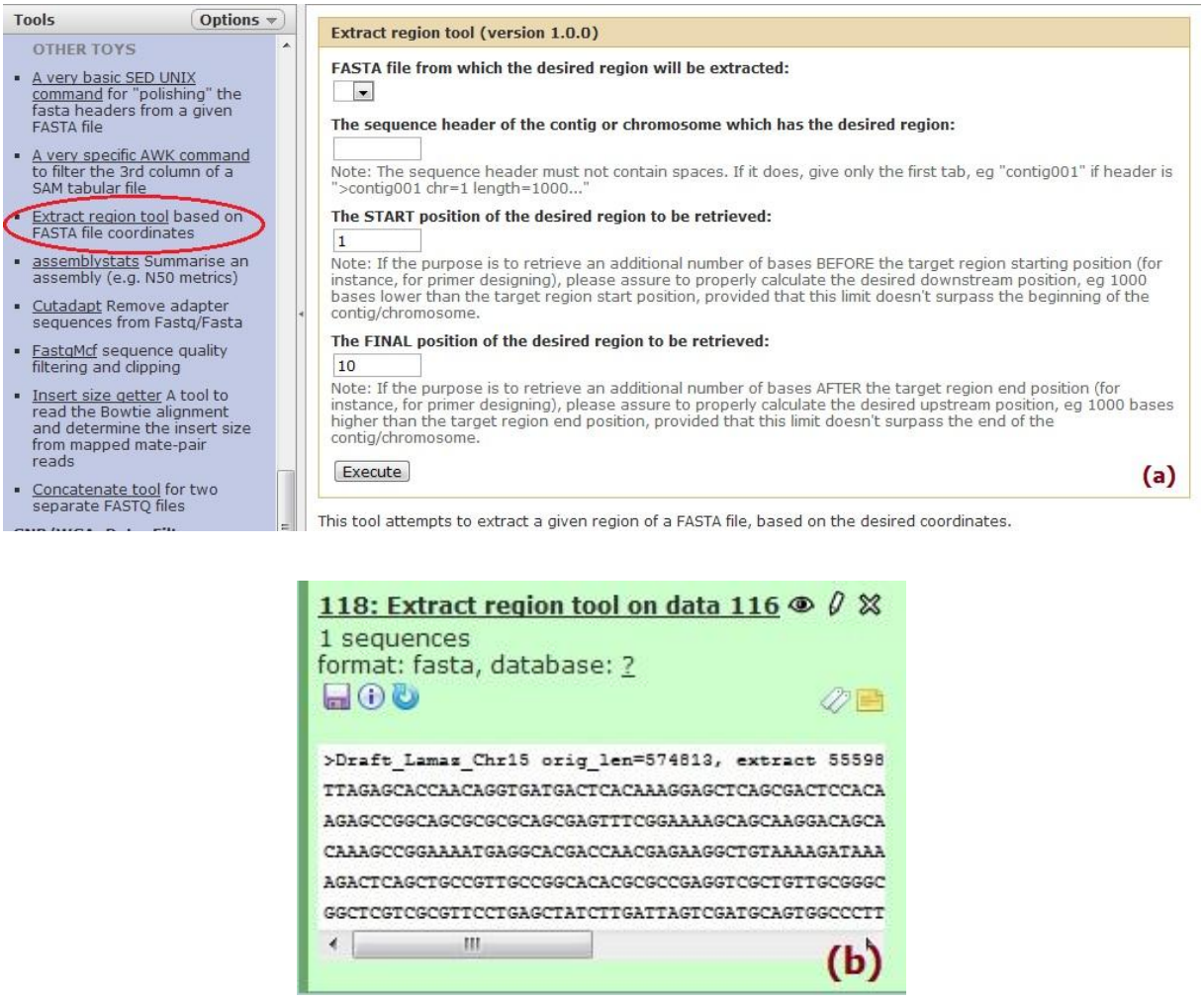


Figura 5.15 - (a) Tela da ferramenta *Extract region tool* do módulo *NGS: LASZLO's Sandbox* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico, da aplicação da ferramenta *Extract region tool*, a partir dos dados fornecidos pelo usuário.

5.2.5. Motivação para os fluxos de trabalho para montagem *de novo*: "teste-piloto" usando o sistema STINGRAY

Paralelamente a este projeto, surgiu a oportunidade de tomar parte em um grupo de trabalho que tinha, como um de seus objetivos, dotar o sistema de integração de recursos de genômica e análise de genes STINGRAY (Wagner et al., 2007) com a capacidade de receber e processar dados provenientes de tecnologias NGS (em fase de elaboração)[75], a partir da inclusão de programas montadores do tipo *de novo*, também de uso livre. Com isso, vislumbrou-se, também, a possibilidade de elaborar fluxos de trabalho básicos adicionais na plataforma da instância local GALAXY aqui descrita, os quais pudessem, entretanto, tirar vantagem da maior flexibilidade de alteração e personalização oferecida por esta última, visando eventuais futuros refinamentos. A combinação dos programas montadores funcionou, portanto, como um outro "teste-piloto" para os demais fluxos de trabalho básicos implementados neste projeto e é brevemente detalhada a seguir.

O sistema STINGRAY, de uso consolidado e efetivamente em produção, oferecia uma menor margem de manobra para mudanças radicais. Ainda que sua modificação fosse primeiramente realizada em ambiente de desenvolvimento, para depois ser replicada no seu equivalente de produção, quanto menos fossem as alterações, melhor isso seria. Assim, optou-se por utilizar o software MIRA para processar dados de 454 e Illumina, devido à sua versatilidade em trabalhar com diferentes tecnologias NGS. No caso dos dados de ABI SOLiD™, o software ABI SOLiD™ *de novo accessory tools 2.0* (Applied Biosystems, 2010) foi o escolhido, por ser recomendado pelo próprio fabricante do sequenciador, estar disponível publicamente e já incorporar uma espécie de *pipeline* embutido em um único pacote, o qual já empregava o pacote Velvet (Zerbino; Birney, 2008). Tais programas, portanto, foram integrados ao sistema STINGRAY, habilitando-o a receber os dados dessas três tecnologias NGS e a processá-los através das rotinas de montagem *de novo* dos referidos programas.

Independentemente da tecnologia NGS, a ideia era a de que, após o recebimento e processamento das *short reads*, o sistema STINGRAY inserisse os *contigs* gerados, já como *clusters*, em seu banco de dados. A partir daí, outras análises poderiam ser realizadas. Em sua concepção original, a solução só podia receber leituras provenientes de sequenciamento

[75] Wagner G, Jardim R, Tschoeke DA, Loureiro DR, Ocaña KACS, Ribeiro A, Emmel VE, Probst CM, Pitaluga A, Vicente ACP, Grisard EC, Cavalcanti MC, Campos MLM, Mattoso M, Dávila AMR. STINGRAY: System for Integrated Genomic Resources and Analysis, artigo em preparação para submissão ao periódico BMC Bioinformatics.

Sanger as quais, após submetidas ao processo de *clusterização* (usualmente executado com o programa CAP3 (Huang; Madan, 1999)), eram, então, devidamente agrupadas, formando os *clusters* (Guimarães; Cavalcanti, 2009). Cabe ressaltar que, principalmente no caso das leituras curtas, muito dificilmente, por si só, elas ofereceriam ao STINGRAY a possibilidade de analisá-las. Somente quando estivessem sob a forma de *contigs* é que elas seriam de utilidade para as rotinas de análise de sequência, comprovando o benefício proporcionado pelo processo de montagem idealizado.

As Figuras 5.16 e 5.17 exibem exemplos de telas da aplicação STINGRAY já preparadas para receber alguns tipos de dados de NGS. Tal como também explorado neste trabalho da instância local GALAXY "personalizada", o conceito de parametrização previamente sugerida ao usuário, com o intuito de tornar a ferramenta mais amigável, também pode ser observado.

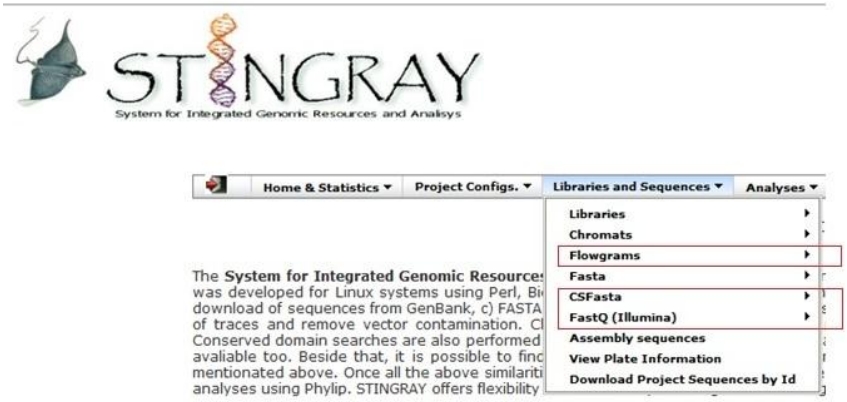


Figura 5.16 - Menu de entrada de dados de NGS na aplicação STINGRAY: opções "Flowgrams" para dados de 454, "CSFasta" para dados de SOLiD™ e "FastQ (Illumina)" para dados de Solexa/Illumina.

Figura 5.17 - Tela de entrada para dados de SOLiD™ com sugestões de parametrização para o usuário.

5.2.6. Sequência de etapas do fluxo de trabalho básico para montagem *de novo* a partir de dados de Solexa/Illumina no protótipo *LASZLO @ GALAXY*

Para a construção deste fluxo de trabalho básico, os critérios de sugestões de encadeamento de programas de uso livre pelos próprios fabricantes de sequenciadores (Illumina, Inc.; 2010a, 2010b) e de existência de *wrapper* disponível na própria plataforma GALAXY (no caso, o pacote Velvet) foram aplicados. A Illumina, por exemplo, propõe, como fluxo de trabalho básico para a montagem *de novo*, a partir dos dados obtidos em uma corrida no sequenciador Genome Analyzer, uma abordagem como a mostrada na Figura 5.18.

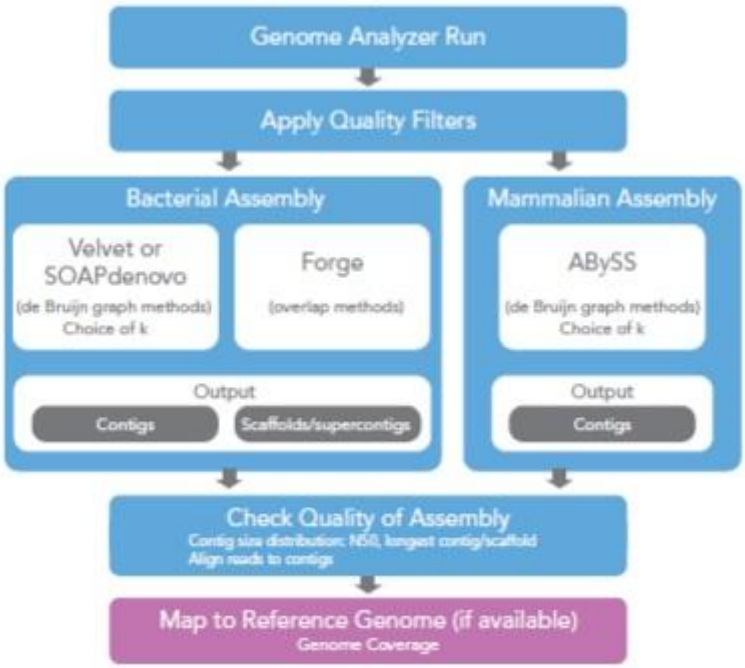


Figura 5.18 - Fluxograma proposto para Illumina para montagem do tipo *de novo*.
Fonte: Modificado de Illumina, Inc., 2010b.

Como a instância local, ao contrário da plataforma pública GALAXY, já vem munida com um *wrapper* para o montador Velvet, o fluxo proposto na figura poderia ser mais facilmente obtido. Velvet foi um dos primeiros montadores para leituras curtas e, atualmente, é um dos mais utilizados. Ele aplica uma abordagem de grafos de Bruijn, na qual cada nó representa os *k-mers*[76] presentes em um conjunto de leituras. Os nós (ou vértices) são ligados entre si quando seus respectivos *k*-mers são consecutivos (sobreposição de *k* - 1 posições) (Figura 5.19). Após a construção do grafo, este é simplificado por uma etapa de correção de erros que remove arestas e nós errôneos (de baixa frequência) e, em seguida, ele é percorrido

[76] Pequenas sequências cujos tamanhos, em bases, são delimitados pelo parâmetro *k* escolhido, sendo *k* menor do que o tamanho da leitura.

por uma abordagem de Caminho Euleriano[77], a qual corresponde à montagem obtida. Velvet também leva em consideração as informações de leituras pareadas (Illumina, Inc., 2010b). Conforme Paszkiewicz e Studholme (2012), para a montagem de um conjunto de dados de genoma completo, a partir de uma biblioteca de tamanho único de inserto, Velvet pode ser uma boa opção, especialmente para pequenos genomas de até, aproximadamente, 40 Mb. Uma vez a literatura apontando um tamanho de genoma de 32.8 Mb para o organismo *Leishmania major* (Bañuls et al., 2007), por exemplo, isso também foi levado em consideração para a utilização da ferramenta Velvet na tentativa de montagem *de novo* dos dados disponíveis de *Leishmania amazonensis*, além dos outros fatores e critérios mencionados mais acima.

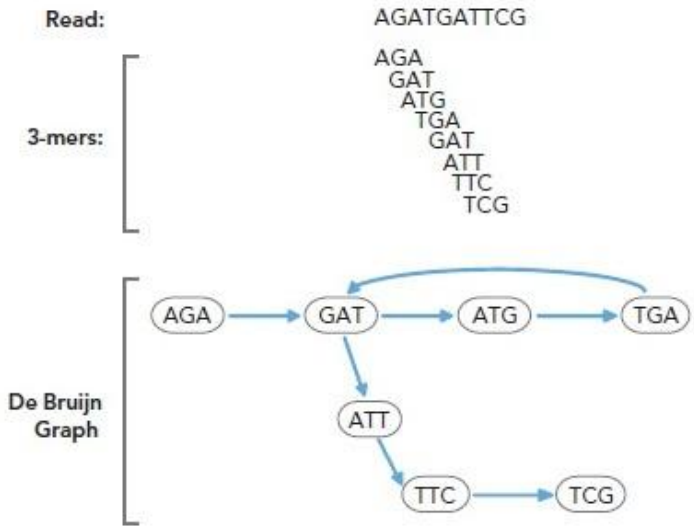


Figura 5.19 - Abordagem de grafo de Bruijn empregada pelo programa Velvet. Na figura, k = 3. O tamanho da sobreposição entre os nós é k - 1 = 2. As setas azuis indicam a ordem dos k-mers e suas sobreposições.
Fonte: Modificado de Illumina, Inc., 2010b.

5.2.6.1 Carga dos arquivos "lane_2_1.txt" e "lane_2_2.txt" e avaliação de qualidade das leituras

Toda a sequência de ferramentas empregada no fluxo de trabalho anterior — *Upload File*, *FASTQ Groomer*, *FASTQ Summary Statistics*, *Boxplot*, *FASTQ joiner*, *FASTQ Trimmer* e *FASTQ splitter* — pode ser, da mesma forma, aplicada ao fluxo de trabalho básico de montagem *de novo* para dados de Illumina.

[77] "Caminho em um grafo que visita cada aresta apenas uma vez."
Fonte: http://pt.wikipedia.org/wiki/Caminho_euleriano.

5.2.6.2 Instalação do pacote Velvet

Para prover funcionalidade aos *wrappers* já existentes na instância local GALAXY, foi instalada a versão 1.2.03 do pacote, seguindo-se as instruções de administração e configuração de instância local para NGS, já mencionadas neste trabalho, e, em seguida, as instruções fornecidas com a documentação do programa.

5.2.6.3 Preparação dos dados de Solexa/Illumina para o programa montador Velvet

Para uso apropriado dos dados pareados em formato FASTQ pelo programa Velvet, era necessária uma manobra de mesclagem dos arquivos anteriormente gerados pela ferramenta *splitter*, uma vez que o pacote só trata arquivos pareados, corretamente, quando cada integrante do par é visto imediatamente após o seu correspondente (Figura 5.20).

```
@HWI-EAS210R_0001:6:1:3:663#CGATGT/1
TGTTCTTATTGGACCAGAANGAGGATTTAGTGAAGAAG
+HWI-EAS210R_0001:6:1:3:663#CGATGT/1
dgggggggggfgggddbb]_BTTSSSfffddddddd^`cc
@HWI-EAS210R_0001:6:1:3:663#CGATGT/2
CTACTCTCTAACAGAAAGCAGATAGTCAAACCGTGTAA
+HWI-EAS210R_0001:6:1:3:663#CGATGT/2
d`Yd`fff`dffefde^cffee^^^Ic_cceeee^WcS
```

Figura 5.20 - Disposição aceita pelo programa Velvet para arquivos de leituras pareadas no formato FASTQ.
Fonte: Modificado de Molecular Evolution, 2012.[78]

O programa Velvet já traz um *script* específico para fazer isso. Porém, para incluir essa funcionalidade na plataforma, um *wrapper* teve de ser escrito para poder adaptá-lo de maneira apropriada. A ferramenta resultante foi incluída no módulo *NGS: LASZLO's Sandbox*, dentro de um bloco de ferramentas denominado *NGS: DE NOVO ASSEMBLY TOYS*, especialmente organizado para abrigar os recursos desse tipo de montagem. Sua interface gráfica pode ser vista na Figura 5.21.

[78] http://www.molecularevolution.org/resources/activities/velvet_and_bowtie_activity.

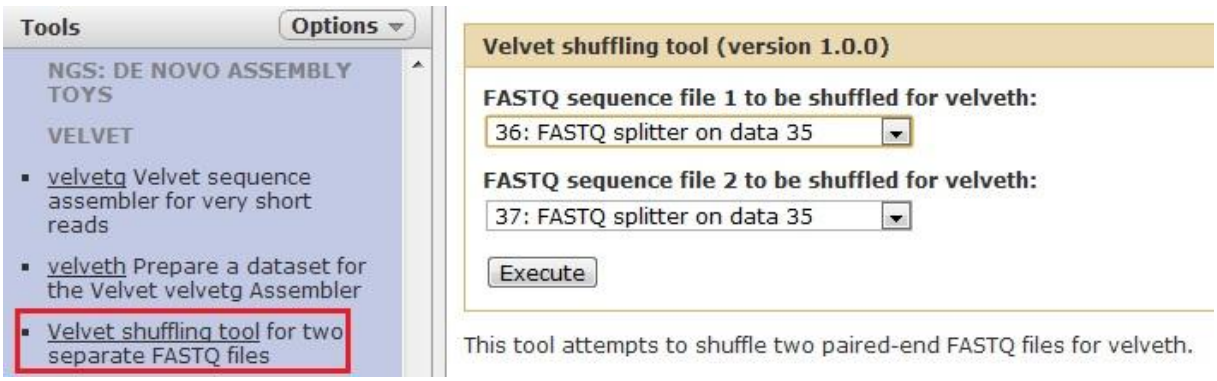


Figura 5.21 - Interface gráfica da ferramenta *Velvet shuffling tool* e detalhe de sua guia de acesso no painel de ferramentas.

Basicamente, utiliza-se cada um dos campos da ferramenta para inserir os arquivos provenientes da ferramenta *splitter* ou, caso não sejam aplicadas etapas de melhoria de qualidade por poda de bases, diretamente os arquivos FASTQ originais.

5.2.6.4 A montagem de novo a partir de dados de Solexa/Illumina com o programa Velvet no protótipo LASZLO @ GALAXY

Com os arquivos pareados estando devidamente preparados para a utilização, passou-se à etapa de montagem com o programa Velvet. Dois subprogramas principais compõem o referido pacote: (1) *velveth*, o qual lê os arquivos de sequências e produz uma tabela *hash*[79] e dois conjuntos de arquivos denominados *Roadmaps* e *Sequences* (algo como "Mapas" e "Sequências", em português), para uso pelo subprograma subsequente (2) *velvetg*, responsável por construir os grafos de Bruijn e executar a montagem. Da mesma forma, um *wrapper* específico, desenvolvido por James Johnson (Universidade de Minnesota, EUA), é disponibilizado, no pacote original da instância local, para cada uma desses programas. As duas interfaces são mostradas nas Figuras 5.22 e 5.23, respectivamente.

[79] Estrutura de dados especial, que associa chaves de pesquisa a valores, com o objetivo de, a partir de uma chave simples, fazer uma busca rápida visando a obtenção do valor desejado (http://pt.wikipedia.org/wiki/Tabela_de_dispersão).

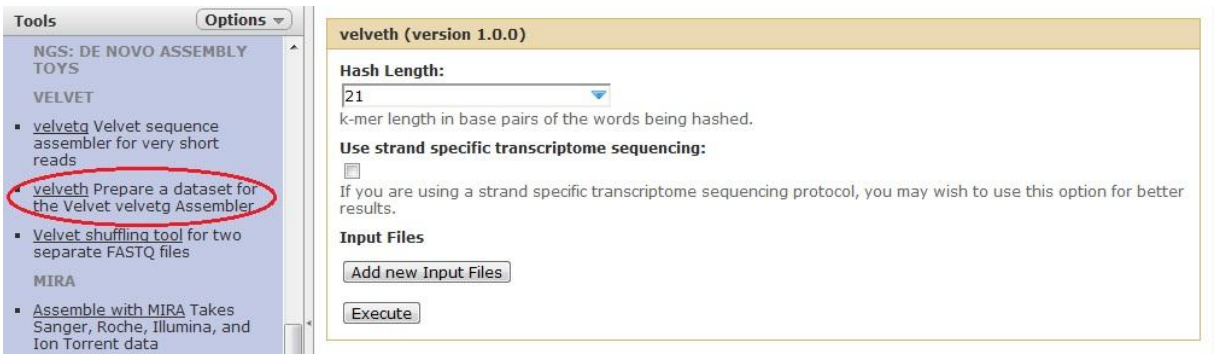


Figura 5.22 - Interface gráfica da ferramenta *velveth*, de autoria de James Johnson, e detalhe de sua guia de acesso no painel de ferramentas. Na instância local *LASZLO @ GALAXY*, a interface da ferramenta foi movida para o bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*, parte integrante do módulo *NGS: LASZLO's Sandbox*.

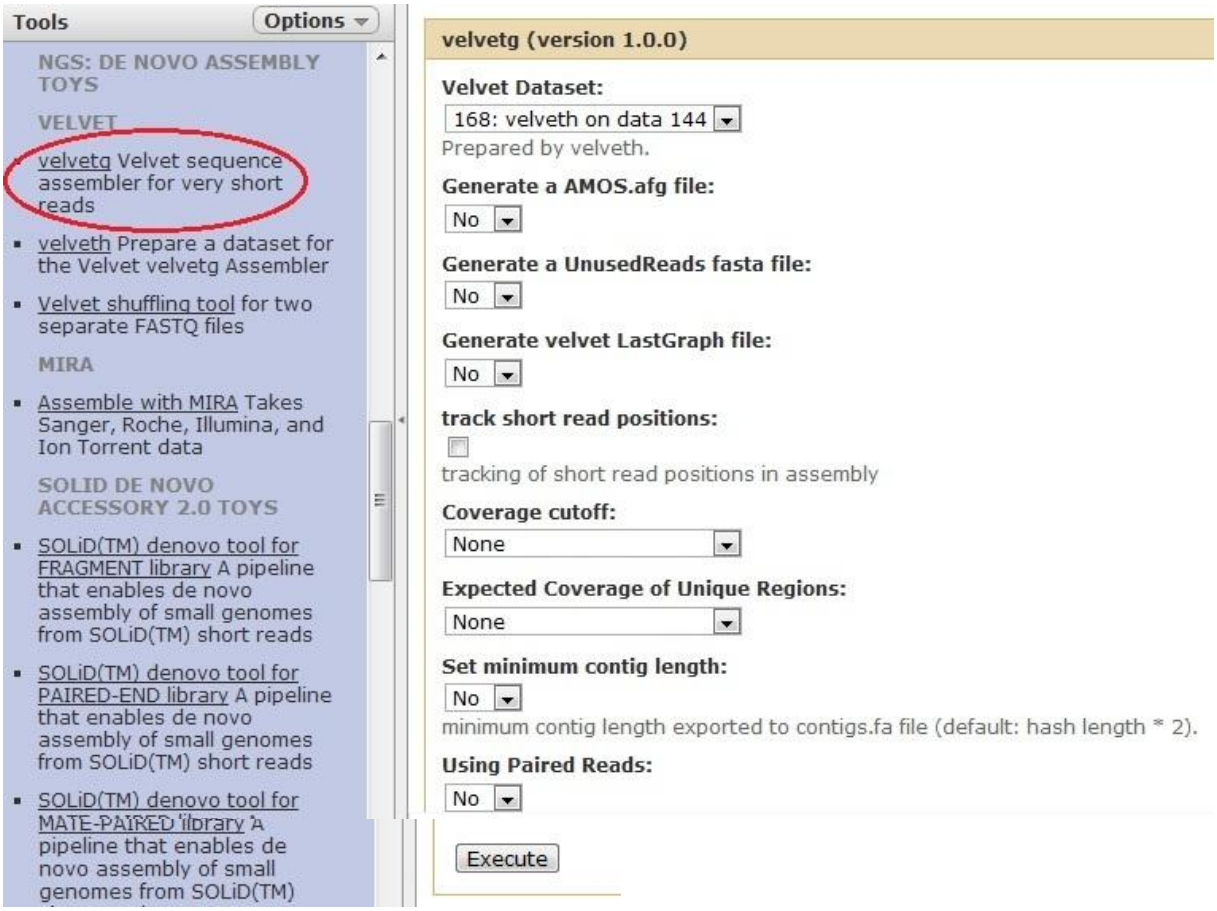


Figura 5.23 - Interface gráfica da ferramenta *velvetg*, de autoria de James Johnson, e detalhe de sua guia de acesso no painel de ferramentas. Na instância local *LASZLO @ GALAXY*, a interface da ferramenta foi movida para o bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*, parte integrante do módulo *NGS: LASZLO's Sandbox*.

Conforme as recomendações de Illumina, Inc. (2010b), um ponto crítico durante a montagem de genomas é a otimização de parâmetros, tais como cobertura, tamanho do inserto das leituras pareadas e qualidade dos dados. Ambas as interfaces oferecem informações auxiliares ao usuário a respeito dos parâmetros que podem ser utilizados, informações essas

baseadas na vasta documentação do software Velvet[80]. Na ferramenta *velveth*, por exemplo, existe o parâmetro *Hash length* (ou "Tamanho do *hash* ou *k-mer*"), o qual indica o tamanho, em pares de bases, das combinações de sequências que serão usadas para gerar o grafo de montagem posterior. Já a ferramenta *velvetg* permite a utilização de parâmetros padrão ou avançados, tais como o tamanho do inserto, por exemplo (a opção por parâmetros avançados é habilitada ao se informar que leituras pareadas estão sendo usadas). Ainda conforme Illumina, Inc. (2010b), alguns pontos devem ser levados em consideração, quando do momento de escolha de parâmetros para a montagem *de novo*. Por exemplo:

- O tamanho mais eficiente para o valor de *k-mer* de uma dada montagem é determinado pela cobertura, o tamanho da leitura, bem como a taxa de erros nos dados. O valor de *k-mer* apresenta influência significativa sobre a qualidade da montagem e, apesar de estimativas poderem ser feitas, ele é difícil de ser determinado, tipicamente sendo obtido a partir do teste de uma faixa restrita de valores. A experiência recomenda, no entanto, que tal valor não deve ser menor que a metade do tamanho da leitura. Recomenda-se, também, o uso de valores ímpares, de maneira a evitar sequências palindrômicas, as quais podem criar ambiguidades no grafo de Bruijn, tornando difícil a sua resolução[81]. Miller et al. (2010), por exemplo, explicam que um palíndromo é uma sequência de DNA que contém a sua própria sequência reversa complementar e que, por esse motivo, sequências desse tipo induzem caminhos que retornam a si próprios nos grafos de montagem. Destacam, porém, que pelo menos um montador — Velvet — evita esse problema de maneira elegante, por exigir um valor ímpar para *k* (a extensão do *k-mer*), já que não há possibilidade de um *k-mer* de valor ímpar encontrar correspondência em sua sequência reversa complementar;
- Quanto ao tamanho do inserto, em linhas gerais, bibliotecas com tamanhos maiores usualmente geram montagens menos fragmentadas e com *contigs* mais longos, dependendo dos elementos repetitivos do genoma do organismo em questão. Recomenda-se, também, a combinação de bibliotecas de insertos longos com as de insertos curtos, porém de grande cobertura, para que seja obtida uma cobertura suficiente.

Após o processo de poda de bases, os arquivos de leituras pareadas (separados pela ferramenta *splitter*), ficaram com sequências de 48 pares de bases de tamanho. Então, para

[80] Disponível em http://www.ebi.ac.uk/~zerbino/velvet/Manual.pdf.
[81] Conforme explicação presente, por exemplo, em: http://www.homolog.us/blogs/2011/09/27/k-mer-sizes-for-genome-assembly/.

testar a funcionalidade do fluxo de trabalho básico para montagem *de novo* de dados de Illumina, foram realizados testes consecutivos de montagem utilizando os seguintes valores de *k-mer*, na ferramenta *velveth*: 25, 29, 31, 33 e 35. Na ferramenta *velvetg*, foi informado que o conjunto de leituras era do tipo pareado, mas, para fins de simplificação, o restante dos parâmetros disponíveis foi deixado em sua condição padrão.

5.2.6.5 Captura de informações estatísticas sobre os resultados da montagem de novo a partir de dados de Solexa/Illumina no protótipo LASZLO @ GALAXY

Para a captura de informações estatísticas simples a respeito das montagens obtidas, foi adaptado o *wrapper* da ferramenta *assemblystats*, disponível na página da comunidade de desenvolvimento GALAXY, já citada, e desenvolvido por Konrad Paszkiewicz (Universidade de Exeter, Reino Unido), como parte do bloco de ferramentas *OTHER TOYS* do módulo *NGS: LASZLO's Sandbox*. Ao ser aplicada a ferramenta, a qual trabalha especificamente com as informações implícitas no arquivo de "Contigs" (gerado, por padrão, por *velvetg*), os resultados estatísticos puderam ser, então, colhidos para cada rodada da ferramenta. A Figura 5.24 traz a interface gráfica da ferramenta *assemblystats*.

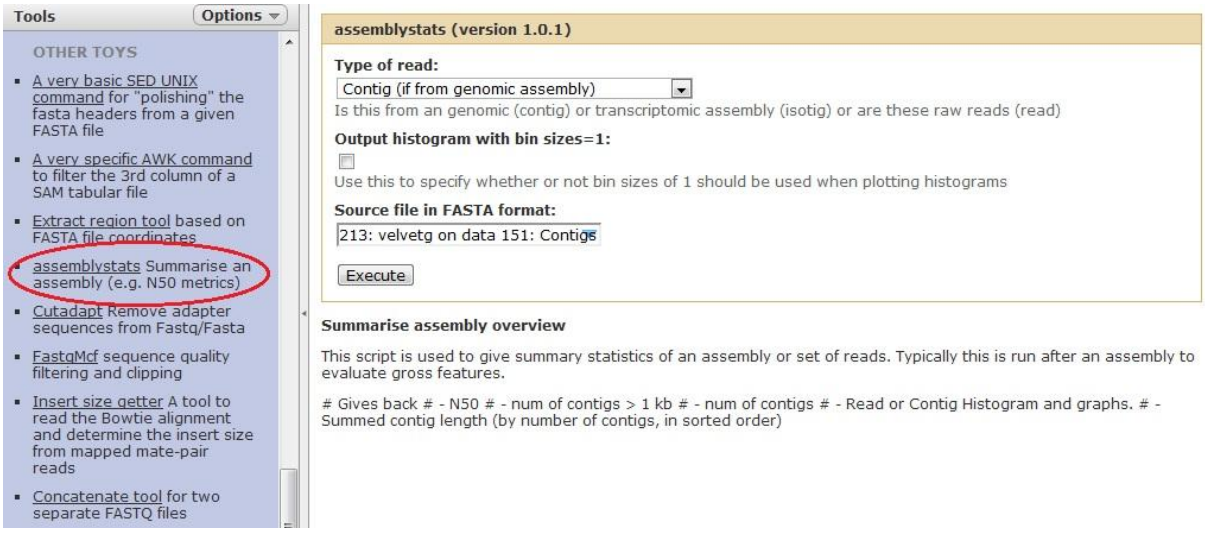


Figura 5.24 - Interface gráfica da ferramenta *assemblystats*, de autoria de Konrad Paszkiewicz, adaptada para a instância local *LASZLO @ GALAXY* e detalhe de sua guia de acesso no painel de ferramentas.

A Tabela 5.2 traz os resultados colhidos, com a ferramenta *assemblystats*, para o conjunto de dados que produziu o maior valor de N50[82].

[82] N50 é uma medida estatística que representa o tamanho do menor *scaffold* ou *contig* acima do qual 50% de um rascunho de montagem de genoma estaria representado (Baker, 2012). Em outras palavras, ao serem ordenados todos os *contigs*, do mais longo ao mais curto, até que o tamanho acumulado exceda 50% do tamanho total de todas as sequências, o tamanho do último *contig* adicionado é o N50. Em geral, quanto maior o valor N50, maior a qualidade do genoma montado (https://wiki.nbic.nl/index.php/Raw_results_of_NGS_de_novo_assembly).

Tabela 5.2 - Informações do relatório produzido pela ferramenta *assemblystats* em relação aos dados de montagem *de novo* de *Leishmania amazonensis* para o conjunto de dados que apresentou o maior valor N50.

Statistics for contig lengths	
Min contig length	61
Max contig length	133,041
Mean contig length	4006.07
Standard deviation of contig length	8422.24
Median contig length	475
N50 contig length	16,671
Statistics for numbers of contigs	
Number of contigs	7,457
Number of contigs >= 1kb	3,005
Number of contigs in N50	515
Statistics for bases in the contigs	
Number of bases in all contigs	29,873,230
Number of bases in contigs >= 1kb	28,813,316
GC Content of contigs	58.99%
Simple Dinucleotide repeats	
Number of contigs with over 70% dinucleotide repeats	0.03% (2 contigs)
AT	0.01% (1 contig)
CG	0.00% (0 contigs)
AC	0.00% (0 contigs)
TG	0.01% (1 contigs)
AG	0.00% (0 contigs)
TC	0.00% (0 contigs)
Simple mononucleotide repeats	
Number of contigs with over 50% mononucleotide repeats	0.00% (0 contigs)
AA	0.00% (0 contigs)
TT	0.00% (0 contigs)
CC	0.00% (0 contigs)
GG	0.00% (0 contigs)

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

A Figura 5.25, produzida pela mesma ferramenta, ilustra as estatísticas relacionadas aos tamanhos de *contigs* obtidos.

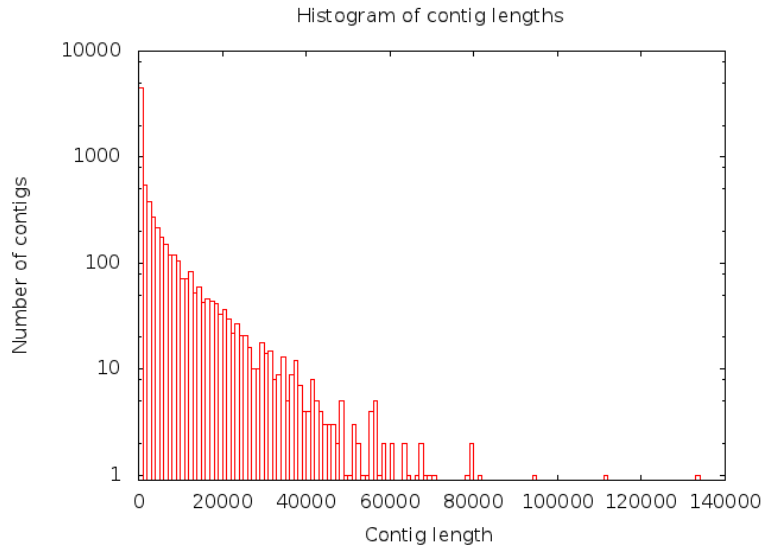


Figura 5.25 - Histograma dos tamanhos de *contigs* para a montagem *de novo*, dos dados de sequenciamento de *Leishmania amazonensis*, para o valor de N50 = 16671.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Já a Figura 5.26, também produzida pela ferramenta, exibe a soma dos tamanhos dos *contigs* produzidos.

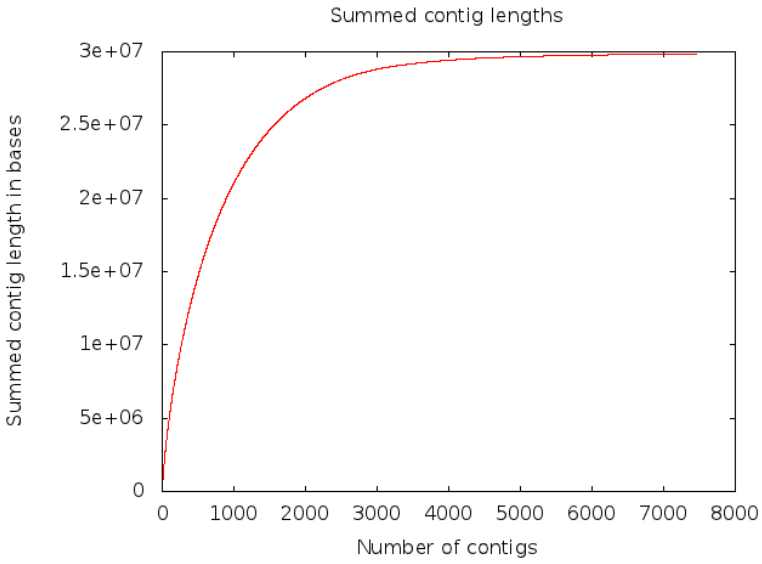


Figura 5.26 - Soma dos tamanhos de *contigs* para a montagem *de novo*, dos dados de sequenciamento de *Leishmania amazonensis*, para o valor de N50 = 16671.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.6.6 Monitoração do consumo de memória durante a montagem de novo a partir de dados de Solexa/Illumina no protótipo LASZLO @ GALAXY

Para a montagem *de novo* dos dados de Solexa/Illumina utilizados, o consumo de memória, capturado pela ferramenta MRTG, alcançou picos de aproximadamente 60 GB, conforme mostrado na Figura 5.27.

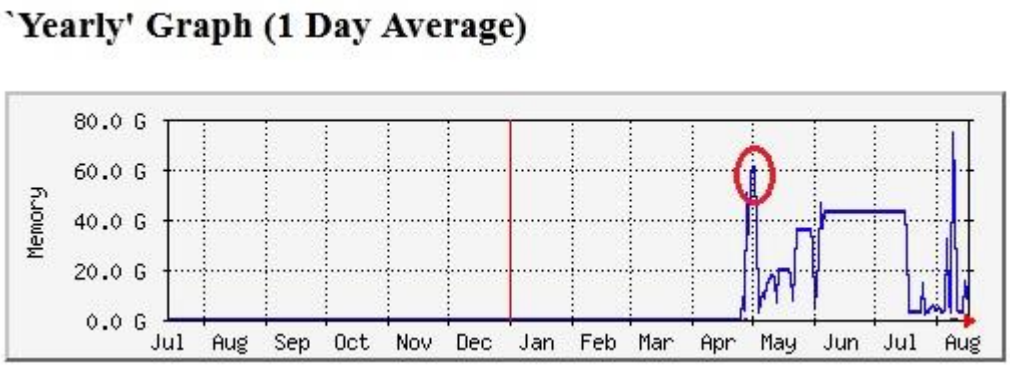


Figura 5.27 - Consumo de memória do servidor durante a montagem *de novo* dos dados de Solexa/Illumina para *L. amazonensis*. A resolução do gráfico obtido (diária) não se refere a um experimento específico (por exemplo, um determinado valor de *k-mer*), mas sim ao dia em que foram executados os testes consecutivos de montagens com diferentes *k-mers*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.7. Sequência de etapas do fluxo de trabalho básico para montagem *de novo* a partir de dados de ABI SOLiD™ no protótipo *LASZLO @ GALAXY*

Tal como no fluxo de trabalho proposto para o sistema STINGRAY, procurou-se incorporar, à plataforma da instância local GALAXY, a ferramenta disponibilizada pela Applied Biosystems (2010) para a montagem *de novo* (SOLiD™ *de novo accessory tools 2.0*), obedecendo, assim, ao critério de projeto de encadeamento de programas de uso livre disponibilizado pelos próprios fabricantes de sequenciadores. Tal ferramenta embute um *pipeline*, sendo recomendada para a montagem de genomas de até 30 Mpb, a partir de leituras curtas originadas na plataforma ABI SOLiD™ e com cobertura de sequenciamento de 50 vezes ou mais. Conforme a documentação do pacote, o *pipeline*, escrito em linguagem Perl, é bastante automatizado e foi concebido para simplificar e otimizar os parâmetros necessários, em prol da maior facilidade de uso e desempenho. As principais características descritas na documentação do pacote, as quais foram implementadas nesta versão do trabalho, são listadas a seguir:

- Por padrão, os dados de entrada para o programa são arquivos de leituras produzidas no sistema SOLiD™, em formato .csfasta, e os respectivos arquivos com os valores de qualidade dessas leituras, em formato .qual;

- Uma etapa de amostragem de um subconjunto de leituras é realizada para estimar os melhores parâmetros para as rotinas de trabalho subsequentes;
- Correção de erro pré-montagem realizada pela ferramenta embutida *SOLiD™ Accuracy Enhancement Tool* (SAET v.2.2);
- Conversões de formatos internas, entre as etapas, realizadas automaticamente;
- Mecanismo de montagem representado pelo software Velvet, na versão 0.7.55[83];
- Fechamento de lacunas entre *contigs*, para a formação de *scaffolds*, no caso de dados pareados, realizada pela ferramenta *Assembly Assistant for SOLiD™* (ASID v.1.0). Essa ferramenta também executa a conversão da sequência-consenso do espaço de cores para o espaço de bases;
- Os programas SAET e ASID podem ser executados em multiprocessamento (suas tarefas podem ser divididas entre diversos processadores ou núcleos de processadores);

A Figura 5.28 exibe as etapas cumpridas pela ferramenta *de novo accessory tools 2.0*, tal como implementada na instância local *LASZLO @ GALAXY*[84].

[83] http://www.ebi.ac.uk/~zerbino/velvet/. Independentemente do pacote *de novo acessory tools 2.0*, o programa Velvet necessita ser instalado e especificamente compilado para trabalhar em espaço de cores, conforme instruções disponibilizadas com o próprio pacote Velvet. Além disso, a documentação do pacote *de novo accessory tools 2.0* oferece instruções a respeito da correta integração entre ambos os programas.

[84] O programa *SOLiD™ de novo accessory tools 2.0*, quando instalado de forma independente, permite que as etapas de *rsampling*, SAET e *analysis* sejam opcionais. Nesta primeira versão do trabalho, para fins de simplificação das tarefas de desenvolvimento e para tirar proveito da existência da ferramenta de correção de erro embutida SAET, os *wrappers* criados forçaram a obrigatoriedade de uso de todas as etapas. Foi retirada nesta versão, também, a etapa de alinhamento das leituras (parte da etapa de análise).

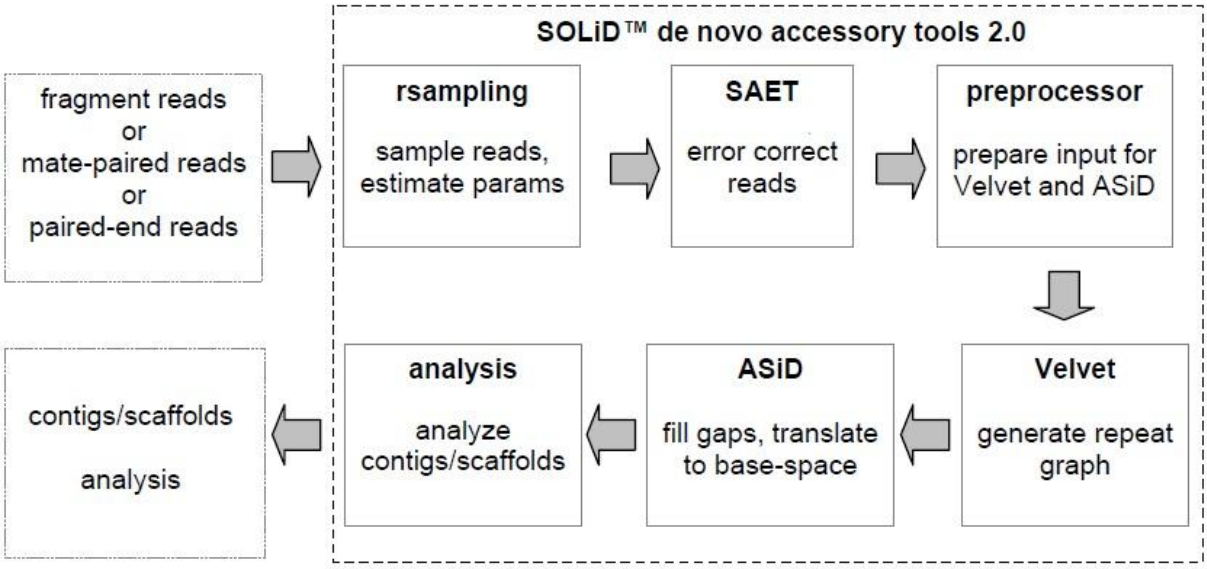


Figura 5.28 - Etapas da ferramenta *de novo accessory tools* idealizadas para a instância local *LASZLO @ GALAXY*.
Fonte: Modificado de Applied Biosystems, 2010.
Nota: Os termos da figura foram mantidos em inglês, tal como no original.

5.2.7.1 Criação dos wrappers do pacote SOLiD™ de novo accessory tools 2.0 para o protótipo LASZLO @ GALAXY

Na inexistência de *wrappers* disponíveis para o pacote *de novo accessory tools 2.0* na plataforma pública, no pacote distribuído da instância local original, ou na comunidade de desenvolvimento GALAXY, três interfaces gráficas foram criadas — uma para montagem de fragmentos únicos, outra para a montagem de dados *paired-end* e outra para a montagem de dados *mate-pair* — e incluídas no módulo *NGS: LASZLO's Sandbox*, sob um novo bloco de ferramentas denominado *SOLID DE NOVO ACCESSORY 2.0 TOYS*, especialmente organizado para abrigar tais recursos. Além disso, os *scripts* originais da ferramenta *de novo accessory tools 2.0*, tal como incentivado pela própria documentação do produto, foram alterados para que pudessem ter sua atuação expandida à instância local personalizada. As Figuras 5.29, 5.30 e 5.31 trazem os aspectos das interfaces gráficas criadas para esta fase do projeto.

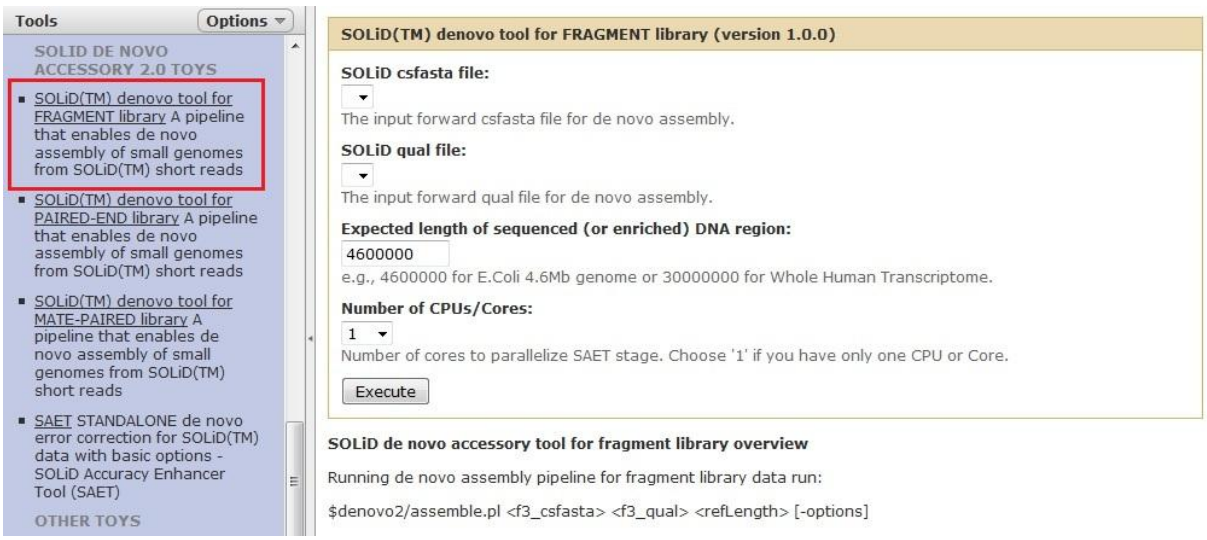


Figura 5.29 - Interface gráfica da ferramenta *SOLiD(TM) denovo tool for FRAGMENT library*, na instância local *LASZLO @ GALAXY*, e detalhe de sua guia de acesso no painel de ferramentas.

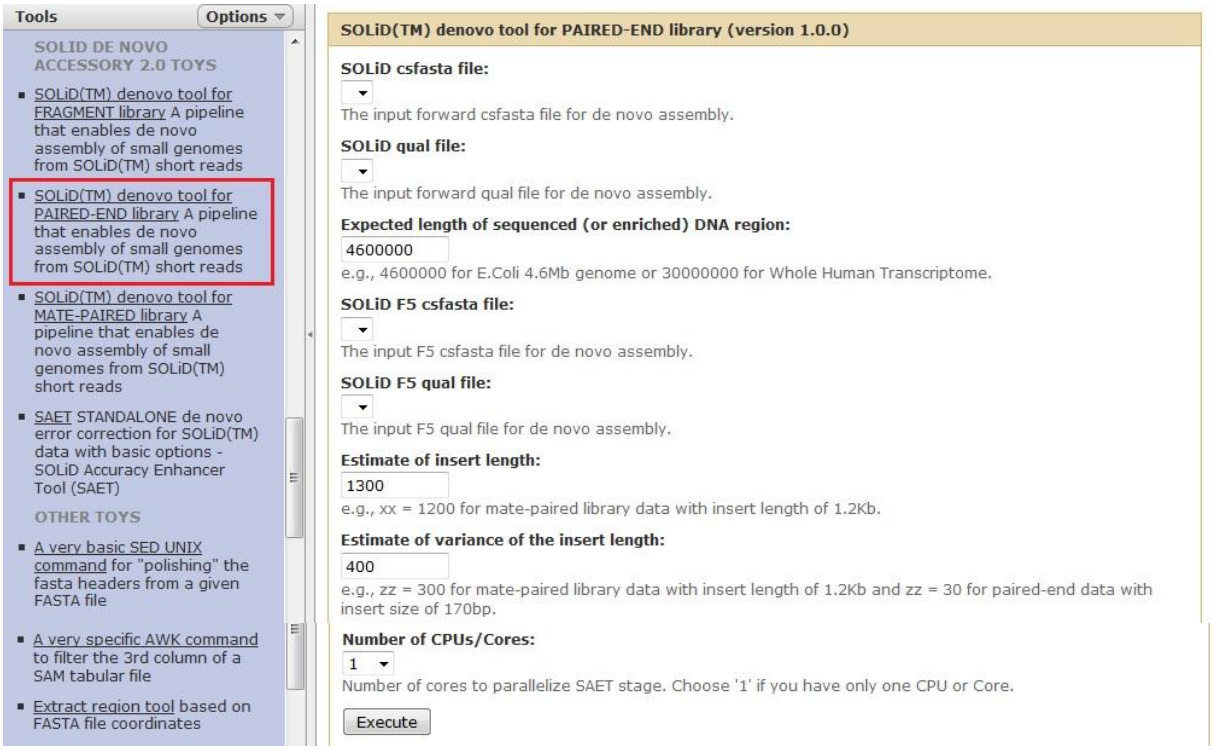


Figura 5.30 - Interface gráfica da ferramenta *SOLiD(TM) denovo tool for PAIRED-END library*, na instância local *LASZLO @ GALAXY*, e detalhe de sua guia de acesso no painel de ferramentas.

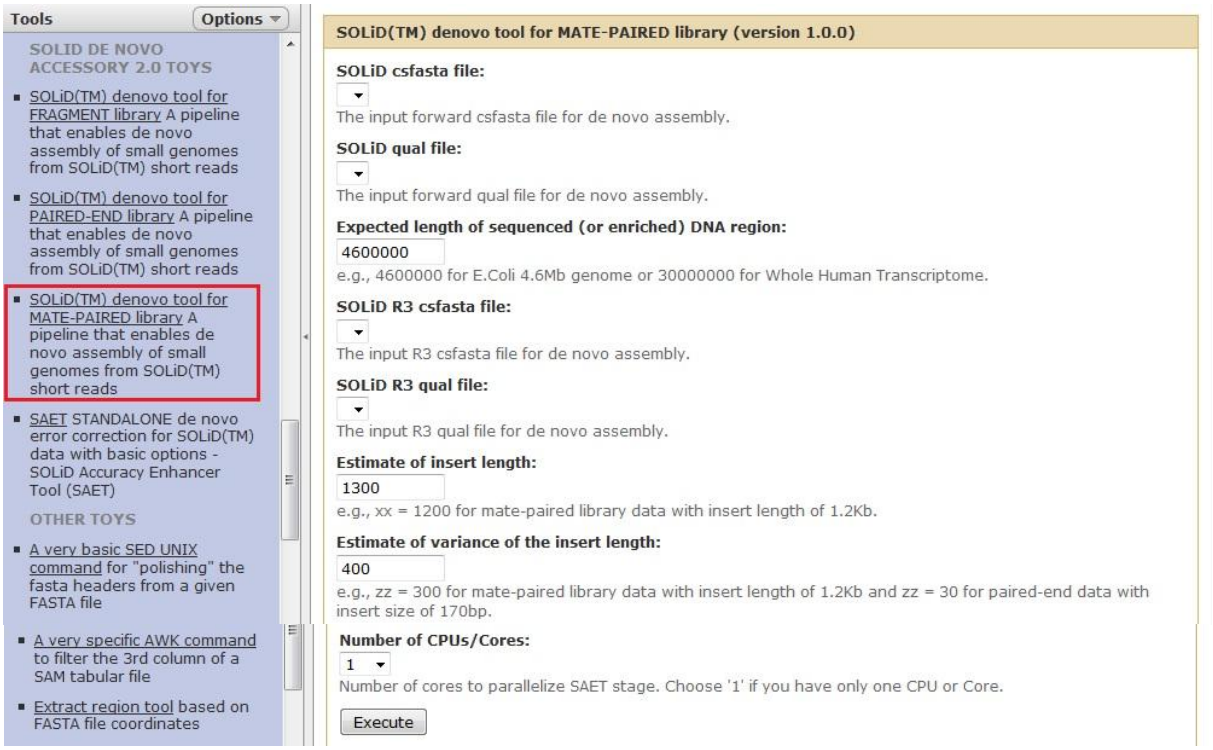


Figura 5.31 - Interface gráfica da ferramenta *SOLiD(TM) denovo tool for MATE-PAIRED library*, na instância local *LASZLO @ GALAXY*, e detalhe de sua guia de acesso no painel de ferramentas.

Pode-se observar, em cada uma das interfaces criadas, a existência do campo *Number of CPUs/Cores* (ou "Número de CPUs/Núcleos"). Uma vez que a máquina de desenvolvimento do protótipo possuía 16 núcleos de processamento (2 processadores com 8 núcleos em cada um), a possibilidade de pararelização das tarefas nas etapas SAET e ASID não foi, assim, descartada. Uma das alterações nos *scripts* objetivou oferecer ao usuário a opção por escolher a quantidade de núcleos de processamento que deveriam ser usados, quando da utilização da ferramenta. Cabe ressaltar que, dependendo da infra-estrutura computacional a ser utilizada, tais *scripts* podem ser facilmente retrabalhados para comportar o número apropriado de núcleos de processamento do recurso de *hardware* específico ou, até mesmo, ter a sua lógica de programação mais refinada para lidar com essa informação de maneira dinâmica.

5.2.7.2 Carga e avaliação de qualidade dos arquivos de E. coli DH10B

Os arquivos "ecoli_600x_F3.csfasta", "ecoli_600x_F3.qual", "ecoli_600x_R3.csfasta" e "ecoli_600x_R3.qual" foram inseridos na plataforma através da ferramenta *Upload File*, da guia *Get Data*, já citada. Em seguida, para avaliar a qualidade das leituras, foram usadas, em conjunto, sobre os arquivos de valores de qualidade .qual, as ferramentas baseadas no pacote

FASTX-Toolkit[85] (de Assaf Gordon), *Compute quality statistics* e *Draw quality score boxplot*, do bloco *AB-SOLID DATA*, *guia NGS: QC and manipulation*, já integrantes da instância local. Os seguintes resultados, exibidos nas Figuras 5.32 e 5.33, foram produzidos.

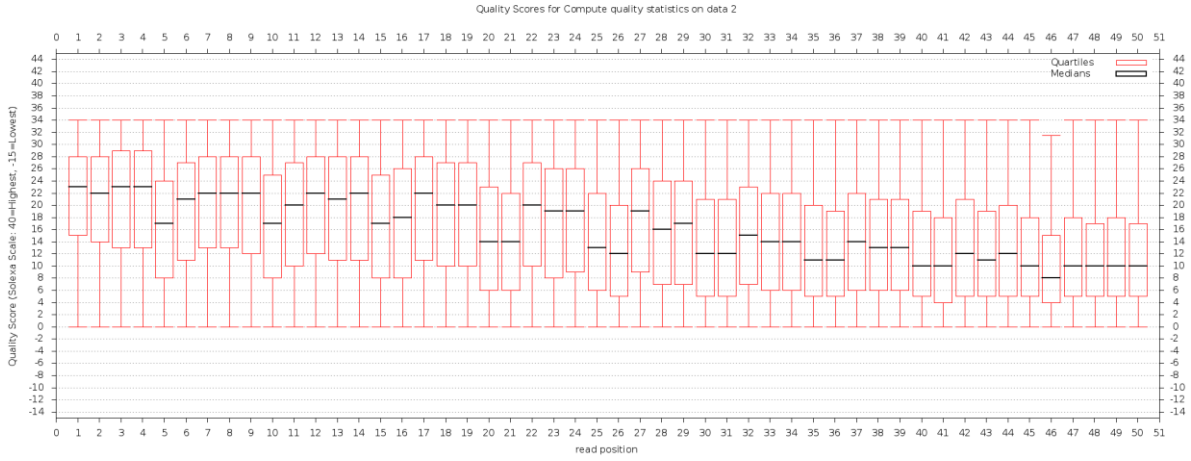


Figura 5.32 - Resultado da avaliação dos valores de qualidade das leituras diretas contidas no arquivo "ecoli_600x_F3.qual".
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

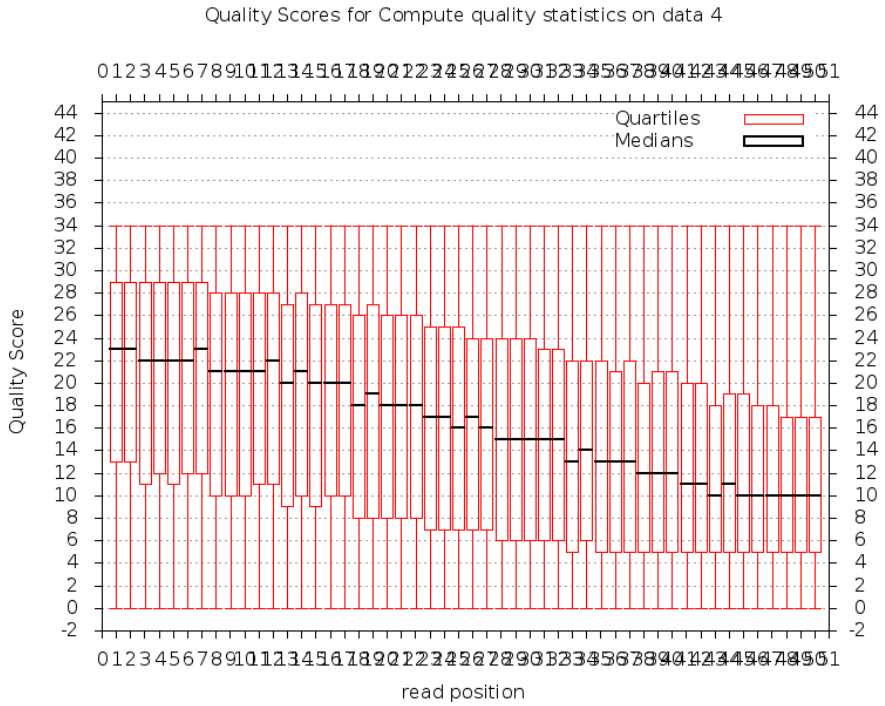


Figura 5.33 - Resultado da avaliação dos valores de qualidade das leituras reversas contidas no arquivo "ecoli_600x_R3.qual".
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

[85] Disponível em http://hannonlab.cshl.edu/fastx_toolkit/index.html.

Uma vez que os arquivos seriam tratados pela ferramenta SAET, nenhuma operação de poda de bases ou correção de qualidade foi realizada, apesar de ter sido identificada queda de qualidade nas posições finais das leituras, sendo esta ainda mais abrupta para o caso das leituras reversas. Dois experimentos de montagem foram, então, realizados, tal como se segue.

5.2.7.3 Experimento com dados de SOLiD™ usando biblioteca de fragmentos únicos no protótipo LASZLO @ GALAXY

Apenas os arquivos "ecoli_600x_F3.csfasta" e "ecoli_600x_F3.qual" foram entregues ao processamento pela ferramenta *SOLiD(TM) denovo tool for FRAGMENT library*. O campo *Expected length of sequenced (or enriched) DNA region* (ou "Tamanho esperado da região de DNA sequenciada (ou enriquecida)") foi deixado com o valor padrão "**4600000**", por se tratar de *E. coli*, e, no campo *Number of CPUs/Cores*, foi selecionada a opção referente ao número "**16**", equivalente aos 16 núcleos de processamento do servidor.

A ferramenta especialmente criada para a instância local *LASZLO @ GALAXY*, nesta fase do projeto, foi programada para gerar dois arquivos resultantes: um arquivo de *contigs* em formato FASTA e um outro arquivo de estatísticas básicas sobre a montagem realizada. Entretanto, para fins de padronização da demonstração dos resultados deste trabalho, a ferramenta *assemblystats*, mais completa, foi empregada sobre o arquivo de *contigs* gerado. A Tabela 5.3 mostra os resultados obtidos.

Tabela 5.3 - Informações do relatório produzido pela ferramenta *assemblystats* em relação à montagem *de novo* da biblioteca de fragmentos de *E. coli* a partir de dados de SOLiD™.

Statistics for contig lengths	
Min contig length	180
Max contig length	9,845
Mean contig length	1481.98
Standard deviation of contig length	1201.04
Median contig length	1,134
N50 contig length	2,148
Statistics for numbers of contigs	
Number of contigs	3,248
Number of contigs >= 1kb	1,792
Number of contigs in N50	731
Statistics for bases in the contigs	
Number of bases in all contigs	4,813,466
Number of bases in contigs >= 1kb	4,002,148
GC Content of contigs	51.42%
Simple Dinucleotide repeats	
Number of contigs with over 70% dinucleotide repeats	0.00% (0 contigs)
AT	0.00% (0 contigs)
CG	0.00% (0 contigs)
AC	0.00% (0 contigs)
TG	0.00% (0 contigs)
AG	0.00% (0 contigs)
TC	0.00% (0 contigs)
Simple mononucleotide repeats	
Number of contigs with over 50% mononucleotide repeats	0.00% (0 contigs)
AA	0.00% (0 contigs)
TT	0.00% (0 contigs)
CC	0.00% (0 contigs)
GG	0.00% (0 contigs)

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Para fins ilustrativos a respeito dos números da tabela anterior, a Figura 5.34 exibe o histograma gerado pela ferramenta *assemblystats*, referente aos tamanhos de *contigs* obtidos na montagem.

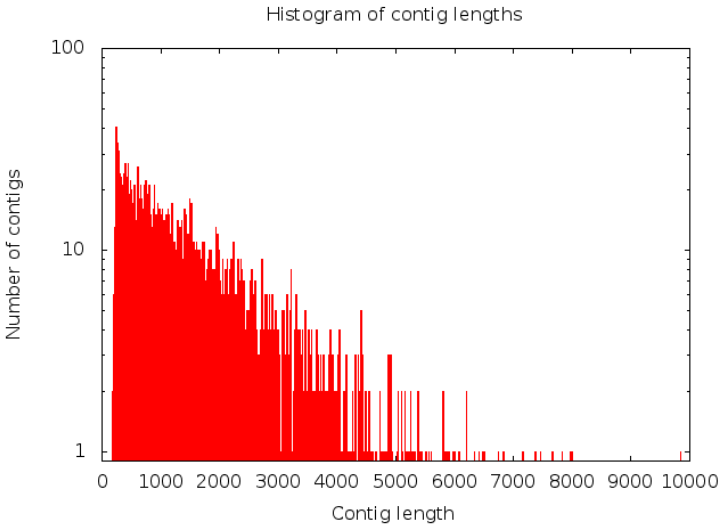


Figura 5.34 - Histograma dos tamanhos de *contigs* gerados na montagem da biblioteca de fragmentos de *E. coli* a partir de dados de SOLiD™.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.7.4 Experimento com dados de SOLiD™ usando biblioteca MATE-PAIRED no protótipo LASZLO @ GALAXY

No caso deste experimento, os quatro arquivos disponíveis da corrida de sequenciamento, "ecoli_600x_F3.csfasta", "ecoli_600x_F3.qual", "ecoli_600x_R3.csfasta" e "ecoli_600x_R3.qual", foram entregues à ferramenta *SOLiD(TM) denovo tool for MATE-PAIRED library*. O campo *Expected length of sequenced (or enriched) DNA region* foi deixado com o valor padrão "**4600000**", por se tratar de *E. coli*, e, no campo *Number of CPUs/Cores*, foi selecionada a opção referente ao número "**16**", equivalente aos 16 núcleos de processamento do servidor. Além disso, seguindo a informação presente na documentação da Applied Biosystems, a respeito das características da biblioteca de *E. coli*, os campos *Estimate of insert length* (ou "Estimativa para o tamanho do inserto") e *Estimate of variance of the insert length* (ou "Estimativa para a variância do tamanho do inserto") foram preenchidos, respectivamente, com os valores "**1200**" e "**300**".

A ferramenta especialmente criada para a instância local *LASZLO @ GALAXY*, nesta fase do projeto, foi programada para gerar quatro arquivos resultantes: um arquivo de *contigs* em formato FASTA e seu correspondente arquivo de estatísticas básicas sobre a montagem realizada e um arquivo de *scaffolds*, também em formato FASTA, devidamente acompanhado por seu arquivo de informações estatísticas básicas. Entretanto, conforme dito, para fins de padronização da demonstração dos resultados deste trabalho, a ferramenta *assemblystats*, mais

completa, foi empregada sobre os arquivos de *contigs* e *scaffolds* gerados. A Tabela 5.4 mostra os resultados obtidos.

Tabela 5.4 - Informações do relatório produzido pela ferramenta *assemblystats* em relação à montagem *de novo* da biblioteca *MATE-PAIRED* de *E. coli* a partir de dados de SOLiD™.

Statistics for contig lengths	***Contigs***	***Scaffolds***
Min contig length	101	112
Max contig length	61,432	303,759
Mean contig length	4763.39	12122.27
Standard deviation of contig length	6762.25	45728.24
Median contig length	1,867	242
N50 contig length	11,013	191,315
Statistics for numbers of contigs		
Number of contigs	949	373
Number of contigs >= 1kb	544	57
Number of contigs in N50	120	10
Statistics for bases in the contigs		
Number of bases in all contigs	4,520,454	4,521,606
Number of bases in contigs >= 1kb	4,384,246	4,428,096
GC Content of contigs	50.74%	50.73%
Simple Dinucleotide repeats		
Number of contigs with over 70% dinucleotide repeats	0.00% (0 contigs)	0.00% (0 contigs)
AT	0.00% (0 contigs)	0.00% (0 contigs)
CG	0.00% (0 contigs)	0.00% (0 contigs)
AC	0.00% (0 contigs)	0.00% (0 contigs)
TG	0.00% (0 contigs)	0.00% (0 contigs)
AG	0.00% (0 contigs)	0.00% (0 contigs)
TC	0.00% (0 contigs)	0.00% (0 contigs)
Simple mononucleotide repeats		
Number of contigs with over 50% mononucleotide repeats	0.00% (0 contigs)	0.00% (0 contigs)
AA	0.00% (0 contigs)	0.00% (0 contigs)
TT	0.00% (0 contigs)	0.00% (0 contigs)
CC	0.00% (0 contigs)	0.00% (0 contigs)
GG	0.00% (0 contigs)	0.00% (0 contigs)

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Para fins ilustrativos a respeito dos números da tabela anterior, as Figuras 5.35 e 5.36 exibem os histogramas gerados pela ferramenta *assemblystats*, referentes aos tamanhos de *contigs* e *scaffolds* obtidos na montagem.

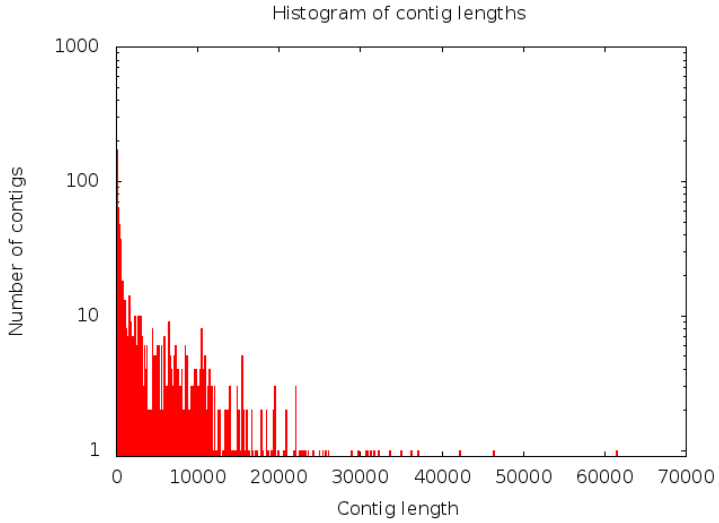


Figura 5.35 - Histograma dos tamanhos de *contigs* gerados na montagem da biblioteca *MATE-PAIRED* de *E. coli* a partir de dados de SOLiD™.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

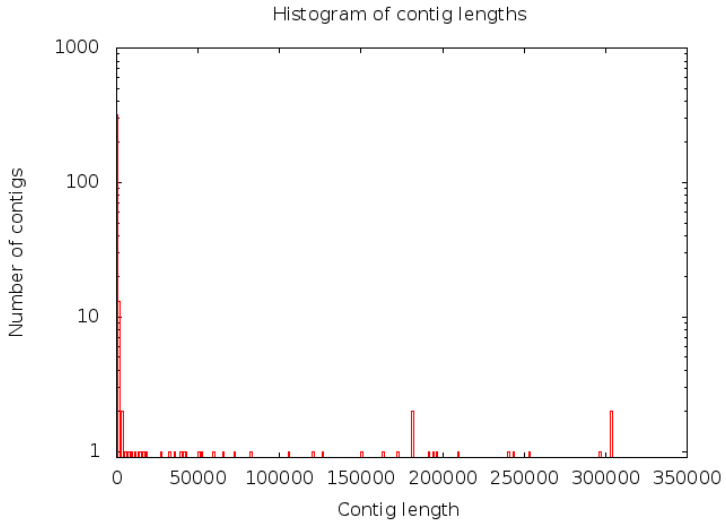


Figura 5.36 - Histograma dos tamanhos de *scaffolds* gerados na montagem da biblioteca *MATE-PAIRED* de *E. coli* a partir de dados de SOLiD™.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

A Figura 5.37 exibe os gráficos, também produzidos pela ferramenta *assemblystats*, referente às respectivas somas de tamanhos de *contigs* (*scaffolds*, para o caso do gráfico mais à direita), como uma outra forma disponível para o usuário analisar os dados produzidos.

Pode-se observar que um menor número de *contigs* (ou *scaffolds*) se faz necessário para alcançar a soma total de bases que compreendem a montagem, à medida em que são usadas leituras pareadas em lugar de fragmentos únicos e, também, quando há a possibilidade de ser utilizada uma etapa de resolução de *scaffolds*, tal como proporcionado pela ferramenta ASID do *pipeline*.

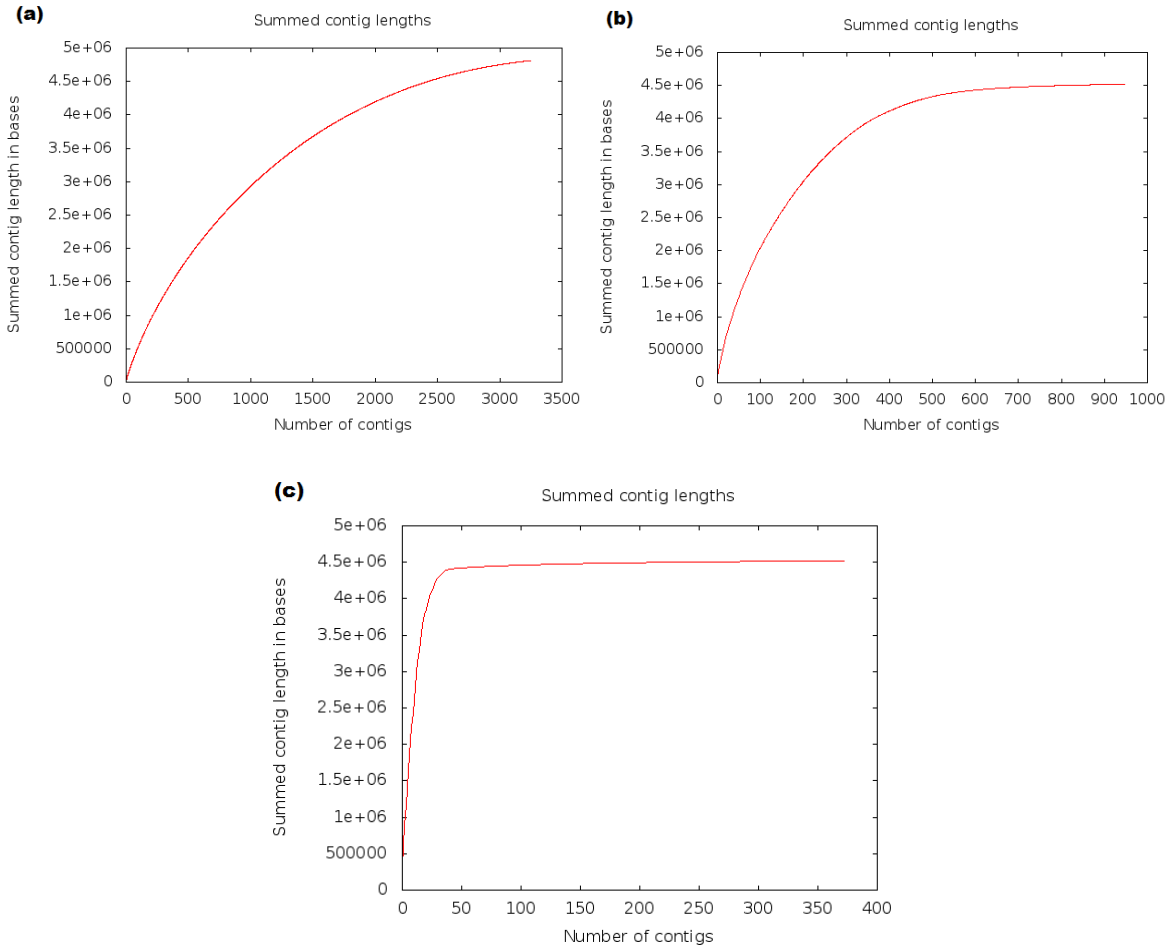


Figura 5.37 - Somas dos tamanhos de *contigs* e *scaffolds* gerados nas montagens das bibliotecas de fragmentos únicos (a) e *MATE-PAIRED* de *E. coli* (b) para *contigs* e (c) para *scaffolds* a partir de dados de SOLiD™.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Na documentação da Applied Biosystems (2010), sobre a ferramenta *SOLiD™ de novo accessory tools 2.0*, é informado que, em média, o tamanho do *contig* N50 varia de:

- 2 a 3,5 kpb para montagens a partir de bibliotecas de fragmentos longas de 50 pb;
- 3 a 4 kpb para montagens a partir de bibliotecas *paired-end* de 50 x 25 pb;
- 15 a 20 kpb para montagens a partir de bibliotecas *mate-pair* de 50 x 50 pb. Ainda para leituras do tipo *mate-pair*, o tamanho do *scaffold* N50 varia de 0,2 a 1 Mpb.

Assim, os resultados obtidos podem ser considerados como aproximados ao que foi obtido pela Applied.

5.2.7.5 Paralelização da etapa SAET no protótipo LASZLO @ GALAXY

Como exemplo de resultado da paralelização de tarefas executadas pela etapa SAET, a Figura 5.38 é exibida. Nela, pode-se observar os 16 núcleos, dos dois processadores do servidor, sendo exigidos, simultaneamente, em quase 100%, pelo rol de processos referentes à ferramenta, comprovando o funcionamento do parâmetro *Number of CPUs/Cores* sugerido na interface gráfica para o usuário.

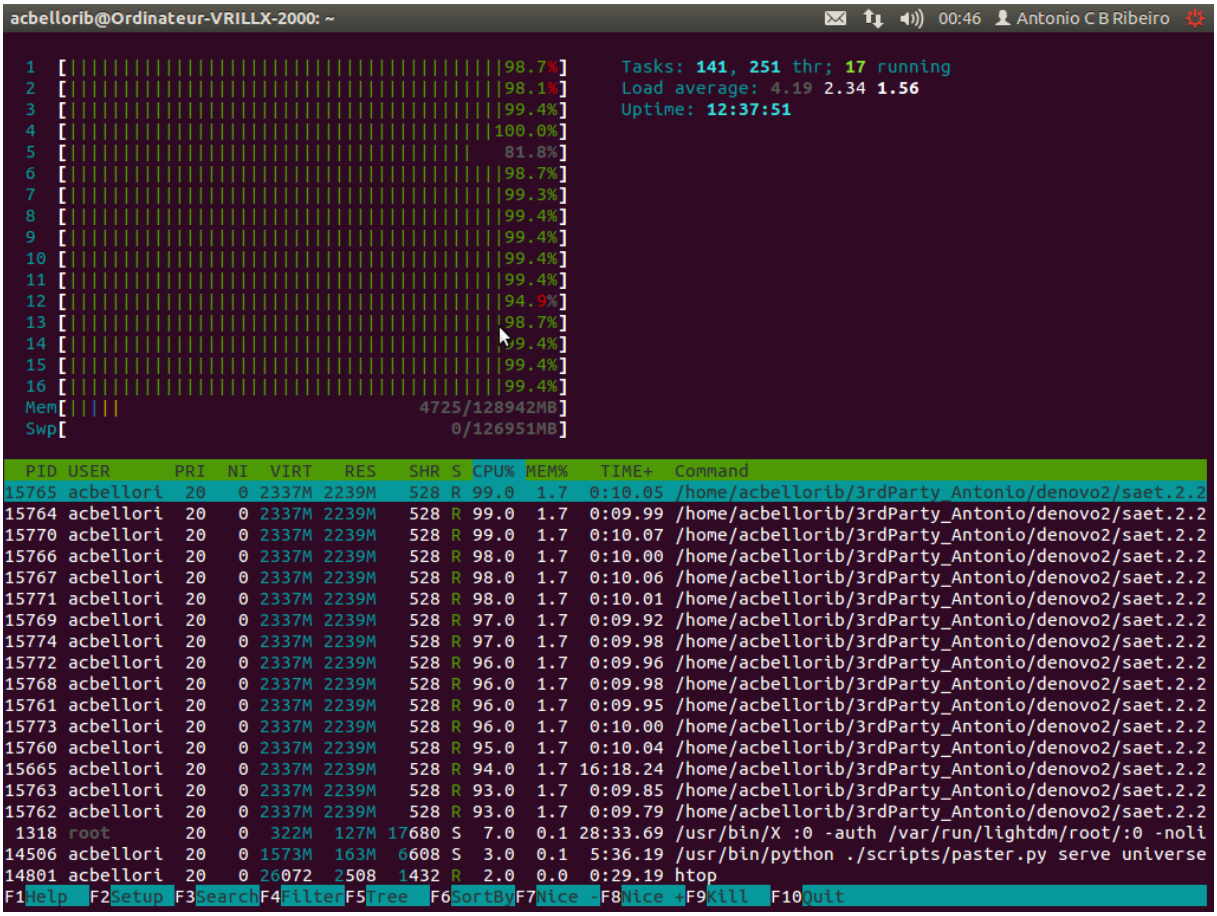


Figura 5.38 - Paralelização da etapa interna SAET no servidor do protótipo.

5.2.7.6 Monitoração do consumo de memória durante as montagens de novo a partir de dados de SOLiD™

A execução da ferramenta *SOLiD(TM) denovo tool for FRAGMENT library*, para o conjunto de dados de *E. coli*, consumiu cerca de 24 GB de memória nos maiores picos (Figura 5.39), conforme capturado pela ferramenta MRTG.

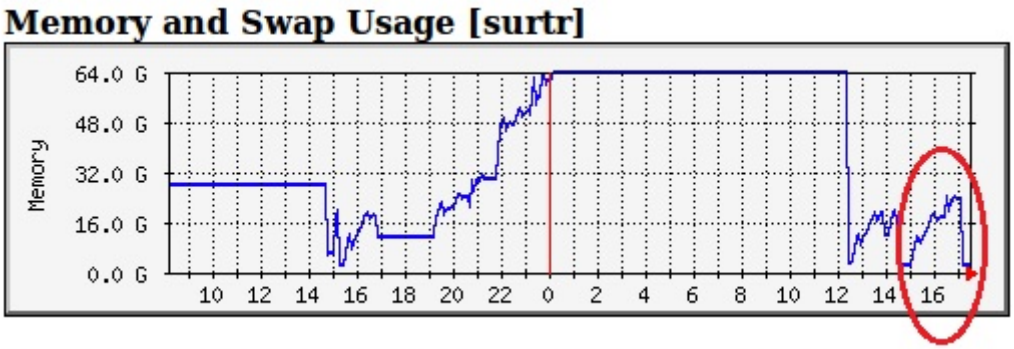


Figura 5.39 - Consumo de memória do servidor durante a montagem *de novo* dos dados de SOLiD™ para a biblioteca de fragmentos únicos de *E. coli*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

No caso da ferramenta *SOLiD(TM) denovo tool for MATE-PAIRED library*, o consumo alcançou cerca de 44 GB de memória nos maiores picos (Figura 5.40).

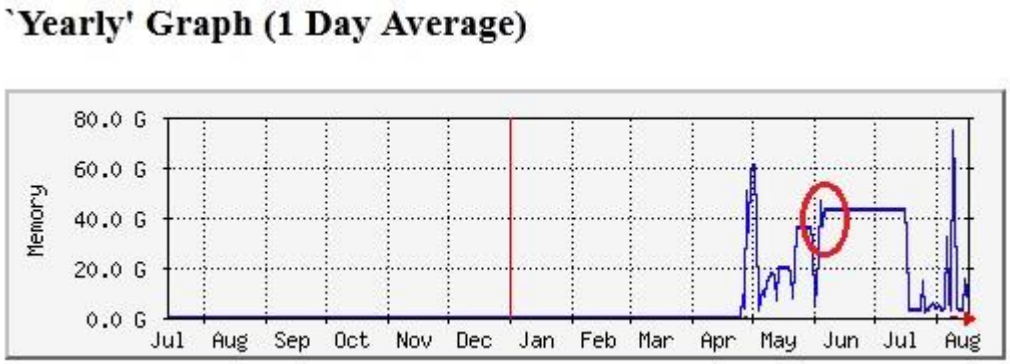


Figura 5.40 - Consumo de memória do servidor durante a montagem *de novo* dos dados de SOLiD™ para a biblioteca *MATE-PAIRED* de *E. coli*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.8. Sequência de etapas do fluxo de trabalho básico para montagem *de novo* a partir de dados de 454 no protótipo *LASZLO @ GALAXY*

Seguindo a mesma linha de atuação da alteração do sistema STINGRAY, o programa MIRA foi a opção escolhida para a montagem *de novo* de dados provenientes da tecnologia 454. A escolha foi facilitada pela existência de um *wrapper* disponível na comunidade de desenvolvimento GALAXY, de autoria de Peter Cock (James Hutton Institute, Reino Unido), o que imediatamente nos permitiu obedecer ao correspondente critério de desenvolvimento e, também, pelo fato do pacote ser versátil, capaz de trabalhar com dados originados em várias tecnologias NGS (454, Solexa/Illumina, Ion Torrent, PacBio) e, também, com a tecnologia Sanger. Assim, mais um critério de desenvolvimento pôde ser atendido. O programa MIRA, na sua versão original independente, é um montador/alinhador para projetos de genomas completos e de ESTs, especificamente desenvolvido para lidar com elementos repetitivos em

dados genômicos e com SNPs em dados de EST, com grande acurácia. Por trabalhar com diversas tecnologias, permite a execução de montagens híbridas, ou seja, aquelas em que dados de diferentes tecnologias são trabalhados em conjunto (Chevreux, 2010).

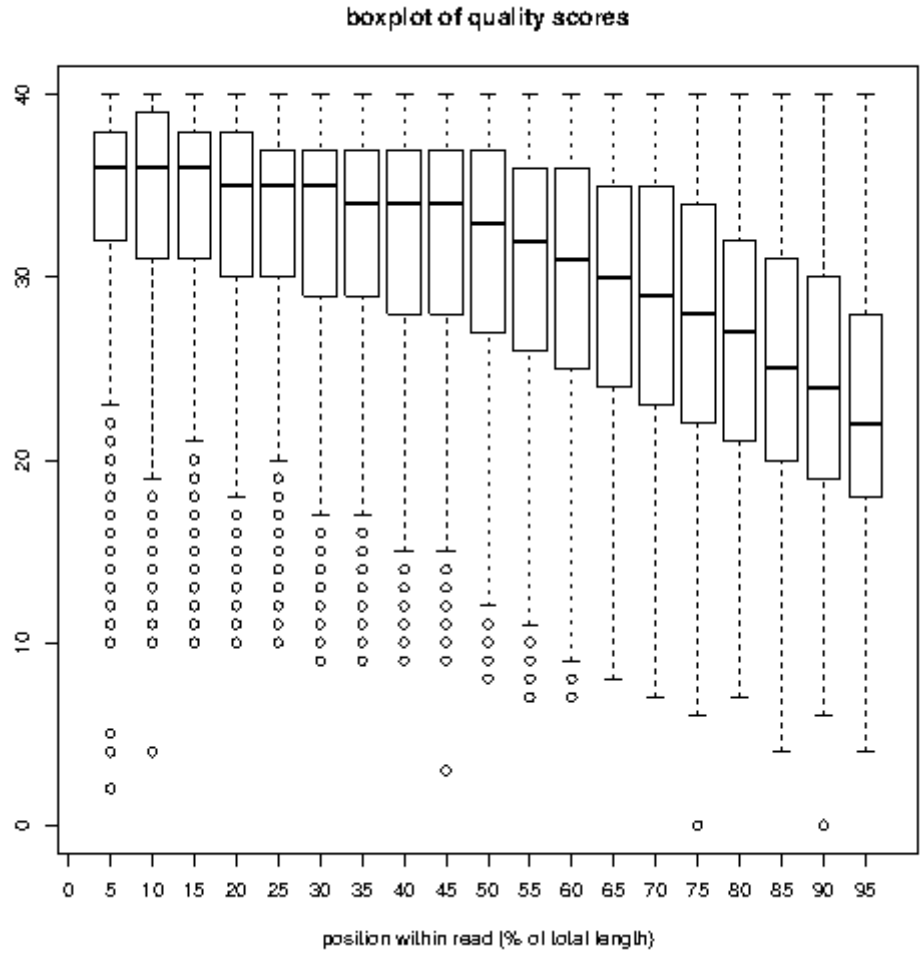
5.2.8.1 Instalação do pacote MIRA

Para prover a devida funcionalidade ao *wrapper* obtido na comunidade de desenvolvedores GALAXY, foi instalada a versão 3.4.0 do pacote MIRA, seguindo-se as instruções de administração e configuração de instância local para NGS e, em seguida, as instruções fornecidas com a documentação do programa[86].

5.2.8.2 Carga e avaliação de qualidade do arquivo de 454

Após ser inserido por intermédio da ferramenta *Upload File*, o formato SFF de 454 já tem condições de ser manipulado pela instância local GALAXY, empregando-se a ferramenta já embutida *SFF converter*, da guia *Convert Formats*. Na condição padrão de parametrização dessa ferramenta, o formato SFF é convertido em três outros arquivos: um arquivo das leituras em formato FASTA, um arquivo com os valores de qualidade das leituras, também em formato FASTA, e um arquivo com informações referentes às leituras, em formato XML. De posse do arquivo com os valores de qualidade, este foi submetido à uma breve avaliação pela ferramenta *Build base quality distribution*, do bloco *ROCHE-454 DATA*, *guia NGS: QC and manipulation*, já integrantes da plataforma da instância local. A Figura 5.41 exibe os resultados obtidos.

[86] Disponível em http://sourceforge.net/projects/mira-assembler/files/MIRA/stable/



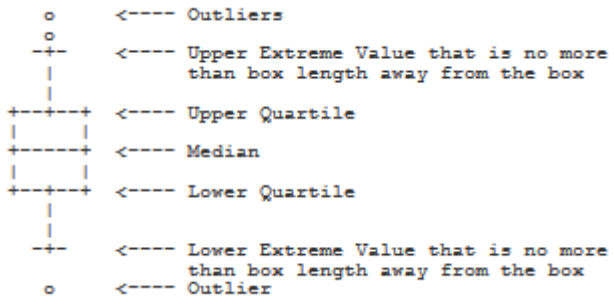


Figura 5.41 - Resultado da avaliação dos valores de qualidade de 454 contidos no arquivo "SRR066482 QUAL". A ferramenta *Build base quality distribution*, a qual também pode ser usada para avaliar dados de SOLiD™ e Illumina, exibe no eixo horizontal, para o caso específico de 454, a posição relativa (em %) das qualidades das leituras, uma vez que a tecnologia produz leituras com diferentes tamanhos.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

As leituras apresentavam bons valores de qualidade (mediana acima de 20, na escala PHRED) ao longo de toda a faixa de posições relativas do gráfico, então, optou-se por processar os arquivos, em um primeiro momento, sem nenhuma operação de poda de bases ou ajuste de qualidade.

5.2.8.3 A montagem de novo a partir de dados de 454 com o programa MIRA no protótipo LASZLO @ GALAXY

A Figura 5.42 exibe a tela do *wrapper* do programa MIRA, após a sua devida instalação, configuração e organização dentro do bloco de ferramentas *NGS: DE NOVO ASSEMBLY TOYS*, do módulo *NGS: LASZLO's Sandbox*.

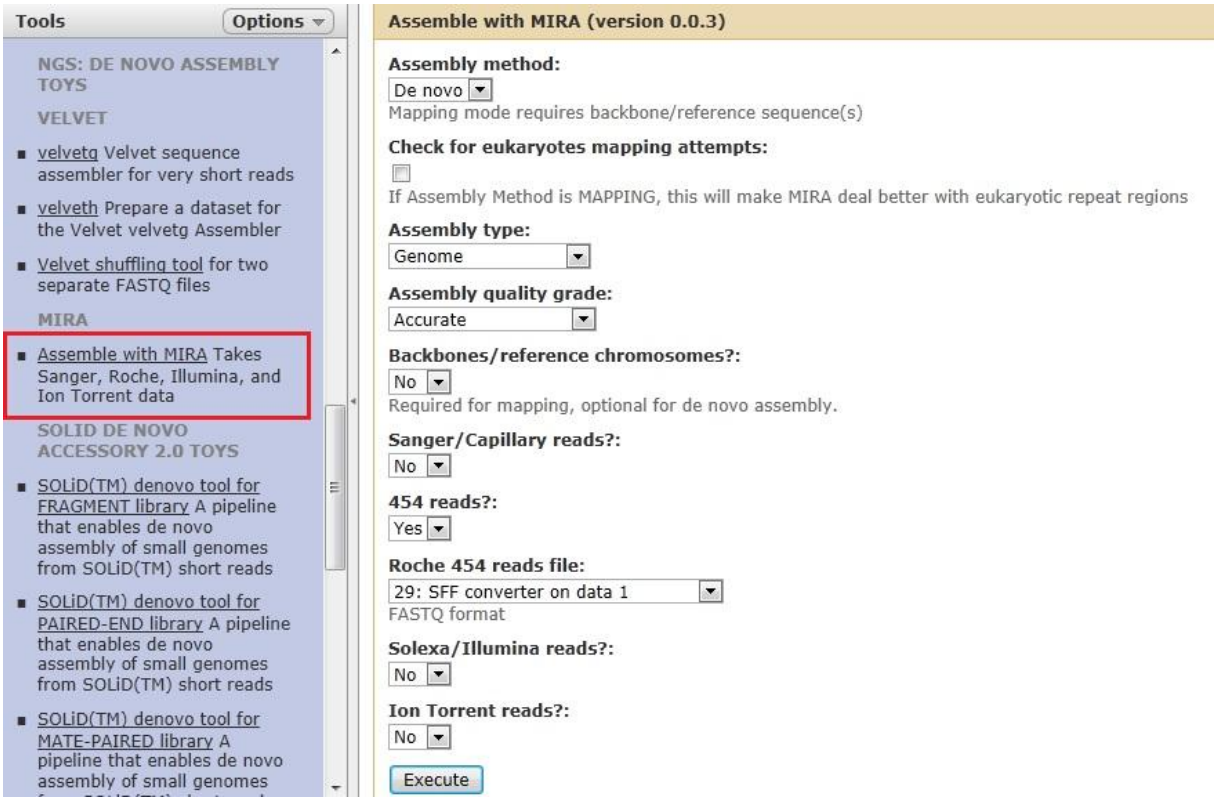


Figura 5.42 - Interface gráfica da ferramenta *Assemble with MIRA*, na instância local *LASZLO @ GALAXY*, e detalhe de sua guia de acesso no painel de ferramentas, no bloco *NGS: DE NOVO ASSEMBLY TOOLS* do módulo *NGS: LASZLO's Sandbox*.

Tal *wrapper* foi concebido de maneira a aceitar arquivos somente no formato FASTQ. Assim, a ferramenta *SFF converter* foi uma vez mais usada, sobre os dados originais de 454, para possibilitar a aquisição de um arquivo no devido formato FASTQ. Para isso, seus parâmetros originais foram alterados da seguinte maneira:

(1) Guia *Convert formats* → Ferramenta *SFF converter*;
(2) No menu suspenso *Completely remove ends with low qual and/or adaptor sequence* (ou "Remover completamente extremidades com baixa qualidade e/ou sequências de adaptadores"), a linha referente à opção "**Yes**" (ou "Sim") foi selecionada;
(3) A caixa de seleção *Do you want FASTQ file instead of FASTA + FASTA quality file?* (ou "Você deseja um arquivo FASTQ ao invés de um conjunto FASTA + FASTA com valores de qualidade?"), foi marcada;
(4) Botão *Execute* pressionado.

Com isso, o arquivo FASTQ necessário foi obtido. A Figura 5.43 exibe a ferramenta *SFF converter* com seus parâmetros alterados de maneira a produzi-lo.

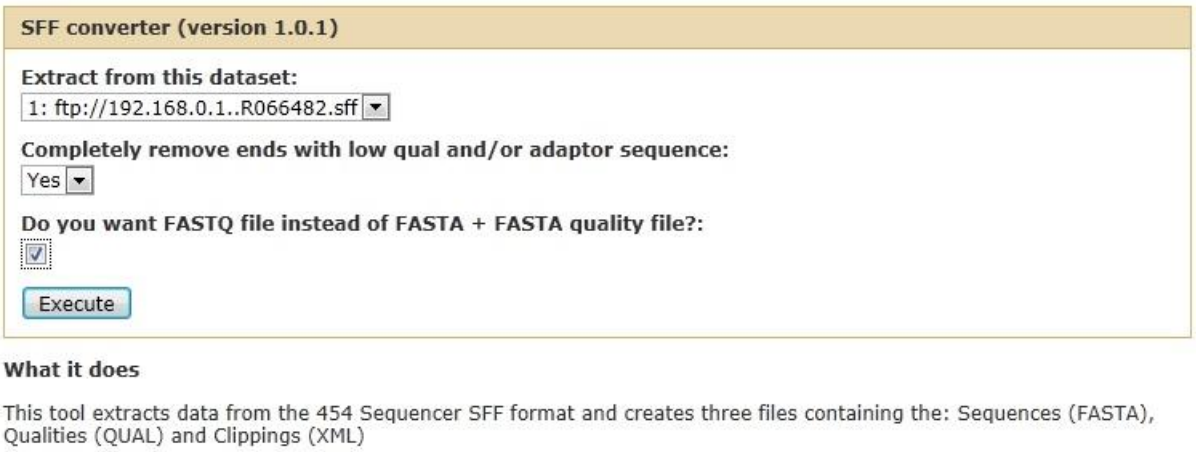


Figura 5.43 - Interface gráfica da ferramenta *SFF converter* com parâmetros alterados para a geração do arquivo FASTQ necessário como entrada para o *wrapper* do programa montador MIRA.

A ferramenta *Assembly with MIRA* foi, então, alimentada com esse arquivo FASTQ correspondente aos dados de 454 SRR066482, a partir da indicação de que as leituras se tratavam de produtos da tecnologia 454 (linha "**Yes**" selecionada no campo *454 reads?* (ou "Leituras de 454?")). Além disso, os outros parâmetros foram mantidos com valores padrão, basicamente informando que a montagem seria do tipo *de novo* (linha "**De novo**" selecionada no campo *Assembly method* (ou "Método de montagem")), de genoma (linha "**Genome**" selecionada no campo *Assembly type* (ou "Tipo de montagem")) e com qualidade acurada (linha "**Accurate**" selecionada no campo *Assembly quality grade* (ou "Grau de qualidade da montagem")).

5.2.8.4 Captura de informações estatísticas sobre os resultados da montagem de novo a partir de dados de 454 no protótipo LASZLO @ GALAXY

O arquivo de *contigs* resultante foi, em seguida, avaliado com a ferramenta *assemblystats*. A Tabela 5.5 mostra os resultados obtidos.

Tabela 5.5 - Informações do relatório produzido pela ferramenta *assemblystats* em relação à montagem *de novo* dos dados de 454 referentes ao organismo *P. papatasi*.

Statistics for contig lengths	
Min contig length	61
Max contig length	15,819
Mean contig length	664.74
Standard deviation of contig length	406.69
Median contig length	600
N50 contig length	716
Statistics for numbers of contigs	
Number of contigs	6,961
Number of contigs >= 1kb	741
Number of contigs in N50	2,279
Statistics for bases in the contigs	
Number of bases in all contigs	4,627,237
Number of bases in contigs >= 1kb	1,034,626
GC Content of contigs	34.06%
Simple Dinucleotide repeats	
Number of contigs with over 70% dinucleotide repeats	0.00% (0 contigs)
AT	0.00% (0 contigs)
CG	0.00% (0 contigs)
AC	0.00% (0 contigs)
TG	0.00% (0 contigs)
AG	0.00% (0 contigs)
TC	0.00% (0 contigs)
Simple mononucleotide repeats	
Number of contigs with over 50% mononucleotide repeats	0.00% (0 contigs)
AA	0.00% (0 contigs)
TT	0.00% (0 contigs)
CC	0.00% (0 contigs)
GG	0.00% (0 contigs)

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Para fins ilustrativos a respeito dos números da tabela anterior, a Figura 5.44 exibe o histograma gerado pela ferramenta *assemblystats*, referente aos tamanhos de *contigs* obtidos na montagem.

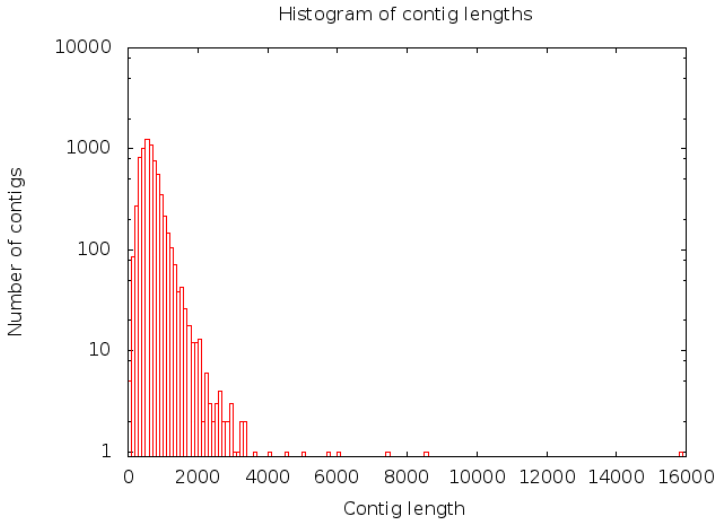


Figura 5.44 - Histograma dos tamanhos de *contigs* gerados na montagem dos dados de 454 referentes ao organismo *P. papatasi*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Já a Figura 5.45 exibe a soma dos tamanhos dos *contigs* produzidos.

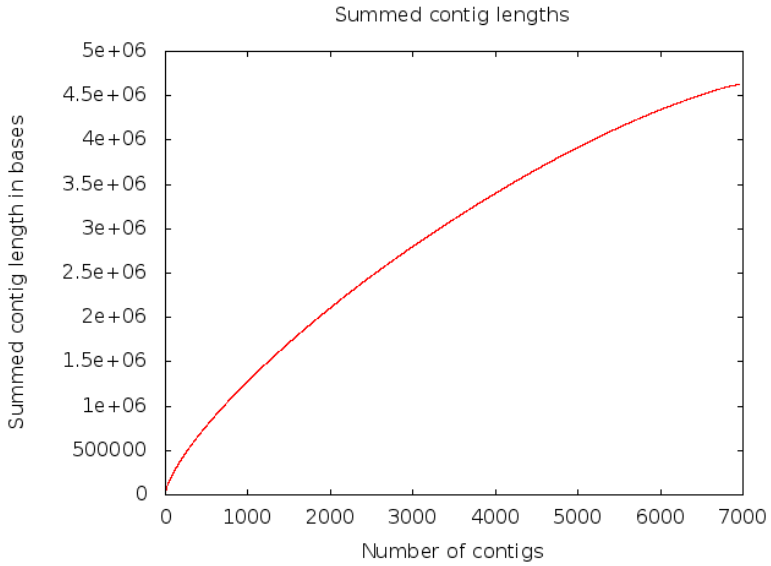


Figura 5.45 - Soma dos tamanhos de *contigs* para a montagem *de novo* dos dados de sequenciamento de *P. papatasi*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.8.5 Monitoração do consumo de memória durante a montagem de novo a partir de dados de 454 no protótipo LASZLO @ GALAXY

Para a montagem *de novo* dos dados de 454 utilizados, o consumo de memória, capturado pela ferramenta MRTG, alcançou um pico de 32 GB, conforme mostrado na Figura 5.46.

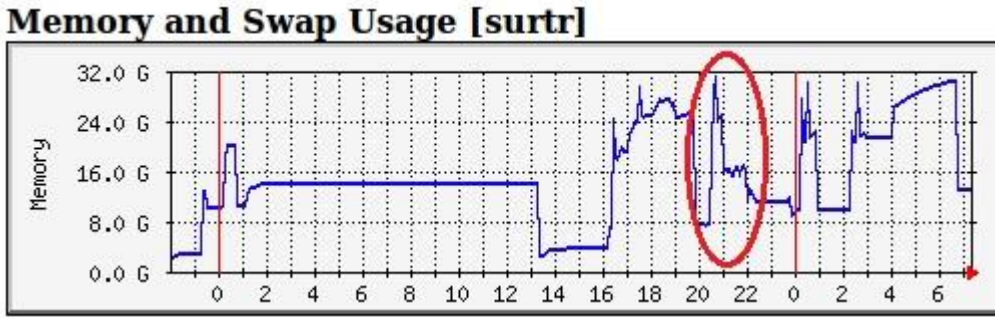


Figura 5.46 - Consumo de memória do servidor durante a montagem *de novo* dos dados de 454 para *P. papatasi*.
Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.2.9. Sugestões de fluxos de trabalho básicos para montagem com auxílio de genoma de referência a partir de dados de ABI SOLiD™ e 454 no protótipo *LASZLO @ GALAXY*

Tal como no caso da tecnologia Solexa/Illumina, a instância local GALAXY já vem com *wrappers* prontos para a realização das tarefas usuais de alinhamento/mapeamento de genomas para ABI SOLiD™ e 454, dentro da guia *NGS: Mapping*. Para ABI SOLiD™, por exemplo, existe a possibilidade de se trabalhar com os mapeadores Bowtie, BWA ou PerM, sendo necessária, apenas, a devida instalação e configuração do(s) pacote(s) correspondentes, tal como já foi aqui comentado. Da mesma maneira, para dados de 454, o mapeamento pode ser feito com o programa LASTZ (Harris, 2007). De fato, sugestões de roteiros básicos também são fornecidas na comunidade GALAXY para o mapeamento de dados de ABI SOLiD™[87] e 454[88], também sob a forma de tutoriais. A combinação dos passos descritos nesses tutoriais, com um roteiro do tipo usado neste trabalho (para mapeamento dos dados de Solexa/Illumina), pode servir de base para a elaboração de fluxos de trabalhos básicos apropriados para essas tarefas. Por essa razão, em vez da realização e detalhamento de experimentos específicos para essas duas tecnologias, que tão somente iriam reproduzir os passos já exemplificados pelos desenvolvedores da plataforma GALAXY, optou-se por resumir as eventuais etapas que poderiam compor os fluxos desse tipo, tal como se segue:

- ABI SOLiD™:
    - carga dos arquivos originais SOLiD™ e avaliação de qualidade por meio das ferramentas já exemplificadas neste trabalho;
    - uma vez que os mapeadores disponíveis para SOLiD™, na plataforma

[87] Tutoriais 8 - *SOLiD Single End* (http://screencast.g2.bx.psu.edu/quickie8_solid_single_end/flow.html), 9 - *SOLiD Mate Pair* (http://screencast.g2.bx.psu.edu/quickie9_solid_mate_pair/flow.html) e 10 - *Mapping Against a Custom Reference Genome* (http://screencast.g2.bx.psu.edu/quickie10_custom_genome/flow.html).
[88] Tutorial 15 - *454 Mapping Single End* (http://screencast.g2.bx.psu.edu/quickie_15_lastz_frag/flow.html).

GALAXY, só trabalham com o formato FASTQ, conversão dos arquivos originais de SOLiD™ para FASTQ, por meio da ferramenta *Convert SOLiD output to fastq* (ou "Converter saída SOLiD para fastq"), pertencente ao bloco de ferramentas *AB-SOLID DATA* da guia *NGS: QC and manipulation*;

  - mapeamento das leituras com um dos mapeadores disponíveis para a tecnologia SOLiD™ (tal como ilustrado, por exemplo, no mapeamento dos dados de Illumina);
  - uso das ferramentas SAMTools para as devidas filtragens e manipulações dos dados para as etapas de análise subsequentes.

- 454:
  - carga dos arquivos originais 454 e avaliação de qualidade por meio das ferramentas já exemplificadas neste trabalho;
  - mapeamento do arquivo original de 454, em formato FASTA, com a ferramenta LASTZ;
  - uso das ferramentas SAMTools ou da combinação de ferramentas dos grupos *Statistics* (ou "Estatísticas"), *Filter and Sort* (ou "Filtrar e classificar") e *Join, Subtract and Group* (ou "Unir, subtrair e agrupar"), para as devidas filtragens e manipulações dos dados para as etapas de análise subsequentes.

5.3. O módulo *NGS: LASZLO's Sandbox*

As ferramentas elaboradas para estender a funcionalidade da instância local GALAXY podem ser consideradas como resultados práticos deste trabalho, uma vez que boa parte não é contemplada pela plataforma pública e, tampouco, no pacote que pode ser obtido para instalação e configuração de uma instância local. Em resumo, portanto, o módulo *NGS: LASZLO's Sandbox* foi provido dos seguintes blocos de ferramentas nesta versão do protótipo:

- Bloco *MORE SAMTOOLS TOYS*, ferramenta *SAMTools pileup-to-fastQ converter*, implementada para a realização de conversão do formato pileup para o FASTQ;
- Bloco *OTHER TOYS*, ferramenta *Extract region tool*, implementada para a extração, dentro dos dados de montagem, de uma determinada região de interesse que tenha sido identificada com a ferramenta *NCBI BLAST+ blastn* da plataforma;
- Bloco *OTHER TOYS*, ferramenta *assemblystats*, adaptada a partir de *wrapper* pré-existente para levantamento de dados estatísticos sobre montagem *de novo*;
- Bloco *NGS: DE NOVO ASSEMBLY TOYS*, ferramenta *Velvet shuffling tool*,

implementada para fazer a preparação dos dados pareados em formato FASTQ para a utilização pelo programa Velvet;

- Bloco *NGS: DE NOVO ASSEMBLY TOYS*, ferramentas *velveth* e *velvetg*, já integrantes da instância local, tiveram suas guias de acesso reorganizadas para que, pela interface gráfica, ficassem atreladas a este bloco de ferramentas, facilitando a sua devida localização pelo usuário;
- Bloco *SOLID DE NOVO ACCESSORY 2.0 TOYS*, ferramentas *SOLiD(TM) denovo tool for FRAGMENT library*, *SOLiD(TM) denovo tool for PAIRED-END library* e *SOLiD(TM) denovo tool for MATE-PAIRED library*;
- Bloco *NGS: DE NOVO ASSEMBLY TOYS*, ferramenta *Assembly with MIRA*, adaptada a partir de *wrapper* pré-existente para montagem *de novo* de dados de 454;

Os *wrappers* originalmente adaptados ou desenvolvidos podem ser encontrados no Apêndice F deste trabalho.

5.4. Sequências de *Leishmania amazonensis*

Durante a execução do projeto, duas montagens básicas usando os fluxos de trabalho propostos e dados de sequenciamento reais do organismo *Leishmania amazonensis* foram realizadas. Os resultados dessas montagens foram disponibilizados ao Laboratório de Biologia Computacional e Sistemas, do Instituto Oswaldo Cruz, e também foram usados como auxiliares em uma primeira avaliação do genoma do referido organismo (Tschoeke et al., 2011). Com isso, o objetivo de produzir esboços do genoma do organismo *Leishmania amazonensis*, para futuras análises, pôde ser concretizado.

5.5. Resultados obtidos com a combinação de ferramentas *NCBI BLAST+ blastn* e ferramenta *Extract region tool* do módulo *NGS: LASZLO's Sandbox*

A combinação dessas ferramentas foi idealizada para auxiliar o profissional da área de ciências da vida, cujo conhecimento de informática esteja ainda mais limitado a operações do tipo "apontar e clicar", na obtenção de respostas para questões biológicas do tipo: "a partir dos arquivos com dados de sequenciamento NGS para um determinado organismo, é possível encontrar regiões no genoma montado que sejam similares à de um gene conhecido em outra espécie próxima? E, na eventualidade de encontrar essa região similar, é possível "pinçá-la",

dos dados de montagem, acrescida de faixas limitadas de nucleotídeos *upstream* e *downstream* adjacentes para auxiliar, por exemplo, o desenho de *primers* para a realização de outras análises pela biologia experimental?".

Para verificar se tais questões poderiam ser respondidas na prática, as sequências de genes já identificados em outras espécies de leishmania foram confrontadas com o resultado do esboço de montagem obtido para a *Leishmania amazonensis*. Antes disso, porém, a título de um simples controle, foi realizada uma consulta com um gene já conhecido da própria espécie, de número de acesso AY370533[89]. Tal consulta, com a ferramenta *NCBI BLAST+ blastn* gerou o seguinte resultado resumido (Tabela 5.6)[90].

Tabela 5.6 - Relatório resumido da ferramenta *NCBI BLAST+ blastn* para o gene conhecido de *Leishmania amazonensis* de número de acesso AY370533 em relação aos dados de montagem de *Leishmania amazonensis*.

Query	Length (kbp)	Subject	Length (kbp)	Identities	Gaps	Strand
AY370533.1 (controle)	1545	Draft_Lamaz_19	655043	1544/1545 (99%)	0/1545 (0%)	Plus/Plus

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Apenas um único nucleotídeo, na sequência de consulta com 1545 pares de bases de extensão, apresentou discordância em relação à sequência "montada": a posição de número 369 da sequência do gene conhecido é ocupada por uma timina, enquanto que, na correspondente posição, de número 627362 do cromossomo 19 da sequência montada, foi encontrada uma citosina. Tal mudança, por exemplo, poderia eventualmente caracterizar a ocorrência de um SNP.

Tendo sido obtido sucesso na consulta dessa sequência de gene conhecido, o próximo passo para validar a combinação das ferramentas seria simular a necessidade de capturar, na sequência montada, a região correspondente ao gene, acrescida de regiões *upstream* e *downstream* adjacentes. Foram, então, anotadas as posições inicial (626994) e final (628538), no cromossomo da sequência montada, que perfaziam a faixa equivalente à do gene, a partir das informações do relatório da ferramenta *NCBI BLAST+ blastn*. Simulando-se, então, a necessidade de obter 1000 pares de bases anteriores e posteriores a essas coordenadas, foram inseridos os valores 625994 e 629538, para a devida extração da região de interesse correspondente[91], ou seja, a região equivalente ao gene conhecido, flanqueada, em cada uma

[89] Disponível em http://www.ncbi.nlm.nih.gov/nuccore/AY370533.
[90] No Apêndice G deste trabalho pode ser encontrado, como exemplo, o relatório completo da referida consulta.
[91] No Apêndice H deste trabalho pode ser encontrada, como exemplo, a referida região "extraída" da sequência montada de *Leishmania amazonensis*.

das extremidades, por 1000 bases adicionais. Isso comprovou o funcionamento básico proposto para a ferramenta *Extract region tool* do módulo *NGS: LASZLO's Sandbox*.

A partir daí, a mesma estratégia foi empregada para auxiliar a profissional Adriana Degrossoli em suas pesquisas, durante o período de execução do projeto. Foram realizadas consultas (e respectivas "extrações" de regiões de interesse adicionadas de 1000 bases em cada uma das extremidades) envolvendo outros quatro genes conhecidos de outras espécies de leishmania, cujos resultados[92] de consulta são descritos abaixo (Tabela 5.7).

Tabela 5.7 - Relatório resumido da ferramenta *NCBI BLAST+ blastn* para quatro genes conhecidos de outras espécies de leishmania em relação aos dados de montagem de *Leishmania amazonensis*.

Query	**Length** (kbp)	**Subject**	**Length** (kbp)	**Identities**	**Gaps**	**Strand**
Gene conhecido #1 de *L. major*	~1	Draft_Lamaz_15	574813	(93%)	(0%)	Plus/Minus
Gene conhecido #1 de *L. mexicana*	~2	Draft_Lamaz_15	574813	(98%)	(0%)	Plus/Plus
Gene conhecido #2 de *L. mexicana*	~0,9	Draft_Lamaz_08	1671952	(99%)	(0%)	Plus/Plus
Gene conhecido #3 de *L. mexicana*	~0,5	Draft_Lamaz_08	1671952	(99%)	(0%)	Plus/Plus

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

Como um dos resultados práticos da aplicação das ferramentas citadas, um par de *primers* pôde ser desenhado, a partir da primeira sequência extraída (resultante da comparação do gene conhecido de *Leishmania major* com o rascunho de montagem do genoma de *Leishmania amazonensis*). Com os referidos *primers* e o DNA genômico de *L. amazonensis*, um experimento de PCR foi conduzido, o qual culminou com a obtenção de um produto de aproximadamente 1 kpb, coincidindo com o tamanho previsto teoricamente (1022 pb, no caso específico do experimento) e indicando uma boa possibilidade quanto à exatidão da sequência fornecida. O produto da reação foi separado eletroforeticamente, em gel de agarose a 1% em TAE, e visualizado com brometo de etídeo, conforme mostrado na Figura 5.47 abaixo.

[92] Para preservar os dados de pesquisa originais da profissional Adriana Degrossoli, os genes conhecidos foram rotulados de maneira genérica e os respectivos tamanhos de suas sequências foram apresentados em valores aproximados. O objetivo é, tão somente, demonstrar o alto percentual de identidade e o baixo percentual de *gaps* que foram obtidos em relação aos tamanhos das sequências envolvidas.

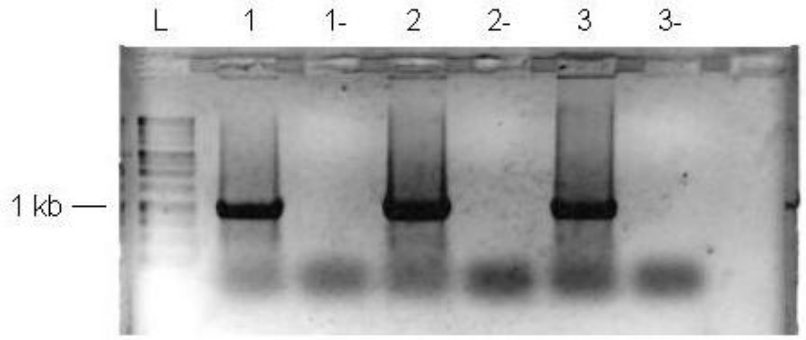


Figura 5.47 - Separação eletroforética de experimento de PCR baseado nos *primers* desenhados a partir da sequência extraída dos dados de montagem de *L. amazonensis*, pela ferramenta *Extract region tool* do módulo *NGS: LASZLO's Sandbox*, e DNA genômico de *L. amazonensis*. Legenda: L → *DNA ladder*; 1, 2 e 3 → produto da reação variando-se a concentração de $MgCl_2$; 1-, 2- e 3- → controles negativos das reações correspondentes, nas quais não foi adicionado o DNA genômico de *L. amazonensis*.
Fonte: Degrossoli, 2011. (comunicação pessoal)[93]

A título de verificação, o mesmo tipo de consulta foi realizado sobre os dados de montagem usando a abordagem *de novo*. Foram escolhidos, como dados do genoma montado, aqueles que apresentaram o maior valor N50 (16761, para o *hash length* de valor 31). Mesmo no caso desse tipo de montagem, mais dificultada pela ausência de um genoma de referência para auxiliar o processo, todas as consultas foram, da mesma forma, bem sucedidas, tal conforme mostrado na Tabela 5.8 a seguir.

[93] Dados gentilmente cedidos por Adriana Degrossoli (Laboratório de Bioquímica de Proteínas e Peptídeos, Instituto Oswaldo Cruz). Degrossoli A. Re: gene leishmania. [mensagem pessoal]. Mensagem recebida por acbellorib@gmail.com em 26 de maio de 2011.

Tabela 5.8 - Relatório resumido da ferramenta *NCBI BLAST+ blastn* para o gene conhecido de controle (de *L. amazonensis*) e os quatro genes conhecidos de outras espécies de leishmania em relação aos dados de montagem sob a abordagem *de novo* de *Leishmania amazonensis*.

Query	Length (kbp)	Subject	Length	Identities	Gaps	Strand
AY370533.1 (controle)[94]	1545	NODE_3014_length_25030_cov_21.060007	25060	1544/1545 (99%)	0/1545 (0%)	Plus/Plus
Gene conhecido #1 de *L. major*	~1	NODE_863_length_27299_cov_18.568556	27329	(93%)	(0%)	Plus/Plus
Gene conhecido #1 de *L. mexicana*	~2	NODE_863_length_27299_cov_18.568556	27329	(98%)	(0%)	Plus/Minus
Gene conhecido #2 de *L. mexicana*	~0,9	NODE_5885_length_16808_cov_18.446871	16838	(99%)	(0%)	Plus/Plus
Gene conhecido #3 de *L. mexicana*	~0,5	NODE_3126_length_3201_cov_20.154015	3231	(99%)	(0%)	Plus/Plus

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

5.6. Potencial de anotação automática dos dados gerados no protótipo *LASZLO @ GALAXY*

O desenvolvimento deste trabalho também previa a possibilidade do protótipo vir a ser utilizado como um gerador de dados ou "módulo de montagem de genomas" para outros sistemas, visando análises futuras. Com isso, foi realizado um teste básico, no sistema STINGRAY, de carga dos arquivos de *contigs* produzidos em cada montagem, de maneira a validar a alimentação correta de seu banco de dados. Em seguida, foi realizada uma análise simples de similaridade dos dados montados contra dados anotados do banco do STINGRAY, produzindo os seguintes resultados (Tabela 5.9), o que sugere um bom potencial de anotação da produção obtida no protótipo.

[94] No Apêndice I deste trabalho pode ser encontrado, como exemplo, o relatório completo da referida consulta.

Tabela 5.9 - Relatório resumido das estatísticas de projeto da plataforma STINGRAY para os arquivos de *contigs* alimentados em seu banco de dados e após rodada de análise de similaridade com a ferramenta *blastn*.

Assembly file	**Number of reads**	**Number of clusters**	**Sum of consensus length** (bp)	**Clusters avg. length** (bp)	**BLAST Cluster with hits**	**BLAST Cluster with NO hits**
L. amazonensis (Reference)	34	34	28764924	846027	34 (100%)	0 (0%)
L. amazonensis (*de novo*)	7457	7457	29873230	4006	4132 (55%)	3325 (45%)
E. coli (fragment, *de novo*)	3248	3248	4813466	1482	3225 (99%)	23 (1%)
E. coli (contigs, *de novo*)	949	949	4520454	4763	943 (99%)	6 (1%)
E. coli (scaffolds, *de novo*)	373	373	4521606	12122	368 (99%)	5 (1%)
P. papatasi (*de novo*)	6961	6960	4627142	665	3263 (47%)	3697 (53%)

Nota:
As informações do relatório foram mantidas conforme o original em inglês.

6. Considerações finais

- O conceito do trabalho intitulado *LASZLO @ GALAXY* era o de tentar oferecer, por meio da combinação de programas de uso livre, uma alternativa para abordar o problema de montagem de genomas (a partir de dados de NGS), ao mesmo tempo oferecendo ao usuário final uma maneira mais amigável de fazê-lo. Os fluxos de trabalho empregados, típicos para o tratamento de dados das tecnologias NGS e normalmente implementados via linha de comando e por usuários com conhecimentos avançados, puderam ser transportados para um sistema com interface *Web*, já contemplando sugestões de parametrização. Por esse motivo, se tornaram mais adequados à utilização por usuários com conhecimentos de informática não tão avançados e, consequentemente, mais acostumados com ações do tipo "apontar e clicar";

- As montagens realizadas no protótipo, para os dados do organismo *Leishmania amazonensis*, apresentaram bons resultados e puderam ser aproveitadas para auxiliar um primeiro processo de avaliação do genoma do referido organismo;

- Nesta primeira fase do trabalho, não se objetivou realizar comparações ou análises minuciosas a respeito das qualidades das montagens obtidas, mas sim observar se a abordagem proposta conseguiria produzir resultados satisfatórios para uso em análises posteriores por eventuais usuários finais. De fato, todas as montagens produzidas demonstraram bom potencial de utilização para análises subsequentes, principalmente considerando que a maioria foi realizada a partir da parametrização padrão das ferramentas, ou seja, sem qualquer refinamento dos valores de parâmetros. Alterações de parâmetros e comparações das respectivas qualidades das montagens obtidas, para cada novo cenário de configuração de parâmetros a ser testado, são tarefas previstas para fases subsequentes de desenvolvimento do protótipo;

- A montagem *de novo* dos dados de *Leishmania amazonensis*, realizada com a ferramenta Velvet, apesar de satisfatória, fez o programa consumir cerca de 60 GB de memória RAM do servidor, utilizando valores de *hash* (*k-mer*) próximos ao valor padrão de 31. Os limites de *hash* podem ser elevados, mas isso implica em um consumo maior de memória. Como possíveis trabalhos futuros específicos para esses dados de montagem, vislumbram-se, ainda: (i) uso de valores maiores de *hash* com a própria ferramenta Velvet, explorando um pouco mais a capacidade do servidor e (ii) integração e incorporação de ferramentas como ABySS ou ALLPATHS/ALLPATHS2, com o objetivo de reduzir os requerimentos de memória e, possivelmente, incrementar, simultaneamente, os valores de *k-mer*;

- A ferramenta *Extract region tool*, pertencente ao módulo *NGS: LASZLO's Sandbox*, criada para atender a um requisito específico de uma profissional da área de ciências da vida, se mostrou bastante eficaz para auxiliá-la quanto à captura, em dados de montagem de genoma, de regiões equivalentes de genes conhecidos de espécies próximas, facilitando o trabalho de desenho de *primers*. Em um primeiro momento, foi idealizada a sua utilização somente a partir das informações fornecidas pela ferramenta *NCBI BLAST+ blastn*. Uma vez que a instância local GALAXY já traz outros recursos de BLAST similares (embutidos na seção *NCBI BLAST+*), outras variações de utilização da ferramenta *Extract region tool* podem ser futuramente idealizadas. A ferramenta também pode ser aprimorada no sentido de oferecer mais campos de coordenadas inicial e final ao usuário, de maneira que diferentes regiões possam ser extraídas a partir de uma única varredura no conjunto de dados de montagem. Outro aprimoramento viável seria dotá-la da capacidade de gerenciar automaticamente a quantidade de bases possível de ser extraída das regiões *upstream* e *downstream* à região de interesse, em relação aos limites do *contig* escolhido pelo usuário;

- A utilização do sistema GALAXY como núcleo do protótipo se mostrou apropriada para garantir flexibilidade a este último, no sentido de acomodar futuras alterações e refinamentos. Por exemplo, a inclusão de novas ferramentas ou alteração das existentes — tal como as mencionadas no item anterior, por exemplo — pode sempre ser feita de maneira rápida, fácil e independente, sem a necessidade de alterar outras partes funcionais do sistema;

- Também a metodologia de *Extreme Programming* se mostrou flexível para acomodar alterações em tempo de projeto e para proporcionar incremento de funcionalidades ao longo de sua execução;

- Pela propriedade de usabilidade demonstrada pelo protótipo, este apresenta grande potencial para vir a ser alçado à condição de um serviço *Web* institucional ou, em menor escala, como um serviço para atender a um determinado laboratório ou grupo mais limitado de usuários. Uma vez devidamente configurado para esse fim[95] e acomodado em uma infra-estrutura adequada de hardware, pode-se vislumbrá-lo como um grande aliado do usuário, no que diz respeito ao tratamento do grande volume de dados de NGS que continuamente vêm sendo gerados;

- Como demais trabalhos futuros para uma próxima etapa de desenvolvimento estão previstas a adequação do protótipo para receber dados de outras tecnologias NGS — por exemplo, Ion Torrent, PacBio, Oxford Nanopore, etc. —, bem como a implementação de ferramentas para a análise de dados de RNA-Seq. Outra ferramenta cuja implantação é vislumbrada é o *script* VelvetOptimizer[96], o qual possibilitaria a otimização dos parâmetros de montagem a serem empregados pelo programa Velvet. Também, após receberem mais alguns refinamentos, é prevista a publicação dos *wrappers* criados para a ferramenta *SOLiD™ de novo accessory tools 2.0* na página de desenvolvedores da comunidade GALAXY.

[95] O protótipo foi implementado sobre uma instância local básica. Isso significa dizer, por exemplo, que o banco de dados utilizado é o SQLite, um banco simples para tarefas básicas de desenvolvimento e que não tolera ações concorrentes. No sentido de obter a escalabilidade necessária a um serviço institucional, as recomendações descritas no endereço http://wiki.g2.bx.psu.edu/Admin/Config/Performance/Production%20Server devem ser seguidas. Lá, por exemplo, será orientada a troca do SQLite para um sistema gerenciador de banco de dados mais robusto, como PostgreSQL ou MySQL, de maneira a possibilitar a concorrência de uso da plataforma.
[96] Disponível em http://bioinformatics.net.au/software.velvetoptimiser.shtml.

Referências bibliográficas

Adams MD, Celniker SE, Holt RA, Evans CA, Gocayne JD, Amanatides PG, et al. The genome sequence of *Drosophila melanogaster*. Science 2000;287:2185-95.

Afgan E, Goecks J, Baker D, Coraor N, Nekrutenko A, Taylor J, et al. Galaxy - a gateway to tools in e-Science. In: Yang K, editor. Guide to e-Science: next generation scientific research and discovery. [local desconhecido]: Springer, in press 2011. 35 p.

Alberts B, Bray D, Johnson A, Lewis J, Raff M, Roberts K, et al. Fundamentos da biologia celular: uma introdução à biologia molecular da célula. Porto Alegre: Artmed; 1999.

Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. Biologia molecular da célula. 4. ed. Porto Alegre: Artmed; 2004.

Alonso DP. Utilização de marcadores moleculares no estudo populacional de *Leishmania infantum chagasi* no Brasil [dissertation]. Botucatu: Universidade Estadual Paulista; 2011. 59 p. Portuguese.

Altschul SF, Gish W, Miller W, Myers EW, Lipmanl DJ. Basic local alignment search tool. J. Mol. Biol. 1990;215(3):403-10.

Andersson P. Install and configure SNMP on Ubuntu [*homepage* na Internet]. [local desconhecido]: An It-Slave in the digital saltmine; 05 fev 2009 [atualizada em 30 abr 2012; acesso em 20 mar 2012]. Disponível em: http://www.it-slav.net/blogs/2009/02/05/install-and-configure-snmp-on-ubuntu/.

Ansorge WJ, inventor; EMBL Heidelberg, cessionário. Process for sequencing nucleic acids without gel sleving media on solid support and DNA chips (Verfahren zur Sequenzierung von Nukleinsauren ohne Gele). German Patent Application DE 41 41 178 A1 and Corresponding Worldwide Patent Applications. 1991.

Ansorge WJ. Next-generation DNA sequencing techniques. New Biotech. 2009;25(4):195-203.

Applied Biosystems. Applied Biosystems SOLiD™ 3 Plus System: de novo assembly protocol [Internet]. Carlsbad (CA): Life Technologies Corporation; maio 2010 [data de atualização desconhecida; acesso em 01 dez 2010]. Disponível em: http:// http://solidsoftwaretools.com/gf/project/.

Arruda P. Genômica vegetal. In: Mir L, organizador editorial. Genômica. São Paulo: Editora Atheneu; 2004. p. 95-103.

Baker M. *De novo* genome assembly: what every biologist should know. Nat. Methods 2012;9(4):333-7.

Bañuls A, Hide M, Prugnolle F. *Leishmania* and the Leishmaniases: a parasite genetic update and advances in taxonomy, epidemiology and pathogenicity in humans. Adv. Parasitol. 2007;64:1-109.

Bao H, Xiong Y, Guo H, Zhou R, Lu X, Yang Z, et al. MapNext: a software tool for spliced and unspliced alignments and SNP detection of short sequence reads. BMC Genomics 2009;10 (Suppl 3):S13.

Bao S, Jiang R, Kwan W, Wang B, Ma X, Song Y. Evaluation of next-generation sequencing software in mapping and assembly. J. Hum. Genet. 2011;56:406-14. Retraction in: Bao S, Jiang R, Kwan W, Wang B, Ma X, Song Y. Evaluation of next-generation sequencing software in mapping and assembly. J. Hum. Genet. 2011;1-9.

Barr J. Live and let license [*homepage* na Internet]. [local desconhecido]: IT World; c1994-2012 [22 maio 2001; acesso em 05 jul 2012]. Disponível em: http://www.itworld.com/LWD010523vcontrol4.

Batzoglou S, Jaffe DB, Stanley K, Butler J, Gnerre S, Mauceli E, et al. ARACHNE: a whole-genome shotgun assembler. Genome Res. 2002;12:177-89.

Bayer M, Milne I, Stephen G, Shaw P, Cardle L, Wright F, Marshall D. Comparative visualization of genetic and physical maps with Strudel. Bioinformatics 2011;27(9):1307-8.

Beck K. Extreme Programming explained: embrace change. 2. ed. [local desconhecido]: Addison-Wesley; 2004.

Bennett ST, Barnes C, Cox A, Davies L, Brown C. Toward the 1,000 dollars human genome. Pharmacogenomics 2005;6(4):373-82.

Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, et al. Accurate whole human genome sequencing using reversible terminator chemistry. Nature 2008;456:53-9.

BISTI - Biomedical Information Science and Technology Initiative. NIH working definition of bioinformatics and computational biology [Internet]. Bethesda (MD); 17 jul 2000 [data de atualização desconhecida; acesso em 10 jul 2011]. Disponível em: http://www.bisti.nih.gov/docs/CompuBioDef.pdf.

Blankenberg D, Taylor J, Schenk I, He J, Zhang Y, Ghent M, et al. A framework for collaborative analysis of ENCODE data: making large-scale analyses biologist-friendly. Genome Res. 2007;17(6):960-4.

Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, et al. Galaxy: a web-based genome analysis tool for experimentalists. Curr. Prot. Bioinf. 2010 Jan; Chapter 19: Unit 19.10.1-21.

Blankenberg D, Gordon A, Von Kuster G, Coraor N, Taylor J, Nekrutenko, et al. Manipulation of FASTQ data with Galaxy. Bioinformatics 2010;26(14):1783-5.

Blankenberg D, Taylor J, Nekrutenko, et al. Making whole genome multiple alignments usable for biologists. Bioinformatics 2011;27(17):2426-8.

Boetzer M, Henkel CV, Jansen HJ, Butler D, Pirovano W. Scaffolding pre-assembled contigs using SSPACE. Bioinformatics 2011;27(4):578-9.

Böcker S. Sequencing from compomers: using mass spectrometry for DNA *de novo* sequencing of 200+ nt. J. Comput. Biol. 2004;11(6):1110-34.

Braslavsky I, Hebert B, Kartalov E, Quake SR. Sequence information can be obtained from single DNA molecules. Proc. Natl. Acad. Sci. U.S.A. 2003;100(7):3960-4.

Brown TA. Genética: um enfoque molecular. 3. ed. Rio de Janeiro: Editora Guanabara Koogan; 1999.

Bryant Jr DW, Wong W, Mockler TC. QSRA — a quality-value guided *de novo* short read assembler. BMC Bioinformatics 2009;10:69.

Butler J, MacCallum I, Kleber M, Shlyakhter IA, Belmonte MK, Lander ES, et al. ALLPATHS: De novo assembly of whole-genome shotgun microreads. Genome Res. 2008;18(5):810-20.

Campbell NA, Reece JB, Urry LA, Cain ML, Wasserman SA, Minorsky PV, et al. Biologia. 8. ed. Porto Alegre: Artmed; 2010.

Campos L. HTML: rápido e prático. Goiânia: Terra; 2004.

Carvalho MCCG, Silva DCG. Sequenciamento de DNA de nova geração e suas aplicações na genômica de plantas. Cienc. Rural 2010;40(3):735-44.

Catanho M. Desenvolvimento de abordagens computacionais e ferramentas para a análise comparativa de genomas microbianos [master's thesis]. Rio de Janeiro: Instituto Oswaldo Cruz; 2005. 77 p. Portuguese.

Chaisson M, Pevzner P, Tang H. Fragment assembly with short reads. Bioinformatics 2004;20(13):2067-74.

Chaisson MJ, Pevzner PA. Short read fragment assembly of bacterial genomes. Genome Res. 2007;18:324-30.

Chaisson MJ, Brinza D, Pevzner PA. De novo fragment assembly with short mate-paired reads: Does the read length matter? Genome Res 2009;19:336-46.

Chapman JA, Ho I, Sunkara S, Luo S, Schroth GP, Rokhsar DS. Meraculous: *De Novo* Genome Assembly with Short Paired-End Reads. PLoS ONE 2011;6(8):e23501.

Chen Y, Souaiaia T, Chen T. PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. Bioinformatics 2009;25(19):2514-21.

Chevreux B, Wetter T, Suhai S. Genome Sequence Assembly Using Trace Signals and Additional Sequence Information. Computer Science and Biology: Proceedings of the German Conference on Bioinformatics (GCB) 99; 1999 Oct 4-6; Hannover, Germany. p. 45-56.

Chevreux B, Pfisterer T, Drescher B, Driesel AJ, Müller WEG, Wetter T, et al. Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs. Genome Res. 2004;14:1147-59.

Chevreux B. Sequence assembly with MIRA3: the definitive guide [*homepage* na Internet]. [local desconhecido]: Sourceforge.net; c2010 [data de atualização desconhecida; acesso em 05 jan 2011]. Disponível em: http://mira-assembler.sourceforge.net/docs/DefinitiveGuideToMIRA.html.

Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 2009;25:1422-3.

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res. 2010;38(6):1767-71. Epub 2009 Dec 16.

Cox AJ 2007. ELAND: Efficient large-scale alignment of nucleotide databases. Illumina, San Diego, CA.

Dahlö M. Illumina sequencing [Internet]. Uppsala: Uppnex, Uppsala Universitet; 18 out 2010 [18 out 2010; acesso em 15 jun 2012]. Disponível em https://www.uppnex.uu.se/uppnex-book/technologies/solexa-sequencing.

Dale JW, Schantz M, Plant N. From genes to genomes: concepts and applications of DNA technology. 3. ed. Oxford: John Wiley-Blackwell; 2012.

Danen V. Graphing router usage with MRTG [*homepage* na Internet]. [local desconhecido]: TechRepublic; 08 maio 2006 [atualizada em 09 maio 2006; acesso em 20 mar 2012]. Disponível em: http://www.techrepublic.com/article/graphing-router-usage-with-mrtg/6068756/.

Danen V. Monitor your memory usage with MRTG [*homepage* na Internet]. [local desconhecido]: TechRepublic; 27 out 2006 [atualizada em 29 out 2006; acesso em 20 mar 2012]. Disponível em: http://www.techrepublic.com/article/monitor-your-memory-usage-with-mrtg/6130300/.

Davies K. Seu genoma por mil dólares: a revolução no sequenciamento do DNA e a nova era da medicina personalizada. São Paulo: Companhia das Letras; 2011.

Dayarian A, Michael TP, Sengupta AM. SOPRA: Scaffolding algorithm for paired reads via statistical optimization. Bioinformatics 2010;11:345.

De Faria RA. Treinamento avançado em XML: desvende os poderosos recursos desta linguagem. São Paulo: Digerati Books; 2005.

De la Bastide M, McCombie WR. Assembling genomic DNA sequences with PHRAP. Curr. Protoc. Bioinformatics 2007; Chapter 11:Unit11.4.

Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, Salzberg SL. Alignment of whole genomes. Nucleic Acids Res. 1999;27(11):2369-76.

Delcher AL, Phillippy A, Carlton J, Salzberg SL. Fast algorithms for large-scale genome alignment and comparison. Nucleic Acids Res. 2002;30(11):2478-83.

Dicionário eletrônico Houaiss da língua portuguesa - versão 1.0. Rio de Janeiro: Editora Objetiva; 2001.

Dinov ID, Torri F, Macciardi F, Petrosyan P, Liu Zhizhong, Zamanyan A, et al. Applications of the pipeline environment for visual informatics and genomics computations. BMC Bioinformatics 2011;12(304):1-20.

Dohm JC, Lottaz C, Borodina T, Himmelbauer H. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. Genome Res. 2007;17(11):1697-706. Epub 2007 Oct 1.

Dressman D, Yan H, Traverso G, Kinzler KW, Vogelstein B. Transforming single DNA molecules into fluorescent magnetic particles for detection and enumeration of genetic variations. Proc. Natl. Acad. Sci. U.S.A. 2003;100(15):8817-22.

Drmanac R, Labat I, Brukner I, Crkvenjakov R. Sequencing of megabase plus DNA by hybridization: theory of the method. Genomics 1989;4:114-28.

Droege M, Hill B. The Genome Sequencer FLX™ System: longer reads, more applications, straight forward bioinformatics and more complete data sets. J. Biotech. 2008;136:3-10.

Durfee T, Nelson R, Baldwin S, Plunkett III G, Burland V, Mau B, et al. The complete genome sequence of *Escherichia coli* DH10B: insights into the biology of a laboratory workhorse. J. Bacteriol. 2008;190(7):2597-606.

Earl D, Bradnam K, John JS, Darling A, Lin D, Fass J, et al. Assemblathon 1: A competitive assessment of de novo short read assembly methods. Genome Res. 2011;21:2224-41.

Ewing B, Hillier L, Wendl MC, Green P. Base-calling of automated sequencer traces using Phred. I. Accuracy Assessment. Genome Res. 1998;8:175-85.

Ewing B, Green P. Base-calling of automated sequencer traces using Phred. II. Error probabilities. Genome Res. 1998;8:186-94.

Flicek P, Birney E. Sense from sequence reads: methods for alignment and assembly. Nat. Methods 2009;6(11s):S6-S12.

Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, et al. Galaxy: a platform for interactive large-scale genome analysis. Genome Res. 2005;15(10):1451-5.

Gibas C, Jambeck P. Desenvolvendo bioinformática. Rio de Janeiro: Campus; 2001.

Glenn TC. Field guide to next-generation DNA sequencers. Mol. Ecol. Resour. 2011;11:759-69.

Goecks J, Nekrutenko A, Taylor J, et al. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 2010;11(8):R86.

Gomes MR. Reconstrução *in silico* das vias de processamento da informação genética nos Tritryps (*Trypanosoma cruzi*, *Trypanosoma brucei* e *Leishmania major*): busca por análogos funcionais [master's thesis]. Rio de Janeiro: Instituto Oswaldo Cruz; 2010. 92 p. Portuguese.

Griffiths AJF, Wessler SR, Carroll SB, Doebley J. Introduction to genetic analysis. 10. ed. New York: W. H. Freeman and Company; 2012.

Gugik G. Código aberto e software livre não significam a mesma coisa! [*homepage* na Internet]. [local desconhecido]: Tecmundo; 13 mar 2009 [13 mar 2009; acesso em 05 jul 2012]. Disponível em: http://www.tecmundo.com.br/linux/1739-codigo-aberto-e-software-livre-nao-significam-a-mesma-coisa-.htm.

Guimarães GS. Sequenciamento de DNA. In: Carvalho CV, Ricci G, Affonso R, organizadoras. Guia de práticas em biologia molecular. São Caetano do Sul: Yendis Editora; 2010. p. 133-44.

Guimarães MP, Cavalcanti MCR. Proveniência de dados na área de Bioinformática. M.S.C. 2009;2:1-30.

Hadfield J. Google maps for next-gen facilities [*homepage* na Internet]. [local desconhecido]: SEQanswers; 5 maio 2009 [atualizada em 13 ago 2009; acesso em 09 jul 2012]. Disponível em: http://seqanswers.com/forums/showthread.php?t=1656.

Hadfield J. Google maps update: 1442 instruments in 486 labs [*homepage* na Internet]. [local desconhecido]: SEQanswers; 21 fev 2011 [atualizada em 17 maio 2012; acesso em 09 jul 2012]. Disponível em: http://seqanswers.com/forums/showthread.php?t=9591.

Hadfield J, Loman N. Next generation genomics: world map of high-throughput sequencers [*homepage* na Internet]. [local desconhecido]: omicsmaps.com; c2009-2012 [data de atualização desconhecida; acesso em 09 jul 2012]. Disponível em: http://omicsmaps.com.

Hall N. Advanced sequencing technologies and their wider impact in microbiology. J. Exp. Biol. 2007;209:1518-25.

Harris RS. Improved pairwise alignment of genomic DNA [dissertation]. University Park (PA): Penn State, The Pennsylvania State University; 2007. 74 p.

Hemrajani A. Desenvolvimento ágil em Java com Spring, Hibernate e Eclipse. São Paulo: Pearson Prentice Hall; 2007.

Henson J, Tischler G, Zemin N. Next-generation sequencing and large genome assemblies. Pharmacogenomics 2012;13(8):901-15.

Hernandez D, François P, Farinelli L, Osteras M, Schrenzel J. De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. Genome Res. 2008;18(5):802-9.

Hillman-Jackson J, Clements D, Blankenberg D, Taylor J, Nekrutenko A, et al. Using Galaxy to perform large-scale interactive data analyses. Curr. Prot. Bioinf. 2012 Jun;Unit 10.5.

Holland RCG, Down TA, Pocock M, Prlic A, Huen D, James K, et al. BioJava: an open-source framework for bioinformatics. Bioinformatics 2008;24:2096-7.

Homer N, Merriman B, Nelson SF. BFAST: An Alignment Tool for Large Scale Genome Resequencing. PLoS ONE 2009;4(11):e7767.

Horner DS, Pavesi G, Castrignanò T, De Meo PD, Liuni S, Sammeth M, et al. Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing. Brief. Bioinform. 2009;2(2):181-97.

Hoon S, Ratnapu KK, Chia J, Kumarasamy B, Juguang X, Clamp M, et al. Biopipe: a flexible framework for protocol-based bioinformatics analysis. Genome Res. 2003;13:1904-15.

Hossain MS, Azimi N, Skiena S. Crystallizing short-read assemblies around seeds. BMC Bioinformatics 2009;10(Suppl 1):S16.

Huang X, Madan A. Cap3: a dna sequence assembly program. Genome Res. 1999;9:868-77.

Huang X, Yang SP. Generating a genome assembly with PCAP. Curr. Prot. Bioinf. 2005;Chapter 11:Unit 11.3.

Hunkapiller T, Kaiser RJ, Hoop BF, Hood L. Large-scale and automated DNA sequence determination. Science 1991;254:59-67.

Hutchison III CA. DNA sequencing: bench to bedside and beyond. Nucleic Acids Res. 2007;35(18):6227-37.

Hyman ED. A new method of sequencing DNA. Anal. Biochem. 1988;174:423-36.

Illumina, Inc. DNA sequencing with Solexa technology [Internet]. San Diego (CA): Illumina, Inc.; 2007 [data de atualização desconhecida; acesso em 20 jul 2012]. Disponível em: http://www.plantsciences.ucdavis.edu/bit150/2006/JD_Lecture/Lecture%201%20Databases/Solexa_DNAsequencing.pdf.

Illumina, Inc. *De novo* assembly with the Genome Analyzer [Internet]. San Diego (CA): Illumina, Inc.; 2010 [27 jan 2009; acesso em 03 jan 2011]. Disponível em: http://www.illumina.com/documents/products/technotes/technote_denovo_assembly.pdf.

Illumina, Inc. *De novo* assembly using Illumina reads [Internet]. San Diego (CA): Illumina, Inc.; 2010 [13 out 2009; acesso em 03 jan 2011]. Disponível em: http://www.illumina.com/Documents/products/technotes/technote_denovo_assembly_ecoli.pdf.

Jaffe DB, Butler J, Gnerre S, Mauceli E, Lindblad-Toh K, Mesirov JP, et al. Whole-genome sequence assembly for mammalian genomes: Arachne 2. Genome Res. 2003;13:91-6.

Janitz M, editor. Next-Generation genome sequencing: towards personalized medicine. Weinheim: Wiley-VCH; 2008.

Jeck WR, Reinhardt JA, Baltrus DA, Hickenbotham MT, Magrini V, Mardis ER, et al. Extending assembly of short DNA sequences to handle error. Bioinformatics 2007;23(21): 2942-4.

Jiang H, Wong WH. SeqMap: mapping massive amount of oligonucleotides to the genome. Bioinformatics 2008;24:2395-6.

Karow J. Leerink report: about 900 next-gen sequencers deployed to date; market poised for growth [*homepage* na Internet]. [local desconhecido]: Silicon Investor; 30 jun 2009 [atualizada em 03 jul 2009; acesso em 16 maio 2011]. Disponível em: http://siliconinvestor.advfn.com/readmsg.aspx?msgid=25756735.

Kent WJ. BLAT — the BLAST-like alignment tool. Genome Res. 2002;12(4):656-64.

Kircher M, Stenzel U, Kelso J. Improved base calling for the Illumina Genome Analyzer using machine learning strategies. Genome Biol. 2009; 10(8):R83. Epub 2009 Aug 14.

Klassen JL, Currie CR. Gene fragmentation in bacterial draft genomes: extent, consequences and mitigation. BMC Genomics 2012;13:14.

Klug WS, Cummings MR, Spencer CA. Concepts of genetics. 8. ed. New Jersey: Pearson Education, Inc; 2006.

Koboldt D. Making the leap: Maq to BWA [*homepage* na Internet]. [local desconhecido]: MassGenomics; 23 dec 2009 [atualizada em 29 dez 2009; acesso em 20 jul 2012]. Disponível em: http://massgenomics.org/2009/12/making-the-leap-maq-to-bwa.html.

Koboldt D. Fast, efficient short read alignment with gaps: Bowtie 2 [*homepage* na Internet]. [local desconhecido]: MassGenomics; 12 apr 2012 [atualizada em 12 abr 2012; acesso em 20 jul 2012]. Disponível em: http://massgenomics.org/2012/04/fast-efficient-short-read-alignment-with-gaps-bowtie-2.html.

Koskovsky PS, Wadhawan S, Chiaromonte F, Ananda G, Chung W, Taylor J, et al. Windshield splatter analysis with the Galaxy metagenomic pipeline. Genome Res. 2009;19(11):2144-53.

Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10:R25.

Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat. Methods 2012;9(4):357-9.

Lazarus R, Taylor J, Qiu W, Nekrutenko A. Toward the commoditization of translational genomic research: design and implementation features of the Galaxy genomic workbench. Summit Translat. Bioinforma. 2008;2008:56-60.

Leamon JH, Lee WL, Tartaro KR, Lanza JR, Sarkis GJ, deWinter AD, et al. A massively parallel PicoTiterPlate™ based platform for discrete picoliter-scale polymerase chain reactions. Electrophoresis 2003;24:3769-77.

Lemos M. Workflow para Bioinformática [dissertation]. Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro; 2004. Portuguese.

Li H, Ruan J, Durbin R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. 2008;18:1851-8.

Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009;25(14):1754-60.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009;25:2078-9.

Li R, Li Y, Kristiansen K, Wang J. SOAP: short oligonucleotide alignment program. Bioinformatics 2008;24(5):713-4.

Li R, Yu C, Li Y, Lam T, Yiu S, Kristiansen K, Wang J. SOAP2: an improved ultrafast tool for short read alignment. Bioinformatics 2009;25(15)1966-7.

Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, et al. De novo assembly of human genomes with massively parallel short read sequencing. Genome Res. 2010;20:265-72.

Lin H, Zhang Z, Zhang MQ, Ma B, Li M. Zoom! zillions of oligos mapped. Bioinformatics 2008, 24(21):2431-7.

Liu L, Li Y, Li S, Hu N, He Y, Pong R, et al. Comparison of Next-Generation Sequencing systems. J. Biomed. Biotechnol. 2012; 2012: Article ID 251364, 11 pages, doi:10.1155/2012/251364.

Lunter G, Goodson M. Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads. Genome Res. 2010;21:000-000.

Luscombe NM, Greenbaum D, Gerstein M. What is bioinformatics? A proposed definition and overview of the field. Methods Inf. Med. 2001;40:346-58.

MacCallum I, Przybylski D, Gnerre S, Burton J, Shlyakhter I, Gnirke A, et al. ALLPATHS 2: small genomes assembled accurately and with high continuity from short paired reads, Genome Biol. 2009;10:R103.

Macmillan Publishers Limited. Human genome: genomes by the thousand [editorial]. Nature. 2010;467:1026-7.

Madabhushi RS. Separation of 4-color DNA sequencing extension products in noncovalently coated capillaries using low viscosity polymer solutions. Electrophoresis 1998;19:224-30.

Mardis ER. The impact of next-generation sequencing technology on genetics. Trends Genet. 2008;24(3):133-141.

Mardis ER. Next-Generation DNA sequencing methods. Annu. Rev. Genomics Hum. Genet. 2008;9:387-402.

Mardis ER. New strategies and emerging technologies for massively parallel sequencing: applications in medical research. Genome Med. 2009;1(4):40.1-4.

Mardis ER. A decade's perspective on DNA sequencing technology. Nature 2011;470:198-203.

Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, et al. Genome sequencing in microfabricated high-density picolitre reactors. Nature 2005;437(7057):376-80. Epub 2005 Jul 31 Cited in PubMed; PMID 16056220.

Martínez-Alcántara A, Ballesteros E, Feng C, Rojas M, Koshinsky H, Fofanov VY, et al. PIQA: Pipeline for Illumina G1 Genome Analyzer data Quality Assessment. Bioinformatics 2009;25(18):2438-9.

Meldrum D. Automation for genomics, part one: preparation for sequencing. Genome Res. 2000;10:1081-92.

Meldrum D. Automation for genomics, part two: sequencers, microarrays, and future trends. Genome Res. 2000;10:1288-303.

Metzker ML. Sequencing technologies: the next generation. Nature Rev. Genet. 2010;11:31-46.

Micklos DA, Freyer GA, Crotty DA. A ciência do DNA. 2. ed. Porto Alegre: Artmed; 2005.

Microsoft PRESS®. Dicionário de informática. 3. ed. Rio de Janeiro: Campus; 1998.

Miller JR, Delcher AL, Koren S, Venter E, Walenz BP, Brownley A, et al. Aggressive assembly of pyrosequencing reads with mates. Bioinformatics 2008;24(24):2818-24.

Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. Genomics 2010;95:315-27.

Miller W, Rosenbloom K, Hardison RC, Hou M, Taylor J, Raney B, et al. 28-way vertebrate alignment and conservation track in the UCSC Genome Browser. Genome Res. 2007;17(12):1797-808.

Milne I, Shaw P, Stephen G, Bayer M, Cardle L, Thomas WTB, et al. Flapjack — graphical genotype visualization. Bioinformatics 2010;26(24):3133-4.

Mitra RD, Shendure J, Olejnik J, Edyta-Krzymansk-Olejnik, Church GM. Fluorescent *in situ* sequencing on polymerase colonies. Anal. Biochem. 2003;320:55-65.

Mota Filho JE. Descobrindo o Linux: entenda o sistema operacional GNU/Linux. 2. ed. São Paulo: Novatec Editora; 2007.

Mullikin JC, Ning Z. The Phusion Assembler. Genome Res. 2002;13(1):81-90.

Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP, Flanigan MJ, et al. A whole-genome assembly of Drosophila. Science 2000;287:2196-2204.

NCBI. SRA Handbook [*homepage* na Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010- [15 jun 2012; acesso em 04 abr 2012]. Disponível em: http://www.ncbi.nlm.nih.gov/books/NBK47528/.

Ning Z, Cox AJ, Mullikin JC. SSAHA: A fast search method for large DNA databases. Genome Res. 2001;11:1725-9.

Nyrén P, Lundin A. Enzymatic method for continuous monitoring of inorganic pyrophosphate synthesis. Anal. Biochem. 1985;151:504-9.

Oliva G. Genômica estrutural: de genes a novos fármacos. In: Mir L, organizador editorial. Genômica. São Paulo: Editora Atheneu; 2004. p. 163-82.

Olson M. New methods for assembly and validation of large genomes [master's thesis]. Notre Dame (IN): University of Notre Dame; 2009. 80 p.

Ondov BD, Varadarajan A, Passalacqua KD, Bergman NH. Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications. Bioinformatics 2008;24(23):2776-7.

Ossowski S, Schneeberger K, Clark RM, Lanz C, Warthmann N, Weigel D. Sequencing of natural strains of *Arabidopsis thaliana* with short reads. Genome Res. 2008;18(12):2024-33.

Otto TD, Sanders M, Berriman M, Newbold C. Iterative correction of reference nucleotides (iCORN) using second generation sequencing technology. Bioinformatics 2010;26:1704-7.

Pacific Biosciences. Corporate Info [*homepage* na Internet]. Menlo Park (CA): Pacific Biosciences; c2010-2012 [data de atualização desconhecida; acesso em 09 jul 2012]. Disponível em: http://www.pacificbiosciences.com/aboutus/corporate_info/.

Pareek CS, Smoczynski R, Tretyn A. Sequencing technologies and genome sequencing. J. Appl. Genetics 2011; 52:413-35.

Paszkiewicz K, Studholme DJ. *De novo* assembly of short sequence reads. Brief. Bioinform. 2010;2(5):457-72.

Paszkiewicz K, Studholme DJ. High-Throughput Sequencing data analysis software: current state and future developments. In: Rodríguez-Ezpeleta N, Hackenberg M, Aransay AM, editors. Bioinformatics for High Throughput Sequencing. New York: Springer; 2012. p. 231-48.

Pearson WR; Lipman DJ. Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. U.S.A. 1988;85:2444-8.

Peterson TW, Nam SJ, Darby A. Next Gen Sequencing Survey: what laboratory directors are saying about next generation sequencing, GWAS and stimulus [Internet]. [local desconhecido]: J.P.Morgan; 12 maio 2010 [data de atualização desconhecida; acesso em 20 jun 2012]. Disponível em: http://www.genomicslawreport.com/wp-content/uploads/2011/04/JP-Morgan-NGS-Report.pdf.

Petterson E, Lundeberg J, Ahmadian A. Generations of sequencing technologies. Genomics 2009;93:105-11.

Pevzner PA, Tang H, Waterman MS. An Eulerian path approach to DNA fragment assembly. Proc. Natl. Acad. Sci. U.S.A. 2001;98(17):9748-53.

Pevzner PA, Tang H. Fragment assembly with double-barreled data. Bioinformatics 2001;17(Suppl. 1):S225-33.

Pitaluga AN. Aspectos moleculares da imunidade de *Lutzomyia longipalpis* (Díptera: Psychodidae) [dissertation]. Rio de Janeiro: Instituto Oswaldo Cruz; 2007. 223 p. Portuguese.

Pop M, Salzberg SL, Shumway M. Genome sequence assembly: algorithms and issues, IEEE Computer 2002;35(7):47-55.

Pop M, Phillippy A, Delcher AL, Salzberg S. Comparative genome assembly. Brief. Bioinform. 2004;5(3):237-48.

Pop M, Kosack DS, Salzberg SL. Hierarchical scaffolding with Bambus. Genome Res. 2004;14(1):149-59.

Pop M, Salzberg SL. Bioinformatics challenges of new sequencing technology. Trends Genet. 2008;24(3):142-9.

Pop M. Genome assembly reborn: recent computational challenges. Brief. Bioinform. 2009;10(4):354-66.

Preparata FP, Upfal E. Sequencing-by-hybridization at the information-theory bound: an optimal algorithm. J. Comput. Biol. 2000;7(3/4):621-30.

Pressman RS. Engenharia de software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH Editora; 2011.

Prober JM, Trainor GL, Dam RJ, Hobbs FW, Robertson CW, Zagursky RJ, et al. A system for rapid DNA sequencing with fluorescent chain-terminating dideoxynucleotides, Science 1987;238:336-41.

Qi J, Zhao F, Buboltz A, Schuster SC. inGAP: an integrated next-generation genome analysis pipeline. Bioinformatics 2010;26(1):127-9.

Quinlan AR, Stewart DA, Strömberg MP, Marth GT. Pyrobayes: an improved base caller for SNP discovery in pyrosequences. Nat. Methods 2008;5(2):179-81.

Ramos RTJ, Carneiro AR, Baumbach J, Azevedo V, Schneider MPC, Silva A. Analysis of quality raw data of second generation sequencers with Quality Assessment Software, BMC Res. Notes 2011;4:130.

Rice P, Longden I, Bleasby A. EMBOSS: the European Molecular Biology Open Software Suite. Trends Genet. 2000;16:276-7.

Richter DC, Schuster SC, Huson DH. *OSLay: optimal syntenic layout of unfinished assemblies*. Bioinformatics 2007 Jul 1;23(13):1573-9. Epub 2007 Apr 26.

Rodríguez-Ezpeleta N, Hackenberg M, Aransay AM, editors. Bioinformatics for High Throughput Sequencing. New York: Springer; 2012.

Ronaghi M, Karamohamed S, Petterson B, Uhlén M, Nyrén P. Real-time DNA sequencing using detection of pyrophosphate release. Anal. Biochem. 1996;242:84-9.

Ronaghi M, Uhlén M, Nyrén P. DNA sequencing: a sequencing method based on real-time pyrophosphate. Science 1998;281(5375):363-5.

Ronaghi M. Pyrosequencing sheds light on DNA sequencing. Genome Res. 2001;11:3-11.

Rothberg JM, Leamon JH. The development and impact of 454 sequencing. Nat. Biotechnol. 2008, 26(10):1117-24.

Ruffalo M, LaFramboise T, Koyutürk M. Comparative analysis of algorithms for next-generation sequencing read alignment. Bioinformatics 2011;27(20):2790-6.

Rumble SM, Lacroute P, Dalca AV, Fiume M, Sidow A, Brudno M. SHRiMP: accurate mapping of short color-space reads. PLOS Comput. Biol. 2009;5(5):e1000386.

Rutherford K, Parkhill J, Crook J, Horsnell T, Rice P, Rajandream M, et al. Artemis: sequence visualization and annotation. Bioinformatics 2000;16:944-5.

Saade J. Bash: guia de consulta rápida. São Paulo: Novatec Editora; 2001.

Salmela L, Mäkinen V, Välimäki N, Ylinen J, Ukkonen E. Fast scaffolding with small independent mixed integer programs. Bioinformatics 2011;27(23):3259-65.

Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc. Natl. Acad. Sci. U.S.A. 1977;74(12):5463-7.

Santos FR, Bonatto SL. Genética de populações humanas. In: Mir L, organizador editorial. Genômica. São Paulo: Editora Atheneu; 2004. p. xxvii-xxxix.

Santos WG. Desenvolvendo software de forma autônoma. Rio de Janeiro: Editora Ciência Moderna; 2010.

Sasson AS. From millions to one: theoretical and concrete approaches to *de novo* assembly using short read DNA sequences [dissertation]. New Brunswick (NJ): Rutgers, The State University of New Jersey; 2010. 112 p.

Schadt EE, Turner S, Kasarskis A. A window into third-generation sequencing. Hum. Mol. Gen. 2010;19(2):R227-R240.

Schatz MC. The missing graphical user interface for genomics. Genome Biol. 2010;11(8):128. Epub 2010 Aug 25.

Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics 2011;27:863-4.

Schuster SC. Next-generation sequencing transforms today's biology. Nat. Methods 2008;5(1):16-8.

SEQanswers. Google maps compilation of NGS instruments [*homepage* na Internet]. [local desconhecido]: SEQanswers; 23 mar 2010 [atualizada em 23 mar 2010; acesso em 09 jul 2012]. Disponível em: http://seqanswers.com/forums/showthread.php?s=f548ae2fa2ea4c827d8e573487fe0a02&t=4462.

Setubal J, Meidanis J. Introduction to computational molecular biology. Boston: PWS Publishing Company; 1997.

Shendure J, Porreca GJ, Reppas NB, Lin X, McCutcheon JP, Rosenbaum AM, et al. Accurate multiplex polony sequencing of an evolved bacterial genome. Science 2005;309:1728-32.

Shendure J, Ji H. Next-generation DNA sequencing. Nat. Biotechnol. 2008;26(10):1135-45.

Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJM, Birol I. ABySS: A parallel assembler for short read sequence data. Genome Res. 2009;19:1117-23.

Simpson JT, Durbin R. Efficient *de novo* assembly of large genomes using compressed data structures. Genome Res. 2012;22(3):549-56.

Skrabanek L. Using Galaxy for NGS Analyses [Internet]. New York (NY): Institute for computational biomedicine, Weill Medical College of Cornell University; c2012 [jun 2012; acesso em 20 jul 2012]. Disponível em: http://chagall.med.cornell.edu/galaxy/GalaxyWorkshopNotes.pdf.

Smeds L. Extract region from fasta sequence [Internet]. Uppsala: Uppnex, Uppsala Universitet; 24 fev 2011 [24 fev 2011; acesso em 15 abr 2012]. Disponível em https://www.uppnex.uu.se/content/extract-region-fasta-sequence.

Smith AD, Xuan Z, Zhang MQ. Using quality scores and longer reads improves accuracy of Solexa read mapping. BMC Bioinformatics 2008;9:128.

Smith LM, Sanders RJK, Hughes P, Dood C, Connel CR, Heiner C, et al. Fluorescence detection in automated DNA sequence analysis. Nature 1986;321: 674-9.

Sommer DD, Delcher AL, Salzberg SL, Pop M. Minimus: a fast, lightweight genome assembler. BMC Bioinformatics 2007;8:64.

Sommerville I. Engenharia de software. 8 ed. São Paulo: Pearson Prentice Hall; 2007.

Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, et al. The BioPerl toolkit: Perl modules for the life sciences. Genome Res. 2002;12:1611-8.

Sterky F, Lundeberg J. Sequence analysis of genes and genomes. J. Biotech. 2000;76:1-31.

Sutton GG, White O, Adams MD, Kerlavage AR. TIGR Assembler: A new tool for assembling large shotgun sequencing projects. Genome Science and Technology 1995;1(1):9-19.

Swerdlow H, Gesteland R. Capillary gel electrophoresis for rapid, high resolution DNA sequencing. Nucleic Acids Res. 1990;18(6):1415-9.

Swerdlow H, Wu S, Harke H, Dovichi NJ. Capillary gel electrophoresis for DNA sequencing: laser-induced fluorescence detection with the sheath flow cuvette. J. Chromatogr. 1990;516:61-7.

Taylor J, Schenk I, Blankenberg D, Nekrutenko A. Using Galaxy to perform large-scale interactive data analysis. Curr. Prot. Bioinf. 2007 Sep;19:10.5.1-10.5.25.

Teles VM. Extreme programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. São Paulo: Novatec editora; 2006.

The Genome Institute. Phlebotomus papatasi [Internet]. St. Louis (MO): School of Medicine, Washington University; c1993-2012 [13 set 2010; acesso em 15 jul 2012]. Disponível em http://genome.wustl.edu/genomes/view/phlebotomus_papatasi.

The SAM Format Specification Working Group. The SAM format specification (v1.4-r985) [Internet]. [local desconhecido]: The SAM Format Specification Working Group; 7 set 2011 [7 set 2011; acesso em 27 jun 2012]. Disponível em http://samtools.sourceforge.net/SAM1.pdf.

Thompson JF, Milos PM. The properties and applications of single-molecule DNA sequencing. Genome Biol. 2011;12(217):1-10.

Thudi M, Li Y, Jackson SA, May GD, Varshney RK. Current state-of-art of sequencing technologies for plant genomics research. Brief. Func. Genomics 2012;2(1):3-11.

Trapnell C, Pachter L, Salzberg SL. TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 2009;25(9):1105-11.

Tsai IJ, Otto TD, Berriman M. Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. Genome Biol. 2010;11:R41.

Tschoeke DA. Desenvolvimento de um sistema integrado para genotipagem de protozoários patogênicos utilizando-se genes ortólogos universais [master's thesis]. Rio de Janeiro: Instituto Oswaldo Cruz; 2010. 148 p. Portuguese.

Tschoeke DA, Guedes H, Jardim R, Boité MC, Nunes GL, R Antonio, et al. Initial analysis of the *Leishmania amazonensis* genome [Apresentação ao XX Congreso Latinoamericano de Parasitología y XV Congreso Colombiano de Parasitología y Medicina Tropical; 2011 set. 27; Bogotá, Colômbia].

Valouev A, Ichikawa J, Tonthat T, Stuart J, Ranade S, Peckham H, et al. A high-resolution, nucleosome position map of *C. elegans* reveals a lack of universal sequence-dietated positioning. Genome Res. 2008;18:1051-63.

Wagner G. Geração e análise comparativa de sequências genômicas de *Trypanosoma rangeli* [master's thesis]. Rio de Janeiro: Instituto Oswaldo Cruz; 2006. 105 p. Portuguese.

Wagner G, Soriano K, Jucá H, Belloze KT, Tschoeke DA, Geronimo GA, et al. STINGRAY: System for Integrated Genomic Resources and Analysis. [Apresentação no International Workshop on Genomic Databases (IWGD'07); 2007, Angra dos Reis, Brasil].

Walker MR, Rapley R. Route maps in gene technology. Cambridge: Blackwell Science Ltd.; 1997.

Warren RL, Sutton GG, Jones SJM, Holt RA. Assembling millions of short DNA sequences using SSAKE. Bioinformatics 2007;23(4):500-1.

Waterston RH, Lander ES, Sulston JE. On the sequencing of the human genome. Proc. Natl. Acad. Sci. U.S.A. 2002;99(6):3712-6.

Watson JD, Berry A. DNA: o segredo da vida. São Paulo: Companhia das Letras; 2005.

Weiss VA. Estratégias de finalização e montagem do genoma da bactéria diazotrófica endofítica *Herbaspirillum seropedicae* SmR1 [master's thesis]. Curitiba: Universidade Federal do Paraná; 2010. 70 p. Portuguese.

WHO. Control of Leishmaniases. Geneva: World Health Organization; 1990.

Wikipédia. Análise de sistemas [*homepage* na Internet]. [local desconhecido]: Wikipédia; 02 ago 2012 [02 ago 2012; acesso em 23 ago 2012]. Disponível em: http://pt.wikipedia.org/wiki/An%C3%A1lise_de_sistemas.

Wold B, Myers RM. Sequence census methods for functional genomics. Nat. Methods 2008;5(1):19-21.

Xiong, Jin. Essential bioinformatics. New York: Cambridge University Press; 2006.

Zerbino D, Birney E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res. 2008;18(5):821-9.

Zhang J, Chiodini R, Badr A, Zhang G. The impact of next-generation sequencing on genomics. J. Genet. Genomics 2011;38(3):95-109.

Zhang W, Chen J, Yang Y, Tang Y, Shang J, Shen B. A practical comparison of *de novo* genome assembly software tools for Next-Generation Sequencing technologies. PLoS ONE 2011;6(3):e17915.

Zhang Z, Schwartz S, Wagner L, Miller W. A greedy algorithm for aligning DNA sequences. J. Comput. Biol. 2000;7(1-2):203-14.

URLs:

http://www.ncbi.nlm.nih.gov/genbank - Acesso em 15 jun 2012.

http://www.ncbi.nlm.nih.gov/sra - Acesso em 15 jun 2012.

http://www.454.com/ - Acesso em 10 jun 2012.

http://www.appliedbiosystems.com/ - Acesso em 10 jun 2012.

http://www.roche-applied-science.com/ - Acesso em 10 jun 2012.

http://www.illumina.com/ - Acesso em 10 jun 2012.

http://www.polonator.org - Acesso em 10 jun 2012.

http://www.helicosbio.com - Acesso em 10 jun 2012.

http://www.iontorrent.com/technology - Acesso em 10 jun 2012.

http://www.pacificbiosciences.com/ - Acesso em 10 jun 2012.

http://omicsmaps.com - Acesso em 7 jun 2012.

http://omicsmaps.com/stats - Acesso em 7 jun 2012.

https://www.uppnex.uu.se/system/files/imagecache/for_lightbox/fig3_1.jpg - Acesso 7 de jun 2012.

http://solidsoftwaretools.com/gf/project/denovo/ - Acesso em 20 dez 2010.

http://bioinformatics.bc.edu/marthlab/Mosaik - Acesso em 8 abr 2012.

http://solidsoftwaretools.com/gf/project/corona/ - Acesso em 20 dez 2010.

ftp://ftp.sanger.ac.uk/pub/users/zn1/phusion2/ - Acesso em 28 ago 2012.

http://www.lge.ibi.unicamp.br/zorro/ - Acesso em 20 mai 2011.

http://sourceforge.net/apps/mediawiki/amos/index.php?title=Minimus2 - Acesso em 20 mai 2011.

http://sourceforge.net/apps/mediawiki/amos/index.php?title=AMOS - Acesso em 20 mai 2011.

http://seqanswers.com/wiki/Software/list - Acesso em 10 jul 2012.

http://www.uni-tuebingen.de/modeling/Mod_Leish_Intro_de.html - Acesso em 10 jun 2012.

http://enfermagem-sae.blogspot.com.br/2009/04/leishmaniose-visceral-ou-calazar.html - Acesso em 15 jul 2012.

http://genome.wustl.edu/data.cgi - Acesso em 5 jun 2012.

http://genome.wustl.edu/pub/organism/Invertebrates/Phlebotomus_papatasi/assembly/Phlebotomus_papatasi-2.0/ASSEMBLY - Acesso em 5 jun 2012.

http://genome.wustl.edu/pub/organism/Invertebrates/Phlebotomus_papatasi/assembly/Phlebotomus_papatasi-2.0/README - Acesso em 5 jun 2012.

http://programmers.stackexchange.com/questions/59713/best-development-methodology-for-one-person - Acesso em 10 maio 2012.

http://agilemodeling.com - Acesso em 8 abr 2012.

http://extremeprogramming.org - Acesso em 8 abr 2012.

http://www.fsf.org - Acesso em 10 jul 2012.

http://opensource.org - Acesso em 10 jul 2012.

http://galaxyproject.org - Acesso em 10 mar 2012.

http://usegalaxy.org - Acesso em 10 mar 2012.

http://getgalaxy.org - Acesso em 13 mar 2012.

http://wiki.g2.bx.psu.edu/PublicGalaxyServers - Acesso em 10 mar 2012.

http://wiki.g2.bx.psu.edu/Support - Acesso em 10 mar 2012.

http://wiki.g2.bx.psu.edu/Mailing%20Lists - Acesso em 10 mar 2012.

http://en.wikipedia.org/wiki/FASTQ_format - Acesso em 10 mar 2011.

http://maq.sourceforge.net/fastq.shtml - Acesso em 10 abr 2010.

http://tritrypdb.org/tritrypdb - Acesso em 20 dez 2010.

http://www.solidsoftwaretools.com/gf/project/ecoli2x50/ - Acesso em 20 dez 2010.

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR066/SRR066482/SRR066482.sra - Acesso em 20 dez 2010.

http://user.list.galaxyproject.org/How-to-determine-version-of-Galaxy-from-main-page-td4223309.html - Acesso em 13 mar 2012.

http://www.ubuntu-br.org/download - Acesso em 10 mar 2012.

http://oss.oetiker.ch/mrtg/ - Acesso em 10 mar 2012.

http://maq.sourceforge.net/ - Acesso em 20 abr 2010.

http://sourceforge.net/projects/maq/files/ - Acesso em 10 mar 2012.

http://samtools.sourceforge.net - Acesso em 10 mar 2012.

http://sourceforge.net/projects/samtools/files/ - Acesso em 10 mar 2012.

http://www.sanger.ac.uk/resources/software/artemis/ - Acesso em 20 abr 2010.

http://www.r-project.org - Acesso em 20 abr 2010.

http://xyala.cap.ed.ac.uk/GenePool/ - Acesso em 20 abr 2010.

http://wiki.g2.bxpsu.edu/Admin/Tools/Tool%20Dependencies - Acesso em 10 mar 2012.

http://mira-assembler.sourceforge.net/ - Acesso em 22 dez 2010.

http://www.ebi.ac.uk/~zerbino/velvet/ - Acesso em 22 dez 2010.

http://stingray.biowebdb.org - Acesso em 22 dez 2010.

http://toolshed.g2.bx.psu.edu/ - Acesso em 10 mar 2012.

http://wiki.g2.bx.psu.edu/Learn - Acesso em 10 mar 2012.

http://maq.sourceforge.net/maq-man.shtml - Acesso em 10 abr 2012.

http://sourceforge.net/tracker/?func=detail&aid=2824334&group_id=191815&atid=938893 - Acesso em 10 abr 2012.

http://www.open-bio.org/wiki/Main_Page - Acesso em 10 abr 2012.

http://www.bioruby.org - Acesso em 10 abr 2012.

http://bioinf.scri.ac.uk/tablet/assembly-conversion.html - Acesso em 10 abr 2012.

http://seqanswers.com/wiki/How-to/RNASeq_analysis - Acesso em 10 abr 2012.

http://screencast.g2.bx.psu.edu/quickie_17_ftp_upload/flow.html - Acesso em 12 mar 2012.

http://wiki.g2.bx.psu.edu/Learn/Screencasts - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie10_custom_genome/flow.html - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie_11_illumina_se/flow.html - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie12_illumina_pe/flow.html - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie_13_fastq_basic/flow.html - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie_14_fastq_adv/flow.html - Acesso em 10 mar 2012.

http://pt.wikipedia.org/wiki/Parsing - Acesso em 10 mar 2012.

http://wiki.g2.bx.psu.edu/Admin/Data%20Integration - Acesso em 10 mar 2012.

http://wiki.g2.bx.psu.edu/Admin/NGS%20Local%20Setup - Acesso em 10 mar 2012.

http://bio-bwa.sourceforge.net/ - Acesso em 10 jul 2012.

http://genome.ucsc.edu/FAQ/FAQformat.html#format5.1 - Acesso em 10 mar 2012.

http://main.g2.bx.psu.edu/ - Acesso em 10 mar 2012.

http://wiki.g2.bx.psu.edu/Admin/Tools/Tool%20Config%20Syntax - Acesso em 10 mar 2012.

http://pt.wikipedia.org/wiki/Caminho_euleriano - Acesso em 11 ago 2012.

http://www.molecularevolution.org/resources/activities/velvet_and_bowtie_activity - Acesso em 20 mar 2012.

http://pt.wikipedia.org/wiki/Tabela_de_dispersão - Acesso em 17 jul 2012.

http://www.ebi.ac.uk/~zerbino/velvet/Manual.pdf - Acesso em 10 maio 2012.

http://www.homolog.us/blogs/2011/09/27/k-mer-sizes-for-genome-assembly/ - Acesso em 10 maio 2012.

https://wiki.nbic.nl/index.php/Raw_results_of_NGS_de_novo_assembly - Acesso em 10 maio 2012.

http://hannonlab.cshl.edu/fastx_toolkit/index.html - Acesso em 10 maio 2012.

http://sourceforge.net/projects/mira-assembler/files/MIRA/stable/ - Acesso em 10 maio 2012.

http://screencast.g2.bx.psu.edu/quickie8_solid_single_end/flow.html - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie9_solid_mate_pair/flow.html - Acesso em 10 mar 2012.

http://screencast.g2.bx.psu.edu/quickie_15_lastz_frag/flow.html - Acesso em 10 mar 2012.

http://www.ncbi.nlm.nih.gov/nuccore/AY370533 - Acesso em 10 ago 2011.

http://wiki.g2.bx.psu.edu/Admin/Config/Performance/Production%20Server - Acesso em 10 mar 2012.

http://bioinformatics.net.au/software.velvetoptimiser.shtml - Acesso em 10 ago 2012.

http://perl.org.br/Perldoc/V500807/Perlintro - Acesso em 13 maio 2012.

http://perldoc.perl.org - Acesso em 13 maio 2012.

http://perldoc.perl.org/perlintro.html - Acesso em 13 maio 2012.

http://www.python.org - Acesso em 13 maio 2012.

http://en.wikipedia.org/wiki/Python_(programming_language) - Acesso em 13 maio 2012.

http://www.gnu.org/software/bash/ - Acesso em 13 maio 2012.

http://www.w3schools.com/xml/xml_whatis.asp - Acesso em 13 maio 2012.

http://www.w3schools.com/html/default.asp - Acesso em 13 maio 2012.

http://pt.wikipedia.org/wiki/ASCII - Acesso em 10 mar 2012.

http://www.phrap.org - Acesso em 16 abr 2010.

Apêndices

Apêndice A - Roteiro de utilização para o fluxo de trabalho básico de montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina no protótipo *LASZLO @ GALAXY*

Carga dos arquivos "lane_2_1.txt" e "lane_2_2.txt" com a ferramenta Upload File

(1) Guia *Get Data* → Ferramenta *Upload File*;
(2) Preencher Campo *URL/Text* com a informação: "**ftp://192.168.0.114/lane_2_1.txt"**;
(3) Escolher a opção "**fastq**" no menu suspenso *File Format* (ou "Formato do arquivo");
(4) Pressionar o botão *Execute*;
(5) Após a carga bem sucedida do arquivo, os passos devem ser repetidos para o caminho de rede referente ao segundo arquivo: "**ftp://192.168.0.114/lane_2_2.txt"**.

A Figura A.1 exibe a tela da ferramenta *Upload File* e a correspondente entrada, no painel de histórico dos experimentos do usuário (*History*)[97], referente aos arquivos "lane_2_1.txt" e "lane_2_2.txt".

[97] Na plataforma GALAXY, o painel de histórico foi concebido para manter o registro das entradas do usuário e configurações de parâmetros utilizadas, garantindo, desta forma, que as análises realizadas possam ser precisamente reproduzidas (conceito de reprodutibilidade) (Blankenberg et al., 2011). Cabe destacar que, visualmente falando, o registro das etapas realizadas é mostrado de baixo para cima, o que pode causar estranheza a usuários iniciantes.

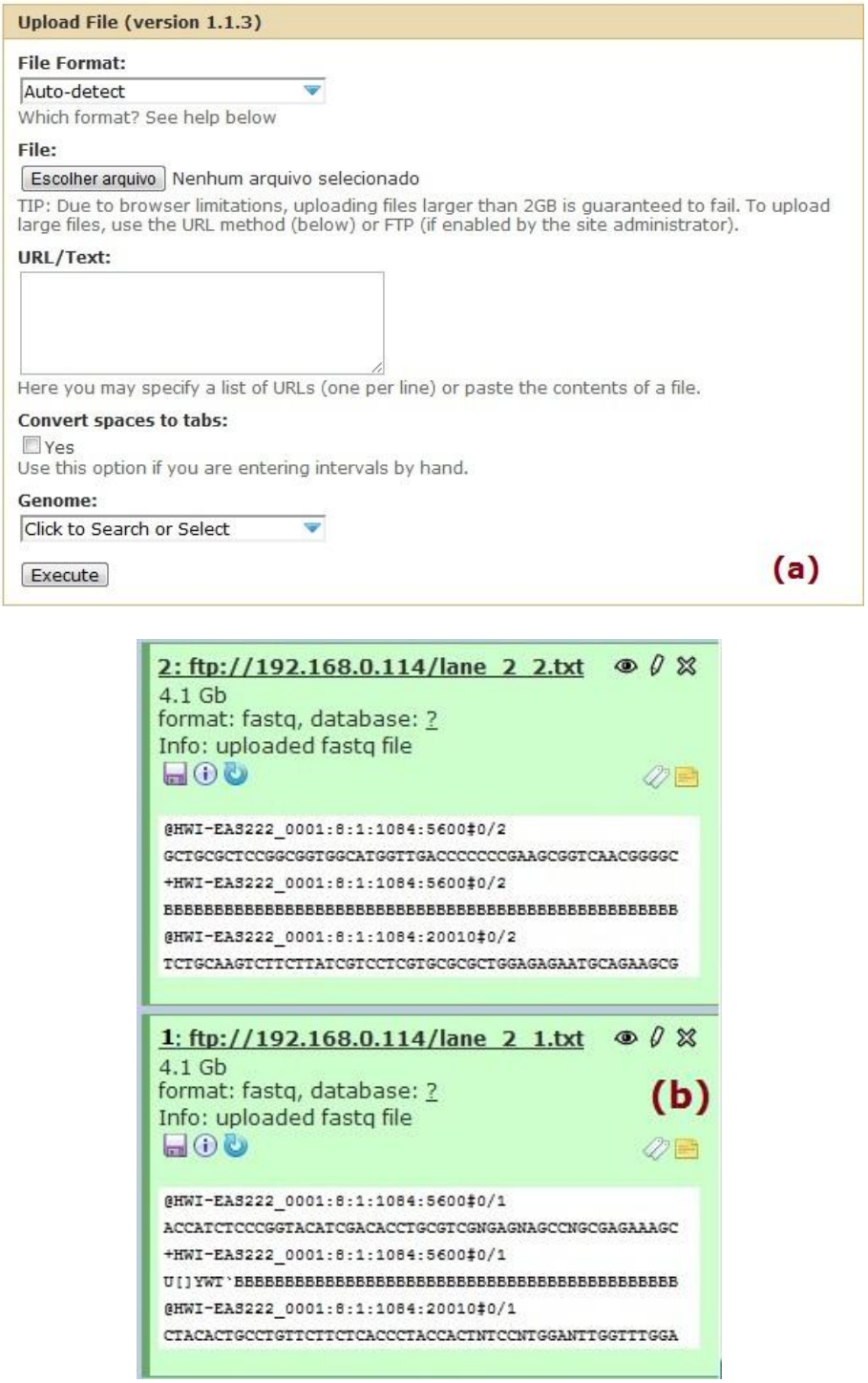


Figura A.1 - (a) Tela inicial da ferramenta *Upload File*. (b) Registro da carga dos arquivos de leituras pareadas "lane_2_1.txt" e "lane_2_2.txt" no painel de histórico do usuário.

Preparação dos dados em formato FASTQ com a ferramenta FASTQ Groomer

(1) Guia *NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ* → Ferramenta *FASTQ Groomer*;
(2) Selecionar a opção referente ao arquivo "**lane_2_1.txt**" no menu suspenso *File to groom* (ou "Arquivo a ser preparado");

(3) Selecionar a opção "**Illumina 1.3-1.7**" no menu suspenso *Input FASTQ quality scores type* (ou "Tipo de valor de qualidade do FASTQ de entrada")[98];
(4) Selecionar a opção "**Show advanced options**" (ou "Exibir opções avançadas") no menu suspenso *Advanced Options* (ou "Opções avançadas"), para exibí-las;
(5) Selecionar a opção "**Sanger (recommended)**" (ou "Sanger (recomendado)") no menu suspenso *Output FASTQ quality scores type* (ou "Tipo de valor de qualidade do FASTQ de saída");
(6) Demais opções da ferramenta podem ser mantidas com seus respectivos valores padrão;
(7) Pressionar o botão *Execute*;
(8) Após a conversão do arquivo para o formato FASTQ Sanger, os passos devem ser repetidos para o segundo arquivo: "**lane_2_2.txt**".

A Figura A.2 exibe a tela da ferramenta *FASTQ Groomer*. Já a Figura A.3, a correspondente entrada, no painel de histórico dos experimentos do usuário, referente à conversão dos arquivos "lane_2_1.txt" e "lane_2_2.txt" para o formato FASTQ Sanger.

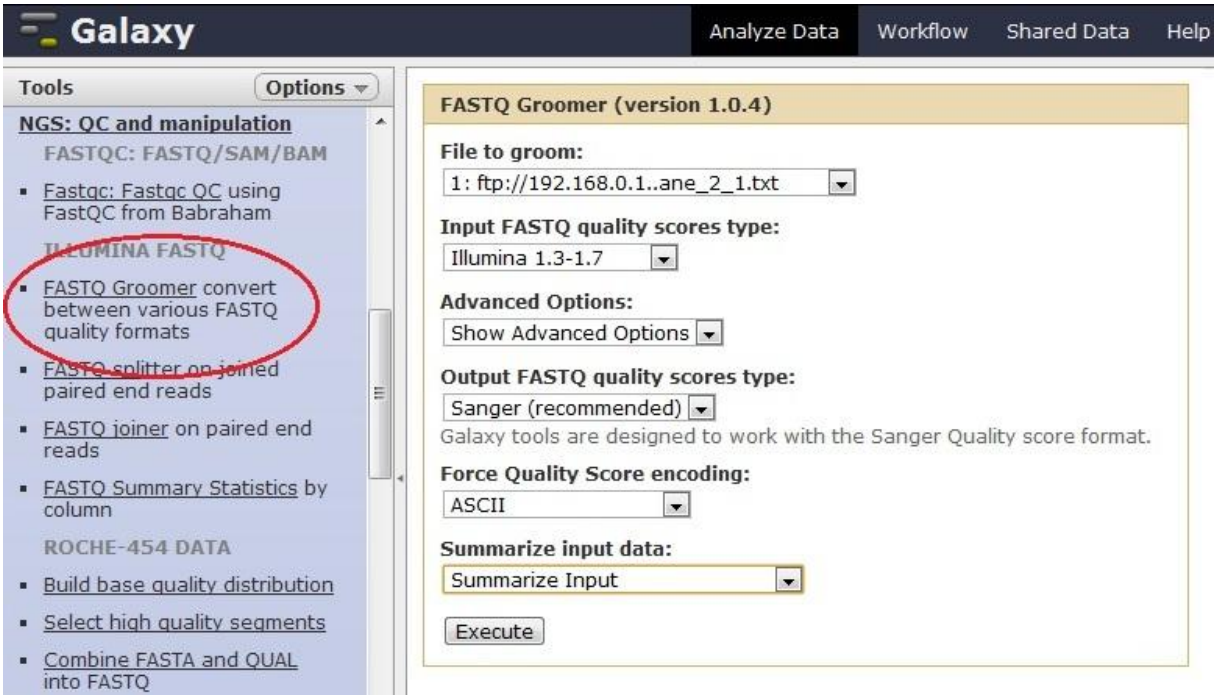


Figura A.2 - (a) Tela da ferramenta *FASTQ Groomer* e indicação de sua respectiva guia no painel de ferramentas.

[98] Uma vez que o arquivo em questão possuía codificação "Illumina 1.5", tal opção deveria ser selecionada.

Figura A.3 - Registro da conversão dos arquivos de leituras pareadas "lane_2_1.txt" e "lane_2_2.txt" para o formato FASTQ Sanger no painel de histórico do usuário.

Análise e refinamento dos valores de qualidade das leituras dos arquivos de entrada com as ferramentas FASTQ Summary Statistics e Boxplot

(1) Guia *NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ* → Ferramenta *FASTQ Summary Statistics*;
(2) No menu suspenso *FASTQ File* (ou "Arquivo FASTQ"), selecionar a opção referente ao primeiro conjunto de dados anteriormente preparado pela ferramenta *Groomer*: "**3: FASTQ Groomer on data 1**";
(3) Pressionar o botão *Execute*;
(4) Após a execução da ferramenta para o primeiro conjunto, os passos devem ser repetidos para o segundo: "**4: FASTQ Groomer on data 2**";
(5) Guia *Graph/Display Data* → Ferramenta *Boxplot*;
(6) No menu suspenso *Quality Statistics File* (ou "Arquivo com estatísticas de qualidade"), selecionar a opção referente à primeira execução da ferramenta *FASTQ Summary Statistics* (passo 2 acima): "**5: FASTQ Summary Statistics on data 3**". Uma vez que a ferramenta *Boxplot* é diretamente compatível com as colunas da ferramenta *FASTQ Summary Statistics*, o restante dos parâmetros pode ser deixado com seus respectivos valores padrão;
(7) Pressionar o botão *Execute*;
(8) Após a conclusão de execução da ferramenta, os passos de 5 a 7 devem ser aplicados ao conjunto referente à segunda rodada da ferramenta estatística (passo 4 acima): "**6: FASTQ Summary Statistics on data 4**".

A Figura A.4 exibe (a) a tela da ferramenta *FASTQ Summary Statistics* e (b) o registro correspondente no painel de histórico dos experimentos do usuário.

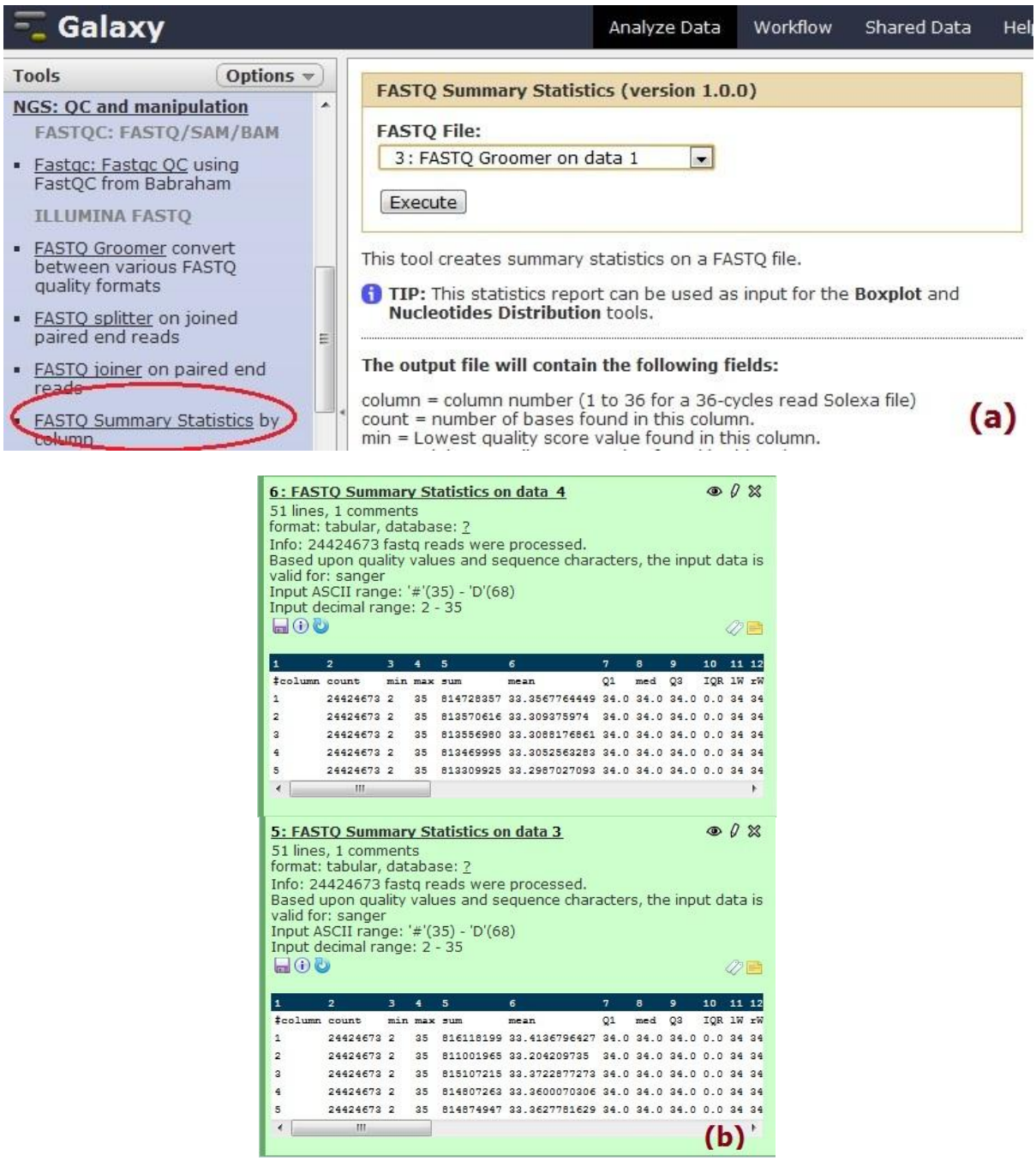


Figura A.4 - (a) Tela da ferramenta *FASTQ Summary Statistics* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *FASTQ Summary Statistics* sobre os arquivos anteriormente preparados pela ferramenta *FASTQ Groomer*.

A Figura A.5 exibe (a) a interface da ferramenta *Boxplot* e (b) as respectivas entradas no histórico do usuário.

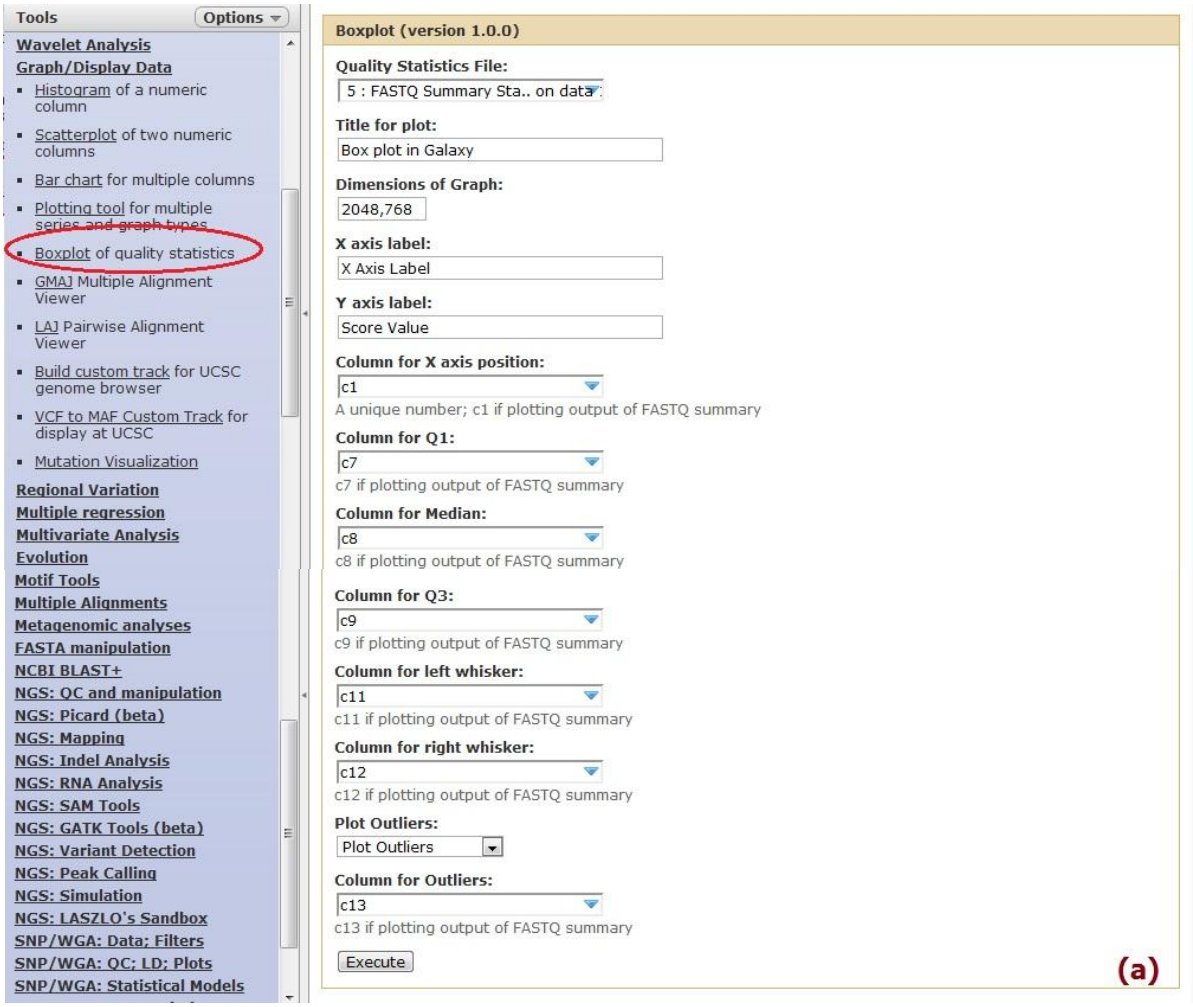


Figura A.5 - (a) Tela da ferramenta *Boxplot* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *Boxplot* sobre os resultados da ferramenta *FASTQ Summary Statistics*.

Preparação de arquivos pareados FASTQ com a ferramenta FASTQ joiner para eventual operação posterior de poda de bases de baixa qualidade

(1) Guia *NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ* → Ferramenta *FASTQ joiner*;
(2) No menu suspenso *Left-hand Reads* (ou "Leituras do primeiro conjunto"), selecionar a opção referente ao primeiro conjunto de dados anteriormente preparado pela ferramenta *Groomer*: "**3: FASTQ Groomer on data 1**";

(3) No menu suspenso *Right-hand Reads* (ou "Leituras do segundo conjunto"), selecionar a opção referente ao segundo conjunto de dados anteriormente preparado pela ferramenta *Groomer*: "**4: FASTQ Groomer on data 2**";
(4) Pressionar o botão *Execute*.

A Figura A.6 exibe (a) a tela da ferramenta *FASTQ joiner* e (b) o registro correspondente no painel de histórico dos experimentos do usuário.

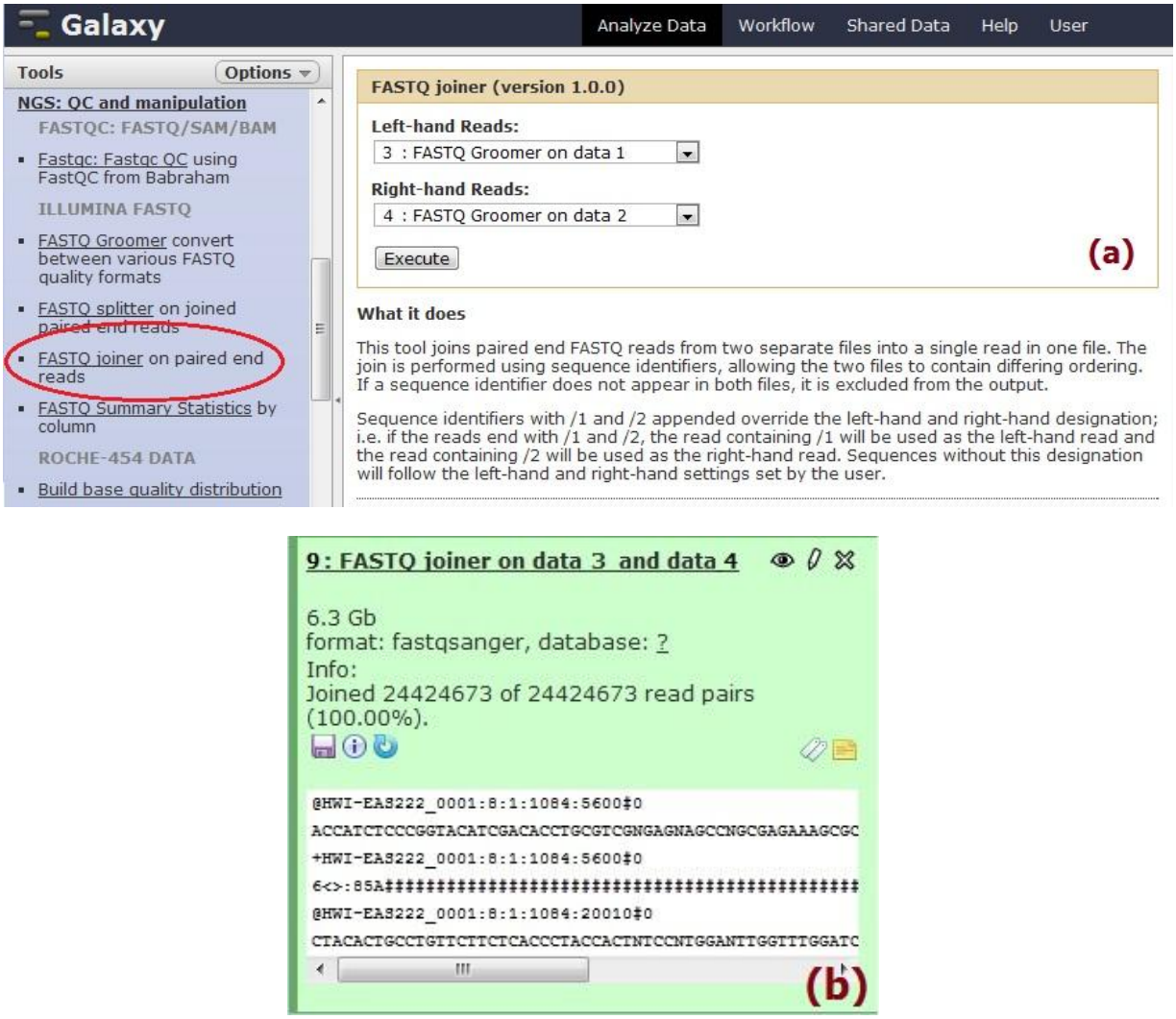


Figura A.6 - (a) Tela da ferramenta *FASTQ joiner* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *FASTQ joiner* sobre os arquivos anteriormente preparados pela ferramenta *FASTQ Groomer*.

Eventual poda de bases de baixa qualidade com a ferramenta FASTQ Trimmer

(1) Guia *NGS: QC and manipulation*, bloco de ferramentas *Generic FASTQ Manipulation* (ou "Manipulação genérica de FASTQ") → Ferramenta *FASTQ Trimmer*;
(2) No menu suspenso *FASTQ file*, selecionar a opção referente ao conjunto de dados produzido no passo anterior pela ferramenta *joiner*: "**9: FASTQ joiner on data 3 and 4**";

(3) No menu suspenso *Define Base Offsets as* (ou "Definir deslocamentos de base como"), manter o valor padrão: "**Absolute Values**" (ou "Valores absolutos")[99];
(4) Inserir, nos campos *Offset from 5' end* (ou "Deslocamento a partir da extremidade 5' ") e *Offset from 3' end* (ou "Deslocamento a partir da extremidade 3' "), os devidos valores no que diz respeito ao número desejado de bases a serem podadas[100];
(5) Os demais valores da ferramenta podem ser mantidos como padrão;
(6) Pressionar o botão *Execute*.

A Figura A.7 exibe (a) a tela da ferramenta *FASTQ Trimmer* e (b) o registro correspondente no painel de histórico dos experimentos do usuário.

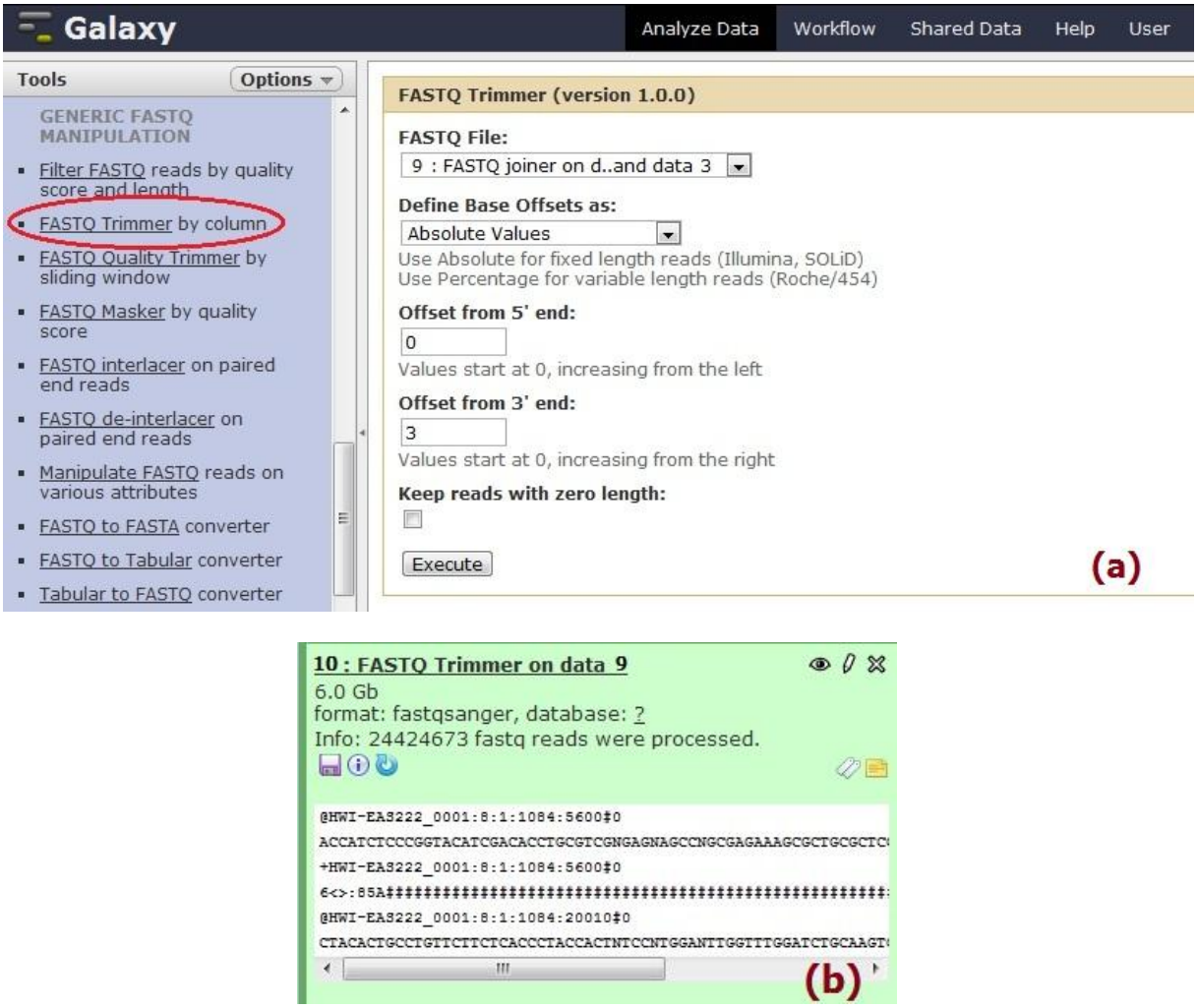


Figura A.7 - (a) Tela da ferramenta *FASTQ Trimmer* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *FASTQ Trimmer* sobre o produto da ferramenta *FASTQ joiner*.

[99] Por se tratar de arquivo de leitura com valor fixo de tamanho (arquivo de Illumina, no caso), o uso de valores absolutos é a opção mais recomendada.
[100] Tal como um arquivo típico da tecnologia Illumina, ambos os conjuntos de leituras pareadas apresentavam alta qualidade na extremidade 5' das leituras e baixa qualidade na extremidade 3'. Desta forma, o campo *Offset from 5' end* foi deixado no valor padrão "**0**" (zero) (de maneira a não haver poda de bases nessa extremidade), enquanto o campo *Offset from 3' end* foi preenchido com o valor "**3**" (de maneira a serem podadas as bases 49-51).

Usando a ferramenta FASTQ splitter para separar arquivos que tenham sido eventualmente unidos pela ferramenta FASTQ joiner

(1) Guia *NGS: QC and manipulation*, bloco de ferramentas *ILLUMINA FASTQ* → Ferramenta *FASTQ splitter*;
(2) No menu suspenso *FASTQ Reads* (ou "Leituras FASTQ"), selecionar a opção referente ao conjunto de dados resultante da ferramenta *Trimmer*: "**10: FASTQ Trimmer on data 9**";
(3) Pressionar o botão *Execute*.

Ao contrário da ferramenta *joiner*, a ferramenta *splitter* gera dois arquivos na saída. A Figura A.8 exibe (a) a tela da ferramenta *FASTQ splitter* e (b) o registro, dos dois arquivos gerados, no painel de histórico dos experimentos do usuário.

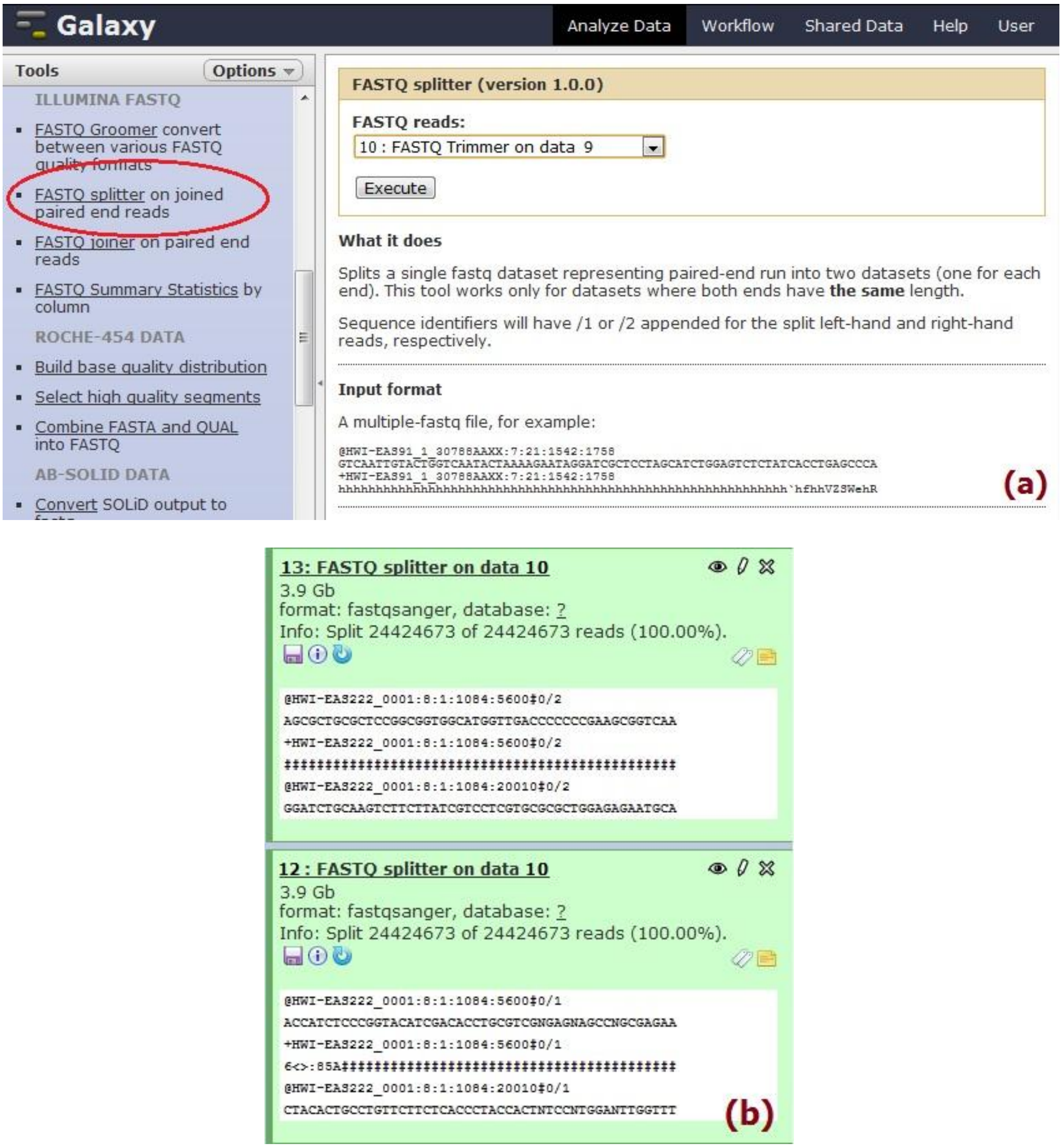


Figura A.8 - (a) Tela da ferramenta *FASTQ splitter* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *FASTQ splitter* sobre o arquivo resultante da ferramenta *FASTQ Trimmer*.

Carga do arquivo com o genoma de referência a partir do próprio computador do usuário (sem necessidade de uso de FTP)

(1) Guia *Get Data* → Ferramenta *Upload File*;
(2) Pressionar o botão *Escolher arquivo* e, a partir da nova janela aberta, selecionar o arquivo com o genoma de referência: "**LmexicanaGenomic_TriTrypDB-4.0.fasta**";
(3) Escolher a opção "**fasta**" no menu suspenso *File Format*;
(4) Pressionar o botão *Execute*.

A Figura A.9 exibe a entrada, no painel de histórico dos experimentos do usuário, referente ao arquivo com o genoma de referência.

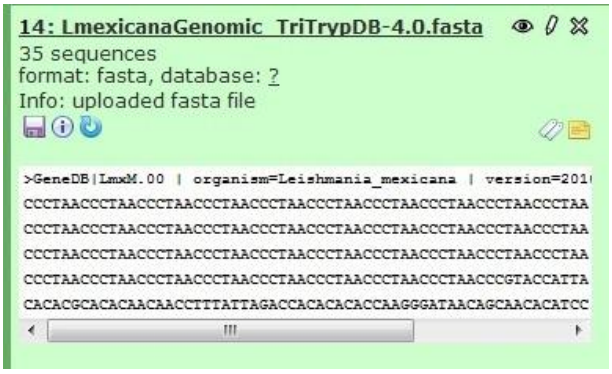


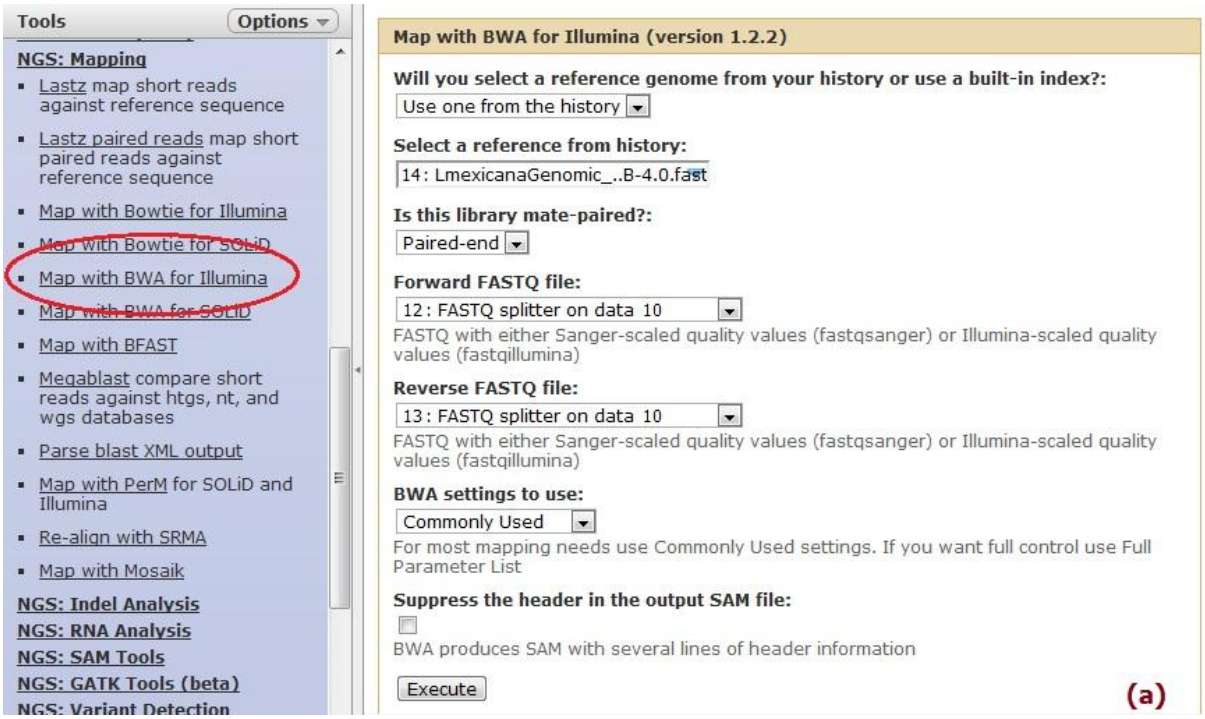
Figura A.9 - Arquivo do genoma de referência inserido no fluxo de trabalho.

Mapeamento das leituras de Solexa/Illumina

(1) Guia *NGS: Mapping* → Ferramenta *Map with BWA for Illumina*;
(2) No menu suspenso *Will you select a reference genome from your history or use a built-in index?* (ou "O genoma de referência será selecionado do histórico ou um índice pronto será usado?"), selecionar a opção "**Use one from the history**" (ou "Usar um do histórico"). Com isso, o menu suspenso *Select a reference from history* (ou "Selecione uma referência do histórico") é habilitado e, nele, deve ser escolhida a opção referente ao arquivo com o genoma de referência: "**14: LmexicanaGenomic_TriTrypDB-4.0.fasta**";
(3) No menu suspenso *Is this library mate-paired?* (ou "Esta biblioteca é *mate-paired*?"), selecionar a opção referente ao valor "**Paired-End**"[101];
(4) No menu suspenso *Forward FASTQ file* (ou "Arquivo FASTQ de leituras diretas"), selecionar a opção referente ao primeiro conjunto de dados proveniente da ferramenta *splitter*: "**12: FASTQ splitter on data 10**";
(5) No menu suspenso *Reverse FASTQ file* (ou "Arquivo FASTQ de leituras reversas"), selecionar a opção referente ao segundo conjunto de dados proveniente da ferramenta *splitter*: "**13: FASTQ splitter on data 10**";
(6) No menu suspenso *BWA settings to use* (ou "Parâmetros BWA a serem usados), pode ser mantida a opção padrão "**Commonly used**" (ou "Comumente usados")[102];
(7) As demais opções podem ser mantidas com seus respectivos valores padrão;
(8) Pressionar o botão *Execute*.

A Figura A.10 exibe (a) a tela da ferramenta *Map with BWA for Illumina* e (b) o resultado de sua execução no painel de histórico dos experimentos do usuário:

[101] Uma vez que os dados eram pareados, tal opção é a mais recomendada.
[102] Para fins de simplificação do fluxo de trabalho básico.



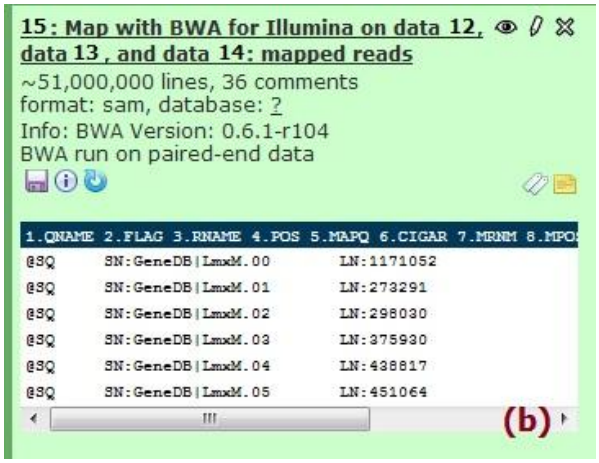


Figura A.10 - (a) Tela da ferramenta *Map with BWA for Illumina* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *Map with BWA for Illumina* sobre os arquivos de leituras provenientes da ferramenta *splitter* e o arquivo com o genoma de referência "LmexicanaGenomic_TriTrypDB-4.0.fasta".

Filtragem de dados com o pacote SAMtools

(1) Guia *NGS: SAM Tools* → Ferramenta *Filter SAM*;
(2) No menu suspenso *Select dataset to filter* (ou "Selecione o conjunto de dados a ser filtrado"), selecionar a opção referente ao arquivo SAM gerado na etapa anterior: "**15: Map with BWA for Illumina on data 12, data 13, and data 14: mapped reads**";
(3) Na seção *Flags* (ou "Sinalizadores") da ferramenta, por meio do botão *Add new Flag* (ou "Adicione novo sinalizador"), incluir, sucessivamente, os seguintes sinalizadores e respectivas configurações de parâmetros:

- "**Read is paired**" (ou "Leitura pareada") no menu suspenso *Type* (ou "Tipo") e selecionar o valor "**Yes**" no campo *Set the states for this flag* (ou "Selecione os estados para este sinalizador");
- "**Read is mapped in a proper pair**" (ou "Leitura mapeada em um par apropriado") no menu suspenso *Type* e selecionar o valor "**Yes**" no campo *Set the states for this flag*;
- "**The read is unmapped**" (ou "Leitura não mapeada") no menu suspenso *Type* e selecionar o valor "**No**" (ou "Não")[103] no campo *Set the states for this flag*;

(4) Pressionar o botão *Execute*.

A Figura A.11 exibe a tela da ferramenta *Filter SAM*. Já a Figura A.12, o resultado de sua execução no painel de histórico dos experimentos do usuário.

[103] A combinação de parâmetros "The read is unmapped"/"No", apesar de soar estranha, é uma forma de assegurar que todas as leituras não mapeadas serão deixadas de fora do filtro, ou, em outras palavras, que somente as leituras mapeadas serão levadas em consideração (12 - *Illumina Paired Ends* (http://screencast.g2.bx.psu.edu/quickie12_illumina_pe/flow.html).

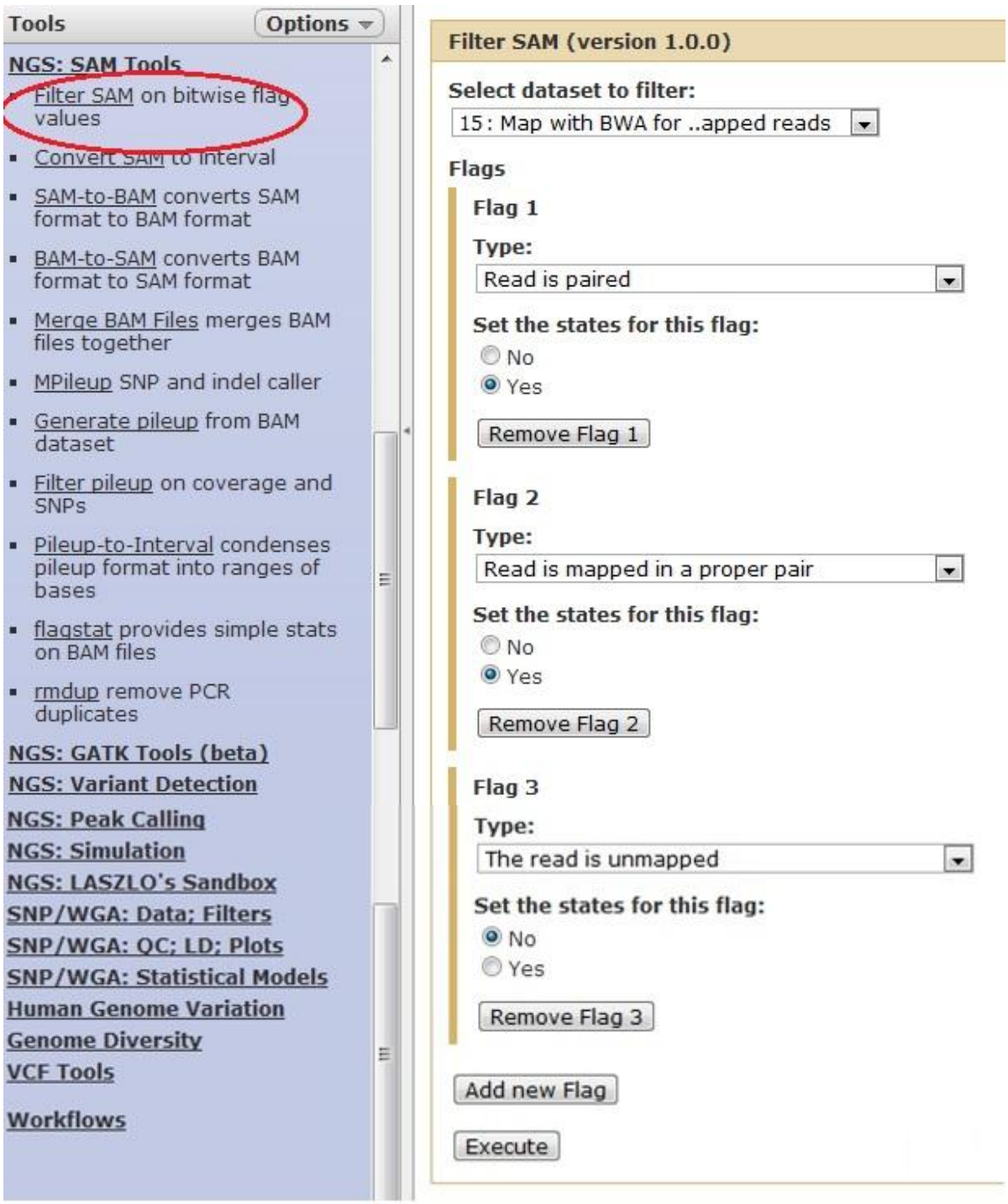


Figura A.11 - Tela da ferramenta *Filter SAM* e indicação de sua respectiva guia no painel de ferramentas.

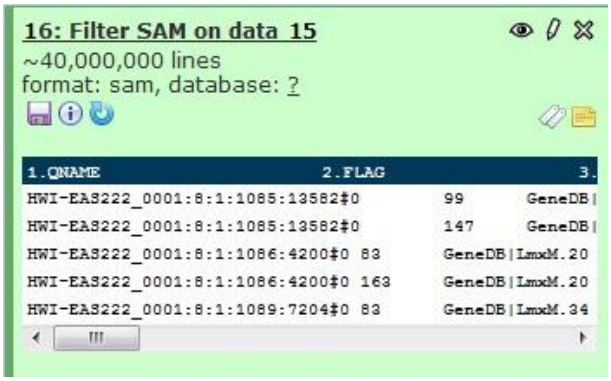


Figura A.12 - Registro, no painel de histórico do usuário, da aplicação da ferramenta *Filter SAM* sobre o arquivo SAM resultante do mapeamento das leituras pela ferramenta BWA.

Conversão do formato SAM para o formato BAM

(1) Guia *NGS: SAM Tools* → Ferramenta *SAM-to-BAM*;
(2) No menu suspenso *Choose the source for the reference list* (ou "Escolher a origem da lista de referência"), selecionar a opção "**History**" (ou "Histórico"). Com isso, o menu suspenso *Using reference file* (ou "Usando o arquivo de referência") é habilitado e, nele, pode ser escolhida a opção referente ao arquivo com o genoma de referência: "**14: LmexicanaGenomic_TriTrypDB-4.0.fasta**";
(3) No menu suspenso *Convert SAM file* (ou "Converter arquivo SAM"), escolher a opção referente ao arquivo obtido na etapa anterior: "**16: Filter SAM on data 15**";
(4) Pressionar o botão *Execute*.

A Figura A.13 exibe (a) a tela da ferramenta *SAM-to-BAM* e (b) o resultado de sua execução no painel de histórico dos experimentos do usuário.

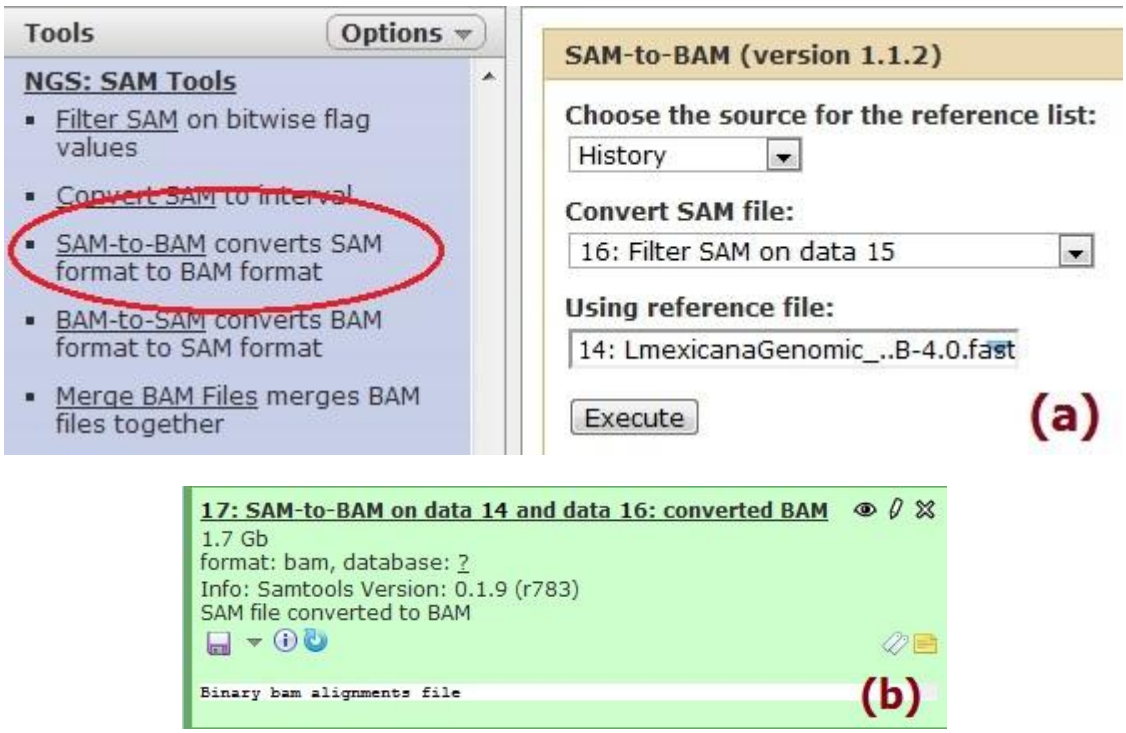


Figura A.13 - (a) Tela da ferramenta *SAM-to-BAM* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *SAM-to-BAM* sobre o arquivo SAM, filtrado pela ferramenta *Filter SAM*, mais o arquivo com o genoma de referência "LmexicanaGenomic_TriTrypDB-4.0.fasta".

Verificação da posição de mapeamento das leituras em relação à referência (formato pileup)

(1) Guia *NGS: SAM Tools* → Ferramenta *Generate pileup*;
(2) No menu suspenso *Will you select a reference genome from your history or use a built-in index?* (ou "Você irá selecionar uma genoma de referência do seu histórico ou usará um índice pronto?"), selecionar a opção "**Use one from history**" (ou "Usar um do histórico"). Com isso, o menu suspenso *Select a reference genome* (ou "Selecione um genoma de referência") é habilitado e, nele, pode ser escolhida a opção referente ao arquivo com o genoma de referência: "**14: LmexicanaGenomic_TriTrypDB-4.0.fasta**";
(3) No menu suspenso *Select the BAM file to generate the pileup file for* (ou "Selecione o arquivo BAM para o qual será gerado o *pileup*"), escolher a opção referente ao arquivo obtido na etapa anterior: "**17: SAM-to-BAM on data 14 and data 16: converted BAM**";
(4) Os outros parâmetros da ferramenta podem ser mantidos como padrão;
(5) Pressionar o botão *Execute*.

A Figura A.14 exibe (a) a tela da ferramenta *Generate pileup* e (b) o resultado de sua execução no painel de histórico dos experimentos do usuário.

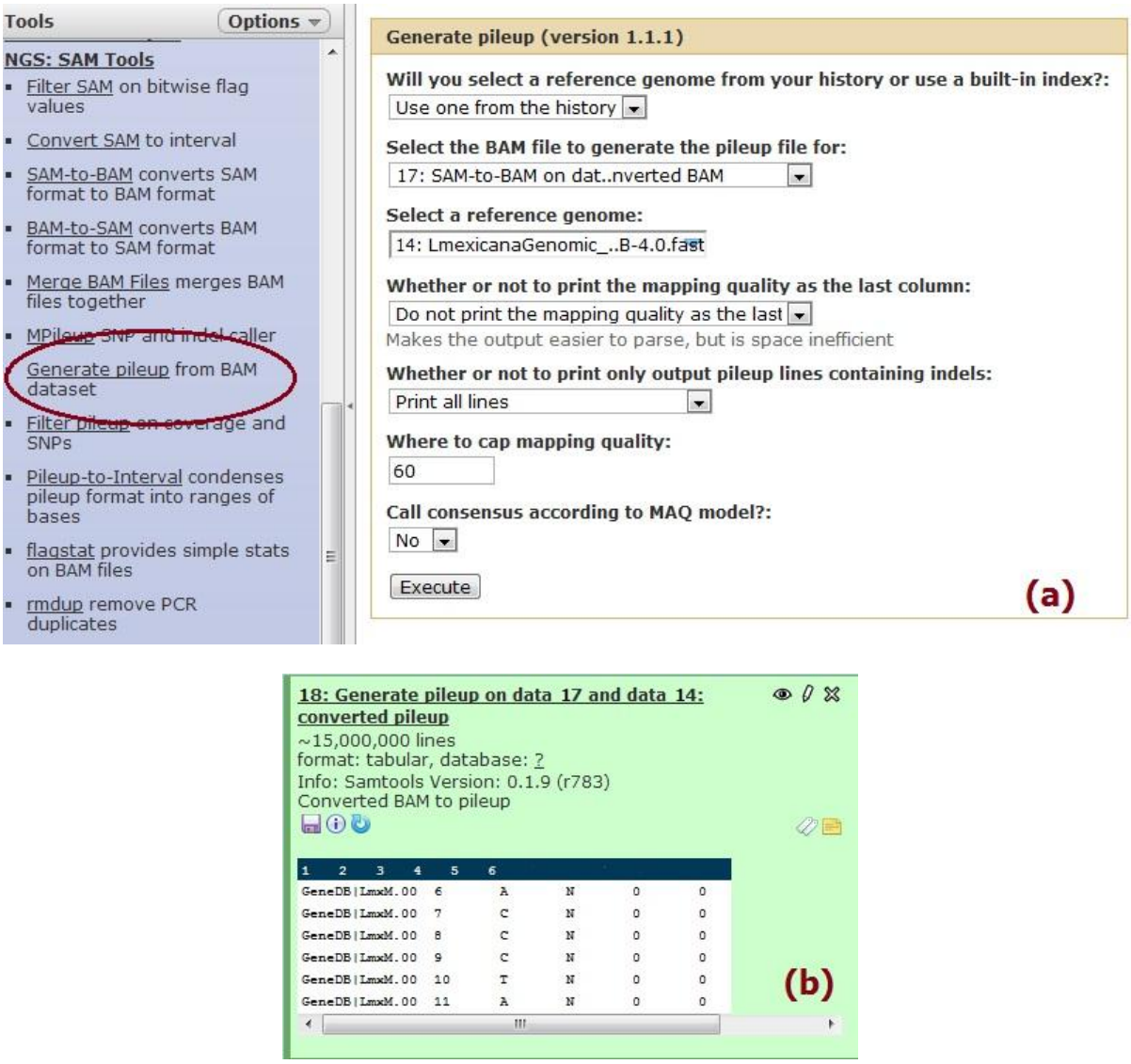


Figura A.14 - (a) Tela da ferramenta *Generate pileup* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *Generate pileup* sobre o arquivo BAM, convertido na etapa anterior, mais o arquivo com o genoma de referência "LmexicanaGenomic_TriTrypDB-4.0.fasta".

Conversão do formato pileup para o formato FASTQ

(1) Guia *NGS: LASZLO's Sandbox* → Bloco de ferramentas *MORE SAMTOOLS TOYS* → Ferramenta *SAMTOOLS pileup-to-fastQ converter*;
(2) No menu suspenso *SAMTOOLS pileup file* (ou "Arquivo *pileup* SAMTOOLS"), selecionar a opção referente ao arquivo de *pileup* produzido na etapa anterior: "**18: Generate pileup on data 17 and data 14: converted pileup**";
(3) Pressionar o botão *Execute*.

A Figura A.15 exibe a tela da ferramenta *SAMTOOLS pileup-to-fastQ converter*, ao passo que o resultado de sua execução, conforme apresentado no painel de histórico dos experimentos do usuário, é mostrado na Figura A.16.

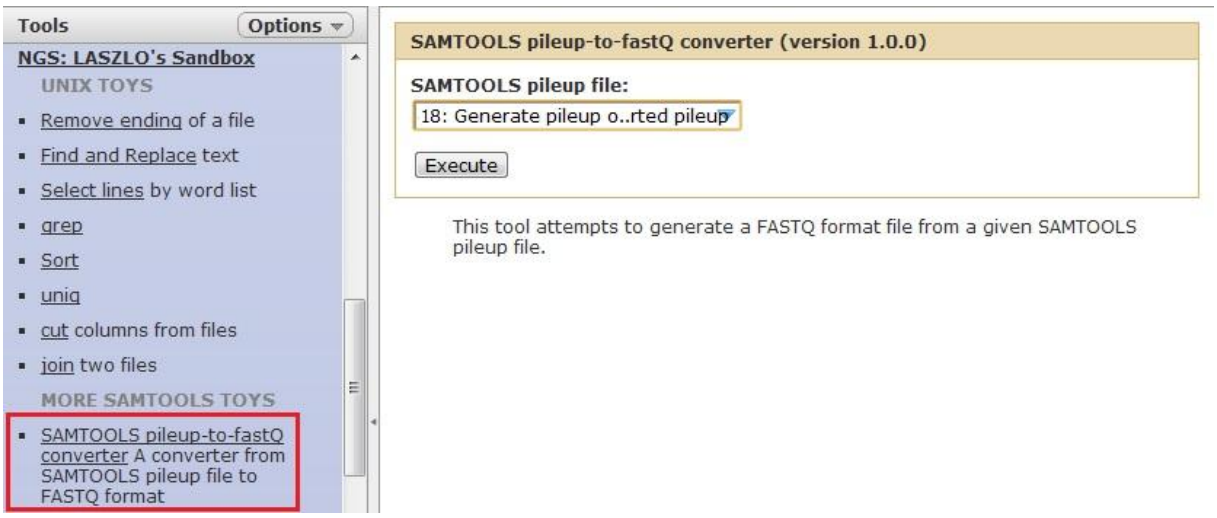


Figura A.15 - Ferramenta *SAMTOOLS pileup-to-fastq converter*.

Figura A.16 - Registro, no painel de histórico do usuário, da execução da ferramenta *SAMTOOLS pileup-to-fastq converter*.

Conversão do formato FASTQ para o formato FASTA

(1) Guia *Convert formats* → Ferramenta *FASTQ to FASTA*;
(2) No menu suspenso *FASTQ file to convert* (ou "Arquivo FASTQ a ser convertido"), selecionar a opção referente ao arquivo gerado na etapa anterior "**19: SAMTOOLS pileup-to-fastQ converter on data 18**";
(3) Pressionar o botão *Execute*.

A Figura A.17 exibe (a) a tela da ferramenta *FASTQ to FASTA* e (b) o resultado de sua execução no painel de histórico dos experimentos do usuário.

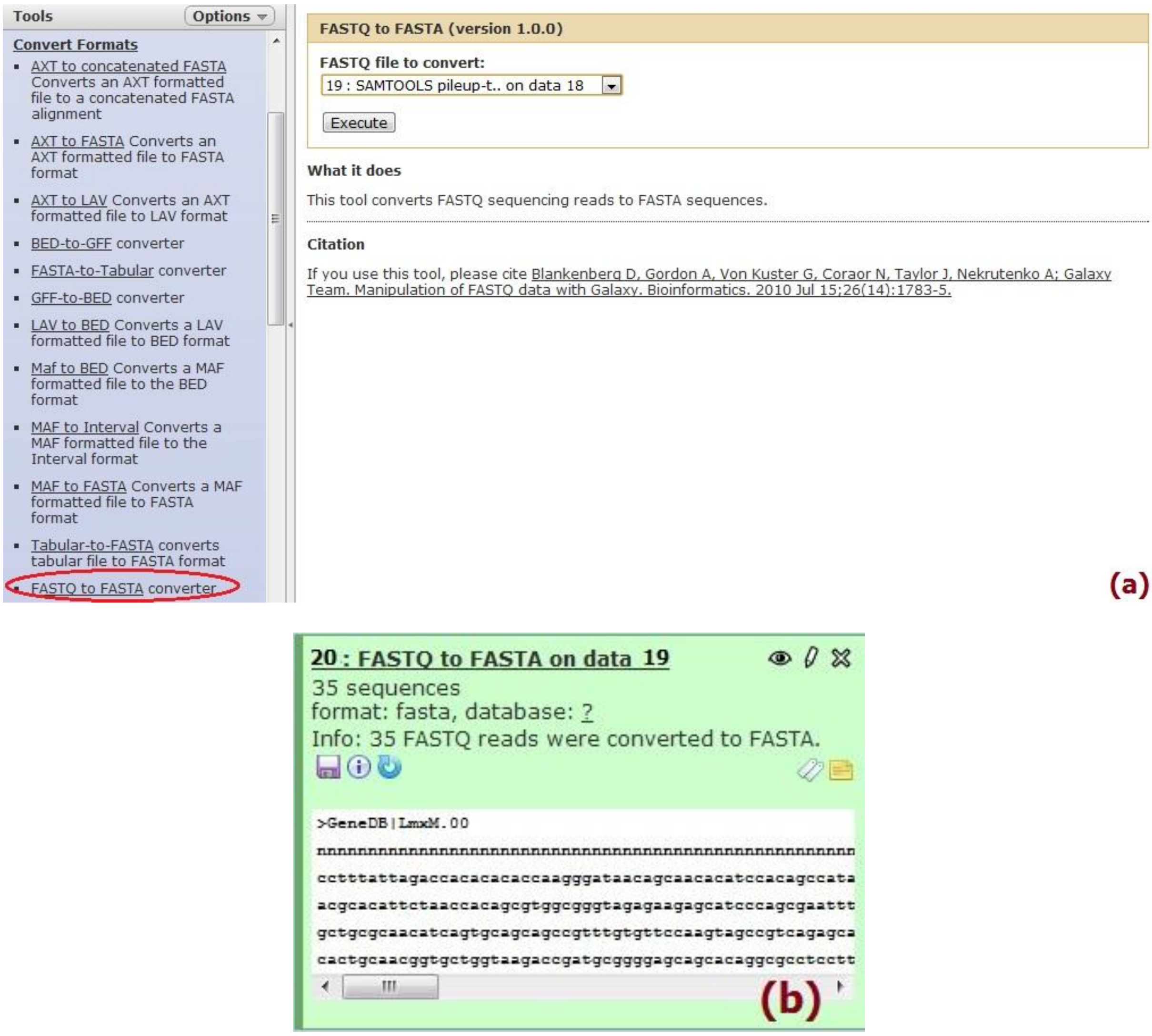


Figura A.17 - (a) Tela da ferramenta *FASTQ to FASTA* e indicação de sua respectiva guia no painel de ferramentas. (b) Registro, no painel de histórico do usuário, da aplicação da ferramenta *FASTQ to FASTA* sobre o arquivo FASTQ gerado pela ferramenta personalizada *SAMTOOLS pileup-to-fastQ converter* na etapa anterior.

Apêndice B - Informações adicionais sobre as linguagens de programação e marcação utilizadas no desenvolvimento do protótipo

Perl[104]

Linguagem de programação de *scripting*, estável, de uso geral e multiplataforma — roda, por exemplo, em ambientes Unix, Linux, Windows, *Macintosh*, etc. — originalmente elaborada para a manipulação e processamento de arquivos em formato de texto, Perl, ou *Practical extraction and report language*, atualmente é utilizada para uma infinidade de tarefas, tais como administração de sistemas, desenvolvimento *Web* e de interfaces gráficas, entre outras. Foi elaborada por Larry Wall, na década de 1980, para ser prática (fácil de usar, eficiente e completa), ao invés de bonita (enxuta, elegante e mínima). A linguagem combina recursos da linguagem C e de utilitários do sistema operacional Unix, como *sed*, *awk* e *sh*, sendo poderosa, também, para lidar com expressões regulares. É relativamente fácil de ser aprendida, de uso livre e inclui suporte tanto para programação procedural, quanto para programação orientada a objetos.

Python[105]

Linguagem de programação desenvolvida no início dos anos 90, por Guido Van Rossum, também de uso geral, livre e multiplataforma — roda em Windows, Linux/Unix e Mac OS X, por exemplo. Sua filosofia de desenvolvimento enfatiza a inteligibilidade do código e, por esse motivo, sua sintaxe é dita como concisa (clara) e expressiva (mais coisas são feitas com menos linhas de código, se comparada a outras linguagens). É usada em vários domínios de aplicação, como desenvolvimento *Web*, acesso a bases de dados, interfaces gráficas (GUIs), aplicações científicas, etc. Um de seus pontos fortes é sua vasta biblioteca padrão, a qual inclui módulos pré-escritos para as mais diversas finalidades.

Bash script[106]

Dois grandes módulos compõem os sistemas operacionais Unix/Linux: o *kernel* (ou núcleo) e o *shell*. O primeiro é responsável pelo gerenciamento de processos e recursos, como memória, CPU e discos, ficando residente em memória enquanto o computador estiver em operação e fornecendo serviços básicos para outros componentes do sistema operacional e para aplicações em execução. O segundo módulo é o responsável por atuar como interface

[104] http://perl.org.br/Perldoc/V500807/Perlintro; http://perldoc.perl.org e http://perldoc.perl.org/perlintro.html.
[105] http://www.python.org; http://en.wikipedia.org/wiki/Python_(programming_language).
[106] http://www.gnu.org/software/bash/.

entre o usuário e o sistema operacional e também é conhecido como interface de linha de comando (CLI). O *shell* interpreta os comandos emitidos pelo usuário e, além disso, implementa uma linguagem de programação que dispõe de comandos de decisão, de controle de fluxo, funções, etc., permitindo a elaboração de pequenos programas, denominados *shell scripts*. Vários *shells* foram desenvolvidos ao longo da história do Unix. O primeiro foi o Bourne shell (sh) e, depois, outros surgiram, como o C shell (csh), o Korn shell (ksh), o Tenex/Tops C shell (tcsh) e o Bourne-Again shell (Bash) aqui mencionado, o qual é o padrão em diversas distribuições Linux (Saade, 2001).

XML[107]

XML significa *eXtensible Markup Language*, algo como "linguagem de marcação extensível", e nada mais é do que uma linguagem de marcação de dados, ou seja, uma linguagem que utiliza um formato específico para descrever os dados de maneira estruturada. Ela foi desenvolvida para estruturar, transportar e armazenar todos os tipos de dados (por exemplo, registros de clientes e catálogos *Web*), com a vantagem de fornecer a extensibilidade — daí seu nome — e a flexibilidade das *tags* (marcações) — as quais não são pré-definidas, podendo ser declaradas, em tempo de programação. O surgimento da Internet, entre outras coisas, difundiu a necessidade de troca constante de informações entre usuários, sistemas e, até mesmo, diretamente entre corporações. XML, então, passou a ganhar destaque como um padrão para a transferência de dados, com capacidade para ser usado em qualquer plataforma e ambiente (De Faria, 2005). Cabe a outro programa entender o que cada *tag* do arquivo XML representa e, com isso, tomar a devida decisão a respeito.

HTML[108]

Enquanto XML foi desenvolvida, como vimos, para transportar e armazenar dados, HTML — sigla para *HyperText Markup Language*, ou "linguagem de marcação de hipertextos" — foi concebida para exibir dados. Trata-se de um padrão mundial, não-proprietário e independente de plataforma, para exibição de documentos (páginas) na Internet. Um documento HTML contém apenas informações sobre a forma como devem ser exibidos textos e imagens na tela. Isso é feito, também, através de *tags* (marcações). Ao abrir uma página HTML em um navegador *Web*, tais marcações são lidas, interpretadas e exibidas conforme as especificações contidas nas *tags* (Campos, 2004).

[107] O URL http://www.w3schools.com/xml/xml_whatis.asp fornece boas explicações sobre a linguagem XML.
[108] O URL http://www.w3schools.com/html/default.asp fornece boas explicações sobre a linguagem HTML.

Apêndice C - Informações adicionais sobre alguns formatos de arquivo tratados no trabalho

FASTQ

O FASTQ é um formato do tipo texto que estende a funcionalidade do formato FASTA (Pearson; Lipman, 1988), armazenando tanto a sequência biológica da leitura (usualmente em nucleotídeos), quanto seus valores de qualidade PHRED correspondentes para cada base, conforme mostrado na Figura C.1.

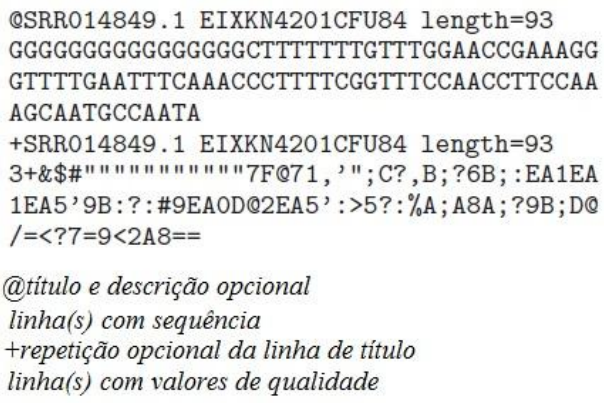


Figura C.1 - Definição do formato FASTQ.
Fonte: Modificado de Cock et al., 2010, p. 1769.

Ele foi originalmente desenvolvido no Wellcome Trust Sanger Institute, mas, atualmente, tem ganhado força como padrão *de facto* para representar os dados de saída de plataformas de alta vazão[109]. Os valores de qualidade PHRED são originalmente derivados do programa de *base-calling* Phred (Ewing et al., 1998, Ewing; Green, 1998), criado na época dos sequenciadores automáticos convencionais e que se tornou o mais utilizado para a realização dessa etapa do processo de sequenciamento. De maneira resumida, quanto menor o valor inteiro de qualidade, maior a probabilidade da base associada a esse valor ter sido atribuída de maneira incorreta. O Quadro C.1 sumariza as relações existentes entre os valores PHRED e suas respectivas probabilidades e acurácias.

[109] http://en.wikipedia.org/wiki/FASTQ_format.

Quadro C.1 - Valores de qualidade PHRED e suas respectivas probabilidades de erro e acurácias.

Valor de qualidade PHRED	Probabilidade de erro na atribuição de base	Acurácia da atribuição de base
10	1 em 10	90%
20	1 em 100	99%
30	1 em 1000	99,9%
40	1 em 10000	99,99%
50	1 em 100000	99,999%

Fonte: Modificado de Skrabanek, 2012, p.8.

Cock et al. (2010) lembram que o surgimento do programa Phred introduziu um novo formato de arquivo, conhecido como QUAL, para armazenar os valores de qualidade. Nesse formato, de estrutura similar ao formato FASTA, os valores de qualidade são apresentados como números inteiros, os quais são separados, entre si, por espaços em branco, conforme ilustrado na Figura C.2.

```
>SRR014849.1 EIXKN4201CFU84 length=93
18 10 5 3 2 1 1 1 1 1 1 1 1 1 1 22 37
31 22 16 11 6 1 26 34 30 11 33 26 30 21
33 26 25 36 32 16 36 32 16 36 32 20 6
24 33 25 30 25 2 24 36 32 15 35 31 17
36 32 20 6 25 29 20 30 25 4 32 26 32 23
32 26 30 24 33 26 35 31 14 28 27 30 22
28 24 27 17 32 23 28 28
```

Figura C.2 - Exemplo de formato QUAL.
Fonte: Extraído de Cock et al, 2010, p.1768.

A possibilidade de armazenar os valores de qualidade PHRED como um único caractere (ou *byte*) e sem a necessidade de espaços em branco entre eles, por meio da utilização de caracteres específicos do código ASCII[110], caracterizou o formato FASTQ como sendo uma maneira simples de codificação, porém de significativa eficiência, em termos de requisitos de espaço, quando comparada com a abordagem do conjunto de arquivos FASTA-QUAL. Cabe frisar aqui, entretanto, um aspecto importante sobre o formato FASTQ: pelo fato dele ter surgido sem uma padronização rígida, três variantes de codificação (incompatíveis entre si) acabaram sendo criadas, ao longo do tempo, para caracterizar as qualidades das bases. Basicamente, tais variantes foram ocasionadas pela existência de três maneiras diferentes de se calcular a probabilidade de erro de atribuição de base e de existirem, também, diferentes formas dessa probabilidade estar representada por algum caractere do código ASCII. Tais variantes são referidas como Sanger, Solexa e Illumina e suas características principais são resumidas no Quadro C.2 (Cock et al., 2010; Skrabanek, 2012).

[110] http://pt.wikipedia.org/wiki/ASCII.

Quadro C.2 - Variantes do formato FASTQ.

Descrição ; nome OBF	Caracteres ASCII		Valor de qualidade	
	Faixa	*Offset**	Tipo de cálculo	Faixa
Padrão Sanger / Illumina 1.7+ `fastq-sanger`	33 a 126	33	PHRED	0 a 93
Solexa / Illumina inicial (< 1.3) `fastq-solexa`	59 a 126	64	Solexa	-5 a 62
Illumina 1.3+ `fastq-illumina`	64 a 126	64	PHRED	0 a 62

Fonte: Modificado de Skrabanek, 2012, p.8.
Nota:
* *Offset* (ou "deslocamento") significa a diferença entre a faixa de valores decimais do código ASCII e a faixa de valores (decimais) de qualidade.

Complementando a informação do quadro anterior, a Figura C.3 ilustra, para cada uma das variantes do formato FASTQ, os caracteres ASCII (e seus respectivos valores decimais) utilizados.

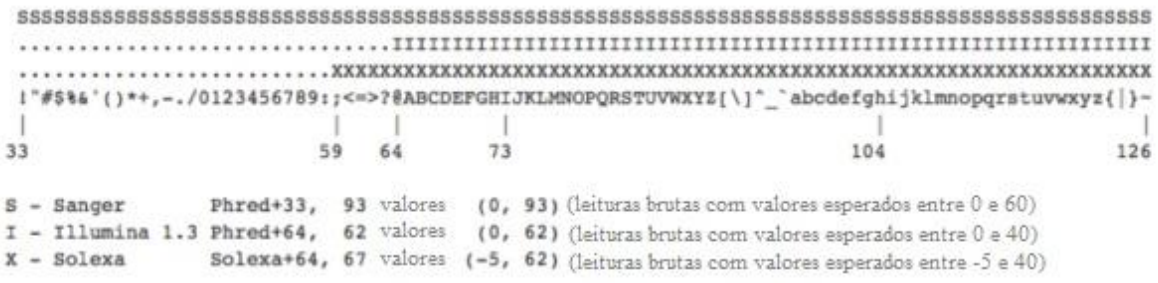


Figura C.3 - Variantes do formato FASTQ em relação ao código ASCII.
Fonte: Modificado de Skrabanek, 2012, p.8.

Apesar de similares, portanto, as variantes do formato FASTQ podem gerar resultados incorretos se analisadas por ferramentas incompatíveis. Desta forma, isso foi levado em consideração neste trabalho.

SAM

O formato SAM, como anteriormente mencionado, foi concebido para armazenar as informações referentes a uma operação de alinhamento, de sequências avulsas quaisquer ou de leituras de sequenciamento, contra um genoma de referência. Basicamente, conforme o exemplo mostrado na Figura C.4, ele consiste de uma seção de cabeçalho e de uma seção com informações sobre o alinhamento.

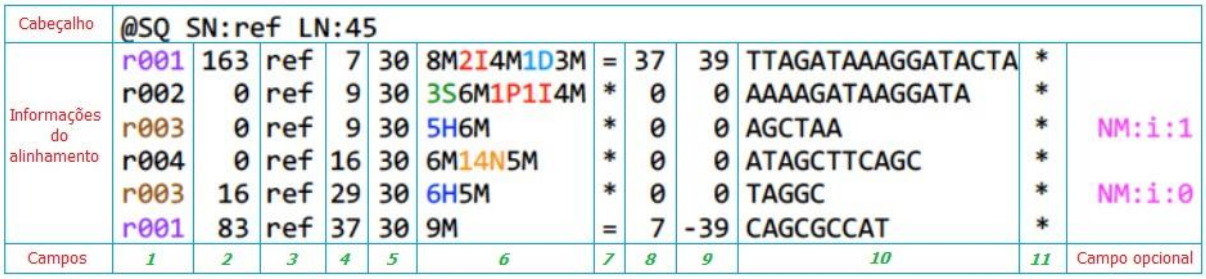


Figura C.4 - Exemplo de formato SAM.
Fonte: Modificado de Skrabanek, 2012, p.29.

As linhas da seção de cabeçalho diferem das da outra seção por serem iniciadas com o caractere "@". Todas as linhas do formato são delimitadas por tabulação e toda linha da seção de alinhamento possui onze campos mandatórios (Quadro C.3) e um número variável de campos opcionais.

Quadro C.3 - Campos mandatórios do formato SAM.

Número	Nome	Descrição
1	QNAME	Nome (NAME) de consulta da leitura ou do par de leituras.
2	FLAG	Sinalizador (FLAG) de "*bit* inteligente" (*bitwise*) com informações sobre o mapeamento da leitura (ver Quadro C.4).
3	RNAME	Nome (NAME) da sequência de referência. Deve corresponder a uma linha @SQ na seção de cabeçalho.
4	POS	POSição mais à esquerda de início de alinhamento da leitura (baseada em coordenada de posição inicial = 1). Valor atribuído como "0" para o caso de uma leitura não-mapeada e, portanto, sem coordenada.
5	MAPQ	Qualidade de MAPeamento (escala PHRED). Valor baseado nas qualidades das bases da leitura mapeada.
6	CIGAR	Cadeia de caracteres CIGAR estendida (operações: MIDNSHP=X), que traz informações mais detalhadas sobre o alinhamento.
7	RNEXT	Campo usado no caso de leituras pareadas; significa o nome da sequência de referência para a próxima leitura (a segunda leitura do par). Recebe valor '=' se tiver a mesma referência da primeira leitura).
8	PNEXT	Campo usado no caso de leituras pareadas, significa a posição de alinhamento da próxima leitura (a segunda leitura do par).
9	TLEN	Campo usado no caso de leituras pareadas, significa o tamanho observado para o fragmento, definido pela extensão do segmento da referência que foi alinhado.
10	SEQ	A SEQuência da leitura alinhada.
11	QUAL	QUALidade da sequência alinhada (código ASCII da qualidade de base + 33; o mesmo do formato FASTQ Sanger).

Fonte: Compilado de Li H et al., 2009, p.2079; The SAM Format Specification Working Group, 2011, p. 3 e Skrabanek, 2012, p.27. Tradução nossa.
Nota:
A página do pacote SAMTools[111] abriga documentos contendo as especificações completas do formato SAM.

Do exposto, observa-se a variedade de informações que podem ser armazenadas com o formato. Neste fluxo de trabalho, mais especificamente o campo FLAG, o qual inclui

[111] http://samtools.sourceforge.net.

informações a respeito do mapeamento de uma leitura individual, foi utilizado. Por usar o conceito *bitwise* (algo como "bit inteligente"), ele é um campo capaz de armazenar diversos valores lógicos (sinalizadores) como uma série curta de bits, ao mesmo tempo em que pode acessar cada bit separadamente (Skrabanek, 2012). O Quadro C.4 mostra as informações que o campo FLAG pode armazenar.

Quadro C.4 - Campo FLAG e suas informações intrínsecas.

Valor hexadecimal	**Valor binário**	**Descrição**
0x1	00000000001 (1)	A leitura é pareada.
0x2	00000000010 (2)	Ambas as leituras em um par foram mapeadas "apropriadamente", conforme o programa alinhador (por exemplo, na orientação correta, considerando-se uma em relação à outra).
0x4	00000000100 (4)	A leitura não foi mapeada.
0x8	00000001000 (8)	A segunda leitura do par (*mate*) não foi mapeada.
0x10	00000010000 (16)	O complemento reverso da leitura teve de ser usado.
0x20	00000100000 (32)	O complemento reverso da segunda leitura do par (*mate*) teve de ser usado.
0x40	00001000000 (64)	A leitura é a primeira do par.
0x80	00010000000 (128)	A leitura é a segunda do par.
0x100	00100000000 (256)	O alinhamento é secundário (uma leitura com correspondências parciais pode apresentar múltiplos registros de alinhamento primário).
0x200	01000000000 (512)	A leitura foi reprovada pela verificação de qualidade da plataforma/fornecedor.
0x400	10000000000 (1024)	Duplicata ótica ou de PCR.

Fonte: Compilado de The SAM Format Specification Working Group, 2011, p. 3 e Skrabanek, 2012, p.28. Tradução nossa.

Nota:

Em uma corrida com leituras únicas, os únicos sinalizadores possíveis serão:

0 Nenhum sinalizador *bitwise* foi atribuído. A leitura foi mapeada à fita direta.

4 A leitura não foi mapeada.

16 A leitura foi mapeada na fita reversa.

Apêndice D - Características do projeto em relação à metodologia *XP*

Quadro D.1 - Alguns valores da *XP* utilizados e características do trabalho correspondentes.

Alguns valores da *XP* utilizados	Significado[112]	Características correspondentes do trabalho
Comunicação	Manter comunicação constante com o cliente, priorizando as funcionalidades que representem maior valor possível para o negócio.	Sistema desenvolvido de forma autônoma, com requisitos determinados pelos "clientes" imediatos dos resultados das eventuais montagens realizadas (no caso, orientadores do projeto e uma usuária da área de ciências da vida). Comunicação e *feedback* constantes, inerentes à essa relação de proximidade e prestação de contas, estavam previstas.
Feedback	Presumir a necessidade de mudanças nos requisitos e ser capaz de responder a elas. Além disso, escutar/fornecer *feedback* do/ao cliente.	
Simplicidade	Foco na simplicidade do software e do processo de desenvolvimento. Sempre que possível, trabalhar ativamente para eliminar a complexidade do sistema. A simplicidade deve ser mantida em todas fases do projeto.	O tempo para a realização do projeto seria relativamente curto e, além disso, ele, de antemão, já envolvia uma razoável complexidade por causa da diversidade de tecnologias e programas de NGS disponíveis. Desta forma, seguindo a linha de pensamento de Hemrajani (2007), a premissa foi por não "reinventar a roda". Existindo uma boa solução, já pronta e que pudesse ser utilizada, esta o seria.
Coragem (pode ser entendida, também, como **disciplina**)	Para abraçar mudanças e o trabalho de qualidade e para seguir estritamente a abordagem de desenvolvimento simples, sem sucumbir a pressões para a elaboração do projeto pensando em futuros requisitos.	O tópico NGS envolve mudanças constantes. O desenvolvimento deveria ser apoiado em alguma metodologia que comportasse possíveis mudanças de requisitos também.

Quadro D.2 - Algumas práticas da *XP* utilizadas e características do trabalho correspondentes.

(continua)

Algumas práticas da *XP* utilizadas	Significado[112]	Características correspondentes do trabalho
Jogo do planejamento	Atividade de planejamento que inicia com o levantamento de requisitos, priorizando as funcionalidades. O cliente identifica prioridades e os desenvolvedores as estimam. O planejamento conduz à criação de um conjunto de "histórias" (fragmentos de funcionalidades, também denominados *histórias de usuários*) registradas em cartões fáceis de serem manipulados pelo cliente e desenvolvedor.	A existência de dados de NGS reais para um determinado organismo e a demanda por sua análise foi o que motivou a ideia de se disponibilizar alguma ferramenta que funcionasse como um serviço para realizar a montagem básica desse genoma e que, posteriormente, pudesse ser estendida para outros organismos e outras tecnologias NGS. Isso serviu de base para o levantamento dos primeiros requisitos formais.
Pequenas versões	Pequenas versões funcionais são disponibilizadas ao longo do processo, mais precisamente, ao final de cada iteração. Após o devido teste de aceitação, passa-se à iteração seguinte, caracterizando, dessa maneira, o	O trabalho, por si só, já previa o incremento de funcionalidades ao longo do tempo, pois deveria ser criado um fluxo de trabalho para cada tipo de abordagem de montagem e para cada tipo de tecnologia NGS. Cada fluxo de trabalho foi, então, considerado como um ciclo

[112] Compilado a partir das referências mencionadas nesta seção, porém com detalhes adaptados ao desenvolvimento autônomo (Santos, 2010).

Quadro D.2 - Algumas práticas da *XP* utilizadas e características do trabalho correspondentes.

(conclusão)

Algumas práticas da *XP* utilizadas	**Significado**	**Características correspondentes do trabalho**
Pequenas versões	desenvolvimento iterativo e incremental.	de iteração para a liberação de uma "versão" do sistema.
Metáfora	No contexto da *XP*, uma *metáfora* é "uma história que todos — clientes, programadores e gerentes — podem contar sobre como um sistema funciona" (Pressman, 2011 apud Beck, 2004). Ela procura facilitar a comunicação com o cliente, entendendo sua visão por meio do uso dos cartões de histórias já citados.	Repetindo o que foi dito no item "Jogo do planejamento", a história primária do sistema era a de este ser capaz, inicialmente, de oferecer um serviço para realizar montagens básicas de genomas a partir de diferentes tecnologias NGS.
Projeto simples	Fazer código exato para implementar cada funcionalidade da forma mais sinples possível, se atendo apenas àquilo que já se conhece no presente, sem considerar *menu* e *design* muito elaborados ou preparação para futuras funcionalidades através de generalizações. Se o projeto tiver de ser melhorado, ele poderá ser refabricado mais tarde.	O autor do trabalho não tem a formação de programador. Assim, o trabalho deveria ser desenvolvido com habilidades básicas de programação. A ideia de seguir um projeto simples e de realizar codificações básicas pareceu ser a mais recomendada para garantir, com razoável segurança, a execução do projeto. Conforme dito anteriormente, a própria premissa do trabalho era a de utilizar ferramentas já disponíveis e liberadas para uso. Assim, sempre que possível, o projeto priorizaria utilizar soluções simples e prontas, em vez das complexas ou daquelas que demandariam muito tempo de desenvolvimento para, ao final, atingir propósito similar.
Time coeso	A equipe de desenvolvimento é formada pelo cliente e pela equipe de desenvolvimento.	Como explicado no Quadro D.1, itens "Comunicação" e "*Feedback*", poucas eram as pessoas envolvidas no desenvolvimento inicial do projeto, caracterizando um "time de desenvolvimento" coeso.
Testes de aceitação	No desenvolvimento autônomo, o sistema será construído de acordo com o que for especificado pelos requisitos, no cartões de histórias e na comunicação constante com o cliente. Os testes representam a forma de o próprio software implementado fornecer *feedback* ao projeto.	Os testes seriam automaticamente realizados ao fim de cada implementação de fluxo de trabalho. Basicamente, o sistema deveria ser capaz de realizar uma montagem básica, usando dados reais de sequenciamento NGS, para cada fluxo de trabalho implementado.
Refabricação	Processo que permite a melhoria contínua da programação, com o mínimo de introdução de erros e mantendo a compatibilidade com o código já existente. Refabricar melhora a clareza do código, dividindo-o em módulos mais coesos e de maior reaproveitamento, evitando a duplicação de código-fonte.	Estava previsto o contínuo desenvolvimento do sistema, mesmo após a conclusão desta etapa do trabalho. Então, a "refabricação" ou "retrabalho" seriam perfeitamente aceitáveis, se adequando ao que é pregado pela metodologia *XP*.
Integração contínua	Sempre que se produzir uma nova funcionalidade, ela deve ser integrada rapidamente à versão mais atual do sistema.	Tal como exposto no item "Pequenas versões", a implementação bem sucedida de um determinado fluxo de trabalho de montagem já estaria incorporando a funcionalidade, ao sistema, de maneira automática.

Apêndice E - Representações esquemáticas dos fluxos de trabalho produzidos

Representação esquemática do fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina extraída do protótipo

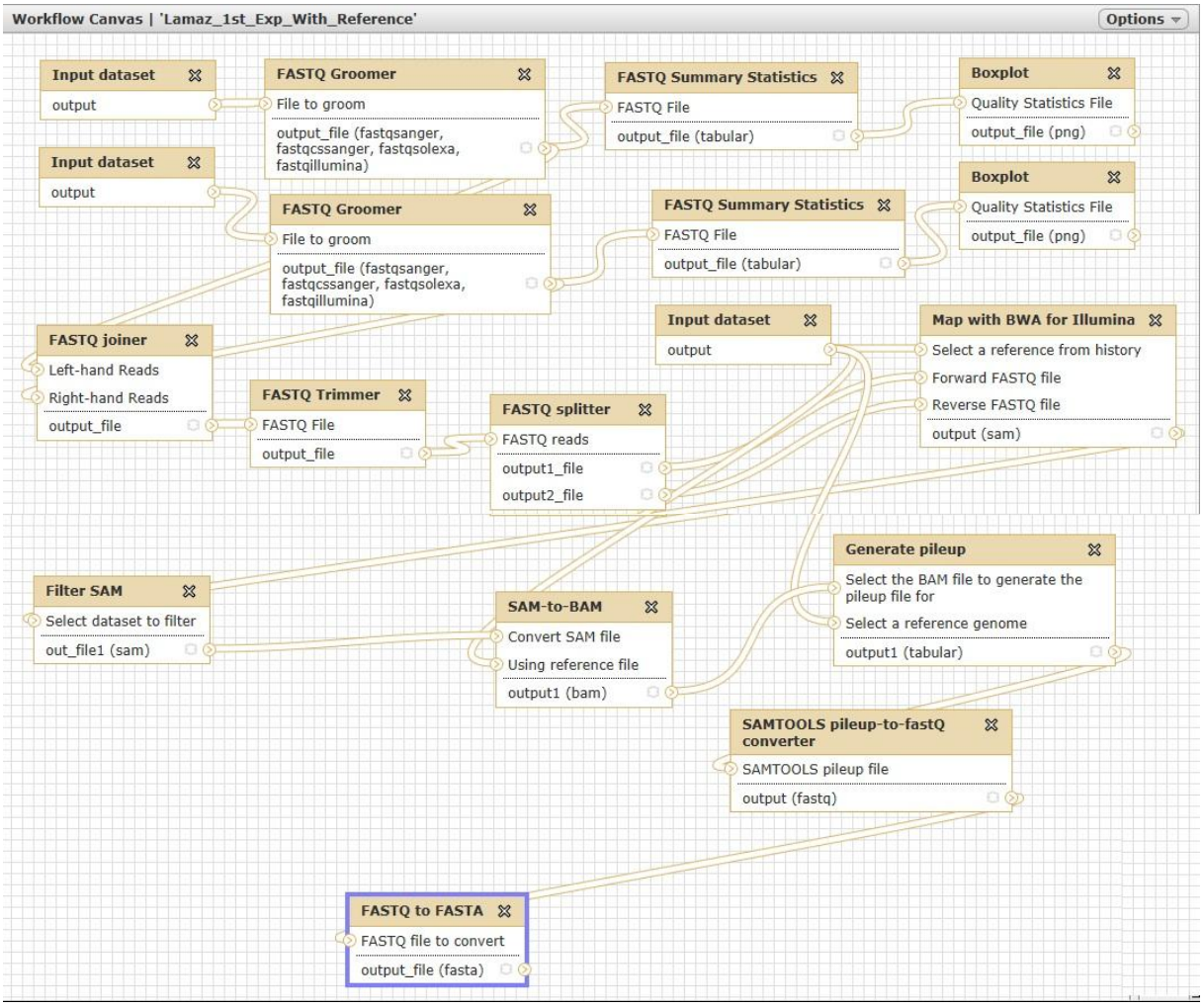


Figura E.1 - Fluxo de trabalho para montagem com auxílio de genoma de referência a partir de dados de Solexa/Illumina.

Representação esquemática do fluxo de trabalho para montagem de novo a partir de dados de Solexa/Illumina extraída do protótipo

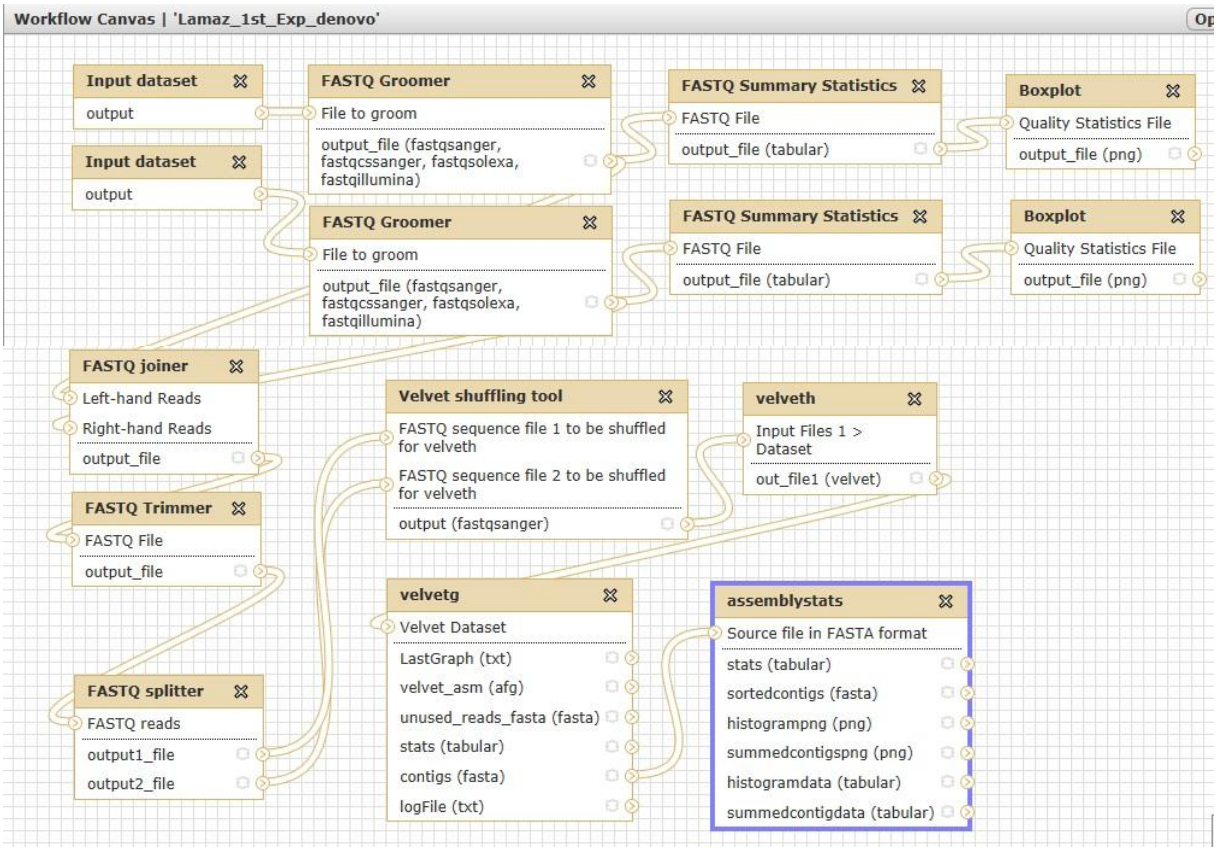


Figura E.2 - Fluxo de trabalho para montagem *de novo* a partir de dados de Solexa/Illumina.

Representação esquemática do fluxo de trabalho para montagem de novo a partir de dados de biblioteca de fragmentos de ABI SOLiD™ extraída do protótipo

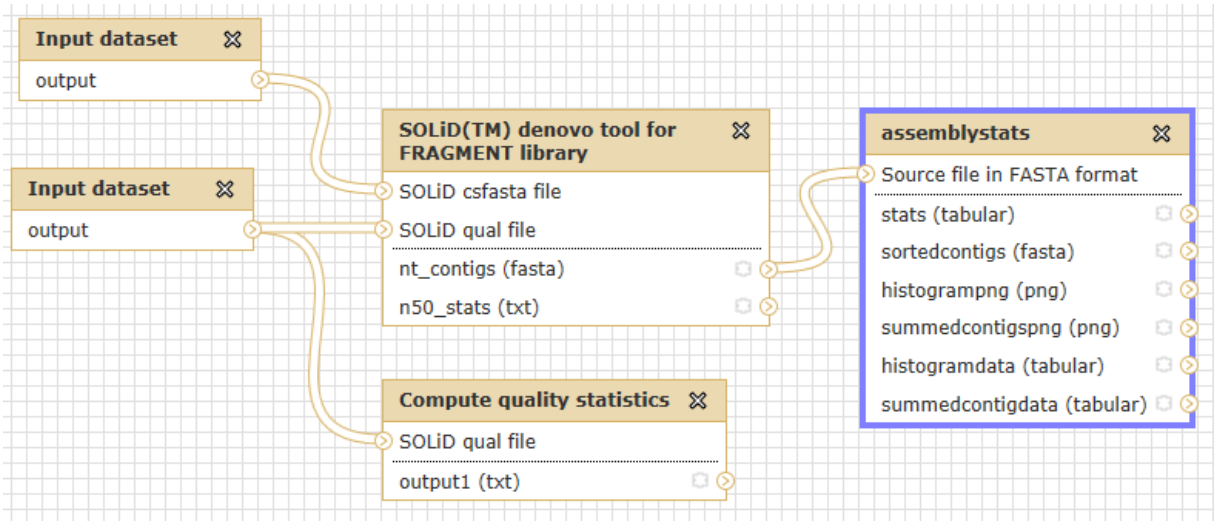


Figura E.3 - Fluxo de trabalho para montagem de novo a partir de dados de fragmentos de ABI SOLiD™.

Representação esquemática do fluxo de trabalho para montagem de novo a partir de dados de biblioteca MATE-PAIRED de ABI SOLiD™ extraída do protótipo

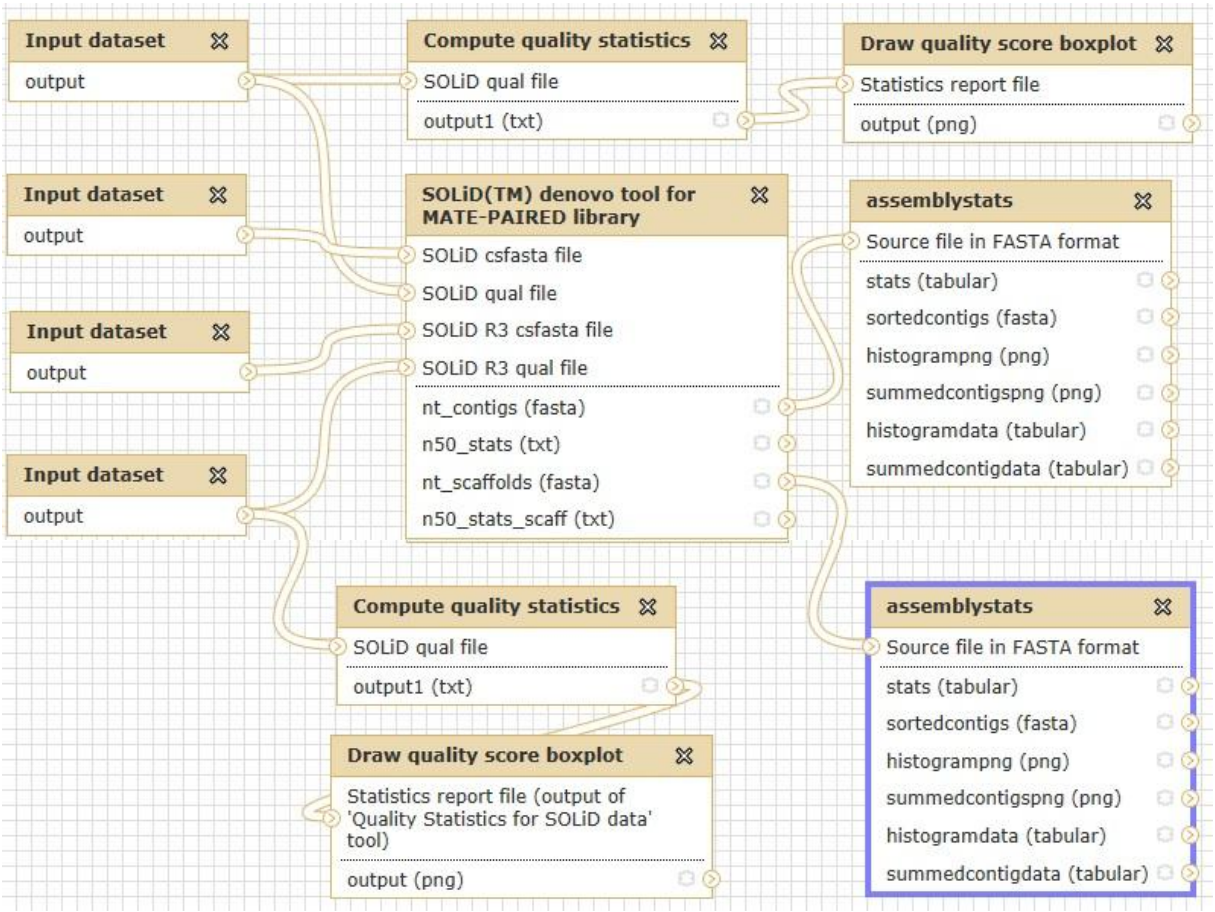


Figura E.4 - Fluxo de trabalho para montagem *de novo* a partir de dados de biblioteca *MATE-PAIRED* de ABI SOLiD™.

Representação esquemática do fluxo de trabalho para montagem de novo a partir de dados de 454 extraída do protótipo

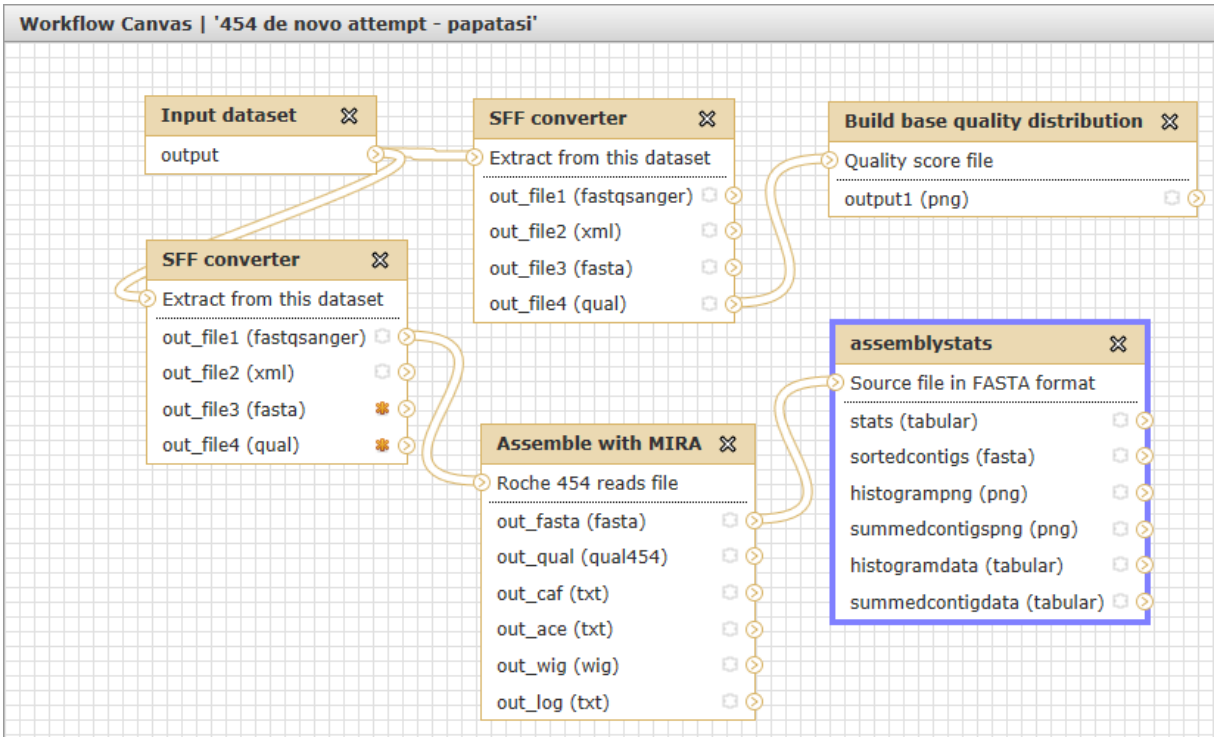


Figura E.5 - Fluxo de trabalho para montagem *de novo* a partir de dados de 454.

Apêndice F - Códigos dos programas *wrappers* adaptados ou desenvolvidos para o protótipo *LASZLO @ GALAXY* nesta etapa do trabalho

Wrappers para a ferramenta SAMTools pileup-to-fastQ converter

Arquivo XML "ngs_sam_pileup2fq.xml"

```
<!--## Galaxy XML for converting SAMTOOLS pileup file to fastq tool.
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
#########################################################################
###################################-->

<tool id="SAMTOOLS_pileup2fq_converter" name="SAMTOOLS pileup-to-fastQ
converter">
        <description>A converter from SAMTOOLS pileup file to FASTQ
format</description>

        <command interpreter="sh">antonio_pileup2fq_wrapper.sh '$input'
'$output'</command>

        <inputs>
                <param name="input" type="data" format="tabular"
label="SAMTOOLS pileup file"/>
        </inputs>

        <outputs>
                <data format="fastq" name="output"/>
        </outputs>

   <help> This tool attempts to generate a FASTQ format file from a given
SAMTOOLS pileup file.</help>
</tool>
<!-- Created by ANTONIO -->
```

Script "antonio_pileup2fq_wrapper.sh"

```
#!/bin/sh
#!/usr/bin/perl -w

##
## Galaxy wrapper for converting SAMTOOLS pileup file to fastq.
##
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
#########################################################################
####################################
##
## command line arguments:
##
##   input_file
##   output_file

INPUT="$1"
OUTPUT="$2"

if [ -z "$OUTPUT" ]; then
        echo "This script should be run from inside galaxy!" >&2
        exit 1
fi

if [ ! -r "$INPUT" ]; then
        echo "error: input file ($INPUT) not found!" >&2
        exit 1
fi
```

```
# Messages printed to STDOUT will be displayed in the "INFO" field in the
galaxy dataset.
# This way the user can tell what was the command

/home/acbellorib/galaxy-dist/tools/ngs_laszlo/samtools.pl pileup2fq -D100
$INPUT > $OUTPUT

exit
```

Wrappers para a ferramenta Extract region tool

Arquivo XML "antonio_extractPartOfFasta.xml"

```
<!--## Galaxy XML for extracting a given region of a FASTA file, based on
the desired coordinates.
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
##########################################################################
##################################-->
<tool id="extractPartOfFasta" name="Extract region tool">
        <description>based on FASTA file coordinates</description>

        <command interpreter="sh">antonio_extractPartOfFasta_wrapper.sh
'$input' '$output' '$contig' '$start' '$end'</command>

        <inputs>
            <param name="input" type="data" format="fasta" label="FASTA
file from which the desired region will be extracted"/>
            <param name="contig" type="text" label="The sequence header of
the contig or chromosome which has the desired region" help="Note: The
sequence he
ader must not contain spaces. If it does, give only the first tab, eg
&amp;quot;contig001&amp;quot; if header is &amp;quot;&amp;gt;contig001 chr=1
length=1000...&amp;quot;"/>
            <param name="start" type="integer" value="1" label="The START
position of the desired region to be retrieved" help="Note: If the purpose
is to re
trieve an additional number of bases BEFORE the target region starting
position (for instance, for primer designing), please assure to properly
calculate the
 desired downstream position, eg 1000 bases lower than the target region
start position, provided that this limit doesn't surpass the beginning of
the contig
/chromosome."/>
            <param name="end" type="integer" value="10" label="The FINAL
position of the desired region to be retrieved" help="Note: If the purpose
is to ret
rieve an additional number of bases AFTER the target region end position
(for instance, for primer designing), please assure to properly calculate
the desire
d upstream position, eg 1000 bases higher than the target region end
position, provided that this limit doesn't surpass the end of the
contig/chromosome."/>
        </inputs>

        <outputs>
                <data format="fasta" name="output"/>
        </outputs>

   <help>This tool attempts to extract a given region of a FASTA file,
based on the desired coordinates.</help>
</tool>
<!-- Created by ANTONIO -->
```

Script "antonio_extractPartOfFasta_wrapper.sh"

```
#!/bin/sh
#!/usr/bin/perl -w

##
## Galaxy wrapper for extracting a given region of a FASTA file based on
its coordinates.
##
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
##############################################################################
####################################
##
## command line arguments:
##
##   input_file
##   output_file
##   contig
##   start
##   end

INPUT="$1"
OUTPUT="$2"
CONTIG="$3"
START="$4"
END="$5"

if [ -z "$OUTPUT" ]; then
        echo "This script should be run from inside galaxy!" >&2
        exit 1
fi

if [ ! -r "$INPUT" ]; then
        echo "error: input file ($INPUT) not found!" >&2
        exit 1
fi

# Messages printed to STDOUT will be displayed in the "INFO" field in the
galaxy dataset.
# This way the user can tell what was the command

/home/acbellorib/galaxy-dist/tools/ngs_laszlo/extractPartOfFasta.pl $INPUT
$CONTIG $START $END > $OUTPUT

exit
```

Script "extractPartOfFasta.pl"

```
#!/usr/bin/perl

# # # # # #
# extractPartOfFasta.pl
# written by Linnéa Smeds 3 Feb 2011
##############################################################################
##
# Adapted by Antonio Ribeiro for testing purposes in GALAXY local instance
# Reference: https://www.uppnex.uu.se/content/extract-region-fasta-sequence
##############################################################################
##
# ================================================
# Extract a specific region from a sequence. Takes a
# fasta file (can contain several sequences), a sequence
# header and start and end positions.
# NB! The sequence header must not contain spaces. If
# it does, give only the first tab, eg "contig001" if
# header is ">contig001 chr=1 length=1000..."
# ================================================
# Usage: extractPartOfFasta.pl <fastafile> <seq name> <start> <end>
#
```

```
#
# Example: extractPartOfFasta.pl mySeq.fa contig4 200 299 > contig4_pos200-
299.fa
#

use strict;
use warnings;


# Input parameters
my $fasta = $ARGV[0];
my $name = $ARGV[1];
my $start = $ARGV[2];
my $end = $ARGV[3];

my $noOfBases = $end-$start+1;

open(FAS, $fasta);
my ($seq, $head, $seqFlag) = ("", "", "off");
while(<FAS>) {
        if(/>/) {
                if($seqFlag eq "on" && $seq ne "") {
                        my $len = length($seq);
                        print "$head orig_len=$len, extract $start-$end
($noOfBases bp)\n";
                        my $substr = substr($seq, $start-1, $noOfBases);
                        my @seqParts = split(/(.{100})/, $substr);
                        for my $seqs (@seqParts) {
                                unless($seqs eq "") {
                                        print $seqs."\n";
                                }
                        }
                        $seq="";
                        $seqFlag="off";
                }
 my @tab = split(/\s+/, $_);
                if($tab[0] eq ">$name") {
                        $seqFlag = "on";
                        $head = $tab[0];
                }
        }
        else {
                if($seqFlag eq "on") {
                        chomp($_);
                        $seq.=$_;
                }
        }
}
if($seqFlag eq "on" && $seq ne "") {
        my $len = length($seq);
        print "$head orig_len=$len, extract $start-$end ($noOfBases bp)\n";
        my $substr = substr($seq, $start-1, $noOfBases);
        my @seqParts = split(/(.{100})/, $substr);
        for my $seqs (@seqParts) {
                unless($seqs eq "") {
                        print $seqs."\n";
                }
        }
        $seq="";
        $seqFlag="off";
}
```

Wrappers para a ferramenta Velvet shuffling tool

Arquivo XML "antonio_velvetShufflerFastq.xml"

```
<!--## Galaxy XML for shuffling two paired-end FASTQ files for velveth.
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
##########################################################################
##################################-->
<tool id="velvetShuffleFastq" name="Velvet shuffling tool">
        <description>for two separate FASTQ files</description>

        <command interpreter="sh">antonio_velvetShufflerFastq_wrapper.sh
'$input1' '$input2' '$output'</command>

        <inputs>
            <param name="input1" type="data" format="fastqsanger"
label="FASTQ sequence file 1 to be shuffled for velveth"/>
            <param name="input2" type="data" format="fastqsanger"
label="FASTQ sequence file 2 to be shuffled for velveth"/>
        </inputs>

        <outputs>
                <data format="fastqsanger" name="output"/>
        </outputs>

   <help>This tool attempts to shuffle two paired-end FASTQ files for
velveth.</help>
</tool>
<!-- Created by ANTONIO -->
```

Script "antonio_velvetShufflerFastq_wrapper.sh"

```
#!/bin/sh
#!/usr/bin/perl -w

##
## Galaxy wrapper for shuffling two paired-end FASTQ files for velveth.
##
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
##########################################################################
####################################
##
## command line arguments:
##
##   filenameA
##   filenameB
##   filenameOut

INPUT1="$1"
INPUT2="$2"
OUTPUT="$3"

if [ -z "$OUTPUT" ]; then
        echo "This script should be run from inside galaxy!" >&2
        exit 1
fi

if [ ! -r "$INPUT1" ]; then
        echo "error: input file ($INPUT1) not found!" >&2
        exit 1
fi

if [ ! -r "$INPUT2" ]; then
        echo "error: input file ($INPUT2) not found!" >&2
        exit 1
fi
```

```
# Messages printed to STDOUT will be displayed in the "INFO" field in the
galaxy dataset.
# This way the user can tell what was the command

/home/acbellorib/galaxy-dist/tools/ngs_laszlo/shuffleSequences_fastq.pl
$INPUT1 $INPUT2 $OUTPUT

exit
```

Wrappers para a ferramenta SOLiD(TM) denovo tool for FRAGMENT library

Arquivo XML "SOLiD_denovo_fragment.xml"

```
<!--## Galaxy XML for adapting the SOLiD (TM) de novo accesssory tools 2.0
pipeline for fragment library.
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
##########################################################################
##################################-->
<tool id="solid_denovo_fragment" name="SOLiD(TM) denovo tool for FRAGMENT
library" version="1.0.0">
  <description>A pipeline that enables de novo assembly of small genomes
from SOLiD(TM) short reads</description>
  <command interpreter="python">
    SOLiD_denovo_fragment.py
          '$nt_contigs.extra_files_path' '$csfasta' '$qual' '$refLength'
'$numcores' '$nt_contigs' '$n50_stats'
  </command>
  <inputs>
    <param name="csfasta" type="data" format="csfasta" label="SOLiD csfasta
file" help="The input forward csfasta file for de novo assembly."/>
        <param name="qual" type="data" format="qualsolid" label="SOLiD qual
file" help="The input forward qual file for de novo assembly."/>
    <param name="refLength" value="4600000" type="integer" label="Expected
length of sequenced (or enriched) DNA region" help="e.g., 4600000 for
E.Coli 4.6Mb
 genome or 30000000 for Whole Human Transcriptome."/>
    <param label="Number of CPUs/Cores" name="numcores" type="select"
help="Number of cores to parallelize SAET stage. Choose '1' if you have
only one CPU or
 Core.">
      <option value="1" selected="yes">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
      <option value="6">6</option>
      <option value="7">7</option>
      <option value="8">8</option>
      <option value="9">9</option>
      <option value="10">10</option>
      <option value="11">11</option>
      <option value="12">12</option>
      <option value="13">13</option>
      <option value="14">14</option>
      <option value="15">15</option>
      <option value="16">16</option>
    </param>
  <!--<requirements>
    <requirement type="package">saet_mp</requirement>
  </requirements> -->
  </inputs>
  <outputs>
    <data format="fasta" name="nt_contigs" label="${tool.name} on
${on_string}: Resulting nt_contigs.fa file after de novo assembly tool"/>
    <data format="txt" name="n50_stats" label="${tool.name} on
${on_string}: Resulting n50.stats.txt file after de novo assembly tool"/>
  </outputs>
  <help>
```

```
**SOLiD de novo accessory tool for fragment library overview**

Running de novo assembly pipeline for fragment library data run:

$denovo2/assemble.pl &lt;f3_csfasta&gt; &lt;f3_qual&gt; &lt;refLength&gt;
[-options]

Input:



f3_csfasta - csfasta/fasta file with reads.

 f3_qual    - filename with quality values. Notice that order of reads in
csfasta file should be the same as in quality value file.

 refLength  - expected length of sequenced DNA region, e.g., 4600000 for
E.Coli 4.6Mb genome.

Output:

 &lt;outdir&gt;/nt_contigs.fa                       - fasta file with
assembled base-space contigs.

 &lt;outdir&gt;/analysis/nt_contigs/n50.stats.txt  - file containing base-
space contigs analysis.
  </help>
</tool>
```

Arquivo "SOLiD_denovo_fragment.py"

```
#!/usr/bin/env python
#Python wrapper for adapting the SOLiD (TM) de novo accessory tools 2.0
pipeline for fragment library in Galaxy.
#Adapted by Antonio, based on the "velvetg_wrapper.py" for Galaxy.
import pkg_resources
import logging, os, string, sys, tempfile, glob, shutil, types, urllib
import shlex, subprocess
from optparse import OptionParser, OptionGroup
from stat import *

log = logging.getLogger( __name__ )

assert sys.version_info[:2] >= ( 2, 4 )

def stop_err( msg ):
    sys.stderr.write( "%s\n" % msg )
    sys.exit()

def __main__():
    #Parse Command Line
    s = 'SOLiD_denovo_fragment.py:  argv = %s\n' % (sys.argv)
    argcnt = len(sys.argv)
    #html_file = sys.argv[1]
    working_dir = sys.argv[1]
    csfasta = sys.argv[2]
    qual = sys.argv[3]
    refLength = sys.argv[4]
    numcores = sys.argv[5]
    nt_contigs = sys.argv[6]
    n50_stats = sys.argv[7]
    #inputs = string.join(sys.argv[9:],' ')
    # print >> sys.stderr, cmdline # so will appear as blurb for file
    #try: # for test - needs this done
    #    os.makedirs(working_dir)
    #except Exception, e:
    #    stop_err( 'Error running SOLiD_denovo_fragment.py ' + str( e ) )
```

```
    cmdline = '%s/assemble_Antonio.pl %s %s %s -outdir %s -numcores %s' %
(os.path.dirname(sys.argv[0]), csfasta, qual, refLength, working_dir,
numcores)
    try:
        proc = subprocess.Popen( args=cmdline, shell=True,
stderr=subprocess.PIPE )
        returncode = proc.wait()
        # get stderr, allowing for case where it's very large
        stderr = ''
        buffsize = 1048576
        try:
            while True:
                stderr += proc.stderr.read( buffsize )
                if not stderr or len( stderr ) % buffsize != 0:
                    break
        except OverflowError:
            pass
        if returncode != 0:
            raise Exception, stderr
    except Exception, e:
        stop_err( 'Error running SOLiD_denovo_fragment.py ' + str( e ) )

    out = open(nt_contigs,'w')
    nt_contigs_path = os.path.join(working_dir,'nt_contigs.fa')
    for line in open( nt_contigs_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(n50_stats,'w')
    n50_stats_path =
os.path.join(working_dir,'analysis/nt_contigs/n50.stats.txt')
    for line in open( n50_stats_path ):
        out.write( "%s" % (line) )
    out.close()

if __name__ == "__main__": __main__()
```

Wrappers para a ferramenta SOLiD(TM) denovo tool for PAIRED-END library

Arquivo XML "SOLiD_denovo_pe.xml"

```
<!--## Galaxy XML for adapting the SOLiD (TM) de novo accesssory tools 2.0
pipeline for paired-end library.
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
#########################################################################
#################################-->
<tool id="solid_denovo_pe" name="SOLiD(TM) denovo tool for PAIRED-END
library" version="1.0.0">
  <description>A pipeline that enables de novo assembly of small genomes
from SOLiD(TM) short reads</description>
  <command interpreter="python">
    SOLiD_denovo_pe.py
                '$nt_contigs.extra_files_path' '$csfasta' '$qual'
'$refLength' '$f5_csfasta' '$f5_qual' '$ins_length'
                '$ins_length_sd' '$numcores' '$nt_contigs' '$n50_stats'
'$nt_scaffolds' '$n50_stats_scaff'
  </command>
  <inputs>
    <param name="csfasta" type="data" format="csfasta" label="SOLiD csfasta
file" help="The input forward csfasta file for de novo assembly."/>
        <param name="qual" type="data" format="qualsolid" label="SOLiD qual
file" help="The input forward qual file for de novo assembly."/>
    <param name="refLength" value="4600000" type="integer" label="Expected
length of sequenced (or enriched) DNA region" help="e.g., 4600000 for
E.Coli 4.6Mb
 genome or 30000000 for Whole Human Transcriptome."/>
    <param name="f5_csfasta" type="data" format="csfasta" label="SOLiD F5
csfasta file" help="The input F5 csfasta file for de novo assembly."/>
```

```
        <param name="f5_qual" type="data" format="qualsolid" label="SOLiD
F5 qual file" help="The input F5 qual file for de novo assembly."/>
    <param name="ins_length" value="1300" type="integer" label="Estimate of
insert length" help="e.g., xx = 1200 for mate-paired library data with
insert len
gth of 1.2Kb."/>
    <param name="ins_length_sd" value="400" type="integer" label="Estimate
of variance of the insert length" help="e.g., zz = 300 for mate-paired
library dat
a with insert length of 1.2Kb and zz = 30 for paired-end data with insert
size of 170bp."/>
    <param label="Number of CPUs/Cores" name="numcores" type="select"
help="Number of cores to parallelize SAET stage. Choose '1' if you have
only one CPU or
 Core.">
      <option value="1" selected="yes">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
      <option value="6">6</option>
      <option value="7">7</option>
      <option value="8">8</option>
      <option value="9">9</option>
      <option value="10">10</option>
      <option value="11">11</option>
      <option value="12">12</option>
      <option value="13">13</option>
      <option value="14">14</option>
      <option value="15">15</option>
      <option value="16">16</option>
    </param>
    <!--<requirements>
    <requirement type="package">saet_mp</requirement>
  </requirements> -->
  </inputs>
  <outputs>
    <data format="fasta" name="nt_contigs" label="${tool.name} on
${on_string}: Resulting nt_contigs.fa file after the de novo assembly
tool"/>
    <data format="txt" name="n50_stats" label="${tool.name} on
${on_string}: Resulting n50.stats.txt file for contigs after the de novo
assembly tool"/>
    <data format="fasta" name="nt_scaffolds" label="${tool.name} on
${on_string}: Resulting nt_scaffolds.fa file after the de novo assembly
tool"/>
    <data format="txt" name="n50_stats_scaff" label="${tool.name} on
${on_string}: Resulting n50.stats.txt file for scaffolds after the de novo
assembly tool
"/>
  </outputs>
  <help>
**SOLiD de novo accessory tool for paired-end library overview**

Running de novo assembly pipeline for paired-end library data run:

$denovo2/assemble.pl &lt;f3_csfasta&gt; &lt;f3_qual&gt; &lt;refLength&gt; -
f5 f5_csfasta -f5qv f5_qual -ins_length xx -ins_length_sd zz [-options]

Input:

 f3_csfasta - csfasta/fasta file with reads.

 f3_qual    - filename with quality values (if available). notice, that
order of reads in csfasta file should be the same as in quality value file.

 refLength  - expected length of sequenced DNA region, e.g., 4600000 for
E.Coli 4.6Mb genome.

Options required for paired-end data:

 -f5 f5_csfasta     -  csfasta file with 2-base encoded F5 reads.
```

```
 -f5qv f5_qual      -  file with quality values for F5 reads. Notice, that
order of reads in csfasta file should be the same as in quality values
file.

Options required for mate-paired or paired-end data:

-ins_length xx      -  estimate of insert length, e.g., xx=1200 for mate-
paired library data with insert length 1.2Kb.

-ins_length_sd zz   -  estimate of variance of the insert length, e.g.,
zz=300 for mate-paired library data with insert length 1.2Kb and zz=30 for
paired-end
 data with insert size 170bp.

Output:

 &lt;outdir&gt;/nt_contigs.fa                        - fasta file with
assembled base-space contigs.

 &lt;outdir&gt;/analysis/nt_contigs/n50.stats.txt   - file containing base-
space contigs analysis.

 &lt;outdir&gt;/nt_scaffolds.fa                      - fasta file with
assembled base-space scaffolds (for mate-paired or paired-end data).

 &lt;outdir&gt;/analysis/nt_scaffolds/n50.stats.txt - file containing base-
space scaffolds analysis.
  </help>
</tool>
```

Arquivo "SOLiD_denovo_pe.py"

```
#!/usr/bin/env python
#Python wrapper for adapting the SOLiD (TM) de novo accessory tools 2.0
pipeline for paired-end library in Galaxy.
#Adapted by Antonio, based on the "velvetg_wrapper.py" for Galaxy.
import pkg_resources
import logging, os, string, sys, tempfile, glob, shutil, types, urllib
import shlex, subprocess
from optparse import OptionParser, OptionGroup
from stat import *

log = logging.getLogger( __name__ )

assert sys.version_info[:2] >= ( 2, 4 )

def stop_err( msg ):
    sys.stderr.write( "%s\n" % msg )
    sys.exit()

def __main__():
    #Parse Command Line
    s = 'SOLiD_denovo_pe.py:  argv = %s\n' % (sys.argv)
    argcnt = len(sys.argv)
    #html_file = sys.argv[1]
    working_dir = sys.argv[1]
    csfasta = sys.argv[2]
    qual = sys.argv[3]
    refLength = sys.argv[4]
    f5_csfasta = sys.argv[5]
    f5_qual = sys.argv[6]
    ins_length = sys.argv[7]
    ins_length_sd = sys.argv[8]
    numcores = sys.argv[9]
    nt_contigs = sys.argv[10]
    n50_stats = sys.argv[11]
    nt_scaffolds = sys.argv[12]
    n50_stats_scaff = sys.argv[13]
```

```
    #inputs = string.join(sys.argv[9:],' ')
    # print >> sys.stderr, cmdline # so will appear as blurb for file
    #try: # for test - needs this done
    #    os.makedirs(working_dir)
    #except Exception, e:
    #    stop_err( 'Error running SOLiD_denovo_mp.py ' + str( e ) )
    #cmdline = '%s/assemble.pl %s %s %s -r3 %s -r3qv %s -ins_length %s -
ins_length_sd %s > /dev/null' % (os.path.dirname(sys.argv[0]), csfasta,
qual, refLeng
th, r3_csfasta, r3_qual, ins_length, ins_length_sd)
    cmdline = '%s/assemble.pl %s %s %s -f5 %s -f5qv %s -ins_length %s -
ins_length_sd %s -outdir %s -numcores %s' % (os.path.dirname(sys.argv[0]),
csfasta, qu
al, refLength, f5_csfasta, f5_qual, ins_length, ins_length_sd, working_dir,
numcores)
    try:
        proc = subprocess.Popen( args=cmdline, shell=True,
stderr=subprocess.PIPE )
        returncode = proc.wait()
        # get stderr, allowing for case where it's very large
        stderr = ''
        buffsize = 1048576
        try:
            while True:
                stderr += proc.stderr.read( buffsize )
                if not stderr or len( stderr ) % buffsize != 0:
                    break
        except OverflowError:
            pass
        if returncode != 0:
            raise Exception, stderr
    except Exception, e:
        stop_err( 'Error running SOLiD_denovo_mp.py ' + str( e ) )

    out = open(nt_contigs,'w')
    nt_contigs_path = os.path.join(working_dir,'nt_contigs.fa')
    for line in open( nt_contigs_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(n50_stats,'w')
    n50_stats_path =
os.path.join(working_dir,'analysis/nt_contigs/n50.stats.txt')
    for line in open( n50_stats_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(nt_scaffolds,'w')
    nt_scaffolds_path = os.path.join(working_dir,'nt_scaffolds.fa')
    for line in open( nt_scaffolds_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(n50_stats_scaff,'w')
    n50_stats_scaff_path =
os.path.join(working_dir,'analysis/nt_scaffolds/n50.stats.txt')
    for line in open( n50_stats_scaff_path ):
        out.write( "%s" % (line) )
    out.close()

if __name__ == "__main__": __main__()
```

Wrappers para a ferramenta SOLiD(TM) denovo tool for MATE-PAIRED library

Arquivo XML "SOLiD_denovo_mp.xml"

```
<!--## Galaxy XML for adapting the SOLiD (TM) de novo accesssory tools 2.0
pipeline for mate-paired library.
## written by Antonio Ribeiro for testing purposes in GALAXY local instance
##########################################################################
##################################-->
<tool id="solid_denovo_mp" name="SOLiD(TM) denovo tool for MATE-PAIRED
library" version="1.0.0">
  <description>A pipeline that enables de novo assembly of small genomes
from SOLiD(TM) short reads</description>
  <command interpreter="python">
    SOLiD_denovo_mp.py
                '$nt_contigs.extra_files_path' '$csfasta' '$qual'
'$refLength' '$r3_csfasta' '$r3_qual' '$ins_length'
                '$ins_length_sd' '$numcores' '$nt_contigs' '$n50_stats'
'$nt_scaffolds' '$n50_stats_scaff'
  </command>
  <inputs>
    <param name="csfasta" type="data" format="csfasta" label="SOLiD csfasta
file" help="The input forward csfasta file for de novo assembly."/>
        <param name="qual" type="data" format="qualsolid" label="SOLiD qual
file" help="The input forward qual file for de novo assembly."/>
    <param name="refLength" value="4600000" type="integer" label="Expected
length of sequenced (or enriched) DNA region" help="e.g., 4600000 for
E.Coli 4.6Mb
 genome or 30000000 for Whole Human Transcriptome."/>
    <param name="r3_csfasta" type="data" format="csfasta" label="SOLiD R3
csfasta file" help="The input R3 csfasta file for de novo assembly."/>
        <param name="r3_qual" type="data" format="qualsolid" label="SOLiD
R3 qual file" help="The input R3 qual file for de novo assembly."/>
    <param name="ins_length" value="1300" type="integer" label="Estimate of
insert length" help="e.g., xx = 1200 for mate-paired library data with
insert len
gth of 1.2Kb."/>
    <param name="ins_length_sd" value="400" type="integer" label="Estimate
of variance of the insert length" help="e.g., zz = 300 for mate-paired
library dat
a with insert length of 1.2Kb and zz = 30 for paired-end data with insert
size of 170bp."/>
    <param label="Number of CPUs/Cores" name="numcores" type="select"
help="Number of cores to parallelize SAET stage. Choose '1' if you have
only one CPU or
 Core.">
      <option value="1" selected="yes">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
      <option value="6">6</option>
      <option value="7">7</option>
      <option value="8">8</option>
      <option value="9">9</option>
      <option value="10">10</option>
      <option value="11">11</option>
      <option value="12">12</option>
      <option value="13">13</option>
      <option value="14">14</option>
      <option value="15">15</option>
      <option value="16">16</option>
    </param>
    <!--<requirements>
    <requirement type="package">saet_mp</requirement>
  </requirements> -->
  </inputs>
  <outputs>
```

```
    <data format="fasta" name="nt_contigs" label="${tool.name} on
${on_string}: Resulting nt_contigs.fa file after the de novo assembly
tool"/>
    <data format="txt" name="n50_stats" label="${tool.name} on
${on_string}: Resulting n50.stats.txt file for contigs after the de novo
assembly tool"/>
    <data format="fasta" name="nt_scaffolds" label="${tool.name} on
${on_string}: Resulting nt_scaffolds.fa file after the de novo assembly
tool"/>
    <data format="txt" name="n50_stats_scaff" label="${tool.name} on
${on_string}: Resulting n50.stats.txt file for scaffolds after the de novo
assembly tool
"/>
  </outputs>
  <help>
**SOLiD de novo accessory tool for mate-paired library overview**

Running de novo assembly pipeline for mate-paired library data run:

$denovo2/assemble.pl &lt;f3_csfasta&gt; &lt;f3_qual&gt; &lt;refLength&gt; -
r3 r3_csfasta -r3qv r3_qual -ins_length xx -ins_length_sd zz [-options]

Input:

 f3_csfasta - csfasta/fasta file with reads.

 f3_qual    - filename with quality values (if available). notice, that
order of reads in csfasta file should be the same as in quality value file.

 refLength  - expected length of sequenced DNA region, e.g., 4600000 for
E.Coli 4.6Mb genome.

Options required for mate-paired data:

 -r3 r3_csfasta     -  csfasta file with 2-base encoded R3 reads.

 -r3qv r3_qual      -  file with quality values for R3 reads. Notice, that
order of reads in csfasta file should be the same as in quality values
file.

Options required for mate-paired or paired-end data:

-ins_length xx      -  estimate of insert length, e.g., xx=1200 for mate-
paired library data with insert length 1.2Kb.

-ins_length_sd zz   -  estimate of variance of the insert length, e.g.,
zz=300 for mate-paired library data with insert length 1.2Kb and zz=30 for
paired-end
 data with insert size 170bp.

Output:

 &lt;outdir&gt;/nt_contigs.fa                       - fasta file with
assembled base-space contigs.

 &lt;outdir&gt;/analysis/nt_contigs/n50.stats.txt   - file containing base-
space contigs analysis.

 &lt;outdir&gt;/nt_scaffolds.fa                     - fasta file with
assembled base-space scaffolds (for mate-paired or paired-end data).

 &lt;outdir&gt;/analysis/nt_scaffolds/n50.stats.txt - file containing base-
space scaffolds analysis.
  </help>
</tool>
```

Arquivo "SOLiD_denovo_mp.py"

```
#!/usr/bin/env python
#Python wrapper for adapting the SOLiD (TM) de novo accessory tools 2.0
pipeline for mate-paired library in Galaxy.
#Adapted by Antonio, based on the "velvetg_wrapper.py" for Galaxy.
import pkg_resources
import logging, os, string, sys, tempfile, glob, shutil, types, urllib
import shlex, subprocess
from optparse import OptionParser, OptionGroup
from stat import *

log = logging.getLogger( __name__ )

assert sys.version_info[:2] >= ( 2, 4 )

def stop_err( msg ):
    sys.stderr.write( "%s\n" % msg )
    sys.exit()

def __main__():
    #Parse Command Line
    s = 'SOLiD_denovo_mp.py:  argv = %s\n' % (sys.argv)
    argcnt = len(sys.argv)
    #html_file = sys.argv[1]
    working_dir = sys.argv[1]
    csfasta = sys.argv[2]
    qual = sys.argv[3]
    refLength = sys.argv[4]
    r3_csfasta = sys.argv[5]
    r3_qual = sys.argv[6]
    ins_length = sys.argv[7]
    ins_length_sd = sys.argv[8]
    numcores = sys.argv[9]
    nt_contigs = sys.argv[10]
    n50_stats = sys.argv[11]
    nt_scaffolds = sys.argv[12]
    n50_stats_scaff = sys.argv[13]
    #inputs = string.join(sys.argv[9:],' ')
    # print >> sys.stderr, cmdline # so will appear as blurb for file
    #try: # for test - needs this done
    #    os.makedirs(working_dir)
    #except Exception, e:
    #    stop_err( 'Error running SOLiD_denovo_mp.py ' + str( e ) )
    #cmdline = '%s/assemble.pl %s %s %s -r3 %s -r3qv %s -ins_length %s -
ins_length_sd %s > /dev/null' % (os.path.dirname(sys.argv[0]), csfasta,
qual, refLeng
th, r3_csfasta, r3_qual, ins_length, ins_length_sd)
    cmdline = '%s/assemble_Antonio.pl %s %s %s -r3 %s -r3qv %s -ins_length
%s -ins_length_sd %s -outdir %s -numcores %s' %
(os.path.dirname(sys.argv[0]), csf
asta, qual, refLength, r3_csfasta, r3_qual, ins_length, ins_length_sd,
working_dir, numcores)
    try:
        proc = subprocess.Popen( args=cmdline, shell=True,
stderr=subprocess.PIPE )
        returncode = proc.wait()
        # get stderr, allowing for case where it's very large
        stderr = ''
        buffsize = 1048576
        try:
            while True:
                stderr += proc.stderr.read( buffsize )
                if not stderr or len( stderr ) % buffsize != 0:
                    break
        except OverflowError:
            pass
        if returncode != 0:
            raise Exception, stderr
    except Exception, e:
        stop_err( 'Error running SOLiD_denovo_mp.py ' + str( e ) )
```

```
    out = open(nt_contigs,'w')
    nt_contigs_path = os.path.join(working_dir,'nt_contigs.fa')
    for line in open( nt_contigs_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(n50_stats,'w')
    n50_stats_path =
os.path.join(working_dir,'analysis/nt_contigs/n50.stats.txt')
    for line in open( n50_stats_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(nt_scaffolds,'w')
    nt_scaffolds_path = os.path.join(working_dir,'nt_scaffolds.fa')
    for line in open( nt_scaffolds_path ):
        out.write( "%s" % (line) )
    out.close()
    out = open(n50_stats_scaff,'w')
    n50_stats_scaff_path =
os.path.join(working_dir,'analysis/nt_scaffolds/n50.stats.txt')
    for line in open( n50_stats_scaff_path ):
        out.write( "%s" % (line) )
    out.close()

if __name__ == "__main__": __main__()
```

Apêndice G - Relatório da ferramenta *NCBI BLAST+ blastn* para o gene conhecido AY370533.1 (*Leishmania amazonensis*) em relação aos dados de montagem (com auxílio de genoma de referência) de *Leishmania amazonensis*

Nota: As "marcas de texto" coloridas abaixo não foram produzidas pela ferramenta, mas sim pelo programa editor de textos deste trabalho, apenas para fins ilustrativos.

```
BLAST Search Results


BLASTN 2.2.26+

Query= gi|38326706|gb|AY370533.1| Leishmania amazonensis inosine 5'
monophosphate dehydrogenase (impdh) gene, complete cds

Length=1545

Subject= Draft_Lamaz_19

Length=655043

 Score = 2782 bits (3084),  Expect = 0.0
 Identities = 1544/1545 (99%), Gaps = 0/1545 (0%)
 Strand=Plus/Plus

Query  1       ATGGCGACCAACAACGCGAACTACCGTATCAAGACCATCAAGGATGGCTGCACCGCCGAG  60
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  626994  ATGGCGACCAACAACGCGAACTACCGTATCAAGACCATCAAGGATGGCTGCACCGCCGAG  627053

Query  61      GAGCTGTTCCAGGGTGATGGGCTGACGTACAATGACTTTATTATTCTGCCGGGCTTCATC  120
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627054  GAGCTGTTCCAGGGTGATGGGCTGACGTACAATGACTTTATTATTCTGCCGGGCTTCATC  627113

Query  121     GACTTTGGCGCTTCCGATGTGAACATCTCTGGCCAGTTCACGAAGCGCATCCGCCTCCAC  180
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627114  GACTTTGGCGCTTCCGATGTGAACATCTCTGGCCAGTTCACGAAGCGCATCCGCCTCCAC  627173

Query  181     ATCCCGATCGTGTCGTCGCCGATGGACACCATCACGGAGAACGAGATGGCGAAGACAATG  240
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627174  ATCCCGATCGTGTCGTCGCCGATGGACACCATCACGGAGAACGAGATGGCGAAGACAATG  627233

Query  241     GCACTCATGGGCGGCGTCGGGGTGCTGCACAACAACTGCACGGTGGAGCGGCAAGTAGAG  300
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627234  GCACTCATGGGCGGCGTCGGGGTGCTGCACAACAACTGCACGGTGGAGCGGCAAGTAGAG  627293

Query  301     ATGGTGAAGTCGGTGAAGGCGTACCGCAACGGATTCATCTCCAAGCCCAAGTCGGTGCCG  360
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627294  ATGGTGAAGTCGGTGAAGGCGTACCGCAACGGATTCATCTCCAAGCCCAAGTCGGTGCCG  627353

Query  361     CCGAACACTCCCATCAGCAAGATCATCCGCATCAAGGAGGAGAAGGGGATCAGTGGCATT  420
               |||||||| ||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627354  CCGAACACCCCCATCAGCAAGATCATCCGCATCAAGGAGGAGAAGGGGATCAGTGGCATT  627413

Query  421     CTTGTGACGGAGAACGGCGACCCGCACGGCAAGCTGCTTGGCATCGTGTGCACGAAGGAT  480
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627414  CTTGTGACGGAGAACGGCGACCCGCACGGCAAGCTGCTTGGCATCGTGTGCACGAAGGAT  627473

Query  481     ATCGACTACGTAAAGAACAAGGACACACCGGTATCGGCGGTTATGACGCGACGCGAGAAG  540
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627474  ATCGACTACGTAAAGAACAAGGACACACCGGTATCGGCGGTTATGACGCGACGCGAGAAG  627533

Query  541     ATGACGGTGGAGCGTGCACCGATCCAGCTGGAAGAGGCAATGGACGTGCTGAACCGCAGC  600
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627534  ATGACGGTGGAGCGTGCACCGATCCAGCTGGAAGAGGCAATGGACGTGCTGAACCGCAGC  627593

Query  601     CGCTACGGCTACCTGCCCATTGTGAATGAGAACGGCGAGGTCGTCAATCTCTGCTCCCGC  660
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627594  CGCTACGGCTACCTGCCCATTGTGAATGAGAACGGCGAGGTCGTCAATCTCTGCTCCCGC  627653

Query  661     CGCGATGCTGTCCGCGCGCGTGACTACCCACACAGCACACTGGACAAGAGCGGTCGACTC  720
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627654  CGCGATGCTGTCCGCGCGCGTGACTACCCACACAGCACACTGGACAAGAGCGGTCGACTC  627713
```

```
Query  721     ATCTGCGCTGCCGCGACCTCAACGCGCCCGGAGGACAAGCGGCGAGTGGCAACCCTGGCG  780
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627714  ATCTGCGCTGCCGCGACCTCAACGCGCCCGGAGGACAAGCGGCGAGTGGCAACCCTGGCG  627773

Query  781     GAAGTCGGCGTTGATGTTCTGGTTCTGGACAGCTCTCAGGGAAACACGATCTACCAGATC  840
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627774  GAAGTCGGCGTTGATGTTCTGGTTCTGGACAGCTCTCAGGGAAACACGATCTACCAGATC  627833

Query  841     GCTTTCATCAAGTGGGTCAAGTCGACGTATCCGCACCTCGAAGTGGTGGCAGGAAATGTG  900
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627834  GCTTTCATCAAGTGGGTCAAGTCGACGTATCCGCACCTCGAAGTGGTGGCAGGAAATGTG  627893

Query  901     GTGACGCAAGATCAGGCGAAGAACCTCATTGATGCCGGCGCGGACGGTATTCGCATCGGC  960
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627894  GTGACGCAAGATCAGGCGAAGAACCTCATTGATGCCGGCGCGGACGGTATTCGCATCGGC  627953

Query  961     ATGGGCAGCGGGAGCATCTGCATCACGCAGGAGGTTCTAGCTTGCGGTCGTCCTCAGGGC  1020
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  627954  ATGGGCAGCGGGAGCATCTGCATCACGCAGGAGGTTCTAGCTTGCGGTCGTCCTCAGGGC  628013

Query  1021    ACGGCGGTGTACAAGGTGGCACAGTACTGCGCATCTCGCGGCGTTCCGTGTACCGCTGAC  1080
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628014  ACGGCGGTGTACAAGGTGGCACAGTACTGCGCATCTCGCGGCGTTCCGTGTACCGCTGAC  628073

Query  1081    GGCGGTCTTCGCCAGGTCGGCGACATCTGCAAGGCGCTCGCCATCGGTGCCAACTGCGCG  1140
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628074  GGCGGTCTTCGCCAGGTCGGCGACATCTGCAAGGCGCTCGCCATCGGTGCCAACTGCGCG  628133

Query  1141    ATGCTCGGCGGCATGTTGAGTGGCACGACTGAGACGCCTGGCGAGTACTTCTTCAAGGGC  1200
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628134  ATGCTCGGCGGCATGTTGAGTGGCACGACTGAGACGCCTGGCGAGTACTTCTTCAAGGGC  628193

Query  1201    GGCGTGCGGCTGAAGGTGTACCGCGGCATGGGTAGCCTGGAGGCGATGAACCAGGGCAAG  1260
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628194  GGCGTGCGGCTGAAGGTGTACCGCGGCATGGGTAGCCTGGAGGCGATGAACCAGGGCAAG  628253

Query  1261    GAGTCCGGCAAGCGCTACCTCTCCGAGAACGAGGCGGTGCAGGTTGCACAGGGCGTGTCG  1320
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628254  GAGTCCGGCAAGCGCTACCTCTCCGAGAACGAGGCGGTGCAGGTTGCACAGGGCGTGTCG  628313

Query  1321    GGGAGCGTCGTGGATAAGGGCTCTGCCGCGAAGCTCATCGCCTACGTCTCGAAAGGGCTC  1380
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628314  GGGAGCGTCGTGGATAAGGGCTCTGCCGCGAAGCTCATCGCCTACGTCTCGAAAGGGCTC  628373

Query  1381    CAGCAGTCGGCGCAGGACATCGGCGAGATCAGCTTCGACGCGATTCGCGAGAAGATGTAC  1440
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628374  CAGCAGTCGGCGCAGGACATCGGCGAGATCAGCTTCGACGCGATTCGCGAGAAGATGTAC  628433

Query  1441    GCTGGCCAGGTACTTTTCAGCCGCCGCTCCCCCACGGCCCAGGGCGAGGGTGGCGTGCAC  1500
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628434  GCTGGCCAGGTACTTTTCAGCCGCCGCTCCCCCACGGCCCAGGGCGAGGGTGGCGTGCAC  628493

Query  1501    TCGCTTCACAGCTACGAGAAGAAGCTGTTTGCAGCCAAGATGTAA  1545
               |||||||||||||||||||||||||||||||||||||||||||||
Sbjct  628494  TCGCTTCACAGCTACGAGAAGAAGCTGTTTGCAGCCAAGATGTAA  628538

Lambda     K      H
   0.634    0.408    0.912

Gapped
Lambda     K      H
   0.625    0.410    0.780

Effective search space used: 996940440
```

Apêndice H - Região do gene AY370533.1 (*Leishmania amazonensis*) adicionada de 1 kpb *upstream* e *downstream* e "extraída" pela ferramenta *Extract region tool* a partir dos dados de montagem do genoma de *Leishmania amazonensis*

Nota: As "marcas de texto" coloridas abaixo não foram produzidas pela ferramenta, mas sim pelo programa editor de textos deste trabalho, apenas para fins ilustrativos.

>Draft_Lamaz_19 orig_len=655043, extract 625994-629538 (3545 bp)
TCGCTGTCATGTGCCTGTGCCGTTTGAGTTGTTGTTTATAGAGGGCGGCTGTTGGGTTGGTAGTACCTTTCTTGTGCGTGTGTGCGCGCGTATG
AGCGATGACTGCAGTGGGCTGTGCATACCGCTCCAAGTCCTCTCCCGTTTTTACTGCTCTTCCGCTTGCCTGTGGCTTCACCTCACCCGTGCGT
GCGTCGTGGACAACACAGCGGACGGGAGCCGAACGTGCGCaatgatgtacattgtgcACTCCTaccgaaCCACcGAGCAACAGCAGCGGCAGTG
GAAAGGAATGCAAACGAACACTCGAACGCACACCAACAGGAAGAGTACCCGACAAGCACGACATCGCTGGCATCCGCATAGACGTGAAGCGGAG
GAGTAGGCCAAGCCGGGATGGAGGCGAGTCTCACACCATGTGCAAGCCTACGCGAatttgacctttttcatggaaGCGATGAAGGAAAAGAGCG
ACGCCCCTCTGCGTGCAGCTGTGGGCCGATGCATGCTTTTCGCCTCTTTGTGTCCCCTTCACCATGCCGCGTCCCGACTCCTCTGCATCTTGTG
GCGCTGTGCGTGGAGGGAGGGGGGAGAGGCTACACGATATCACCTCCGGTGGCTGTCGTCTGCGGATGCATGCACGCACTTTCGACGCCAAGGC
AGTGGCGCTAAACTATTCTCACGCTTTGACGCCTCatTCCTGTCTCTGCCGTGCGCTCTACGCTGATGCGGACGCTCACACGCGTCCACGTTCG
AAGCTCACTTTTCTCTGTCTTCTCTCTTTTCATTTGGTTGCTTGCGTTGTGCATCTTTCTACAGCGTGGCAGCGACATACTGCGCGTCACTCTC
TTGGCCTCTCTTTCTCGGTCTTTGATTTTATATACCTTATTCCTCTCCCAACTCTCTCGTTCGCCGAGCCTTCCCGCTGTAGCGCAGCCCTTCA
AGGCACATTCGTCCTTCGTGCACccgcttcagttgtgtgtgtgtGTGTGCTATCGACGCAATGGCGACCAACAACGCGAACTACCGTATCAAGA
CCATCAAGGATGGCTGCACCGCCGAGGAGCTGTTCCAGGGTGATGGGCTGACGTACAATGACTTTATTATTCTGCCGGGCTTCATCGACTTTGG
CGCTTCCGATGTGAACATCTCTGGCCAGTTCACGAAGCGCATCCGCCTCCACATCCCGATCGTGTCGTCGCCGATGGACACCATCACGGAGAAC
GAGATGGCGAAGACAATGGCACTCATGGGCGGCGTCGGGGTGCTGCACAACAACTGCACGGTGGAGCGGCAAGTAGAGATGGTGAAGTCGGTGA
AGGCGTACCGCAACGGATTCATCTCCAAGCCCAAGTCGGTGCCGCCGAACACCCCCATCAGCAAGATCATCCGCATCAAGGAGGAGAAGGGGAT
CAGTGGCATTCTTGTGACGGAGAACGGCGACccGCacgGCAAGCTGCTtgGcatcgtgtgcacgaagGATATCGACTACGTAAAGAACAAGGAC
ACACCGGTATCGGCGGTTATGACGCGACGCGAGAAGATGACGGTGGAGCGTGCACCGATCCAGCTGGAAGAGGCAATGGACGTGCTGAACCGCA
GCCGCTACGGCTACCTGCCCATTGTGAATGAGAACGGCGAGGTCGTCAATCTCTGCTCCCGCCGCGATGCTGTCCGCGCGCGTGACTACCCACA
CAGCACACTGGACAAGAGCGGTCGACTCATCTGCGCTGCCGCGACCTCAACGCGCCCGGAGGACAAGCGGCGAGTGGCAACCCTGGCGGAAGTC
GGCGTTGATGTTCTGGTTCTGGACAGCTCTCAGGGAAACACGATCTACCAGATCGCTTTCATCAAGTGGGTCAAGTCGACGTATCCGCACCTCG
AAGTGGTGGCAGGAAATGTGGTGACGCAAGATCAGGCGAAGAACCTCATTGATGCCGGCGCGGACGGTATTCGCATCGGCATGGGCAGCGGGAG
CATCTGCATCACGCAGGAGGTTCTAGCTTGCGGTCGTCCTCAGGGCACGGCGGTGTACAAGGTGGCACAGTACTGCGCATCTCGCGGCGTTCCG
TGTACCGCTGACGGCGGTCTTCGCCAGGTCGGCGACATCTGCAAGGCGCTCGCCATCGGTGCCAACTGCGCGATGCTCGGCGGCATGTTGAGTG
GCACGACTGAGACGCCTGGCGAGTACTTCTTCAAGGGCGGCGTGCGGCTGAAGGTGTACCGCGGCATGGGTAGCCTGGAGGCGATGAACCAGGG
CAAGGAGTCCGGCAAGCGCTACCTCTCCGAGAACGAGGCGGTGCAGGTTGCACAGGGCGTGTCGGGGAGCGTCGTGGATAAGGGCTCTGCCGCG
AAGCTCATCGCCTACGTCTCGAAAGGGCTCCAGCAGTCGGCGCAGGACATCGGCGAGATCAGCTTCGACGCGATTCGCGAGAAGATGTACGCTG
GCCAGGTACTTTTCAGCCGCCGCTCCCCCACGGCCCAGGGCGAGGGTGGCGTGCACTCGCTTCACAGCTACGAGAAGAAGCTGTTTGCAGCCAA
GATGTAAGACGTGCAGGCAGTTCTCCTCCCTTCGCCTACCTGCCCTCCCCTCTCGCATCATCCCCGTGCCCTTCTGTTTGCTGCTGCTTGAGCT
TGAAAACGAACGTGTCTTGTTCGACGACTTGGCGTCTTTCACTGCTCCTGGCCAGCTCAGTGTTGTGGAGCGCACGCACGCAAAAGACGCAGCC
CATACATTccccccccctccccaagagGTCAAGTTGCCCTCACAGGCTACTCTGAGCACGGCTTGGgtgagaccaancccgtcatcTTTTTTT
CTTGGCTGTTGTGTCCTGCGAGCCTGGCGTGCAAGAGGCATTCACCACGTGGCCCCTGAAATGGGGCAGGAGGGCTGCAGAGGTGCACGTGtgc
gcgacttgggggggtagTCAAAGAGAGgagagagaaggggggngggggcGGAGTTGGTCTTGTCTACGCAATATGCTTGTGTGCCGCTCGAACTGC
CCAGTGCGTTTTGTGTGCATCTTCATGGGGAGAGCACCAGCTTCTCATATGGAGAAGACTCATGAGCAACATGAACGTCGTCGGCAACGTTTGG
CAGGTGCCTGGGGAATGAGTCGTCTCTCCGTCTCTCGCACCTGTCTCCGTGTGCGTGATCTGTCTTACGTTTCCTTGCCGAACCTCTCGACCAG
CGAGCCTGCCTCGCCATCGCATGAAGAGGCCTCTTCTCAACGCAAGTTGACacgTAATCTTAATCGACCTAACGCACTCAAGGCATAACATATC
ATATCCCCCCTCGCCTCCGCCTGGCCCGTTGTTTCGCGCCGCATTCCTCGGTATCGCTTTGAGTGAGATACCCACCGACGCGCCATCCCAAGCA
GTACAGGCGtctgcaagcgnncgcgtgccTCTCCCTCTGTTGTTGCTGGCATTTCTTTCGCTACTGGGCTCCTCTCTCCTTTTTATCTGTGTTG
AGTGGCGCTTCGGCTTtgtaaatcttcacgcacgcgcgcACAAACGTGCTGGGGGCCGCCGTCGGGT

Apêndice I - Relatório da ferramenta *NCBI BLAST+ blastn* para o gene conhecido AY370533.1 (*Leishmania amazonensis*) em relação aos dados de montagem *de novo* de *Leishmania amazonensis*

Nota: As "marcas de texto" coloridas abaixo não foram produzidas pela ferramenta, mas sim pelo programa editor de textos deste trabalho, apenas para fins ilustrativos.

```
Query= gi|38326706|gb|AY370533.1| Leishmania amazonensis inosine 5'
monophosphate dehydrogenase (impdh) gene, complete cds

Length=1545

Subject= NODE_3014_length_25030_cov_21.060007

Length=25060

 Score = 2782 bits (3084),  Expect = 0.0
 Identities = 1544/1545 (99%), Gaps = 0/1545 (0%)
 Strand=Plus/Plus

Query  1     ATGGCGACCAACAACGCGAACTACCGTATCAAGACCATCAAGGATGGCTGCACCGCCGAG  60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  6855  ATGGCGACCAACAACGCGAACTACCGTATCAAGACCATCAAGGATGGCTGCACCGCCGAG  6914

Query  61    GAGCTGTTCCAGGGTGATGGGCTGACGTACAATGACTTTATTATTCTGCCGGGCTTCATC  120
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  6915  GAGCTGTTCCAGGGTGATGGGCTGACGTACAATGACTTTATTATTCTGCCGGGCTTCATC  6974

Query  121   GACTTTGGCGCTTCCGATGTGAACATCTCTGGCCAGTTCACGAAGCGCATCCGCCTCCAC  180
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  6975  GACTTTGGCGCTTCCGATGTGAACATCTCTGGCCAGTTCACGAAGCGCATCCGCCTCCAC  7034

Query  181   ATCCCGATCGTGTCGTCGCCGATGGACACCATCACGGAGAACGAGATGGCGAAGACAATG  240
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7035  ATCCCGATCGTGTCGTCGCCGATGGACACCATCACGGAGAACGAGATGGCGAAGACAATG  7094

Query  241   GCACTCATGGGCGGCGTCGGGGTGCTGCACAACAACTGCACGGTGGAGCGGCAAGTAGAG  300
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7095  GCACTCATGGGCGGCGTCGGGGTGCTGCACAACAACTGCACGGTGGAGCGGCAAGTAGAG  7154

Query  301   ATGGTGAAGTCGGTGAAGGCGTACCGCAACGGATTCATCTCCAAGCCCAAGTCGGTGCCG  360
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7155  ATGGTGAAGTCGGTGAAGGCGTACCGCAACGGATTCATCTCCAAGCCCAAGTCGGTGCCG  7214

Query  361   CCGAACACTCCCATCAGCAAGATCATCCGCATCAAGGAGGAGAAGGGGATCAGTGGCATT  420
             |||||||| ||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7215  CCGAACACCCCCATCAGCAAGATCATCCGCATCAAGGAGGAGAAGGGGATCAGTGGCATT  7274

Query  421   CTTGTGACGGAGAACGGCGACCCGCACGGCAAGCTGCTTGGCATCGTGTGCACGAAGGAT  480
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7275  CTTGTGACGGAGAACGGCGACCCGCACGGCAAGCTGCTTGGCATCGTGTGCACGAAGGAT  7334

Query  481   ATCGACTACGTAAAGAACAAGGACACACCGGTATCGGCGGTTATGACGCGACGCGAGAAG  540
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7335  ATCGACTACGTAAAGAACAAGGACACACCGGTATCGGCGGTTATGACGCGACGCGAGAAG  7394

Query  541   ATGACGGTGGAGCGTGCACCGATCCAGCTGGAAGAGGCAATGGACGTGCTGAACCGCAGC  600
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7395  ATGACGGTGGAGCGTGCACCGATCCAGCTGGAAGAGGCAATGGACGTGCTGAACCGCAGC  7454

Query  601   CGCTACGGCTACCTGCCCATTGTGAATGAGAACGGCGAGGTCGTCAATCTCTGCTCCCGC  660
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7455  CGCTACGGCTACCTGCCCATTGTGAATGAGAACGGCGAGGTCGTCAATCTCTGCTCCCGC  7514

Query  661   CGCGATGCTGTCCGCGCGCGTGACTACCCACACAGCACACTGGACAAGAGCGGTCGACTC  720
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7515  CGCGATGCTGTCCGCGCGCGTGACTACCCACACAGCACACTGGACAAGAGCGGTCGACTC  7574

Query  721   ATCTGCGCTGCCGCGACCTCAACGCGCCCGGAGGACAAGCGGCGAGTGGCAACCCTGGCG  780
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7575  ATCTGCGCTGCCGCGACCTCAACGCGCCCGGAGGACAAGCGGCGAGTGGCAACCCTGGCG  7634

Query  781   GAAGTCGGCGTTGATGTTCTGGTTCTGGACAGCTCTCAGGGAAACACGATCTACCAGATC  840
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7635  GAAGTCGGCGTTGATGTTCTGGTTCTGGACAGCTCTCAGGGAAACACGATCTACCAGATC  7694
```

```
Query  841   GCTTTCATCAAGTGGGTCAAGTCGACGTATCCGCACCTCGAAGTGGTGGCAGGAAATGTG  900
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7695  GCTTTCATCAAGTGGGTCAAGTCGACGTATCCGCACCTCGAAGTGGTGGCAGGAAATGTG  7754

Query  901   GTGACGCAAGATCAGGCGAAGAACCTCATTGATGCCGGCGCGGACGGTATTCGCATCGGC  960
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7755  GTGACGCAAGATCAGGCGAAGAACCTCATTGATGCCGGCGCGGACGGTATTCGCATCGGC  7814

Query  961   ATGGGCAGCGGGAGCATCTGCATCACGCAGGAGGTTCTAGCTTGCGGTCGTCCTCAGGGC  1020
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7815  ATGGGCAGCGGGAGCATCTGCATCACGCAGGAGGTTCTAGCTTGCGGTCGTCCTCAGGGC  7874

Query  1021  ACGGCGGTGTACAAGGTGGCACAGTACTGCGCATCTCGCGGCGTTCCGTGTACCGCTGAC  1080
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7875  ACGGCGGTGTACAAGGTGGCACAGTACTGCGCATCTCGCGGCGTTCCGTGTACCGCTGAC  7934

Query  1081  GGCGGTCTTCGCCAGGTCGGCGACATCTGCAAGGCGCTCGCCATCGGTGCCAACTGCGCG  1140
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7935  GGCGGTCTTCGCCAGGTCGGCGACATCTGCAAGGCGCTCGCCATCGGTGCCAACTGCGCG  7994

Query  1141  ATGCTCGGCGGCATGTTGAGTGGCACGACTGAGACGCCTGGCGAGTACTTCTTCAAGGGC  1200
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  7995  ATGCTCGGCGGCATGTTGAGTGGCACGACTGAGACGCCTGGCGAGTACTTCTTCAAGGGC  8054

Query  1201  GGCGTGCGGCTGAAGGTGTACCGCGGCATGGGTAGCCTGGAGGCGATGAACCAGGGCAAG  1260
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  8055  GGCGTGCGGCTGAAGGTGTACCGCGGCATGGGTAGCCTGGAGGCGATGAACCAGGGCAAG  8114

Query  1261  GAGTCCGGCAAGCGCTACCTCTCCGAGAACGAGGCGGTGCAGGTTGCACAGGGCGTGTCG  1320
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  8115  GAGTCCGGCAAGCGCTACCTCTCCGAGAACGAGGCGGTGCAGGTTGCACAGGGCGTGTCG  8174

Query  1321  GGGAGCGTCGTGGATAAGGGCTCTGCCGCGAAGCTCATCGCCTACGTCTCGAAAGGGCTC  1380
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  8175  GGGAGCGTCGTGGATAAGGGCTCTGCCGCGAAGCTCATCGCCTACGTCTCGAAAGGGCTC  8234

Query  1381  CAGCAGTCGGCGCAGGACATCGGCGAGATCAGCTTCGACGCGATTCGCGAGAAGATGTAC  1440
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  8235  CAGCAGTCGGCGCAGGACATCGGCGAGATCAGCTTCGACGCGATTCGCGAGAAGATGTAC  8294

Query  1441  GCTGGCCAGGTACTTTTCAGCCGCCGCTCCCCCACGGCCCAGGGCGAGGGTGGCGTGCAC  1500
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  8295  GCTGGCCAGGTACTTTTCAGCCGCCGCTCCCCCACGGCCCAGGGCGAGGGTGGCGTGCAC  8354

Query  1501  TCGCTTCACAGCTACGAGAAGAAGCTGTTTGCAGCCAAGATGTAA  1545
             |||||||||||||||||||||||||||||||||||||||||||||
Sbjct  8355  TCGCTTCACAGCTACGAGAAGAAGCTGTTTGCAGCCAAGATGTAA  8399



Lambda     K      H
   0.634    0.408    0.912

Gapped
Lambda     K      H
   0.625    0.410    0.780

Effective search space used: 78234
```

Anexos

Anexo A - Tabelas típicas de ferramentas disponíveis para NGS

Lista disponível em Shendure e Ji (2008)

Table 3 Bioinformatics tools for short-read sequencing

Program	Categories	Author(s)	Reference	URL
Cross_match	Alignment	Phil Green, Brent Ewing and David Gordon		http://www.phrap.org/phredphrapconsed.html
ELAND	Alignment	Anthony J. Cox		http://www.illumina.com/
Exonerate	Alignment	Guy S. Slater and Ewan Birney	72	http://www.ebi.ac.uk/~guy/exonerate
MAQ	Alignment and variant detection	Heng Li	37	http://maq.sourceforge.net
Mosaik	Alignment	Michael Strömberg and Gabor Marth		http://bioinformatics.bc.edu/marthlab/Mosaik
RMAP	Alignment	Andrew Smith, Zhenyu Xuan and Michael Zhang	73	http://rulai.cshl.edu/rmap
SHRiMP	Alignment	Michael Brudno and Stephen Rumble		http://compbio.cs.toronto.edu/shrimp
SOAP	Alignment	Ruiqiang Li *et al.*	35	http://soap.genomics.org.cn
SSAHA2	Alignment	Zemin Ning *et al.*	36	http://www.sanger.ac.uk/Software/analysis/SSAHA2
SXOligoSearch	Alignment	Synamatix		http://synasite.mgrc.com.my:8080/sxog/NewSXOligoSearch.php
ALLPATHS	Assembly	Jonathan Butler *et al.*	38	
Edena	Assembly	David Hernandez *et al.*	74	http://www.genomic.ch/edena
Euler-SR	Assembly	Mark Chaisson and Pavel Pevzner	75	
SHARCGS	Assembly	Juliane Dohm *et al.*	76	http://sharcgs.molgen.mpg.de
SHRAP	Assembly	Andreas Sundquist *et al.*	39	
SSAKE	Assembly	René Warren *et al.*	40	http://www.bcgsc.ca/platform/bioinfo/software/ssake
VCAKE	Assembly	William Jeck	77	http://sourceforge.net/projects/vcake
Velvet	Assembly	Daniel Zerbino and Ewan Birney	41	http://www.ebi.ac.uk/%7Ezerbino/velvet
PyroBayes	Base caller	Aaron Quinlan *et al.*	34	http://bioinformatics.bc.edu/marthlab/PyroBayes
PbShort	Variant detection	Gabor Marth		http://bioinformatics.bc.edu/marthlab/PbShort
ssahaSNP	Variant detection	Zemin Ning *et al.*		http://www.sanger.ac.uk/Software/analysis/ssahaSNP

Incomplete list compiled from sources, including http://seqanswers.com/forums/showthread.php?t=43 and http://www.sanger.ac.uk/Users/lh3/seq-nt.html.

34. Quinlan, A.R., Stewart, D.A., Stromberg, M.P. & Marth, G.T. Pyrobayes: an improved base caller for SNP discovery in pyrosequences. *Nat. Methods* 5, 179–181 (2008).
35. Li, R., Li, Y., Kristiansen, K. & Wang, J. SOAP: short oligonucleotide alignment program. *Bioinformatics* 24, 713–714 (2008).
36. Ning, Z., Cox, A.J. & Mullikin, J.C. SSAHA: a fast search method for large DNA databases. *Genome Res.* 11, 1725–1729 (2001).
37. Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* published online, doi:10.1101/gr.078212.108 (19 August 2008).
38. Butler, J. *et al.* ALLPATHS: de novo assembly of whole-genome shotgun microreads.*Genome Res.* 18, 810–820 (2008).
39. Sundquist, A., Ronaghi, M., Tang, H., Pevzner, P. & Batzoglou, S. Whole-genome sequencing and assembly with high-throughput, short-read technologies. *PLoS ONE* 2, e484 (2007).
40. Warren, R.L., Sutton, G.G., Jones, S.J. & Holt, R.A. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics* 23, 500–501 (2007).
41. Zerbino, D.R. & Birney, E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 18, 821–829 (2008).
72. Slater, G.S. & Birney, E. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* 6, 31 (2005).
73. Smith, A.D., Xuan, Z. & Zhang, M.Q. Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC Bioinformatics* 9, 128 (2008).
74. Hernandez, D., Francois, P., Farinelli, L., Osteras, M. & Schrenzel, J. De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. *Genome Res.* 18, 802–809 (2008).
75. Chaisson, M.J. & Pevzner, P.A. Short read fragment assembly of bacterial genomes. *Genome Res.* 18, 324–330 (2008).
76. Dohm, J.C., Lottaz, C., Borodina, T. & Himmelbauer, H. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res.* 17, 1697–1706 (2007).
77. Jeck, W.R. *et al.* Links extending assembly of short DNA sequences to handle error. *Bioinformatics* 23, 2942–2944 (2007).

Table 2: Tools for the analysis of next-generation sequencing data in several application categories

Tool	Website	Category	Platform[a]
ELAND	http://www.illumina.com/pages.ilmn?ID=315	Alignment	GA
Soap	http://soap.genomics.org.cn	Alignment	GA
ZOOM	http://www.bioinfor.com	Alignment	GA, SO
PASS	http://pass.cribi.unipd.it	Alignment	GA, SO, GS
MOM	http://mom.csbc.vcu.edu	Alignment	GA
Vmatch	http://www.vmatch.de/	Alignment	GA
Bowtie	http://bowtie.cbcb.umd.edu	Alignment	GA
CloudBurst	http://cloudburst-bio.sourceforge.net/	Alignment	GA
BWA	http://maq.sourceforge.net/bwa-man.shtml	Alignment	GA
SHRiMP	http://compbio.cs.toronto.edu/shrimp/	Alignment	GA, SO
AB mapreads	http://solidsoftwaretools.com/gf/project/mapreads/	Alignment	SO
MuMRescueLite	http://genome.gsc.riken.jp/osc/english/dataresource/	Alignment	SO
MAQ	http://maq.sourceforge.net	Alignment	GA, SO
SeqMap	http://biogibbs.stanford.edu/~jiangh/SeqMap/	Alignment	GA
RMAP	http://rulai.cshl.edu/rmap/	Assembly	GA
FindPeaks	http://www.bcgsc.ca/platform/bioinfo/software/findpeaks	ChipSeq analysis	GA, SO, GS
F-Seq	http://www.genome.duke.edu/labs/furey/software/fseq	ChipSeq analysis	GA
SISSRS,	http://sissrs.rajajothi.com/	ChipSeq analysis	GA
QuEST	http://www.stanford.edu/~valouev/QuEST/QuEST.html	ChipSeq analysis	GA
MACS	http://liulab.dfci.harvard.edu/MACS/	ChipSeq analysis	GA
Chipseq.peak finder	http://woldlab.caltech.edu/html/software	ChipSeq analysis	GA
ChIPDiff	http://cmb.gis.a-star.edu.sg/ChIPSeq/paperChIPDiff.htm	ChipSeq analysis	GA
CisGenome	http://www.biostat.jhsph.edu/~hji/cisgenome/	ChipSeq analysis	GA
G-Mo.R-Se	http://www.genoscope.cns.fr/externe/gmorse/#Download	Gene annotation	GA
UEA plant sRNA toolkit	http://srna-tools.cmp.uea.ac.uk/	General smallRNA tools	GA
mirDeep	http://www.mdc-berlin.de/rajewsky/miRDeep	mirRNA identification	GA, SO, GS
MirCat	http://srna-tools.cmp.uea.ac.uk/	mirRNA identification	GA
QSRA	http://qsra.cgrb.oregonstate.edu/	short read assembly	GA
ALLPATHS	http://www.broadinstitute.org/science/programs/genome-biology/computational-rd/computational-research-and-development	short read assembly	GA
Velvet	http://www.ebi.ac.uk/~zerbino/velvet/	short read assembly	GA
EDENA	http://www.genomic.ch/edena.php	short read assembly	GA
VCAKE	http://mac.softpedia.com/get/Math-Scientific/VCAKE.shtml	short read assembly	GA, SO, GS
SHARCGS	http://sharcgs.molgen.mpg.de/download.shtml	short read assembly	GA
EULER-SR	http://euler-assembler.ucsd.edu/portal/	short read assembly	GA
SSAKE	http://www.bcgsc.ca/platform/bioinfo/software/ssake/releases/3.2	short read assembly	GA, SO, GS
CLEAVELAND	http://www.bio.psu.edu/people/faculty/Axtell/AxtellLab/Software.html	smallRNA target identification (plants)	GA
POLYBAYES	http://bioinformatics.bc.edu/marthlab/PolyBayes	SNP calling	GS, SO, GS
SLIDER	http://www.bcgsc.ca/platform/bioinfo/software/slider	SNP calling	GA, SO, GS
QPalma	http://www.fml.tuebingen.mpg.de/raetsch/suppl/qpalma	Spliced Read Mapping	GA
Tophat	http://tophat.cbcb.umd.edu/	Spliced Read Mapping; transcript quantification	GA
Erange	http://woldlab.caltech.edu/rnaseq/	Spliced Read Mapping; transcript quantification	GA
FluxCapacitor	http://flux.sammeth.net/	Transcript quantification	GA, SO, GS

A web version of this table, which is continuously updated, can be found at http://mi.caspur.it/ngs/software_review.php.
[a]GA, Illumina; SO, AB SOLID; GS, Roche 454 FLX.

Listas disponíveis em Bao et al. (2011)

Table 1 Tools for the analysis of next generation sequencing data

Program	*Website*	*Open source*	*Quality score involved*	*Mapping strategy*	*Description*	*Ref*
CloudBurst	http://sourceforge.net/apps/mediawiki/cloudburst-bio/index.php?title-CloudBurst	Yes	No	Hash the reads	Either all alignments or the unambiguous best alignment for each read with any number of mismatches or difference would be reported; running time required linearly increase with the number of reads mapped and near linearly decrease as the number of processors increase.	15
Eland	None	No	Yes	Hash the reads	Probably the first read aligner; works only for 32-bp single-end reads by itself, with GAPipeline extending its ability.	
Maq	http://maq.sourceforge.net	Yes	Yes	Hash the reads	Based on a so-called 'spaced seed indexing' strategy, it can efficiently winnow the candidate locations within the reference.	21
RMAP	http://rulai.cshl.edu/rmap/	Yes	Yes	Hash the reads	Can map reads with or without quality scores; supports paired-end reads or bisulfite-treated reads mapping; no limitations on read widths or number of mismatches.	26
SeqMap	http://biogibbs.stanford.edu/~jiangh/SeqMap/	Yes	No	Hash the reads	Maps dozens of millions of reads to a genome with several billions bp length; can deal with mutations, insertions/deletions; supports various input/output formats, command option lines are also available	27
SHRiMP	http://compbio.cs.toronto.edu/shrimp/	Yes	Yes	Hash the reads	SAM output format; supports both letter space and color space reads; allows paired-end reads alignment, parallel computation	23
ZOOM	http://www.bioinfor.com	No	Yes	Hash the reads	Based on spaced-seed strategy; 100% sensitivity for a wide range of read length and mismatches; a single CPU with 6.5 GB memory, is capable to map 15× coverage of a human genome in one day.	22
BFAST	http://sourceforge.net/projects/bfast/files/	Yes	Yes	Hash the genome	Fast and accurate mapping of tags to genome sequences.	28
MOM	http://mom.csbc.vcu.edu/	Yes	No	Hash the genome	No indels are allowed while mapping, but mismatches are tolerant; establishs a seed hash table for exactly matching short seeds between reference sequence and short reads.	29
Mosaik	http://bioinformatics.bc.edu/marthlab/Mosaik	Yes	Yes	Hash the genome	Based on Smith–Waterman algorithm; supports pair-wise alignments and produces reference-guided assemblies with gapped alignments; written in highly portable C++	
SSAHA2	http://www.sanger.ac.uk/resources/software/ssaha2/	Yes	Yes	Hash the genome	Support most sequencing platforms (ABI–Sanger, Roche 454, Illumina–Solexa); wild range of output formats (SAM, CIGAR, PSL, etc.) are available; a separate package for pile-up pipeline analysis and genotype calling is also included.	31
NovoAlign	http://www.novocraft.com	No	Yes	Hash the genome	Allows gaps up to 7 bp on single-end reads, even longer on paired-end reads aligns with up to eight or more mismatches per read, up to 16 on paired-end reads.	
PASS	http://pass.cribi.unipd.it	Yes	Yes	Hash the genome	Improves the execution time and sensitivity; performs fast gapped and ungapped alignments of short reads onto a reference genome; implemented in C++, supported on Linux and Windows	18
PerM	http://code.google.com/p/perm/	Yes	Yes	Hash the genome	High sensitivity and speed contributed by the use of periodic spaced seeds with higher weight; no paired-end mapping available now.	24
ProbeMatch	http://www.cs.wisc.edu/~jignesh/probematch/	Yes	No	Hash the genome	Tolerant for gapped and ungapped alignments with up to three errors; uses gapped *q*-grams and *q*-grams of various patterns to identify target hits to a query sequence.	30
Slider	http://www.bcgsc.ca/platform/bioinfo/software/slider	Yes	No	Merge sorting	High alignment accuracy and efficiency; with probabilities while matching bases, it reduces the percentage of base mismatches; high SNP discovery rate.	32
Slider II	http://www.bcgsc.ca/platform/bioinfo/software/slider	Yes	No	Merge sorting		32
Bowtie	http://bowtie.cbcb.umd.edu	Yes	Yes	BWT-based, index the genome	Borrows a technique called Burrows–Wheeler Transform (BWT), the algorithm is more complicated than Maq's, but more than 30-fold faster.	20
BWA	http://bio-bwa.sourceforge.net/bwa.shtml	Yes	Yes	BWT-based, index the genome	Implements two different algorithms, both based on Burrows–Wheeler Transform (BWT), the first algorithm is based on BWA-short for short queries up to ~200 bp with low error rate(<3%) and supports paired-end reads, the second algorithm, BWA-SW, is designed for long reads with more errors.	25
SOAP2	http://soap.genomics.org.cn/#	Yes	Yes	BWT-based, index the genome	An updated version of SOAP, in super fast and accurate alignment for large amounts of short reads from Illumina; supports a wide range of read length.	19

15 Schatz, M. C. CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics 25, 1363–1369 (2009).
18 Campagna, D., Albiero, A., Bilardi, A., Caniato, E., Forcato, C., Manavski, S. et al. PASS: a program to align short sequences. Bioinformatics 25, 967–968 (2009).
19 Li, R. Q., Li, Y. R., Kristiansen, K. & Wang, J. SOAP: short oligonucleotide alignment program. Bioinformatics 24, 713–714 (2008).
20 Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 10, R25 (2009).
21 Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. 18, 1851–1858 (2008).
22 Lin, H., Zhang, Z., Zhang, M. Q., Ma, B. & Li, M. ZOOM! Zillions of oligos mapped. Bioinformatics 24, 2431–2437 (2008).
23 Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A. & Brudno, M. SHRiMP: accurate mapping of short color-space reads. PLoS Comput. Biol. 5, e1000386 (2009).
24 Chen, Y., Souaiaia, T. & Chen, T. PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. Bioinformatics 25, 2514–2521 (2009).
25 Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25, 1754–1760 (2009).
26 Smith, A. D., Chung, W. Y., Hodges, E., Kendall, J., Hannon, G., Hicks, J. et al. Updates to the RMAP short-read mapping software. Bioinformatics 25, 2841–2842 (2009).
27 Jiang, H. & Wong, W. H. SeqMap: mapping massive amount of oligonucleotides to the genome. Bioinformatics 24, 2395–2396 (2008).
28 Homer, N., Merriman, B. & Nelson, S. F. BFAST: an alignment tool for large scale genome resequencing. PLoS One 4, e7767 (2009).
29 Eaves, H. L. & Gao, Y. MOM: maximum oligonucleotide mapping. Bioinformatics 25, 969–970 (2009).
30 Kim, Y. J., Teletia, N., Ruotti, V., Maher, C. A., Chinnaiyan, A. M., Stewart, R. et al. ProbeMatch: rapid alignment of oligonucleotides to genome allowing both gaps and mismatches. Bioinformatics 25, 1424–1425 (2009).
31 Ning, Z., Cox, A. J. & Mullikin, J. C. SSAHA: a fast search method for large DNA databases. Genome Res. 11, 1725–1729 (2001).
32 Malhis, N., Butterfield, Y. S. N., Ester, M. & Jones, S. J. M. Slider-maximum use of probability information for alignment of short sequence reads and SNP detection. Bioinformatics 25, 6–13 (2009).
33 Weese, D., Emde, A. K., Rausch, T., Doring, A. & Reinert, K. RazerS-fast read mapping with sensitivity control. Genome Res. 19, 1646–1654 (2009).

Table 4 Tools for *de novo* assembly analysis

Program	*Website*	*Strategy*	*NGS platforms*	*Overview*	*Ref*
QSRA	http://qsra.cgrb.oregonstate.edu/	Greedy	Sanger, Solexa	Quality-value guided Short Read Assembler, it is created to take advantage of quality-value scores to handle base call errors.	44
SHARCGS	http://sharcgs.molgen.mpg.de/index.shtml	Greedy	Solexa	Short-read Assembler based on Robust Contig extension for Genome Sequencing, suitable for un-paired reads (25–40 bp) with high coverage.	42
SSAKE	http://www.bcgsc.ca/platform/bioinfo/software/ssake	Greedy	Solexa (SOLiD?, Helicos?)	Short Sequence Assembly by progressive K-mer search and 3′ read Extension, with a prefix tree, it would progressively search for perfect 3′-most k-mers.	41
VCAKE	http://sourceforge.net/projects/vcake/	Greedy	Solexa (SOLiD?, Helicos?)	Verified Consensus Assembly by K-mer Extension, by using high depth coverage, it could assemble millions of short reads even in the presence of sequencing error.	43
CABOG	http://sourceforge.net/apps/mediawiki/wgs-assembler/index.php?title=Main_Page	OLC	Sanger, 454, Solexa	Celera Assembler with the Best Overlap Graph, robust to homopolymer run length uncertainty, high read coverage and heterogeneous read lengths.	45
Edena	http://www.genomic.ch/edena.php	OLC	Solexa	Exact *de novo* Assembler, based on overlap layout paradigm; uniform-length reads are indexed in a prefix array and all perfect, error-free contigs are produced.	46
Newbler	http://contig.wordpress.com/	OLC	454, Sanger	Particularly designed for 454 platforms; customs receive frequent updates; the source code is not generally available.	47
Shorty	http://www.cs.sunysb.edu/~skiena/shorty/	OLC	Helicos, Solexa, SOLiD	Using a few (5–10) seeds of length 300–500 bp to assemble short-paired reads; can accurately estimate intercontig distance from multiple spanning mate pairs.	48
ABySS	http://www.ncbi.nlm.nih.gov/pubmed/19251739	DBG	Solexa, SOLiD	Assembly By Short Sequences, a parallelized sequence assembler.	51
ALLPATHS	ftp://ftp.broadinstitute.org/pub/crd/ALLPATHS/	DBG	Solexa, SOLiD?	Two key concepts in the algorithm: 1). Finding all paths across a given read pair 2). Localization, using pairs to isolate regions of the genome and assemble them.	52
EULER-SR	http://euler-assembler.ucsd.edu/portal/	DBG	Sanger, 454, Solexa, SOLiD	Eulerian approach-based assembler, stated to be the assembler generating optimal short read assemblies of bacterial genomes.	53
SOAPdenovo	http://soap.genomics.org.cn/soapdenovo.html	DBG	Solexa	Has been integrated into the short oligonucleotide alignment program (SOAP) package; designed for large-genome assembly in a cost-effective way.	54
Velvet	http://www.ebi.ac.uk/~zerbino/velvet	DBG	Sanger, 454, Solexa, SOLiD	Ideal for short reads (25–50 bp) and paired-ends reads to produce contigs with significant length; tolerant color space reads.	55

Note: all the items in the fourth column, excluding Shorty, ALLPATH and EULER-SR, which were further checked by the author, were cited from http://en.wikipedia.org/wiki/Sequence_assembly.

41 Warren, R. L., Sutton, G. G., Jones, S. J. M. & Holt, R. A. Assembling millions of short DNA sequences using SSAKE. Bioinformatics 23, 500–501 (2007).
42 Dohm, J. C., Lottaz, C., Borodina, T. & Himmelbauer, H. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. Genome Res. 17, 1697–1706 (2007).
43 Jeck, W. R., Reinhardt, J. A., Baltrus, D. A., Hickenbotham, M. T., Magrini, V., Mardis, E. R. et al. Extending assembly of short DNA sequences to handle error. Bioinformatics 23, 2942–2944 (2007).
44 Bryant Jr, D. W., Wong, W. K. & Mockler, T. C. QSRA: a quality-value guided de novo short read assembler. BMC Bioinformatics 10, 69 (2009).
45 Miller, J. R., Delcher, A. L., Koren, S., Venter, E., Walenz, B. P., Brownley, A. et al. Aggressive assembly of pyrosequencing reads with mates. Bioinformatics 24, 2818–2824 (2008).
46 Hernandez, D., Francois, P., Farinelli, L., Osteras, M. & Schrenzel, J. De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. Genome Res. 18, 802–809 (2008).
47 Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A. et al. Genome sequencing in microfabricated high-density picolitre reactors. Nature 437, 376–380 (2005).
48 Hossain, M. S., Azimi, N. & Skiena, S. Crystallizing short-read assemblies aroundseeds. BMC Bioinformatics 10(Suppl 1), S16 (2009).
51 Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. M. & Birol, I. ABySS: a parallel assembler for short read sequence data. Genome Res. 19, 1117–1123 (2009).
52 Butler, J., MacCallum, I., Kleber, M., Shlyakhter, I. A., Belmonte, M. K., Lander, E. S. et al. ALLPATHS: de novo assembly of whole-genome shotgun microreads. Genome Res. 18, 810–820 (2008).
53 Chaisson, M. J. & Pevzner, P. A. Short read fragment assembly of bacterial genomes. Genome Res. 18, 324–330 (2008).
54 Li, R. Q., Zhu, H. M., Ruan, J., Qian, W. B., Fang, X. D., Shi, Z. B. et al. De novo assembly of human genomes with massively parallel short read sequencing. Genome Res. 20, 265–272 (2010).
55 Zerbino, D. R. & Birney, E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res. 18, 821–829 (2008).

Lista disponível em Zhang J et al. (2011)

Table 3

The alignment, assembly and utility bioinformatic tools for NGS.

Program	Function	Platform	Website
De novo assembly			
Abyss	Alignment/assembly	Illumina	http://www.bcgsc.ca/platform/bioinfo/software/abyss
ALLPATHS	Alignment/assembly	Illumina	http://www.broadinstitute.org/science/programs/genome-biology/crd
AMOScmp	Alignment/assembly	Roche	http://sourceforge.net/projects/amos/files/
ARACHNE	Alignment/assembly	Roche	http://www.broadinstitute.org/science/programs/genome-biology/crd
CAP3	Alignment/assembly	Roche	http://pbil.univ-lyon1.fr/cap3.php
consensus/Seq-Cons	Alignment/assembly	Roche	http://www.seqan.de/downloads/projects.html
Curtain	Alignment/assembly	Illumina/Roche/ABI	http://code.google.com/p/curtain/
Edena	Alignment/assembly	Illumina	http://www.genomic.ch/edena
Euler-SR	Alignment/assembly	Illumina/Roche	http://euler-assembler.ucsd.edu/portal/?q=team
FuzzyPath	Alignment/assembly	Illumina/Roche	ftp://ftp.sanger.ac.uk/pub/zn1/fuzzypath/fuzzypath_v3.0.tgz
IDBA	Alignment/assembly	Illumina	http://www.cs.hku.hk/~alse/idba/
MIRA/MIRA3	Alignment/assembly	Illumina/Roche	http://chevreux.org/projects_mira.html
Newbler	Alignment/assembly	Roche	roche-applied-science.com/
Phrap	Alignment/assembly	Illumina/Roche	http://www.phrap.org/consed/consed.html#howToGet
RGA	Alignment/assembly	Illumina	http://rga.cgrb.oregonstate.edu/
QSRA	Alignment/assembly	Illumina	http://qsra.cgrb.oregonstate.edu/
SHARCGS	Alignment/assembly	Illumina	http://sharcgs.molgen.mpg.de/
SHORTY	Alignment/assembly	ABI	http://www.cs.sunysb.edu/~skiena/shorty/
SHRAP	Alignment/assembly	Roche	By request
SOAPdenovo	Alignment/assembly	Illumina	http://soap.genomics.org.cn
SOPRA	Alignment/assembly	Illumina/ABI	http://www.physics.rutgers.edu/%7Eanirvans/SOPRA/
SR-ASM	Alignment/assembly	Roche	http://bioserver.cs.put.poznan.pl/sr-asm-short-reads-assembly-algorithm
SSAKE	Alignment/assembly	Illumina/Roche	http://www.bcgsc.ca/platform/bioinfo/software/ssake
Taipan	Alignment/assembly	Illumina	http://sourceforge.net/projects/taipan/files/
VCAKE	Alignment/assembly	Illumina/Roche	http://sourceforge.net/projects/vcake
Velvet	Alignment/assembly	Illumina/Roche/ABI	http://www.ebi.ac.uk/%7Ezerbino/velvet
Reference-based assembly			
BFAST	Alignment/assembly	Illumina/ABI	http://sourceforge.net/apps/mediawiki/bfast/index.php?title=Main_Page
Bowtie	Alignment/assembly	Illumina/Roche/ABI	http://bowtie-bio.sourceforge.net
BWA	Alignment/assembly	Illumina/ABI	http://bio-bwa.sourceforge.net/bwa.shtml
CoronaLite	Alignment/assembly	ABI	http://solidsoftwaretools.com/gf/project/corona/
CABOG	Alignment/assembly	Roche/ABI	http://wgs-assembler.sf.net
ELAND/ELAND2	Alignment/assembly	Illumina/ABI	http://www.illumina.com/
EULER	Alignment/assembly	Illumina	http://euler-assembler.ucsd.edu/portal/
Exonerate	Alignment/assembly	Roche	http://www.ebi.ac.uk/~guy/exonerate
EMBF	Alignment/assembly	Illumina	http://www.biomedcentral.com/1471-2105/10?issue=S1
GenomeMapper	Alignment/assembly	Illumina	http://1001genomes.org/downloads/genomemapper.html
GMAP	Alignment/assembly	Illumina	http://www.gene.com/share/gmap
gnumap	Alignment/assembly	Illumina	http://dna.cs.byu.edu/gnumap/
ICON	Alignment/assembly	Illumina	http://icorn.sourceforge.net/
Karma	Alignment/assembly	Illumina/ABI	http://www.sph.umich.edu/csg/pha/karma/
LAST	Alignment/assembly	Illumina	http://last.cbrc.jp/
LOCAS	Alignment/assembly	Illumina	http://www-ab.informatik.uni-tuebingen.de/software/locas
Mapreads	Alignment/assembly	ABI	http://solidsoftwaretools.com/gf/project/mapreads/
MAQ	Alignment/assembly	Illumina/ABI	http://maq.sourceforge.net

Program	Function	Platform	Website
MOM	Alignment/assembly	Illumina	http://mom.csbc.vcu.edu/
Mosaik	Alignment/assembly	Illumina/Roche/ABI	http://bioinformatics.bc.edu/marthlab/Mosaik
mrFAST/mrsFAST	Alignment/assembly	Illumina	http://mrfast.sourceforge.net/
MUMer	Alignment/assembly	ABI	http://mummer.sourceforge.net/
nexalign	Alignment/assembly	Illumina	http://genome.gsc.riken.jp/osc/english/dataresource/
Novocraft	Alignment/assembly	Illumina	http://www.novocraft.com/
PerM	Alignment/assembly	Illumina/ABI	http://code.google.com/p/perm/
RazerS	Alignment/assembly	Illumina/ABI	http://www.seqan.de/projects/razers.html
RMAP	Alignment/assembly	Illumina	http://rulai.cshl.edu/rmap
segemehl	Alignment/assembly	Illumina/Roche	http://www.bioinf.uni-leipzig.de/Software/segemehl/
SeqCons	Alignment/assembly	Roche	http://www.seqan.de/projects/seqcons.html
SeqMap	Alignment/assembly	Illumina	http://biogibbs.stanford.edu/*jiangh/SeqMap/
SHRiMP	Alignment/assembly	Illumina/Roche/ABI	http://compbio.cs.toronto.edu/shrimp
Slider/SliderII	Alignment/assembly	Illumina	http://www.bcgsc.ca/platform/bioinfo/software/slider
SOCS	Alignment/assembly	ABI	http://solidsoftwaretools.com/gf/project/socs/
SOAP/SOAP2	Alignment/assembly	Illumina/ABI	http://soap.genomics.org.cn
SSAHA/SSAHA2	Alignment/assembly	Illumina/Roche	http://www.sanger.ac.uk/Software/analysis/SSAHA2
Stampy	Alignment/assembly	Illumina	http://www.well.ox.ac.uk/~marting/
SXOligoSearch	Alignment/assembly	Illumina	http://synasite.mgrc.com.my:8080/sxog/NewSXOligoSearch.php
SHORE	Alignment/assembly	Illumina	http://1001genomes.org/downloads/shore.html
Vmatch	Alignment/assembly	Illumina	http://www.vmatch.de/
Diagnostics/utilities			
Artemis/ACT	Visualization tool	Illumina/Roche	http://www.sanger.ac.uk/resources/software/artemis/
CASHX	Pipeline	Illumina	http://seqanswers.com/wiki/CASHX
Consed	Visualization tool	Illumina/Roche	http://www.genome.washington.edu/consed/consed.html
EagleView	Visualization tool	Illumina/Roche	http://bioinformatics.bc.edu/marthlab/EagleView
FastQC	Quality assessment	Illumina/ABI	http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/
Gambit	Visualization tool	Illumina/Roche	http://bioinformatics.bc.edu/marthlab/Gambit
Goby	Data management	Illumina/Roche/ABI	http://campagnelab.org/software/goby/
G-SQZ	Data management	Illumina/ABI	http://public.tgen.org/sqz
Hawkeye	Visualization tool	Illumina/Roche	http://amos.sourceforge.net/hawkeye
Hybrid-SHREC	Error Correction	Illumina/Roche/ABI	http://www.cs.helsinki.fi/u/lmsalmel/hybrid-shrec/
IGV	Visualization tool	Illumina	http://www.broadinstitute.org/igv/?q=home
LookSeq	Visualization tool	Illumina/Roche	http://lookseq.sourceforge.net
MagicViewer	Visualization tool	Illumina	http://bioinformatics.zj.cn/magicviewer/
MapView	Visualization tool	Illumina	http://evolution.sysu.edu.cn/mapview/
NGSView	Visualization tool	Illumina/ABI	http://ngsview.sourceforge.net
PIQA	Quality assessment	Illumina	http://bioinfo.uh.edu/PIQA
Reconciliation	Assembly pipeline	Illumina	http://www.genome.umd.edu/software.htm
RefCov	Sequence coverage	Illumina/Roche	http://genome.wustl.edu/tools/cancer-genomics
SAM Tools	Utilities	Illumina/Roche	http://sourceforge.net/projects/samtools/files/
Savant	Visualization tool	Illumina/Roche	http://compbio.cs.toronto.edu/savant/
ShortRead	Quality assessment	Illumina/Roche	http://bioconductor.org/packages/2.6/bioc/html/ShortRead.html
SHREC	Error Correction	Illumina/Roche	http://www.informatik.uni-kiel.de/jasc/Shrec/
Staden Tools (GAP5)	Pipeline	Illumina/Roche	http://sourceforge.net/projects/staden/files/

Program	Function	Platform	Website
Tablet	Visualization tool	Illumina/Roche	http://bioinf.scri.ac.uk/tablet
TagDust	Data cleaning	Illumina	http://genome.gsc.riken.jp/osc/english/software/.
TileQC	Quality assessment	Illumina	http://www.science.oregonstate.edu/~dolanp/tileqc
XMatchView	Visualization tool	Illumina/Roche	http://www.bcgsc.ca/platform/bioinfo/software/xmatchview
Yenta	Visualization tool	Illumina	http://genome.wustl.edu/tools/cancer-genomics
Geneus	Data management	Illumina/ABI	http://www.genologics.com/solutions/research-informatics/

Table 2: Features of some important tools for analysis of NGS data*

Tool/program	Features	References
De novo alignment		
ABySS	*De novo* sequence assembler designed for aligning very short reads. The single-processor version is useful for assembling genomes up to 40–50 Mb in size.	http://www.bcgsc.ca/platform/bioinfo/software/abyss
EULER-SR	Short read *de novo* assembly, uses a *de Bruijn* graph approach.	http://euler-assembler.ucsd.edu/portal/
MIRA2	MIRA (Mimicking Intelligent Read Assembly) is able to perform true hybrid *de novo* assemblies using reads gathered through 454 sequencing technology (GS20 or GS FLX). Compatible with 454, Illumina and Sanger data. Linux OS required.	http://chevreux.org/projects_mira.html
SSAKE	Short Sequence Assembly by K-mer search and 3′-read Extension (SSAKE) for aggressively assembling millions of short nucleotide sequences by progressively searching for perfect 3′-most k-mers using a DNA prefix tree.	http://www.bcgsc.ca/platform/bioinfo/software/ssake
SOAP*denovo*	Part of the SOAP suite (see below).	http://soap.genomics.org.cn/
VCAKE	*De novo* assembly of short reads with robust error correction. An improvement on early versions of SSAKE.	http://sourceforge.net/projects/vcake/
Velvet	*De novo* genomic assembler specially designed for short read sequencing technologies, such as Illumina or 454. Need ~20–25 × coverage and paired reads.	http://www.ebi.ac.uk/%7Ezerbino/velvet/
Alignment to a reference genome		
Bowtie	Ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of 25 million reads per h on a typical workstation with 2 GB of memory. Uses a Burrows–Wheeler-Transformed (BWT) index.	http://bowtie-bio.sourceforge.net/index.shtml
Exonerate	Offers various forms of pairwise alignment of DNA/protein against a reference.	http://www.ebi.ac.uk/~guy/exonerate/
GenomeMapper	A short read mapping tool designed for accurate read alignments. It quickly aligns millions of reads either with ungapped or gapped alignments.	http://100genomes.org/downloads/genomemapper.html
GMAP	For aligning mRNA and EST Sequences.	http://research-pub.gene.com/gmap/
MAQ	Mapping and Assembly with Qualities (renamed from MAPASS2). Particularly designed for Illumina with preliminary functions to handle ABI SOLiD data.	http://sourceforge.net/projects/maq/
PASS	Allows the users to modulate very finely the sensitivity of the alignments.	http://pass.cribi.unipd.it/cgi-bin/pass.pl
RMAP	Assembles 20–64 bp reads to a FASTA reference genome.	http://rulai.cshl.edu/rmap/
SeqMap	Supports up to 5 or more bp mismatches/INDELs. Highly tuneable.	http://seqmap.compbio.iupui.edu/
SHRiMP	Assembles to a reference sequence.	http://compbio.cs.toronto.edu/shrimp/
Slider	An application for the Illumina Sequence Analyzer output that uses the probability files instead of the sequence files as an input for alignment to a reference sequence or a set of reference sequences.	http://www.bcgsc.ca/platform/bioinfo/software/slider
SOAP	SOAP (Short Oligonucleotide Alignment Program) is a program for efficient gapped and ungapped alignment of short oligonucleotides onto reference sequences. The updated version uses a BWT. Can call SNPs and INDELs.	http://soap.genomics.org.cn/
SSAHA	SSAHA (Sequence Search and Alignment by Hashing Algorithm) is a tool for rapidly finding near exact matches in DNA or protein databases using a hash table. Developed at the Sanger Centre by Zemin Ning, Anthony Cox and James Mullikin. C++ for Linux/Alpha.	http://www.sanger.ac.uk/resources/software/
Vmatch	A versatile software tool for efficiently solving large-scale sequence matching tasks.	http://www.vmatch.de/
Zoom	ZOOM is highly accurate, flexible, and user-friendly with speed being a critical priority. Enables to map millions of short reads, emerged by NGS technology, back to the reference genomes, and carry out post-analysis.	http://www.bioinformaticssolutions.com/all-products/zoom/index.php
SNP/Indel Discovery		
ssahaSNP	A polymorphism detection tool that detects homozygous SNPs and indels by aligning shotgun reads to the finished genome sequence. Highly repetitive elements are filtered out by ignoring those k-mer words with high occurrence numbers. More tuned for ABI Sanger reads.	http://www.sanger.ac.uk/resources/software/
PolyBayesShort	This version is specifically optimized for the analysis of large numbers (millions) of high-throughput next-generation sequencer reads, aligned to whole chromosomes of model organism or mammalian genomes.	http://bioinformatics.bc.edu/marthlab/PbShort

Tool/program	Features	References
PyroBayes	A novel base caller for pyrosequences from the 454 Life Sciences sequencing machines. It was designed to assign more accurate base quality estimates to the 454 pyrosequences.	http://bioinformatics.bc.edu/marthlab/PyroBayes
Alpheus	Pair-wise alignments use BioJava MegaBLAST and Java GMAP parsers; alignments to reference databases; variant detection (SNPs and indels)	http://alpheus.ncgr.org/technical-overview.jsp
Transcriptomics		
G-Mo.R-Se	G-Mo.R-Se is a method aimed at using RNA-Seq short reads to build *de novo* gene models.	http://www.genoscope.cns.fr/externe/gmorse/
MapNext	Useful for (i) unspliced alignment and clustering of reads, (ii) spliced alignment of transcriptomic reads, (iii) SNP detection and calculation of SNP frequency from population sequences and (iv) storage of result data into database to make it available for more flexible query and further analyses.	http://evolution.sysu.edu.cn/english/software/mapnext.htm
QPalma	Optimal Spliced Alignments of Short Sequence Reads. Is an easy-to-use and flexible tool to accurately and efficiently align both transcriptome reads (spliced and unspliced) from RNA-Seq experiments against a reference genome.	http://www.fml.tuebingen.mpg.de/raetsch/suppl/qpalma
TopHat	TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions.	http://tophat.cbcb.umd.edu/
Genome annotation/genome browser/alignment viewer/assembly database		
EagleView	An information-rich genome assembler viewer. It can display a dozen different types of information including base quality and flowgram signal.	http://bioinformatics.bc.edu/marthlab/EagleView
LookSeq	LookSeq is a web-based application for alignment visualization, browsing and analysis of genome sequence data. Supports multiple sequencing technologies, alignment sources, and viewing modes; low or high-depth read pileups; and easy visualization of putative single nucleotide and structural variation.	http://www.sanger.ac.uk/resources/software/
MapView	Enables visualization of short reads alignment on desktop computer.	http://evolution.sysu.edu.cn/mapview/
SAM	Sequence Assembly Manager (SAM) is a whole-genome assembly (WGA) management and visualization Tool. It provides a generic platform for manipulating, analyzing and viewing WGA data, regardless of input type.	http://www.bcgsc.ca/platform/bioinfo/software/sam
XMatchView	A visual tool for analyzing cross match alignments.	http://www.bcgsc.ca/platform/bioinfo/software/xmatchview
Miscelleneous		
CNV-Seq	For detection of copy number variation using high-throughput sequencing.	http://tiger.dbs.nus.edu.sg/cnv-seq/
FindPeaks	Perform analysis of ChIP-Seq experiments.	http://www.bcgsc.ca/platform/bioinfo/software/findpeaks
MACS	Model-based Analysis for ChIP-Seq.	http://liulab.dfci.harvard.edu/MACS/
PeakSeq	PeakSeq is a program for identifying and ranking peak regions in ChIP-Seq experiments.	http://info.gersteinlab.org/PeakSeq
SISSRs	For precise identification of genome-wide transcription factor binding sites from ChIP-Seq data.	http://dir.nhlbi.nih.gov/papers/lmi/epigenomes/sissrs/

*Features of different tools/programs have been compiled from their respective websites.

Anexo B - Principais características e estratégias de funcionamento dos algoritmos dos programas "montadores" típicos

Alinhadores/mapeadores contra um genoma de referência[113]

O alinhamento, por si só, é o processo que determina a origem mais provável, em uma sequência genômica, para uma dada leitura de sequenciamento, tomando-se por base o conhecimento da espécie que gerou a sequência. Leituras de sequenciamento também podem ser alinhadas a outros genomas, desde que a distância evolutiva entre os organismos seja apropriada.

No que diz respeito aos dados provenientes de tecnologias NGS, além da questão das leituras curtas que são produzidas, dois outros fatores devem ser tratados pelos algoritmos de alinhamento: o grande volume de dados gerado, o qual requer um uso otimizado de memória e rapidez, e a diferença de perfis de erros presentes nos dados — por exemplo, para a plataforma 454, tendência quanto à existência de *Indels* provocados pela dificuldade de tratamento das regiões homopoliméricas —, em relação às tecnologias tradicionais de sequenciamento (Bao et al., 2011).

Duas técnicas principais são empregadas em grande parte dos programas alinhadores/mapeadores para NGS: (i) implementações baseadas em tabelas *hash* (feitas a partir da sequência de referência ou do conjunto de leituras) e (ii) métodos baseados em BWT (Burrows Wheeler Transform, ou Transformada Burrows Wheeler), os quais criam um índice eficiente da montagem, facilitando as tarefas de busca, com baixa utilização de memória. Normalmente, programas alinhadores seguem uma estratégia de múltiplos passos, visando mapear a sequência de referência de forma acurada. Em um primeiro momento, esforços são realizados no sentido de rapidamente ser identificado, na sequência de referência, um pequeno conjunto de regiões candidatas para as ocorrências dos melhores mapeamentos. Uma vez que as localizações desse pequeno subconjunto tenham sido identificadas, algoritmos de alinhamento menos velozes, porém mais precisos (como, por exemplo, o Smith-Waterman), são executados nesse subconjunto menor.

[113] Esta seção do Anexo B é um compilado de informações do trabalho de Flicek e Birney (2009). Informações complementares ou corroborativas, presentes em outros trabalhos, também aparecem citadas ao longo do texto.

Métodos baseados em tabelas hash

Tais métodos integram a primeira geração de programas alinhadores para NGS e são baseados em uma estrutura de dados — a tabela *hash* (Figura B.1) — capaz de indexar dados complexos e não-sequenciais, de maneira a agilizar as tarefas de busca. Isso é bastante apropriado para o caso das sequências de leituras de DNA, as quais dificilmente contêm todas as possíveis combinações de nucleotídeos, ao passo que, frequentemente, possuem duplicatas. A tabela *hash*, conforme dito, é feita a partir da sequência de referência ou do conjunto de leituras. No primeiro caso, o algoritmo utiliza o conjunto de leituras para varrer a tabela da sequência de referência, enquanto que, no segundo, a sequência de referência é usada para a varredura da tabela do conjunto de leituras. Independentemente do tipo de tabela *hash*, os algoritmos tipicamente a implementam sob a forma de "sementes espacejadas", as quais são regiões da sequência que devem possuir um padrão específico de similaridade (na forma de igualdades e desigualdades), podendo ser expressas através de uma sequência de caracteres binária — por exemplo, na semente espacejada 111010010100110111, os "1s" englobam as posições para as quais uma igualdade é requerida, enquanto que posições cujos conteúdos são indiferentes são representadas pelos "0s" (Horner et al., 2009). Uma vez que as sementes de alinhamento tenham sido usadas na criação da tabela *hash* e associadas às regiões do genoma candidatas aos melhores alinhamentos, um outro algoritmo, mais especializado e acurado, é usado para determinar a posição correta das leituras de sequenciamento em relação ao genoma de referência.

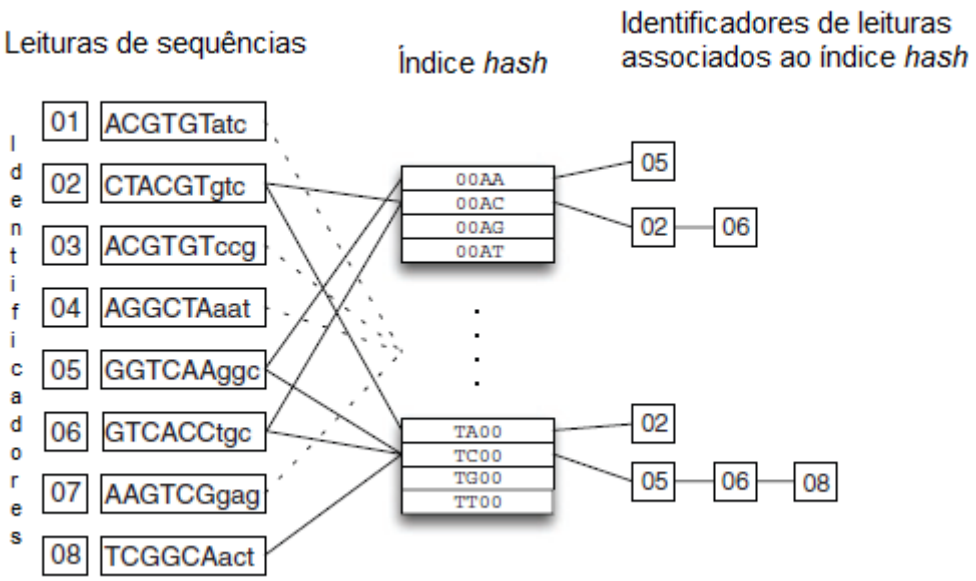


Figura B.1 (Anexo B) - Diagrama esquemático da estratégia de alinhamento baseada em tabela *hash*. As leituras de sequenciamento e seus respectivos identificadores são mostrados, com as regiões que serão usadas para a seleção de sementes em letras maiúsculas, além das sementes de correspondência 0011 e 1100. Os identificadores de leituras são relacionados às sementes através da função *hash* (por exemplo, um número inteiro exclusivo para representar cada semente). Uma vez que a tabela *hash* tenha sido construída para o conjunto de leituras ou o genoma de referência, os dados correspondentes podem ser varridos por meio da mesma função *hash*, resultando em um subconjunto menor a ser alinhado ao genoma.
Fonte: Modificado de Flicek e Birney, 2009, p.S8.

Exemplos de ferramentas que utilizam a abordagem descrita incluem: BFAST, ELAND, MAQ, Mosaik, SeqMap, SHRiMP, SOAP, SSAHA2, ZOOM. Cabe ressaltar que pacotes como ELAND, MAQ, SeqMap, SHRiMP e ZOOM, por exemplo, criam a tabela *hash* a partir do conjunto de leituras, enquanto que BFAST, Mosaik, SOAP e SSAHA2 o fazem a partir do genoma de referência (Bao et al., 2011).

Ainda, como detalhes adicionais a respeito dos principais representantes desta categoria, é válido mencionar que ELAND é um software proprietário, desenvolvido pararelamente à tecnologia de sequenciamento Solexa, o qual, para possibilitar alinhamentos na presença de pequenas desigualdades (permitindo até dois erros por alinhamento (Zhang J et al., 2011)), implementou o conceito de fragmentação das leituras no início do processo (Horner et al., 2009). ELAND, quando comparado a outros programas alinhadores de NGS, no que diz respeito à velocidade, geralmente é considerado um dos mais rápidos disponíveis (Zhang J et al., 2011).

O software MAQ, conforme já adiantado neste trabalho, leva em consideração a qualidade estimada das leituras para o posicionamento destas no alinhamento. Foi um dos

primeiros pacotes para montagem por genoma de referência. Por padrão, seis tabelas *hash* são usadas, permitindo que sequências com duas desigualdades ou menos sejam mapeadas. Também por padrão, MAQ indexa os primeiros 28 pb das leituras. É bastante rápido, versátil, popular e eficiente, embora nem sempre ofereça a garantia de que a melhor correspondência para uma dada leitura tenha sido encontrada. Trabalha tanto com dados de Solexa, quanto de SOLiD™ (Shendure; Ji, 2008; Horner et al., 2009; Zhang J et al., 2011). Foi uma das primeiras ferramentas de alinhamento a oferecer outras funcionalidades para o usuário, como um identificador de SNPs e relatórios estatísticos úteis a respeito do próprio alinhamento (Paszkiewicz; Studholme, 2012).

SeqMap usa como índice o segmento de sequência de caracteres (*substring*) mais longo para o qual é garantida uma correspondência exata. Em seguida, varre o genoma contra esse índice. Também permite alinhamentos com *Indels* (Horner et al., 2009).

SHRiMP inclui uma nova implementação do algoritmo Smith-Waterman, do tipo "espaço de cores para o espaço de bases", compatível com os dados de sequenciamento de codificação em duas bases da plataforma SOLiD™ (Shendure; Ji, 2008). Junto com SOCS, foi uma das primeiras ferramentas de uso livre para mapeamento de dados SOLiD™ contra um genoma de referência (Paszkiewicz; Studholme, 2012).

SOAP permite alinhamentos com ou sem lacunas (*gapped* ou *ungapped alignment*), por meio do mesmo princípio de fragmentação de leituras presente em ELAND. Para acelerar o alinhamento, emprega um algoritmo que faz uso intensivo da memória para verificar a tabela quanto às "sementes", simultaneamente permitindo a poda iterativa das leituras na extremidade 3' (usualmente associada a uma maior presença de erros que impedem o perfeito alinhamento contra o genoma de referência). Nesse cenário, após a poda da leitura, o programa refaz a tentativa de alinhamento até que correspondências sejam encontradas ou que a sequência remanescente fique muito pequena para continuar a ser útil. A principal desvantagem ocasionada pelo recurso é o alto consumo de memória. Ainda, com o intuito de acelerar o processamento, a solução inclui um esquema de "codificação de 2 bits por base" para converter, em valores numéricos, as leituras e o genoma de referência, compactando tais dados em um formato mais eficiente e gerenciável, computacionalmente falando (Shendure; Ji, 2008; Horner et al., 2009).

ZOOM, ferramenta baseada no mesmo princípio de funcionamento do pacote ELAND, difere deste último no fato de que as leituras são indexadas por meio do conceito de "sementes espacejadas" mencionado mais acima.

Métodos baseados em BWT

Tipicamente, utilizam a estrutura de dados conhecida por índice FM (Ferragina-Manzini), na qual, em um primeiro passo, a ordem da sequência do genoma de referência é modificada por meio da Transformada Burrows Wheeler. Esse processo (o qual, a propósito, é facilmente reversível) reordena o genoma de maneira que múltiplas ocorrências de uma mesma sequência apareçam em conjunto na estrutura de dados (Figura B.2). Em um segundo passo, o índice final é criado, sendo este utilizado para o rápido posicionamento das leituras no genoma de referência. Tal como nos métodos baseados em tabela *hash*, uma vez que as leituras tenham sido associadas às regiões do genoma consideradas como as mais prováveis para a ocorrência dos respectivos melhores alinhamentos, outros algoritmos mais sensíveis podem ser empregados para prover o resultado final.

Implementações baseadas em BWT são mais rápidas do que as equivalentes implementadas por meio de tabela *hash*, apesar de ser observada, em contrapartida, uma ligeira perda de sensibilidade no que diz respeito à precisão dos alinhamentos (Flicek; Birney, 2009; Ruffalo et al., 2011).

Exemplos de programas que utilizam essa abordagem são: Bowtie, BWA e SOAP2.

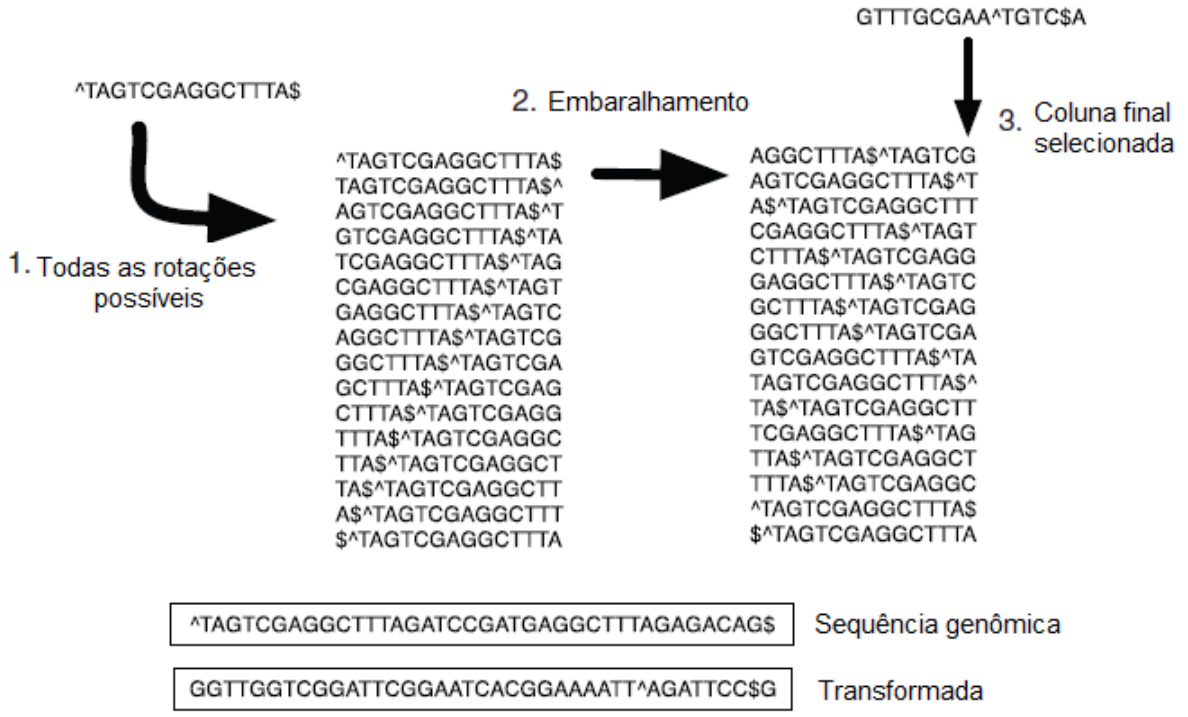


Figura B.2 (Anexo B) - Diagrama esquemático da estratégia de alinhamento baseada em BWT. Para a criação, por exemplo, de uma sequência genômica de 14-mer, são assinalados os pontos inicial e final e construídas todas as rotações possíveis para a dada sequência, colocando-se o primeiro caractere no fim da sequência (etapa 1). Os caracteres ^ e $ marcam, respectivamente, o início e fim da sequência. Uma vez que as sequências tenham sido criadas, elas são embaralhadas (etapa 2). A partir da matriz de sequências embaralhadas, a coluna final é selecionada como a sequência transformada (etapa 3). Tal transformada possui o mesmo tamanho e os mesmos caracteres da sequência original, só que ordenados de maneira diferente. A sequência mais abaixo da figura representa uma longa sequência iniciada pelos mesmos 14-mer, demonstrando o efeito, na sequência transformada, ao ser usada uma sequência original maior.
Fonte: Modificado de Flicek e Birney, 2009, p.S9.

Montadores de novo[114]

O problema da montagem de genomas do tipo *de novo* pode ser encarado, matematicamente, como sendo de difícil solução, pertencendo à classe dos problemas NP-difícil, ou seja, aqueles para os quais nenhuma solução eficiente é conhecida. No entanto, o método é essencial aos esforços para caracterizar a diversidade biológica do planeta, uma vez que abordagens comparativas — tais como as mencionadas anteriormente neste anexo — só podem ser usadas para poucos genomas; aqueles para os quais uma sequência de referência está disponível.

As seguintes técnicas principais são empregadas pela maioria dos programas montadores *de novo*: (i) métodos "gulosos"; (ii) métodos OLC (*Overlap-layout-consensus*) e (iii) Caminho Euleriano.

[114] Esta seção do Anexo B é um compilado de informações do trabalho de Pop (2009). Informações complementares ou corroborativas, presentes em outros trabalhos, também aparecem citadas ao longo do texto.

Especialmente para as estratégias das primeira e segunda categorias, um módulo referido como "computador de sobreposições" leva em consideração todos os alinhamentos pareados existentes entre o conjunto de leituras, sendo um dos componentes mais computacionalmente intensivos. O desempenho desse componente é melhorado por meio de estratégias de indexação, as quais, por exemplo, podem ter seus índices baseados no número de igualdades em uma dada extensão *k* (de *k-mers*). Deste modo, somente leituras que compartilham de um mesmo *k-mer* necessitam ser comparadas na etapa de avaliação das sobreposições. Ainda assim, no caso de regiões repetitivas do genoma, um comportamento quadrático pode vir a ser observado, já que todas as leituras compartilhando de um mesmo *k-mer* terão de ser comparadas, cada uma com todas as demais. Também é importante observar que o uso de leituras provenientes de tecnologias NGS, devido ao grande volume, ocasiona um impacto significativo à etapa de computação de sobreposições. Mais leituras, afinal, podem impor uma complexidade exponencial para os algoritmos responsáveis (Bao et al., 2011). Por isso, a etapa, usualmente, pode ser paralelizada, o que diminui a sua duração em máquinas dotadas de múltiplos processadores (ou núcleos) ou quando executada em *grids* computacionais (quando estes se encontram disponíveis).

Métodos "gulosos"

Algoritmos "gulosos" representam a solução mais simples e intuitiva para o problema de montagem. Nesta abordagem, leituras individuais são reunidas, de maneira iterativa, para formarem *contigs*. O processo se inicia pelas leituras que melhor se sobrepõem — o que ocorre quando o prefixo de uma compartilha de significativa similaridade com o sufixo de outra (Figura B.3(A)) — e termina assim que nenhuma leitura ou *contig* possa mais ser agregado. O critério que determina a melhor sobreposição depende da implementação e, normalmente, envolve fatores como a extensão da sobreposição e o nível de identidade entre as leituras. Já o termo "guloso" se refere ao fato de que o algoritmo toma decisões no sentido de otimizar uma função objetivo local — no caso, a qualidade de sobreposição entre duas leituras — em uma abordagem que pode não atingir a solução ótima global (por exemplo, por sempre processar primeiro a melhor sobreposição, um montador "guloso" pode montar incorretamente regiões repetitivas (Figuras B.3(B) e (C))). Esse tipo de abordagem foi usada por vários montadores de genoma clássicos, para dados provenientes da tecnologia de sequenciamento Sanger, como Phrap (www.phrap.org; de la Bastide; McCombie, 2007), TIGR Assembler (Sutton et al., 1995) e CAP3. No caso da montagem de leituras de NGS, uma estratégia diferente de algoritmo "guloso" tem sido utilizada. Nela, uma leitura ainda não-montada é escolhida para iniciar um *contig*, o qual é repetidamente estendido por leituras

que o sobreponham em sua extremidade 3', até que nenhuma outra extensão seja possível. Em seguida, o processo é repetido na direção 5', usando-se o complemento reverso da sequência do *contig*. A montagem continua de maneira iterativa, a partir da varredura do conjunto de leituras não-montadas, sendo que as leituras são consideradas em ordem decrescente de qualidade, critério este que pode ser representado pela profundidade de sequenciamento ou por uma combinação de fatores como valores de qualidade, profundidade de sequenciamento e presença de, ao menos, uma sobreposição perfeita em relação a uma outra leitura. Uma vez que informações conflitantes (ambíguas) sejam encontradas — por exemplo, duas ou mais leituras que poderiam estender um *contig*, mas que não se sobrepõem (Figura B.3(D)) —, o processo é interrompido para evitar erros de montagem. De fato, erros de montagem causados por elementos repetitivos são evitados com o procedimento, porém menores *contigs* são obtidos (Pop; Salzberg, 2008).

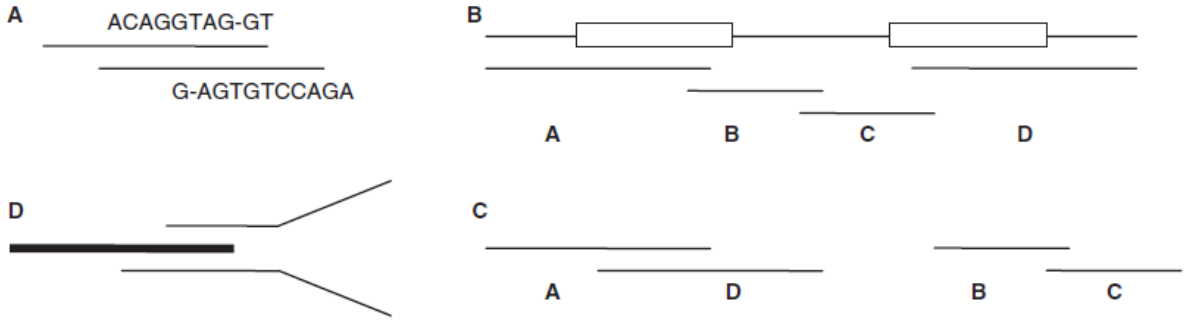


Figura B.3 (Anexo B) - Método "guloso". (A) Sobreposição entre duas leituras — pode ser observado que a similaridade da região de sobreposição não necessita ser perfeita. (B) Montagem correta de um genoma com duas regiões repetitivas (retângulos), a partir das quatro leituras A - D. (C) Montagem produzida pela abordagem "gulosa". Neste caso, as leituras A e D são montadas primeiramente (e incorretamente) porque a sobreposição entre elas é melhor e (D) discordância entre duas leituras (linhas finas) que poderiam estender um dado *contig* (linha grossa), indicando uma potencial fronteira de região repetitiva. O processo de extensão do *contig* é interrompido para evitar erros de montagem.
Fonte: Modificado de Pop, 2009, p.357.

Os pacotes QSRA, SHARCGS, SSAKE e VCAKE, introduzidos para o tratamento de leituras provenientes das tecnologias NGS, são mais exemplos dentre os que utilizam a abordagem aqui descrita (Bao et al., 2011).

SSAKE foi inicialmente elaborado para tratar de leituras curtas, não-pareadas, de tamanho uniforme, baseado na ideia de que uma alta cobertura poderia fornecer um encadeamento de leituras sem erros, caso leituras com erros pudessem ser evitadas no processo. O programa não utiliza um grafo, explicitamente, mas sim uma tabela com as leituras indexadas por seus prefixos, a qual é usada para, de maneira iterativa, procurar as sequências que se sobrepõem perfeitamente à extremidade de um *contig*, respeitando-se um limiar de tamanho pré-definido. Assim, SSAKE escolhe cuidadosamente, dentre múltiplas

leituras com iguais extensões de sobreposição, primeiramente dando preferência às leituras que possuam sobreposição perfeita, de uma extremidade à outra, em outras leituras (o que favorece as leituras sem erros). Em seguida, o pacote detecta quando o conjunto de leituras candidatas apresenta múltiplas extensões, ou seja, particularmente, quando os sufixos de leituras candidatas apresentam diferenças que também são confirmadas em outras leituras (o que equivaleria a encontrar uma ramificação em um grafo). Neste ponto, a extensão do *contig* é interrompida. O pacote pode ser ajustado pelo usuário para assumir um comportamento menos rigoroso. Assim, quando as leituras não satisfazem a um determinado limiar inicial, o programa decrementa esse limiar até que um segundo seja alcançado. Com isso, é possível definir a "agressividade" com a qual o programa faz a extensão através de possíveis regiões de fronteira de elementos repetitivos ou de baixa cobertura. O programa recebeu melhorias para ser capaz de lidar com leituras pareadas e permitir o alinhamento de leituras com pequenas desigualdades entre si (Miller et al., 2010).

SHARCGS também foi elaborado para operar com leituras curtas, não-pareadas, de alta cobertura e tamanho uniforme. O programa agrega fases de pré e pós-processamento ao algoritmo SSAKE básico . O pré-processamento filtra leituras com erros, requerendo um número mínimo de tamanho de sobreposição exata em outras leituras. Um filtro opcional, ainda mais estrito, pode ser aplicado no sentido de manter leituras que se sobreponham e que atinjam um determinado limiar de valor de qualidade. SHARCGS filtra o conjunto bruto de leituras três vezes, usando diferentes parâmetros de filtragem, a cada rodada, para gerar três conjuntos de leituras diferentes. Cada conjunto é montado separadamente por meio do processo iterativo de extensão de *contigs*. Em seguida, no pós-processamento, os três conjuntos de *contigs* obtidos são mesclados através de um algoritmo de alinhamento de sequências, visando aumentar ainda mais a confiabilidade da montagem (Miller et al., 2010; Sasson, 2010).

VCAKE já foi desenvolvido com a capacidade de incorporar correspondências imperfeitas durante o processo de extensão de *contigs* (Miller et al., 2010). Na ausência de um número suficiente de leituras com sobreposições perfeitas, por exemplo, o algoritmo passa a procurar por leituras que contenham uma única desigualdade após a décima posição de base (Sasson, 2010).

QSRA, desenvolvido posteriormente aos demais desta categoria, possui como vantagens a utilização da informação referente aos valores de qualidade das leituras para a

execução do alinhamento e um melhor desempenho em termos de velocidade e qualidade do resultado final (Bao et al., 2011).

Métodos OLC

Nesta estratégia, a montagem é dividida em três estágios. O primeiro é similar à abordagem "gulosa", ou seja, as leituras são comparadas entre si para construir a lista de sobreposições em pares. Tal informação é usada para a construção de um grafo de "sobreposição" (*Overlap*), no qual cada leitura é representada por um nó e uma aresta conectando dois nós representa a ocorrência de sobreposição entre duas leituras correspondentes. A abordagem OLC é um paradigma que trata cada leitura como uma unidade discreta durante a reconstrução de um genoma (Pop; Salzberg, 2008).

No estágio seguinte, o de "arranjo" (*Layout*), o grafo anterior é analisado de maneira a identificar os caminhos que correspondam aos segmentos do genoma que está sendo montado, visando a obtenção de um único percurso que passe somente uma vez por cada nó, usando-se todas as leituras de sequenciamento que tenham sido usadas para alimentar o algoritmo. Tipicamente, uma abordagem hierárquica é usada nesta tarefa, onde, primeiramente, são identificados os segmentos do grafo que podem ser montados sem ambiguidades (representando, portanto, os segmentos de DNA que definitivamente fazem parte do genoma; sendo estes, geralmente, segmentos não-repetitivos ou aqueles completamente contidos em uma única cópia de uma região repetitiva). Bifurcações no grafo, representando ambiguidades — provocadas, por exemplo, por regiões de fronteira entre um elemento repetitivo e regiões genômicas adjacentes às cópias desse elemento ao longo do genoma ou por erros de sequenciamento — são resolvidas por diferentes heurísticas empregadas pelas distintas implementações práticas da estratégia OLC.

A identificação do caminho — também conhecido como "Caminho Hamiltoniano" — na etapa de *Layout* é considerada, também, como um problema NP-difícil. Entretanto, a formulação do grafo possibilita diversas alternativas de análise, as quais não seriam possíveis de serem obtidas pelas abordagens "gulosas" (Figura B.4).

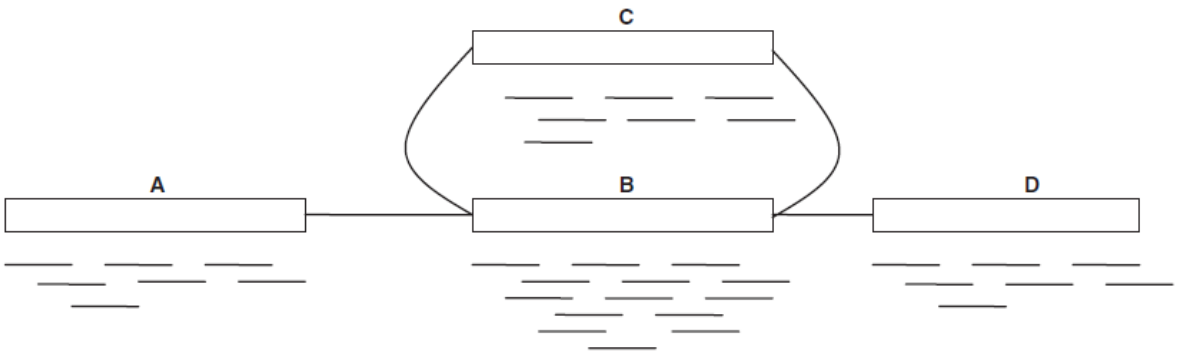


Figura B.4 (Anexo B) - Grafo de sobreposição de um genoma contendo duas cópias de um elemento repetitivo (segmento B) separadas pelo segmento C. A reconstrução correta do genoma é representada pela sequência ABCBD. Estratégias "gulosas" poderiam tanto construir uma montagem fragmentada, interrompida na fronteira do elemento repetitivo B (uma reconstrução possível, por exemplo, seria AB, C, D); quanto omitir a região repetitiva, consequentemente culminando em um erro de montagem (neste caso, por exemplo, a reconstrução poderia ser ABD, C, o que representaria um segmento de DNA inexistente no genoma).
Fonte: Modificado de Pop, 2009, p.358.

O estágio final da estratégia é a obtenção do "consenso" (*Consensus*), ou seja, a determinação da sequência de DNA sugerida pelo arranjo das leituras ao longo do caminho escolhido através do grafo, levando-se também em consideração, para a escolha da leitura mais apropriada, os valores de qualidade das sequências.

Representantes deste tipo de abordagem são, por exemplo, os montadores ARACHNE (Batzoglou et al., 2002; Jaffe et al., 2003), CABOG, Celera Assembler (Myers et al., 2000), Edena, Minimus (Sommer et al., 2007), Newbler, PCAP (Huang; Yang, 2005) e Shorty (Hossain et al., 2009).

A título de informação adicional sobre alguns desses pacotes, cabe comentar que Newbler foi especificamente desenvolvido para lidar com as ambiguidades de tamanho dos trechos homopoliméricos das rodadas de sequenciamento 454 (Bao et al., 2011). Sua primeira versão visava leituras não-pareadas de 100 pb de tamanho, mas, depois, o programa foi revisado para construir *scaffolds* a partir de segmentos pareados. O algoritmo implementa duas vezes a abordagem OLC: na primeira fase, usa as leituras para obter *unitigs* — "mini-montagens", as quais, idealmente, não encontram sobreposições em leituras presentes em outros *unitigs*, e que atuam como *contigs* conservados, preliminares e de alta confiabilidade, servindo como sementes para o restante do processo de montagem. Na segunda fase de OLC, *contigs* maiores são gerados a partir dos *unitigs*. Newbler utiliza a informação de cobertura, quando possível, para corrigir erros de atribuição de bases, além de computar o consenso de *unitigs* e *contigs*, por meio do uso das informações de intensidade de sinal relacionadas a cada ciclo de fluxo, executado na plataforma, para um dado nucleotídeo particular.

CABOG é a revisão do *pipeline* Celera Assembler; este, por sua vez, um algoritmo de abordagem OLC da era Sanger, o qual foi revisado para trabalhar com dados de 454. CABOG descobre sobreposições utilizando sementes compactadas. Ele converte trechos homopoliméricos em bases únicas, de maneira a transpor o problema de incerteza de tamanho relacionado a esses tipos de regiões. O programa monta os primeiros *unitigs* a partir da exclusão de leituras que sejam subsegmentos de outras leituras, já que tais tipos de leituras são mais suscetíveis a falsas sobreposições induzidas por elementos repetitivos. Possui o mesmo esquema de correção de atribuição de bases do software ARACHNE, o qual compara cada leitura com o seu respectivo conjunto de leituras para as quais foram encontradas sobreposições. Qualquer base conflitante com uma preponderância de sobreposições é inferida como sendo um erro de sequenciamento. Não há uma correção da leitura pelo software, mas sim o ajuste das taxas de erro das sobreposições que transpassem o erro inferido. Em seguida, é aplicado um limiar (definido pelo usuário) para as taxas de erro e, das sobreposições remanescentes à execução do filtro de erros e do filtro de mínima extensão de alinhamento, CABOG seleciona a "melhor" sobreposição (aquela que alinha mais bases) por extremidade de leitura. O software constrói um grafo de sobreposição a partir das leituras e "melhores" sobreposições e, nele, monta *unitigs* de máximos percursos simples que sejam livres de ramificações e interseções. Um grafo de *unitigs* também agregando restrições advindas das informações associadas ao pareamento de leituras, é, em seguida, construído. Com ele, CABOG monta *contigs*, a partir de *unitigs*, e os conecta em *scaffolds*. O grafo passa por um processo de refinamento e, finalmente, o programa deriva sequências-consenso através da computação de múltiplos alinhamentos de sequências a partir do conjunto de *scaffolds* mais as leituras (Miller et al., 2010).

Edena foi elaborado para lidar com leituras não-pareadas de tamanho uniforme. A solução não possui um estágio específico de correção de erros antes da construção do grafo. Durante o processo, leituras duplicadas são descartadas e as sobreposições exatas e livres de erros são encontradas. Na abordagem OLC de Edena, o grafo de sequências gerado apresenta os nós como representantes das leituras e as arestas surgem quando o critério de sobreposição é atingido. Durante a montagem, o programa executa diversas fases de correção de erros até a obtenção do grafo final (Miller et al., 2010; Sasson, 2010).

O software Shorty procura lidar com o caso especial em que algumas poucas leituras longas estão disponíveis para atuarem como sementes no recrutamento de leituras curtas e suas contrapartes *mate-pairs*. Por meio de iterações, Shorty faz uso de *contigs* para semear *contigs* ainda mais longos (Miller et al., 2010). De forma inovadora, o pacote estima as

distâncias entre *contigs* a partir das informações de *mate-pairs*, usando algumas poucas sementes de 300 - 500 pb de extensão (Bao et al., 2011). Shorty foi desenvolvido para a montagem de leituras de dados ABI SOLiD™ em "espaço de cores" (Paszkiewicz; Studholme, 2010).

Caminho Euleriano

No início desta estratégia, as leituras são fragmentadas em tamanhos equivalentes ao seu espectro de *k-mer* (Figura B.5(A)), o qual, por sua vez, equivale a uma lista de todos os oligômeros de uma determinada extensão *k* presentes no genoma sendo sequenciado. De fato, um espectro de *k-mer* pode ser encarado como a saída de um experimento de sequenciamento *shotgun* perfeito, no qual todas as leituras possuem o mesmo tamanho *k* e fazem uma amostragem precisa do genoma, sendo o início de cada uma correspondente a cada base presente na sequência. Os dados extraídos das leituras também trazem informações adicionais — por exemplo, a abundância (cobertura) de cada *k-mer*, informação esta que pode ser usada para resolver situações que envolvam elementos repetitivos. O espectro de *k-mer* resultante é então usado para construir um grafo do tipo de Bruijn — um grafo cujos nós representam prefixos e sufixos dos *k-mers* originais, de tamanho *k* - 1, sendo dois nós interligados por uma aresta quando os *k* - 1-*mers* possuem uma sobreposição exata de tamanho *k* - 2 (Figura B.5(B)). Na prática, cada *k-mer* identificado a partir do conjunto de leituras é convertido em uma aresta do grafo de Bruijn e o problema de montagem passa a ser encontrar o caminho que passa por todas as arestas do grafo (considerando que cada *k-mer* está presente no genoma, então sua correspondente aresta deve ser incluída na reconstrução). Tal caminho é conhecido como Caminho Euleriano, o qual dá nome à estratégia.

Na abordagem de Caminho Euleriano, uma vez que, no grafo de Bruijn, as sobreposições estão representadas de maneira implícita por caminhos que interligam duas leituras (Figura B.5(C)), a etapa de computação de sobreposições (existente, por exemplo, na estratégia OLC) não se faz necessária, o que evita a execução de um passo altamente custoso, computacionalmente falando, por usualmente aplicar uma implementação do tipo "todas-contra-todas" na confecção do grafo de sobreposição (Flicek; Birney, 2009). Isso, portanto, pode ser considerada uma vantagem intuitiva desta estratégia em relação à OLC. Além disso, existem diversos algoritmos eficientes para a solução do Caminho Euleriano, ao passo que encontrar o Caminho Hamiltoniano da estratégia OLC é uma tarefa bastante difícil. Entretanto, cabe também observar que um número exponencial de caminhos Eulerianos distintos pode ser encontrado em um grafo, correspondendo às diferentes maneiras que um

genoma pode ser rearranjado em torno de seus elementos repetitivos. Já que a tarefa básica de um montador é encontrar apenas um de todos os possíveis caminhos, equivalendo à correta reconstrução do genoma, a agregação de restrições adicionais à abordagem do Caminho Euleriano, para guiar o algoritmo na determinação desse caminho único correto, também culmina em um problema computacional complexo.

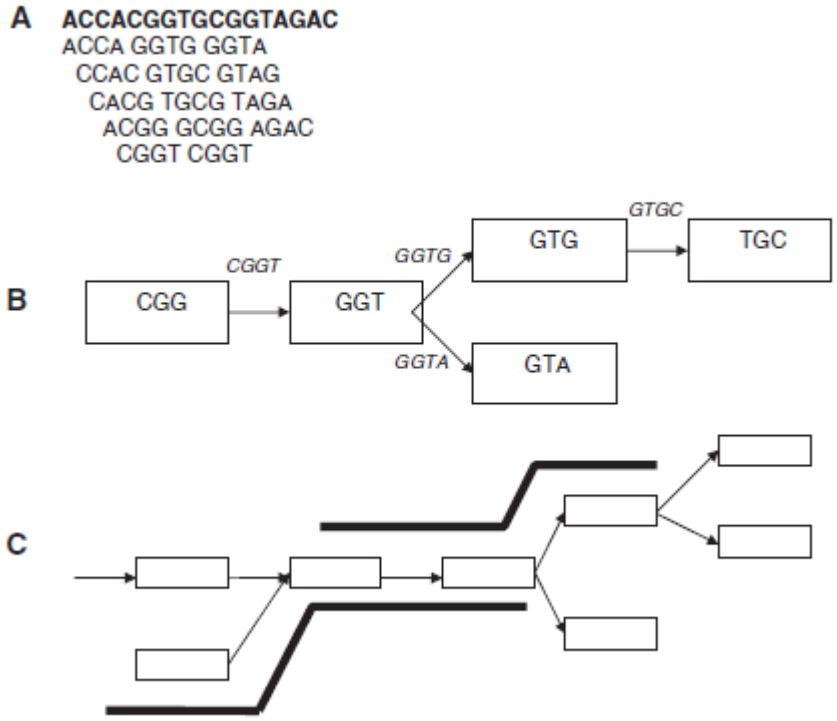


Figura B.5 (Anexo B) - Abordagem por grafo de Bruijn. (A) Espectro de *k-mer* de uma sequência de DNA (em negrito) para um valor de $k = 4$. (B) Seção do grafo de Bruijn correspondente. As arestas aparecem rotuladas pelos respectivos *k-mers*. (C) Sobreposição entre duas leituras (em negrito) que pode ser inferida a partir dos percursos correspondentes através do grafo de Bruijn.
Fonte: Modificado de Pop, 2009, p.359.

Apesar da estratégia ter sido proposta como uma alternativa à OLC para a montagem de dados provenientes da tecnologia Sanger, principalmente pelos montadores da série Euler (Pevzner et al., 2001; Pevzner; Tang, 2001; Chaisson et al., 2004; Chaisson; Pevzner, 2007; Chaisson et al., 2009) ela não foi muito adotada nesse cenário, em parte devido à sua sensibilidade a erros de sequenciamento, os quais provocam a criação de "novos" *k-mers*, o que, por conseguinte, aumenta drasticamente o tamanho e complexidade do grafo de Bruijn resultante. Com isso, sofisticados algoritmos de correção de erro são necessários. No entanto, características presentes nas novas tecnologias de sequenciamento de leituras curtas, como alta cobertura de sequenciamento e padronização do tamanho dos produtos gerados, apresentam um cenário bem mais viável ao uso da abordagem de Caminho Euleriano. Através da fragmentação das leituras originais em segmentos menores, este paradigma é menos

afetado pela questão do menor tamanho das leituras geradas pelas plataformas NGS, ao mesmo tempo em que provê um mecanismo simples para combinar leituras de diferentes tamanhos em cenários de montagens híbridas (Pop; Salzberg, 2008).

Os seguintes pacotes, portanto, são alguns representantes dos que utilizam tal abordagem de montagem: ABySS, ALLPATHS/ALLPATHS2, Euler-SR, Meraculous, SOAPdenovo e Velvet.

Velvet, Euler-SR e ABySS, por exemplo, usam grafos de Bruijn de maneira explícita. ALLPATHS, no entanto, explora um grafo de Bruijn restrito ao comportamento das leituras pareadas. Os métodos diferem quanto ao tratamento de erros e grau de utilização da informação das leituras pareadas, porém contemplando, geralmente, os mesmos estágios: organização de *k-mers*/leituras, criação do grafo, simplificação do grafo, correção de erros e, finalmente, a montagem resultante (Figuras B.6 e B.7). Os montadores mais avançados deste grupo (Velvet, Euler-SR, ABySS e ALLPATHS) podem usar leituras pareadas no intuito de fornecerem montagens longas (Flicek; Birney, 2009; Sasson, 2010).

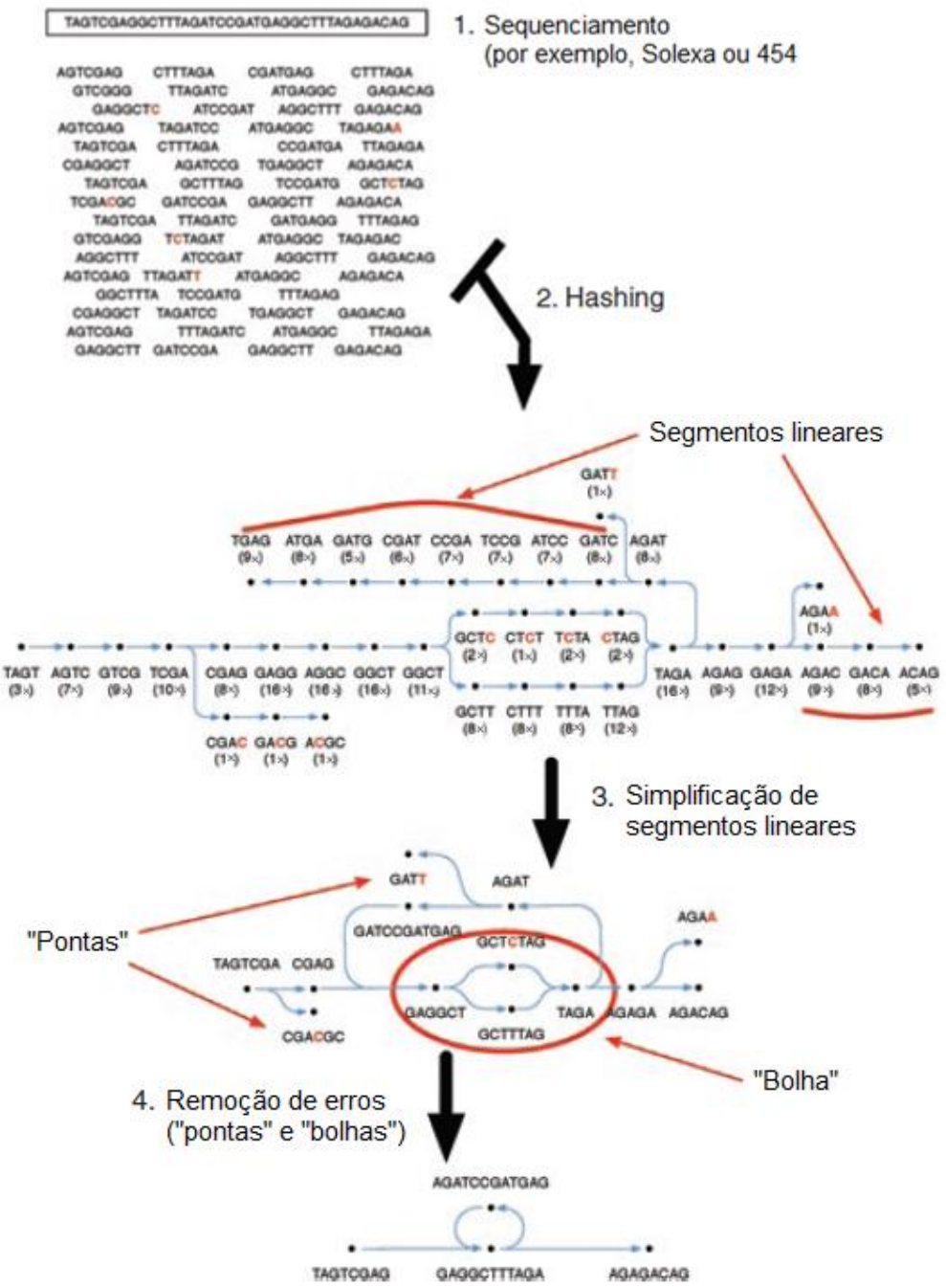


Figura B.6 (Anexo B) - Exemplo de grafo de Bruijn para uma sequência genômica curta, sem polimorfismo. A sequência mostrada no topo da figura representa o genoma, o qual é fragmentado através do sequenciamento *shotgun* em leituras de 7 pb (passo 1). Algumas leituras contêm erros (bases na cor vermelha). No passo 2, os *k-mers* presentes nas leituras (4-*mers* no exemplo) são organizados em nós e a cobertura de cada nó é registrada. Existem segmentos lineares no grafo, ao passo que erros de sequenciamento criam pontos característicos de baixa cobertura. No passo 3, o grafo é simplificado de maneira a combinar nós que estão associados a trechos lineares contínuos em nós únicos e maiores, com os mais variados tamanhos de *k-mer*. Um processo de correção de erros para remover "pontas" e "bolhas", ocasionados por erros de sequenciamento, ocorre no passo 4, criando uma estrutura de grafo final que descreve completa e acuradamente a sequência genômica original.
Fonte: Modificado de Flicek e Birney, 2009, p.S10.

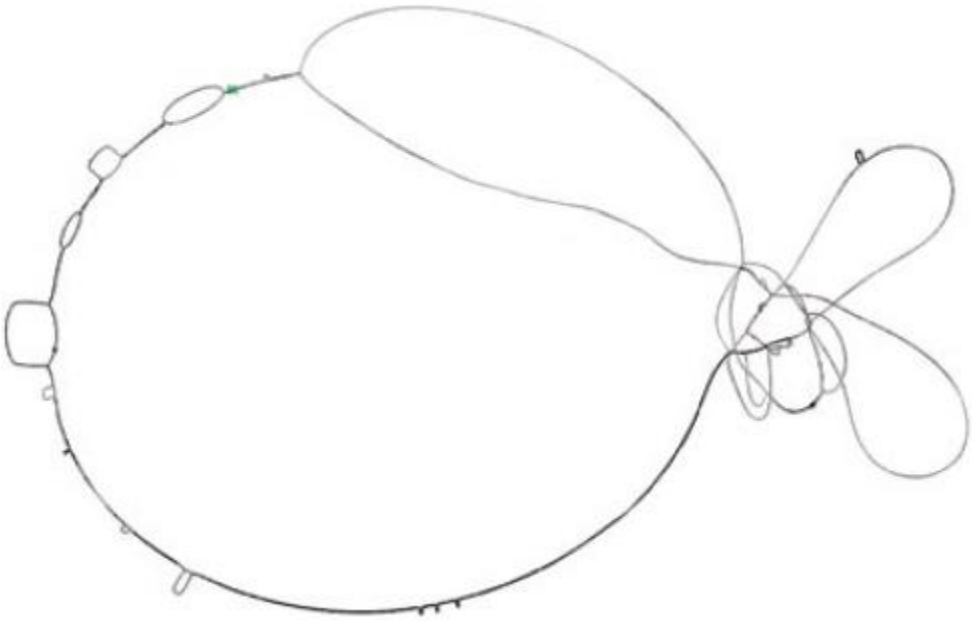


Figura B.7 (Anexo B) - Um grafo de Bruijn completo de um genoma bacteriano. É possível observar a baixa ocorrência de estruturas repetitivas ao longo de todo o genoma.
Fonte: Modificado de Flicek e Birney, 2009, p.S10.

A série de soluções Euler foi iniciada com o desenvolvimento de uma versão para o tratamento de leituras do tipo Sanger. O pacote foi subsequentemente modificado para lidar com leituras de 454, depois com leituras não-pareadas de Solexa/Illumina e, por fim, com leituras pareadas dessa plataforma. Para a correção de erros, antes da construção do grafo, as leituras passam por um processo de filtro. Euler monta, simplifica e compara grafos de Bruijn, a partir de diferentes tamanhos de *k-mers*. Aplica, também, heurísticas para mitigar a complexidade do grafo, provocada por erros de sequenciamento, e explora as informações de extremidades de leituras com baixa qualidade e de pareamento de leituras para corrigi-lo quanto a erros induzidos por elementos repetitivos (Miller et al., 2010).

Velvet é uma implementação completa de montador baseado em grafo de Bruijn, confiável e fácil de ser utilizada. A solução não utiliza um estágio de pré-processamento de correção de erros, embora possua um filtro de leituras para preveni-los. Fazendo uso da topologia do grafo, Velvet tenta encontrar leituras com erros — por exemplo, erros na extremidade da leitura, comuns nas leituras provenientes da tecnologia Illumina, correspondem a pequenas cadeias de *k-mers* que somente se ligam ao restante do grafo como "pontas" soltas, ao passo que erros no meio das leituras proporcionam dois caminhos iniciados e terminados em regiões próximas de sequências similares (Henson et al., 2012) — Velvet também assume uma distribuição de frequência bimodal para a profundidade de cobertura dos *contigs* — picos em baixa cobertura são assumidos como portadores de erros e, com isso, são retirados da montagem. O usuário pode determinar um parâmetro denominado

"coverage cut-off" para especificar o rigor quanto à purga dos *contigs* com baixa cobertura. Além disso, aplica uma outra série de heurísticas no sentido de reduzir a complexidade do grafo obtido. Outros fatores como identidade das sequências e restrições advindas do pareamento de leituras também são levados em consideração para tal. O pacote é recomendado para a montagem *de novo* de leituras curtas pareadas da plataforma Solexa/Illumina. Os requerimentos de memória o tornam menos recomendado para a montagem de genomas grandes e complexos (Miller et al., 2010).

ABySS é uma implementação distribuída, escalável, elaborada para lidar com as limitações de memória impostas pela montagem de genomas grandes e complexos (tais como os de mamíferos) às abordagens de grafos de Bruijn. ABySS distribui o grafo de *k-mer* e suas respectivas computações em um *grid* computacional cuja memória combinada é ampla. Apresenta, como alvo, leituras curtas e pareadas de Solexa/Illumina (Miller et al., 2010; Bao et al., 2011). Em sua etapa de *scaffolding*, usa um número mínimo de cinco pares de leituras como critério padrão para a determinação da ordem e distância entre *contigs* (Paszkiewicz; Studholme, 2010).

ALLPATHS também foi desenvolvido como uma aplicação para genomas grandes, visando leituras curtas e pareadas de Solexa/Illumina. Usa um estágio pré-processador de correção de erros similar ao de Euler, fazendo uso de valores de qualidade para corrigir erros de substituições de bases. Em seguida, um outro componente de pré-processamento, o qual é responsável por gerar "*unipaths*", calcula sobreposições exatas de leituras semeadas por *k-mers*. ALLPATHS procura determinar todos os caminhos de uma leitura a outra coberta por outras leituras. Em seguida, tenta isolar pequenas partes do genoma para montar tais segmentos de forma independente. O algoritmo monta, primeiramente, as sequências locais e, então, agrega as montagens locais à montagem principal, retornando, não apenas uma montagem única, mas todas as possíveis montagens, incluindo ambiguidades (Miller et al., 2010; Sasson, 2010; Bao et al., 2011).

SOAPdenovo combina as abordagens OLC e de grafo de Bruijn em um mesmo pacote, construindo um grafo de *contigs* por meio do método de grafo de Bruijn. SOAPdenovo filtra e corrige leituras usando limiares pré-definidos para frequências de *k-mers*. Constrói o grafo de Bruijn e reduz suas "pontas". Remove "bolhas" através de um algoritmo similar ao de Velvet, com a maior cobertura de leituras determinando o caminho "sobrevivente". Embora sua implementação de grafo de Bruijn seja baseada nas equivalentes de Euler e Velvet, seu grafo ocupa menos espaço de memória. Além disso, restringe o

tamanho de *k-mer* a números ímpares de 13 a 31. Embora *k-mers* mais longos concorram para uma maior confiança, por causa de sua maior taxa de singularidade, o que, consequentemente, possibilitaria a construção de um grafo mais simples, SOAPdenovo restringe esse tamanho, já que maiores coberturas de sequenciamento e tamanhos de leitura seriam necessários para uma montagem bem sucedida. Por fim, SOAPdenovo monta seus *scaffolds* considerando o menor tamanho de inserto, em primeiro lugar, e usa um número mínimo de três pares de leituras como um critério para a determinação da ordem e da distância entre *contigs* (Miller et al., 2010; Paszkiewicz; Studholme, 2010; Sasson, 2010; Bao et al., 2011).

Conforme menção anterior, um fator limitante à montagem de genomas grandes e complexos é o alto consumo de memória dos métodos de grafos de Bruijn. Embora esta seja uma estrutura de dados compacta, todas as implementações fazem uso de algum tipo de estrutura de dados auxiliar ao núcleo do grafo de Bruijn, com o objetivo de mapear as leituras ao grafo. Com isso, muitas implementações que funcionam bem em menor escala (genoma menor do que 50 Megabases), chegam a necessitar de 2 Terabytes de memória real para a montagem de um genoma complexo. Soluções como ABySS e SOAPdenovo, no entanto, conseguiram minimizar o problema. ABySS faz uso de uma estratégia de paralelização MPI-*cluster*, ao passo que SOAPdenovo emprega múltiplas passagens por estruturas de dados compactas, as quais podem ser mantidas em disco para o tratamento de grandes volumes (Flicek; Birney, 2009).

Scaffolding

Nenhuma das estratégias de montagem citadas é capaz de reconstruir um genoma por completo somente a partir das leituras de sequenciamento. Tipicamente, o resultado entregue por muitos montadores consiste de um conjunto, muitas vezes grande, de *contigs* independentes. Um processo denominado *scaffolding* usa outras fontes de informação para o posicionamento relativo desses *contigs* em relação ao genoma. O resultado desse processo são os *scaffolds* — grupos de *contigs* cujas posições relativas são conhecidas, embora não se tenha informação a respeito das sequências de DNA das regiões genômicas que conectam *contigs* adjacentes. Tipicamente, essa estratégia usa as informações de leituras pareadas — dois *contigs* podem ser inferidos como sendo adjacentes, se um componente do par de leituras for montado no primeiro *contig* e o outro membro do par for montado no segundo *contig*. A maioria dos montadores modernos — Euler, ARACHNE, Celera Assembler, Velvet e ALLPATHS, por exemplo —, independentemente da abordagem de montagem utilizada, possui um módulo de *scaffolding*. Tal módulo, tipicamente, segue uma abordagem do tipo

"gulosa", iniciando com a informação mais confiável e incorporando mais dados, de forma iterativa, até que algum tipo de conflito de informação seja encontrado. No pacote Celera Assembler, por exemplo, *contigs* únicos, já bem conectados ao restante da montagem (chamados de *rocks*; algo como "rochas", em português), são processados primeiro. Em seguida, são processados *contigs* que não somente estejam conectados por uma leitura pareada, mas que também sobreponham outros adjacentes (chamados de *stones*; algo como "pedras", em português). Por fim, são levados em consideração os *contigs* soltos que possam ser usados para compor o percurso entre as lacunas (*gaps*) da montagem (chamados de *peebles*; algo como "lascas", em português). Como outro exemplo de implementação, Velvet inicia o processo com *contigs* maiores do que o tamanho da distância entre leituras pareadas, usando o grafo de Bruijn e as leituras pareadas para "caminhar" pelo grafo entre os *contigs*. Já no caso de Euler, este mapeia pares de leituras contra o grafo de Bruijn e, onde um par de leituras corresponder a dois componentes até então desconectados, o algoritmo encontra um caminho entre essas leituras. O tamanho do percurso deve ser aproximadamente igual à distância entre as leituras; critério este que é usado para identificar dados de pareamento incorretos e para fazer a escolha entre caminhos alternativos (Paszkiewicz; Studholme, 2010).

Além dos módulos de *scaffolding* presentes em alguns montadores, existe, também, um número de pacotes de software independentes para a execução dessa tarefa. Para dados de NGS, podem ser considerados como exemplos as soluções Bambus (Pop et al., 2004b), MIP *scaffolder* (Salmela et al., 2011), SOPRA (Dayarian et al., 2010) e SSPACE (Boetzer et al., 2011). Ainda há pouca informação, na literatura atual, que indique o melhor pacote dessa classe em termos de tamanho do *scaffold* gerado ou acurácia (Henson et al., 2012).

Tabela de montadores de novo por Miller et al. (2010)

Table 1
Feature comparison between de novo assemblers for whole-genome shotgun data from next-generation sequencing platforms. OLC refers to the overlap/layout/consensus architecture. DBG refers to the de Bruijn graph architecture. The table is based on the literature cited in the text. It may not reflect the current state of each software package.

Algorithm Feature	Greedy Assemblers	OLC Assemblers	DBG Assemblers
Modeled features of reads			
Base substitutions			Euler, AllPaths, SOAP
Homopolymer miscount		CABOG	
Concentrated error in 3′ end			Euler
Flow space		Newbler	
Color space		Shorty	Velvet
Removal of erroneous reads			
Based on K-mer frequencies			Euler, Velvet, AllPaths
Based on K-mer freq and QV			AllPaths
For multiple values of K			AllPaths
By alignment to other reads		CABOG	
By alignment and QV	SHARCGS		
Correction of erroneous base calls			
Based on K-mer frequencies			Euler, SOAP
Based on Kmer freq and QV			AllPaths
Based on alignments		CABOG	
Approaches to graph construction			
Implicit	SSAKE, SHARCGS, VCAKE		
Reads as graph nodes		CABOG, Newbler, Edena	
K-mers as graph nodes			Euler, Velvet, ABySS, SOAP
Simple paths as graph nodes			AllPaths
Multiple values of K			Euler
Multiple overlap stringencies	SHARCGS		
Approaches to graph reduction			
Filter overlaps		CABOG	
Greedy contig extension	SSAKE, SHARCGS, VCAKE		
Collapse simple paths		CABOG, Newbler	Euler, Velvet, SOAP
Erosion of spurs		CABOG, Edena	Euler, Velvet, AllPaths, SOAP
Transitive overlap reduction		Edena	
Bubble smoothing		Edena	Euler, Velvet, SOAP
Bubble detection			AllPaths
Reads separate tangled paths			Euler, SOAP
Break at low coverage			Velvet, SOAP
Break at high coverage		CABOG	Euler
High coverage indicates repeat		CABOG	Velvet
Special use of long reads		Shorty	Velvet
Graph partitions			
Partition by K-mers			ABySS
Partition by scaffolds			AllPaths
Uses for mate pairs			
Constrain path searches			Euler, Velvet, AllPaths
Guide path selection			Euler, Allpaths
Detect misassembled contigs		CABOG, Shorty	
Merge contigs or fill gaps		CABOG, Shorty	Velvet, ABySS, SOAP
Transitive link reduction		CABOG	SOAP
Detect, avoid repeat contigs		CABOG	Velvet, SOAP
Create scaffolds		CABOG, Shorty	Euler, Velvet, AllPaths, SOAP

Tabela de montadores de novo por Paszkiewicz e Studholme (2010)

Table 2: Software for assembling short sequence reads

Program	Current version (as of 27 February 2010)	References
ABYSS	1.1.2 (15 February 2010)	[63, 64]
ALLPATHS	3 (3 December 2009)	[60, 65, 66]
CLC NGS Cell 3.0	2.2.1 (28 July 2009)	[67]
Curtain[a]	0.0.2 (16 December 2009)	[68]
Edena	2.1.1 (17 March 2008)	[69, 70]
Euler-SR	1.1.12 (30 March 2009)	[71–73]
FuzzyPath	3.1 (8 November 2009)	[74, 75]
Oases[b]	0.1.4 (29 January 2010)	[76]
QSRA	No version number (11 March 2009)	[77, 78]
SASSY	No version number	[17]
SeqCons	No version number (24 September 2009)	[79, 80]
SHARCGS	No version number (19 November 2007)	[82]
SHORTY	2.0 (date unknown)	[83, 84]
SOAPdenovo	2.20 (13 August 2009)	[85, 86]
SOPRA	1 (date unknown)	[87, 88]
SSAKE	3.4 (14 April 2009)	[89, 90]
Taipan	1.0 (15 May 2009)	[91, 92]
VCAKE	1.1 (29 June 2009)	[93, 94]
Velvet	0.7.59 (19 February 2010)	[61, 95, 96]

These software packages are able to perform *de novo* assembly of Illumina short sequence reads with the exception of SHORTY, which is designed to assemble ABI SOLiD colour-space data. Velvet and SOPRA can assemble sequence-space and colour-space data. [a]Curtain is a pipeline, based on Velvet, for hierarchical assembly of short sequence reads in order to overcome memory usage limitations. [b]Oases is specifically designed for assembling transcribed sequences.

17. Imelfort M. Sequence comparison tools. In: Edwards D, Hansen D, Stajich J, (eds). Bioinformatics:Tools and Applications. London: Springer, 2009;13–37.
60. Butler J, MacCallum I, Kleber M, et al. ALLPATHS: de novo assembly of whole-genome shotgun microreads. Genome Res 2008;18:810–20.
61. Zerbino DR, Birney E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 2008;18:821–9.
62. Pevzner PA, Tang H, Tesler G. Denovo repeat classification
and fragment assembly. GenomeRes 2004;14:1786–96.
63. Simpson JT, Wong K, Jackman SD, et al. Birol I. ABySS: a parallel assembler for short read sequence data. GenomeRes 2009;19:1117–23.
64. ABySS. http://www.bcgsc.ca/platform/bioinfo/software/abyss (25 May 2010, date last accessed).
65. Maccallum I, Przybylski D, Gnerre S, et al. ALLPATHS 2: small genomes assembled accurately and with high continuity from short paired reads. Genome Biol 2009;10:R103.
66. ALLPATHS. http://www.broadinstitute.org/science/programs/genome-biology/computational-rd/computational-research-and-development (25 May 2010, date last accessed).
67. CLC bio. http://www.clcbio.com (25 May 2010, date lastaccessed).
68. Curtain. http://code.google.com/p/curtain (25 May 2010, date last accessed).
69. Hernandez D, Franc¸ois P, Farinelli L, et al. De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. Genome Res 2008;18:802–9.
70. Edena. http://www.genomic.ch/edena.php (25 May 2010, date last accessed).
71. Chaisson MJ, Brinza D, Pevzner PA. De novo fragment assembly with short mate-paired reads: does the read length matter? Genome Res 2009;19:336–46.

72. Chaisson MJ, Pevzner PA. Short read fragment assembly of bacterial genomes. GenomeRes 2008;18:324–30.
73. EULSER-SR. http://euler-assembler.ucsd.edu/portal (25 May 2010, date last accessed).
74. Sudbery I, Stalker J, Simpson JT, et al. Deep short-read sequencing of chromosome 17 from the mouse strains A/J and CAST/Ei identifies significant germline variation and candidate genes that regulate liver triglyceride levels. Genome Biol 2009;10:R112.
75. FuzzyPath. ftp://ftp.sanger.ac.uk/pub/zn1/fuzzypath (25 May 2010, date last accessed).
76. Oases. http://www.ebi.ac.uk/_zerbino/oases (25 May 2010, date last accessed).
77. Bryant DW Jr, Wong WK, Mockler TC. QSRA: a qualityvalue guided de novo short read assembler. BMC Bioinformatics 2009;10:69.
78. QSRA. http://qsra.cgrb.oregonstate.edu (25 May 2010, date last accessed).
79. Rausch T, Koren S, Denisov G, et al. A consistency-based consensus algorithm for de novo and reference-guided sequence assembly of short reads. Bioinformatics 2009;25:1118–24.
80. SeqCons. http://www.seqan.de/projects/seqcons.html (25 May 2010, date last accessed).
82. SHARCGS. http://sharcgs.molgen.mpg.de (25 May 2010, date last accessed).
83. Hossain MS, Azimi N, Skiena S. Crystallizing short-read assemblies around seeds. BMC Bioinformatics 2009;10:S16.
84. SHORTY. http://www.cs.sunysb.edu/_skiena/shorty/(25 May 2010, date last accessed).
85. Li R, Zhu H, Ruan J, et al. De novo assembly of human genomes with massively parallel short read sequencing. Genome Res 2010;20:265–72.
86. Short Oligonucleotide Analysis Package. http://soap.genomics.org.cn.
87. Dayarian A, Michael TP, Sengupta AM. SOPRA: an algorithm for high quality de novo assembly of paired reads via statistical optimization. In press.
88. SOPRA. http://www.physics.rutgers.edu/_anirvans/SOPRA/ (25 May 2010, date last accessed).
89. Warren RL, Sutton GG, Jones SJ, Holt RA. Assembling millions of short DNA sequences using SSAKE. Bioinformatics 2007;23:500–1.
90. SSAKE. http://www.bcgsc.ca/platform/bioinfo/software/ssake (25 May 2010, date last accessed).
91. Schmidt B, Sinha R, Beresford-Smith B, Puglisi SJ. A fast hybrid short read fragment assembly algorithm. Bioinformatics 2009;25:2279–80.
92. Taipan. http://taipan.sourceforge.net (25 May 2010, date last accessed).
93. Jeck WR, Reinhardt JA, Baltrus DA, etal. Extending assembly of short DNA sequences to handle error. Bioinformatics 2007;23:2942–4.
94. VCAKE. http://sourceforge.net/projects/vcake (25 May 2010, date last accessed).
95. Zerbino DR, McEwen GK, Margulies EH, Birney E. Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. PLoS One 2009;4:e8407.
96. Velvet. http://www.ebi.ac.uk/_zerbino/velvet/ (25 May 2010, date last accessed).

Tabela de montadores de novo por Henson et al. (2012)

Table 2. Performance and running requirements of popular assemblers.

Assembler	Resources employed for large genome	Test genome	Assembly time	Max. memory usage	Parallel-ised	Data type used and information of PS, RL and MP	Contig N50 (Kb)	Scaffold N50 (Kb)	Assembled size and coverage	General data requirements	Summary comments	Ref.
ABySS	168-core cluster, 2.66-GHz CPU	Human	87 h	~250 Gb (<16 Gb/ node)	Yes	Illumina PE PS 210 bp; RL 2×35–46 bp of 45×	2.4	N/A	2.2 Gb 80.6%	Any short-read libs	Low RAM requirement; good local contig accuracy, but sometimes produces duplicated contigs	[113]
ALLPATHS-LG	48 processors with 512-Gb RAM	Human	25 days	<512 Gb	Yes	Illumina PE PS 180 bp; RL 2×100 bp of 45×; MP 3, 6 and 40 kp of 51×	24	11,543	2.55 Gb 91.1%	Requires one overlapping PE and one MP, can now add long-read libs	Very good on contig continuity and accuracy; restrictive read-library requirements	[114]
CABOG	Computer grid and 16 processors with 256-Gb RAM	T. Devil	Not given	<256 Gb	Yes	Illumina PE PS 300 bp; RL 2×75 bp of 49×; Roche 454 single and MP 6–15 Kb of 8×	11	146.8	2.93 Gb >95%	Accepts both long (Sanger, 454) and short (Illumina, SOLiD) reads	Versatile: accepts multiple data types required computing resources are relatively big	[115]
Phusion2	32 processors with 512 Gb RAM	T. Devil	70 h	<512 Gb	Yes	Illumina PE PS 450 bp; RL 2×100 bp of 85×; MP 3–10 Kb of 10×	28.9	2244.5	2.93 Gb >95%	Any short-read libs	Flexible, good contig continuity but read clustering is sensitive to the evenness of read coverage	[116]
SGA	~100 cores	Human	1417 CPU hours or 6 days	53 Gb	Optional	Illumina PE PS 380 bp; RL 2×100 bp of 48×	9.9	25.1	2.69 Gb 95.4%	Any short-read libs	Low RAM requirement and excellent contig accuracy Contig continuity needs improvement	[117]
SOAPdenovo	32 processors with 512-Gb RAM	Panda	40 h	<256 Gb	Yes	Illumina SE, PE 150 bp, 500 bp; RL 2×45–67 bp of 50×; MP 2.5, 5 and 10 Kb of 23×	40	1220	2.3 Gb >95%	Any short-read libs	User-friendly, efficient assembler, good long-range continuity Contig accuracy needs improvements	[118]

Requirements for, and performance of, large genome assemblers. Figures are given for assemblies reported in publications on each assembler cited in the text (figures and notes on ABySS also use information from private communication with Jared Simpson). Caution is necessary when comparing the figures on assembly quality: it is important to note that the input data in each case is far from equivalent, and details of the calculations of statistics may also vary. Each assembler can be expected to improve over time and so relative performance of older assemblers may be understated (e.g., the ABySS assembly does not show its ability to scaffold). SGA was run with only trivial parallelization and could be run serially. No maximum memory requirement was reported for ALLPATHS-LG and so the total memory of the cluster used is given here [31,36–39,45]. Contig: Contiguous sequence; CPU: Computer processing unit; libs: Libraries; MP: Mate pair, N/A: Not applicable; PE: Paired end; PS: Pair size; RAM: Random access memory; RL: Read length; SGA: String Graph Assembler; T. Devil: Tasmanian Devil.

113 BC Cancer Agency: Genome Sciences Centre. ABySS. www.bcgsc.ca/platform/bioinfo/software/ abyss
114 ALLPATHS-LG. www.broadinstitute.org/software/allpaths-lg/ blog/?page_id=12
115 SourceForge: Celera Assembler. http://sourceforge.net/apps/mediawiki/ wgs-assembler/index.php?title=Main_Page
116 Phusion 2. ftp://ftp.sanger.ac.uk/pub/zn1/phusion2
117 GitHub: SGA. https://github.com/jts/sga
118 SOAP: Short Oligonucleotide Analysis Package. http://soap.genomics.org.cn/soapdenovo.html

Anexo C - Tabela comparativa de alguns programas do tipo *workflow*

Table 1 Comparison of common graphical workflow environments

Workflow Environment	Requires Tool Recompiling	Data Storage	Platform Independent	Client-Server Model	Grid Enabled	Application Area	URL
LONI Pipeline [15]	N	External	Y	Y	Y (DRMAA)	Area agnostic	http://Pipeline.loni.ucla.edu
Taverna [35]	Y (via API)	Internal (MIR)	Y	N	Y (myGRID)	Bioinformatics	http://www.taverna.org.uk
Kepler [14]	Y (via API)	Internal (actors)	Y	N	Y (Ecogrid)	Area agnostic	http://kepler-project.org
Triana [36]	Y	Internal data structure	Y	N	Y (gridLab)	Hetero-geneous Apps	http://www.trianacode.org
Galaxy [37]	N	External	N (Linux, Mac)	Y	N (Cloud EC2)	Bioinformatics	http://usegalaxy.org
Pipeline Pilot	Y	Internal	Y	N	N	Biochemistry	http://accelrys.com/products/pipeline-pilot/
AVS [38]	Y	Internal	Y (platform build)	N	N	Advanced Visualization	http://www.avs.com
VisTrails [39]	Y	Internal	N	N	N	Scientific Visualization	http://www.vistrails.org
Bioclipse [40]	N (plug-ins)	Internal	Y	N	N	Biochemistry Bioinformatics	http://www.bioclipse.net

Y = yes, N = no; Taverna MIR plug-in, MIR = myGrid Information Repository; DRMAA = Distributed Resource Management Application API.