PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

OTIMIZAÇÕES QUALITATIVAS E QUANTITATIVAS NAS FASES DE LEITURA E ANÁLISE EM PIPELINES METAGENÔMICOS

RAQUEL DIAS

Dissertação apresentada como requisição parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. César Augusto F. De Rose

Porto Alegre
2012

Dados Internacionais de Catalogação na Publicação (CIP)

D541o Dias, Raquel
Otimizações qualitativas e quantitativas nas fases de leitura e análise em pipelines metagenômicos / Raquel Dias. – Porto Alegre, 2012.
96 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. César Augusto F. de Rose.

1. Bioinformática. 2. Genética. 3. Análise de Dados. 4. Base de Dados. 5. Biologia Computacional. I. De Rose, César Augusto Fonticielha. II. Título.

CDD 005.74

Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Otimizações Qualitativas e Quantitativas nas Fases de Leitura e Análise em Pipelines Metagenômicos", apresentada por Raquel Dias como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Processamento Paralelo e Distribuído, aprovada em 06/08/2012 pela Comissão Examinadora:

Prof. Dr. César Augusto Fonticielha De Rose – Orientador — PPGCC/PUCRS

Prof. Dr. Luiz Gustavo Leão Fernandes – — PPGCC/PUCRS

Prof. Dr. Eduardo Eizirik - — PPGBCM /PUCRS

Homologada em 13/11/2012, conforme Ata No. 024 pela Comissão Coordenadora.

Prof. Dr. Paulo Henrique Lemelle Fernandes
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P32– sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

AGRADECIMENTOS

O desenvolvimento do presente trabalho foi financiado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e FINEP (Financiadora Nacional de Estudos e Pesquisa), em paralelo com outros projetos. Também foram fornecidos apoio e suporte a partir do Laboratório de Alto Desempenho, da PUCRS, em colaboração com o instituto Idéia.

Gostaria de agradecer aos colegas de trabalho e amigos por todo o apoio que recebi. Sou muito grata também aos meus familiares e principalmente ao meu marido Raul Chipana, pela compreensão, paciência e por todo o incentivo recebido ao longo destes anos. O meu profundo e sincero agradecimento a todas as pessoas que contribuíram para a concretização desta dissertação, estimulando-me intelectual e emocionalmente.

OTIMIZAÇÕES QUALITATIVAS E QUANTITATIVAS NAS FASES DE LEITURA E ANÁLISE EM PIPELINES METAGENÔMICOS

RESUMO

As tecnologias de sequenciamento metagenômico tem avançado rapidamente e a quantidade de dados gerados a partir do sequenciamento em larga escala tem aumentado substancialmente ao longo dos anos. As presentes otimizações e avaliações de desempenho tem foco em algumas das etapas mais críticas e que consomem mais tempo em uma análise metagenômica: pré-processamento, classificação taxonômica e pós - processamento dos resultados de classificação. Otimizações e funções foram implementadas e introduzidas em uma nova arquitetura, PANGEA+, baseada no *pipeline* metagenômico PANGEA. Os principais melhoramentos alcançados com a presente ferramenta foram: suporte a vários formatos de arquivos de entrada e a base de dados taxonômicos do NCBI, novos métodos de classificação de espécies incluídos, análise consenso, implementação de memória distribuída para a fase de classificação de espécies, otimizações de baixa complexidade para o pós-processamento dos resultados de classificação. A avaliação da nova arquitetura, PANGEA+, demonstra melhoramentos consideráveis em várias funcionalidades e, principalmente, na etapa de classificação de espécies, tanto em exatidão quanto em desempenho computacional.

Palavras Chave: PANGEA+, MPI-blastn, NCBI-TaxCollector, análise consenso, pipelines metagenômicos, 16S, classificação de espécies.

QUALITATIVE AND QUANTITATIVE OPTIMIZATIONS IN READ AND ANALYSIS STEPS IN METAGENOMIC PIPELINES

ABSTRACT

Metagenomic sequencing technologies are advancing rapidly and the size of output data from high-throughput genetic sequencing has increased substantially over the years. Our optimizations and performance evaluations are focused in some of the most critical and time-consuming steps of a metagenomic analysis: pre-processing, taxonomic classification assignment and post-processing of classification results. Optimizations and functions were implemented and introduced in a new architecture, PANGEA+, based on the PANGEA metagenomic pipeline. The main improvements of the present tool are: support of new input file formats and NCBI taxonomy database, new species classification methods, consensus analysis, implementation of distributed memory (MPI) for species classification step, and low complexity optimizations for the post-processing of classification results. The evaluation of the new architecture, shows remarkable improvements in many features and, mainly, in the species classification accuracy and performance.

Keywords: PANGEA+, MPI-blastn, NCBI-TaxCollector, consensus analysis, metagenomic pipelines, 16S, species classification.

LISTA DE TABELAS

LISTA DE SIGLAS

CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
DNA – ácido desoxirribonucléico (*deoxyribonucleic acid)*
FINEP - Financiadora Nacional de Estudos e Pesquisa
GPU – Unidade de Processamento Gráfico (*Graphic Processing Unit*)
LAD – Laboratório de Alto Desempenho
MPI - Interface de Troca de Mensagens (*Message Passing Interface*)
PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul
RNA – ácido ribonucléico (ribonucleic acid)

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

Nosso planeta é habitado por um vasto número de células de microrganismos (aproximadamente 10^{30}) [63]. Contudo, a maioria destes (estimada em 99%) são impossíveis de serem cultivados em laboratório [38]. Uma nova área de pesquisa emergiu recentemente devido aos avanços das tecnologias de sequenciamento genético. Conhecida como metagenômica (*metagenomics*), esta linha de pesquisa tem como objetivo compreender a diversidade, variabilidade, abundância, e riqueza de espécies da microbiota ambiental. A metagenômica consiste basicamente no sequenciamento e análise do material genético obtido a partir de uma amostra ambiental. O sequenciamento genético tradicional dependia da cultura de clones como uma fonte de DNA. No entanto, muitos organismos não podem ser isolados e cultivados em laboratório. Sendo uma alternativa para a identificação de organismos incapazes de serem isolados / clonados, a metagenômica surgiu como uma solução para a identificação da biodiversidade ambiental. Operacionalmente, a metagenômica trouxe uma grande inovação, envolvendo o estudo dos genomas e classificação de muitos organismos simultaneamente.

Os dados adquiridos a partir de análises metagenômicas fornecem informações sobre a estrutura, organização, evolução, e origem de organismos; e pode ser utilizada em aplicações científicas em benefício da sociedade e do meio ambiente. Por exemplo, estudos revelam que o perfil de microrganismos que compõem a microbiota intestinal pode estar relacionado a doenças crônicas, como diabetes, obesidade e doença de Chron [5]. Através da metagenômica, esses estudos demonstraram a importância de se identificar interações entre microrganismos e como sua biodiversidade influencia a saúde humana. Além disso, os microrganismos não só interagem com os animais, mas também com as plantas, em vários níveis. A relação entre as plantas e bactérias / fungos envolve muitas características complexas, tais como sinalização química e defesa contra parasitas, que podem estar relacionadas com a saúde da planta e sua produtividade. Estudos metagenômicos sobre comunidades de microbiotas de plantas pode

revelar informações importantes sobre a funcionalidade efeitos benéficos da biodiversidade microbiana sobre o crescimento da planta [19].

Como mencionado, a metagenômica se baseia nas tecnologias de sequenciamento capazes de obter dados genéticos a partir de amostras ambientais. Após a etapa de sequenciamento, ferramentas computacionais são utilizadas para analisar os dados obtidos. Cada estratégia de sequenciamento pode variar em nível de escopo, abrangência ou alvo dos dados genéticos a serem sequenciados, assim como na velocidade do processo de sequenciamento. Atualmente, existem duas principais estratégias de sequenciamento que são utilizadas na metagenômica: (1) metagenômica baseada em 16S (metagenômica baseada em DNA ribossômico ou DNAr) utilizando a tecnologia de sequenciamento Sanger; e (2) análise metagenômica ou análise metatranscriptômica usando tecnologias de sequenciamento de última geração.

Ribossomos (assim como o DNA que os codifica) foram conservados em sua maior parte ao longo do tempo, com um pequeno número de alterações em sua estrutura, mantendo a sua importante função de tradução de RNAm para proteínas. O DNA ribossomal (DNAr) é formado por regiões conservadas, que mantêm a sua funcionalidade, e as regiões hiper-variáveis, que são sujeitas a várias mutações. 16S é um gene ribossomal (16S RNAr) que está presente nas bactérias, geralmente conservado dentro das espécies. As variações no gene 16S são utilizados para a classificação de espécies de bactérias e Archeae a partir do sequenciamento de amostras ambientais de DNA. Uma vantagem em utilizar o gene 16S como alvo para a sequenciamento é que este gene é relativamente pequeno (1,5 kb), tornando-o seu sequenciamento mais rápido quando comparado com outros genes bacterianos [75]. Além disso, uma vez que este está presente apenas em bactérias, é fácil extraí-lo e classificá-lo a partir de amostras ambientais ou médicas, isolando-o de eucariotos (plantas, animais, fungos e protistas têm um gene diferente, conhecido como gene 18S) [11]. Além disso, o seu menor custo de sequenciamento e ampla gama de genes 16S disponíveis nos bancos de dados on-line torna suas comparações acessíveis e viáveis [46][75]. Considerando a diversidade de nomenclaturas encontrada na

literatura, no presente trabalho este procedimento será chamado metagenômica baseada em 16S.

Outra estratégia de sequenciamento de larga escala, conhecida como metatranscriptômica, utiliza tecnologias de nova geração de sequenciamento, a fim de obter o genoma inteiro de organismos a partir de uma amostra, ao contrário do sequenciamento baseado em 16S (que foca somente no sequenciamento de um gene). Por meio do sequenciamento de muitos fragmentos de DNA provenientes dos organismos de uma amostra, este processo consiste na montagem e na reconstrução de todo o genoma dos organismos. O sequenciamento de um genoma inteiro, obviamente, traz mais informações sobre os genes dos organismos, quando comparado com o sequenciamento baseado em 16S. A metatranscriptômica pode acessar informações sobre as taxas de mutação, realizar predição de genes codificantes de proteínas e classificações funcionais para estes genes identificados. No entanto, quando comparado com o método baseado em 16S, o sequenciamento mencionado é muito mais custoso, em nível de tempo de execução e em nível financeiro.

As vantagens do sequenciamento baseado em 16S o revelou como uma boa alternativa para análises metagenômicas. A utilização deste método tem aumentado consideravelmente desde os últimos anos, devido ao seu baixo preço e alta velocidade e sequenciamento. Aliada às novas gerações de tecnologias de sequenciamento, estas técnicas geram centenas de milhares de leituras em um pequeno período de tempo. Assim, a metagenômica baseada em 16S tornou-se uma abordagem acessível e eficiente.

Tecnologias de sequenciamento metagenômico estão avançando rapidamente. Desde os últimos anos, o tamanho dos dados de saída do sequenciamento genético de larga escala tem aumentado substancialmente. As tecnologias de sequenciamento mais recentes evoluíram de uma velocidade de sequenciamento de 10Mb por dia para cerca de 120Gb por dia [29][39]. O pós-processamento dos dados de sequenciamento é indispensável para entender os resultados obtidos. Uma análise metagenômica consiste em um procedimento com várias etapas, que compreendem: filtragem das leituras por qualidade e tamanho, busca das

sequências filtradas contra um banco de dados genéticos com o objetivo de identificar sequências similares conhecidas, atribuição de classificação taxonômica sobre os resultados desta busca, análise estatística das sequências classificadas e não classificadas, resumo e visualização dos resultados. Alguns destes passos devem ser realizados consultando grandes bases de dados, tais como NCBI [22], Greengenes [17] e RDP [13], além de se utilizar um grande número de sequências de entrada. Portanto, ferramentas *standalone* e *online* foram desenvolvidas e passaram a ser utilizadas para realizar uma análise mais rápida e mais automatizada quanto possível. Alguns exemplos de tais ferramentas são: PANGEA [23], Mothur [70], RDP classifier [12], Galaxy [42] and MEGAN [33]. Estas ferramentas, conhecidas como *pipelines* metagenômicos, otimizam o tempo de execução da análise de dados de sequenciamento em larga escala, tentando minimizar a intervenção humana neste processo. *Pipelines* metagenômicos tornaram possíveis muitos estudos metagenômicos [36][37][40][41][73].

Apesar das ferramentas computacionais terem acelerado os estudos metagenômicos, a quantidade de sequências geradas pelas tecnologias de última geração estão aumentando em uma velocidade ainda maior. Estudos realizados por Kahn revelam que a duplicação dos dados de saída de sequenciamentos metagenômicos a cada 9 meses superou as melhorias dos avanços em armazenamento de disco e da computação de alto desempenho [39]. Estes dados podem ser observados na Figura 1. Isso nos aproxima de um cenário onde otimizações computacionais avançadas serão estritamente necessárias para concluir uma análise metagenômica comum.

O presente trabalho foi elaborado a partir de uma análise das principais ferramentas computacionais disponíveis para análises metagenômicas, identificando as suas necessidades, com o objetivo de propor e desenvolver otimizações qualitativas e quantitativas. As otimizações são focadas principalmente em algumas das etapas mais complexas e demoradas de uma análise metagenômica: filtragem das sequencias de entrada, atribuição de classificação taxonômica e pós-processamento dos resultados da classificação. Estas, entre outras etapas, são realizadas pelos *pipelines* metagenômicos com diferentes níveis de automatização, praticidade, desempenho computacional e exatidão. As otimizações foram

implementadas resultando em uma nova arquitetura, denominada PANGEA+. Sua avaliação demonstrou que PANGEA+ é capaz de realizar análises metagenômicas de forma mais rápida e eficiente quando comparada com outras ferramentas.

O restante do presente documento está estruturado da seguinte forma: na seção 2 são revisadas as principais ferramentas utilizadas, conhecidas como *pipelines,* para análises metagenômicas, discutindo suas principais vantagens, desvantagens e necessidades; na seção 3 é descrito o fluxo de trabalho e as principais características das funcionalidades que compõem a nova arquitetura proposta; a seção 4 consiste na avaliação qualitativa e quantitativa de cada uma das novas funcionalidades desenvolvidas; a seção 5 corresponde a concussão do trabalho, abordando os principais avanços alcançados e questões para possíveis trabalhos futuros.

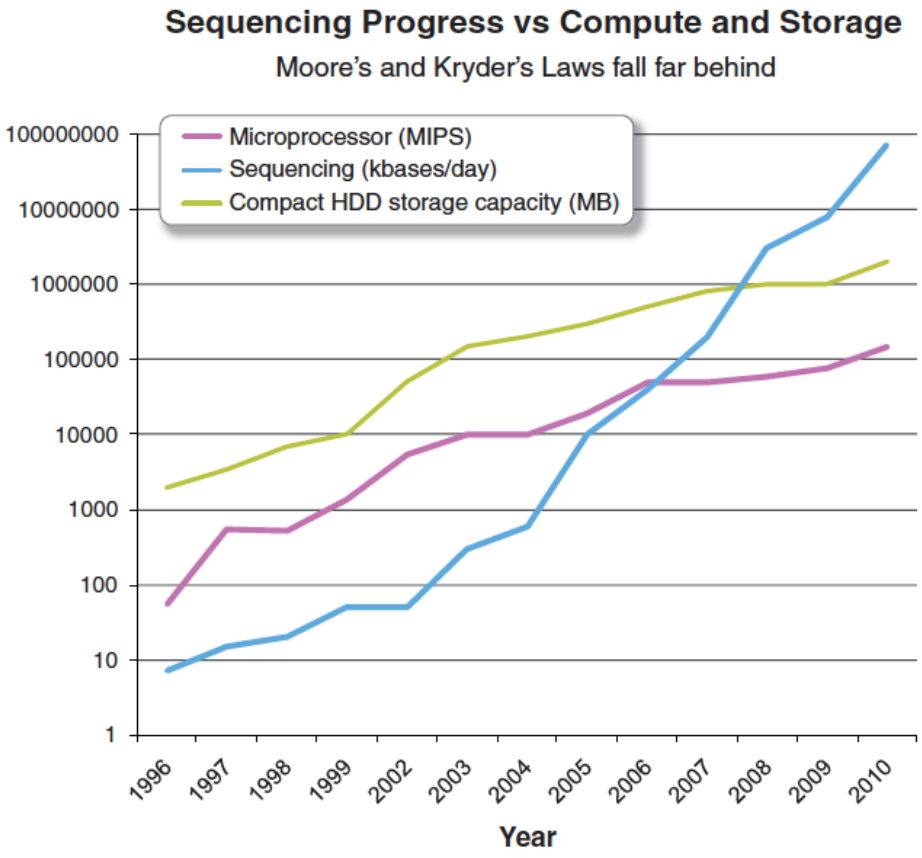

Figura 1. Progresso do sequenciamento contra Progresso de processamento computacional e de armazenamento. Referência: Khan *et al* [39].

2 ESTADO DA ARTE: PIPELINES METAGENÔMICOS

Como mencionado, uma análise metagenômica consiste em um procedimento com várias etapas, que compreendem: filtragem das leituras por qualidade e tamanho, busca das sequências filtradas contra um banco de dados genéticos com o objetivo de identificar sequências similares conhecidas, atribuição de classificação taxonômica sobre os resultados desta busca, análise estatística das sequências classificadas e não classificadas, resumo e visualização dos resultados. Estas etapas se organizam em um fluxo de trabalho com passos que variam em complexidade, automaticidade e tempo de execução. Uma visão resumida deste fluxo de trabalho, com seus tempos de execução relativos aproximados, pode ser observada na Figura 2. Atualmente, o tempo médio de execução da filtragem dos dados de entrada corresponde a menos de 3% em relação ao tempo total de execução. As fases com maior duração de tempo são a classificação de espécies e o pós-processamento dos resultados, que ocupam em média ~43% e ~57%, respectivamente, do tempo total de execução para uma análise metagenômica.

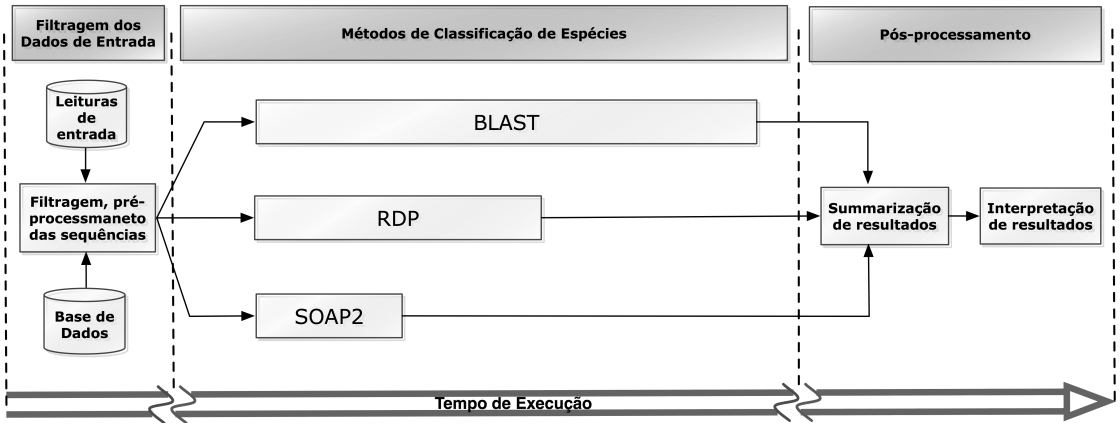

Figura 2. Fluxo de trabalho de uma análise metagenômica, demonstrando as suas principais etapas e seu tempos de execução relativos.

Nesta seção, discutiremos quais são as principais funcionalidades dos *pipelines* metagenômicos disponíveis e se estas abrangem cada uma destas etapas do fluxo de trabalho.

2.1 MEGAN

MEGAN é utilizado para realizar a análise inicial de uma amostra metagenômica. Esta ferramenta deve ser utilizada em conjunto com os outras, sendo que esta não abrange todos os passos da análise metagenômica. Fora do MEGAN, após coletar as leituras obtidas do sequenciamento, é realizada uma busca destas sequências contra uma ou mais bases de dados conhecidas. Esta busca ou alinhamento entre as sequências é feita pelo programa BLAST (Basic Local Alignment Sequence Search) ou um programa similar [28]. Após este passo, o MEGAN pode ser utilizado a fim de processar os resultados da busca e coletar informações sobre as sequências a partir da taxonomia NCBI [22]. O resultado deste processamento é armazenado em um arquivo MEGAN, que contém informações que podem ser utilizadas para a geração de saídas gráficas e estatísticas. Em uma próxima etapa, o usuário pode executar o algoritmo para encontrar o menor ancestral comum (Last Common Ancestor - LCA). Este algoritmo é utilizado para atribuir classificações com base no número de *hits* da busca, elaborar sumários sobre os resultados e construir árvores filogenéticas. Para esta pós-classificação, são considerados genes ribossomais apenas (16S e 18S e 23S RNAr) como base para a comparação.

Na versão atual do MEGAN [33], a análise filogenética consiste em dois novos métodos, uma com base na classificação SEED e outra baseada em KEGG (*Kyoto Encyclopedia of Genes and Genomes*). SEED é um ambiente de análise genômica comparativa e refinamento de classificações de dados genômicos, enquanto o KEGG é um banco de dados de genes ortólogos definidos manualmente e grupos de genomas para a atribuir funções enzimáticas e vias metabólicas aos resultados do sequenciamento [57].

MEGAN é uma ferramenta de certa maneira flexível para o pós-processamento metagenômico, sendo que esta pode ser combinada com programas de busca / alinhamento de sequências diferentes (embora

manualmente) nos passos que precedem sua utilização. MEGAN tem se demonstrado como uma ferramenta eficiente para a identificação e classificação de genes, através de análises de diversos conjuntos de dados (simulados ou experimentais) [56][67][76]. Por outro lado, MEGAN tem uma grande desvantagem porque as etapas de análise iniciais são fragmentadas e são dependentes do trabalho do usuário. Ou seja, MEGAN não possui funções de filtragem de sequências por qualidade, conversão dos dados de entrada e saída para outros formatos, alinhamento de sequências, deixando essa tarefa por conta do usuário. Outra restrição no MEGAN é sua conexão com bancos de dados estritamente online para realizar a sua análise, impossibilitando o usuário de fazer sua análise contra os seus próprios bancos de dados.

2.2 Mothur

Ao contrário do MEGAN, o *pipeline* Mothur tem o objetivo ser um pacote de ferramentas mais completo para analisar os dados de sequenciamento metagenômico. Ele pode ser usado para uma boa parte das etapas de uma análise genômica: filtrar e alinhar as sequências; calcular as suas diferenças; classifica-las; e analisar a biodiversidade. Mothur consiste em um conjunto de várias ferramentas de análise metagenômica já existentes, as quais foram adaptadas para um pacote só. Mothur tem suporte aos dados de saída de ferramentas de classificação, que devem ser executadas separadamente. A maioria das ferramentas de classificação de espécies suportadas pelo Mothur são versões on-line, tais como: Nast, SINA e alinhadores RDP [14][17][18][64]. Entretanto, esta característica pode ser uma desvantagem, sendo que as ferramentas on-line somente aceitam pesquisas contra suas próprias bases de dados, não permitindo que o usuário consulte em suas próprias bases locais.

Entre as principais vantagens de Mothur, está a utilização ferramentas de classificação e análise estatística para identificação das sequências e atribuição de níveis taxonômicos; construção de curvas de amostragem; e estimativa de riqueza e biodiversidade [49][69]. Mothur também compara e

classifica as estruturas de diferentes comunidades e realiza o pós-processamento desses dados para visualização. Para tanto, Mothur gera dendrogramas, mapas de calor, e diagramas de Venn, a fim de adquirir uma visualização mais indicativa e didática dos resultados. No *pipeline* Mothur a maioria das ferramentas foram adaptadas e otimizadas para a obtenção de maior desempenho e flexibilidade. Se comparado com MEGAN, o *pipeline* Mothur é uma ferramenta muito mais completa e automática, uma vez que simplificou muitas etapas de um procedimento de análise metagenômica.

No entanto, há uma desvantagem em seu procedimento de classificação: este requer que seja feito e disponibilizado um alinhamento que servirá de modelo para o programa, no qual o Mothur se baseará para realizar toda a análise. Dependendo de com qual amostra o usuário está trabalhando, este poderá ter que encontrar ou criar o seu próprio alinhamento modelo. Uma vez que Mothur não fornece este recurso, que é necessário para a sua execução, isso implica em um custo de trabalho manual considerável, da parte de quem o utiliza. Um alinhamento automático é um passo inicial e essencial para o procedimento de análise metagenômica e ainda está faltando no pipeline Mothur. É considerado como uma desvantagem fundamental no pipeline Mothur, uma vez que a classificação das espécies é um dos passos críticos em estudos metagenômica.

2.3 Galaxy

Galaxy [42] é um *pipeline* web que inclui várias etapas de um pós-processamento metagenômico, tais como: processamento e controle de qualidade dos dados gerados a partir tecnologias de sequenciamento diferentes, análise estatística e visualização de dados. Através de um servidor, o utilizador pode analisar alinhamentos múltiplos, comparar classificações de genomas e genes, localizar padrões nas amostras, etc. O fluxo de trabalho Galaxy pode ser descrito como um conjunto de funcionalidades diferentes. Inicialmente, ele realiza uma busca de informações taxonômicas ou identificações, contra os dados disponíveis na base de dados do NCBI [35]. Para cada sequência encontrada é atribuído

um número de identificação do GenBank (GI) [3], utilizado pelo mesmo banco de dados do NCBI. Nesta etapa, o MegaBLAST on-line é utilizado [6]. Os resultados obtidos a partir deste procedimento inicial são somados a uma lista de dados sobre os níveis taxonômicos atribuídos a cada sequência identificada. O resultado obtido é um resumo de todas as classificações de sequências, onde cada sequência é comparada com outra hierarquicamente em uma árvore filogenética, mas sem uma representação gráfica ainda.

No passo seguinte, os resultados obtidos na função descrita são convertidos para uma representação gráfica de uma árvore filogenética em formato PDF. Outra função lê os dados gerados pelas etapas anteriores como entrada e gera uma lista resultados únicos encontrados na classificação taxonômica. Em seguida, é realizada uma identificação de qual sequência representa o ancestral menos derivado (ou a base da árvore filogenética) [42]. Por último, o Galaxy testa o número de diferenças significantes entre cada sequência classificada. Ele compara duas sequências usando a pontuação de Poisson para teste de duas amostras, com base no trabalho de Huffman [31] e realiza correções com múltiplos testes.

Galaxy é um *pipeline* completo e apresenta muitos recursos, se comparado com outros *pipelines metagenômicos*, como MEGAN e Mothur. Quando comparado com MEGAN, por exemplo, Galaxy é notavelmente menos exaustivo em relação aos esforços dos usuários para realizar buscas em banco de dados. Recentemente, uma versão local foi desenvolvida, o que permite que usuários e desenvolvedores possam otimizá-lo e usar seus próprios. No entanto, o *pipeline* Galaxy possui três limitações principais: (i) As bases de dados de pesquisa estão limitadas a apenas dois conjuntos de dados NCBI, (ii) que se limita a análise metagenômica baseada em homologia apenas, enquanto existem muitas maneiras diferentes para realizar este estudo comparativo, e (iii) as saídas de pós-processamento para a análise estatística e filogenética estão agrupadas e limitadas a um formato interno, não interpretável por ferramentas estatísticas externas. A falta de flexibilidade, principalmente neste último passo, faz com que o Galaxy se torne dificilmente adaptável para combinar outros programas

estatísticos ou filogenéticos. Consequentemente, este fator dificulta a utilização de métodos de análise alternativos.

2.4 RAST

Rapid Annotation using Subsystems Technology (RAST) é um *pipeline* metagenômico *web [25][55]*. O *pipeline* RAST pode ser descrito em três etapas principais: (1) análise e filtragem de sequências baseada em qualidade de leitura; (2) classificação funcional e de espécies; e (3) análise filogenética. No passo de filtragem das sequências por qualidade, o usuário carrega um arquivo de entrada (apenas o formato de arquivos fasta é suportado). Este arquivo é filtrado pelo tamanho das sequências e sua qualidade da leitura. Este é também conhecido como um passo de normalização, onde as sequências redundantes, curtas e de qualidade baixa são removidas da análise. Assim, as sequências de entrada são preparados para o procedimento de classificação. Todos os arquivos de entrada e saída são classificados em um diretório estruturado, que está disponível para *download* pelo usuário [25].

O passo seguinte refere-se aos procedimentos de classificação de espécies e classificação funcional dos genes (também conhecida como anotação funcional). Para tanto, as sequências passam por um processo de inferência de suas codificações em proteínas, para estimar sua função, através de uma pesquisa BLASTX [6] contra a base de dados SEED [57]. Em paralelo com a busca BLASTX, um procedimento de classificação espécies é efetuado através do BLASTN [6]. Esta pesquisa é realizada utilizando bases de dados de rDNA, como GREENGENES [17], RDP [53] e a base de dados europeia de RNA do gene 16S [79][80]. Os critérios de pesquisa podem variar, dependendo de cada banco de dados.

No terceiro passo a análise filogenética é realizada. Os resultados obtidos nas análises do segundo passo são utilizados para construir a árvore filogenética correspondente as sequências analisadas. A anotação funcional e a análise filogenética são utilizadas para uma reconstrução metabólica inicial, fornecendo informações sobre fluxos metabólicos, reações, enzimas,

etc., as quais as sequências de entrada podem estar relacionadas. Através da interface *web*, o usuário pode visualizar os resultados, através de tabelas resumidas e representações gráficas.

A acessibilidade e visualização de resultados do RAST são uma vantagem deste *pipeline*, se comparado com Mothur e Galaxy. E, ao contrário dos *pipelines* mencionados, RAST é capaz de realizar buscas usando muitos bancos de dados. No entanto, RAST faz uso de um método único de classificação (BLAST), faltando uma comparação entre outras abordagens de classificação. Além disso, RAST ainda carece de flexibilidade em nível de suporte a diversos formatos de entrada, uma vez que é incapaz de ler os arquivos de saída das tecnologias de sequenciamento de nova geração (formatos FASTQ, QSEQS, e a sintaxe de sequencias no formato *Paired-ends* [2][34][56], por exemplo). Outra desvantagem deste *pipeline* é a sua disponibilidade estritamente on-line, uma vez que não existe uma versão local disponível. Isso impossibilita que o usuário tenha um controle maior sobre a análise e não o permite que utilize seus próprios recursos computacionais. Além disso, a versão mais recente do RAST ainda não suporta análise estatística para espécies classificadas e não classificadas [25].

2.5 RDP Pipeline

Ribosomal Database Project (RDP) [13] foi iniciado como um repositório de sequências com qualidade controlada, de rRNA proveniente de bactérias e archeas. Recentemente, tem sido melhorada com a adição de um conjunto de ferramentas de análise e uma nova melhoria na sua estratégia de alinhamento. Estes novos alinhamentos são realizados usando inferências sobre a estrutura secundária para os procedimentos de classificação de espécies, baseando na ferramenta Infernal Aligner [61]. Infernal consiste em um método estocástico de gramática livre de contexto que, como outros métodos probabilísticos, é treinado sob um conjunto de sequências representativas.

Um fator que pode trazer limitações ao sistema de classificação de espécies do RDP é que o sucesso do programa alinhamento depende do conjunto de dados utilizados no passo de treinamento do algoritmo. O conjunto de treinamento padrão, utilizado no RDP, consiste em sequências do gene 16S que são conhecidas (ou seja, determinadas experimentalmente). Estas sequências estão distribuídas em aproximadamente 880 gêneros. Essas sequências de bases nucleotídicas são subdivididas em subsequências de 8 bases (palavras) e então as probabilidades de cada uma destas palavras e suas combinações são calculadas para cada gênero. Assim, quando uma sequência é submetida a base de dados, ela é subdividida e comparada com a estrutura de dados obtida a partir do grupo de treino, e as probabilidades de presença destas palavras são medidas.

Uma das vantagens desse método de classificação é que sua exatidão é grande para sequências que obedecem os padrões do grupo de sequências que foram usadas na fase de treinamento. Entretanto, sua maior limitação ocorre quando a sequência de entrada não corresponde a nenhum dos padrões observados na fase de treinamento. Isso pode causar um decréscimo considerável no nível de exatidão do método, algo que não ocorreria com outros métodos de classificação, como o BLAST, por exemplo. Uma outra desvantagem é que o método de classificação de espécies requer sequências de no mínimo 50 bases de tamanho. Além disso, o RDP on-line somente suporta entradas com até 100.000 sequências, sendo que as novas tecnologias de sequenciamento podem gerar entradas maiores que estas.

Após a classificação, os próximos passos da análise metagenômica se assemelham ao pipeline Mothur e Galaxy, com ferramentas de análise filogenéticas, visualização dos resultados (também disponíveis para download em formato PostScript), análise estatística variabilidade, representatividade e diversidade das sequências, utilizando ferramentas como SPADE [10], EstimateS [15], and R [65].

2.6 Qiime

Qiime (*Quantitative Insights Into Microbial Ecology*) consiste em um pacote de programas para a análise comparativa de comunidades microbiológicas [7][43]. Utilizando sequências e seus dados de qualidade de leitura, as sequências podem ser filtradas, organizadas e convertidas para outros formatos suportados pelas etapas seguintes de análise. Após a filtragem das sequências, o programa se subdivide em um conjunto de várias etapas, que podem ser escolhidas de acordo com as necessidades do usuário. Os principais objetivos, entre as funções realizadas pelo Qiime, consistem em: classificação de espécies, análise filogenética e de biodiversidade, análise funcional e estatística.

Uma das principais inovações deste *pipeline* é a inclusão de novas ferramentas para a comparação entre os microrganismos classificados utilizando informação filogenética. Para realizar esta etapa, é utilizada uma árvore filogenética única que contém as sequências provenientes de pelo menos dois tipos diferentes de amostras ambientais e um arquivo que descreve a amostra de origem de cada sequência. Essa metodologia é aplicada através do programa Unifrac, que foi incorporado ao *pipeline* Qiime recentemente [43][51][52].

Qiime tem a possibilidade de ser paralelizado para adquirir um maior desempenho computacional. Entretanto, este *pipeline* não fornece ainda nenhuma etapa ou função paralelizada (memória compartilhada ou distribuída) e nenhum *script* que faça a quebra dos dados de entrada a serem analisados ou para submeter a carga de trabalho de forma distribuída em um *cluster*. Além disso, a sua forma de instalação, que exige uma máquina virtual personalizada, dificulta a adaptação deste pacote de ferramentas a um *cluster* que já possui seu próprio ambiente computacional.

2.7 PANGEA

No *pipeline* PANGEA [23], as ferramentas são distribuídas localmente, e não em um servidor web como a maioria das ferramentas de pós-processamento, o que permite ao usuário controlar e configurar seu próprio banco de dados, bem como instalar a ferramenta em um cluster computacional de alto desempenho. Todas as etapas são executadas por uma única linha de comando, mas, se necessário, o usuário pode verificar ou voltar em cada resultado referente a cada uma das etapas individualmente. Esta liberdade para controlar e analisar cada subtarefa, mas com automaticidade para executar uma análise completa com apenas uma instrução, é uma das vantagens do PANGEA. Além disso, as distribuições *open source* permitem aos usuários adaptar o programa para o seu próprio banco de dados e estender a análise acrescentando outras ferramentas ou funcionalidades.

PANGEA é uma das ferramentas mais completas entre as revisadas neste trabalho. Este *pipeline* consiste em um conjunto de programas *open source* que processam automaticamente grande parte das etapas de análise de dados metagenômicos. Além disso, o *pipeline* PANGEA pode ser executado em vários ambientes, como Mac OSX, Windows ou Linux.

As principais etapas do pipeline PANGEA são:

1. Filtragem sequências (Trim2.pl): Sequências muito pequenas são removidas e os segmentos de baixa qualidade de leitura são retirados. Esta função foi adaptado a partir do programa Trim2 [30]. O script também apaga informações desnecessárias, como o comprimento da sequência e e comentários sobre a tecnologia de sequenciamento.

2. Divisão por *barcodes (barcode.pl)*: *barcodes* são pequenos segmentos de DNA utilizados para identificar as amostras. Nesta etapa, os *barcodes* são procurados. Os arquivos de entrada no formato fasta são divididos de acordo com a sequência de *barcode* identificada, a fim de organizar e separar cada amostra por sua origem. O resultado será um

conjunto de arquivos fasta unidos por seus *barcodes* em comum. Os arquivos sem *barcodes* separados em um arquivo único diferente.

3. Preparação da base de dados ("formatdb"): Antes de executar a busca das sequências contra uma base de dados, a base de dados deve ser formatada de modo a ser reconhecida pelo programa de alinhamento de sequências, conhecido como BLAST (*Best Local Alignment Search Tool*) [1]. A base de dados suportada pelo BLAST é gerada utilizando o programa "formatdb", que prepara e revisa a sintaxe dos arquivos de entrada para executar a busca das sequências [60].

4. Alinhamento (Megablast): Para a classificação dos organismos presentes nas amostras, os arquivos contendo as sequências, após a separação de *barcodes,* são utilizados para executar o alinhamento Megablast, que é uma versão local do pacote de ferramentas *online* BLAST [1] para alinhamento múltiplo de sequências de DNA. O alinhamento com Megablast pode ser realizado contra arquivos preparados usando TaxCollector [24], que retorna informações adicionais e complementares sobre a classificação taxonômica dos organismos.

O arquivo gerado pelo programa Megablast contém o ID do alvo (nome da sequência de entrada), ID do resultado (identificação do organismo mais próximo encontrado na base de dados, em relação ao alvo), porcentagem de identidade entre o alvo e o resultado, o tamanho da sequência, número de bases desiguais no alinhamento (*miss matches*), número de lacunas inseridas entre as sequências para fazer o alinhamento, início e término da sequência alvo e resultado, valor da pontuação do alinhamento (*bitscore*).

5. Seleção das sequências não classificadas (unclassified selector): Após a execução da classificação pelo Megablast, as sequências não classificadas são separadas pelo *unclassified selector*.

6. Agrupamento de sequências não classificadas: esta etapa é fundamental, já que um dos principais objetivos de uma análise metagenômica é identificar novas espécies (não classificadas ainda). Essas sequências, obtidas no passo anterior são unidas e, em seguida, submetidas ao programa CD-HIT que as agrupas de acordo com sua relação evolutiva. O programa CD-HIT gera um conjunto de sequências não redundantes [49].

Alguns parâmetros podem ser especificados pelo usuário para esta classificação, como o limite de tamanho ou similaridade entre as sequências.

7. Geração de sumários para análise estatística (megaclust e megaclustable):

nesta etapa, os arquivos de saída do Megablast são interpretados e convertidos em um arquivo tabular contendo o agrupamento e a contagem de cada um dos níveis taxonômicos (*operational taxonomic unit* - OTU) encontrados. Esta contagem é feita com base em um limite de valor de identidade e no ID da sequência encontrada pela fase de busca por alinhamento múltiplo.

8. Geração de sumário de OTUs classificadas e não classificadas ("hybridtable"): esta etapa é realizada pela junção dos resultados das etapas 5 e 6.

9. Teste (Qui-quadrado): os resultados obtidos a partir do passo anterior (etapa 8) são testados para determinar se grupos específicos de organismos diferentes entre amostras diferentes (que geralmente são coletadas de ambientes diferentes). Neste passo, a ferramenta “Qui-quadrado” é executada após da etapa 8, automaticamente por um script em R. O teste de significância é realizado para obter um valor de P para a hipótese nula de que não houve diferença entre todos possíveis combinações de pares de amostras.

10. Normalização do número de sequências ("selector") para análise estatística: a normalização dos resultados é feita através da análise de biodiversidade. Esta pode ser realizada com a identificação da amostra com menor número de sequências e utilizando-a como um valor de corte. O programa “selector” realiza esta tarefa.

11. Análise estatística (índice de diversidade de Shannon): os resultados obtidos nas etapas anteriores pode ser avaliada com o cálculo do índice de biodiversidade. O índice de diversidade de Shannon é um entre vários índices de diversidade conhecidos e utilizados para medir a biodiversidade neste tipo de análises. Seus resultados proporcionam informações sobre a distribuição das espécies, tratando as espécies como símbolos e os seus tamanhos populacionais como a sua probabilidade relativa. Esta informação é útil quando se comparam ecossistemas e habitats

semelhantes [71]. O resultado é um estudo comparativo dos dados de biodiversidade de espécies obtidos a partir de ambientes diversos, além de uma rica fonte de dados em formato tabular, adequado para vários programas de análise estatística, tais como R, SPSS, etc [8][59][65].

Os 11 passos descritos descrevem as principais características do *pipeline* PANGEA: pré-processamento de dados (1, 2 e 7), preparação do banco de dados (3), classificação taxonômica das sequências de entrada (4, 5 e 6), tratamento de dados não classificados (6 e 8) e análise estatística (9, 10 e 11). Estes passos fazem PANGEA se tornar um *pipeline* completo com as principais vantagens quando comparado com algumas das principais ferramentas disponíveis atualmente para pós-processamento metagenômico.

2.8 Análise comparativa dos pipelines estudados

Os recursos disponibilizados por cada um dos *pipelines* foram avaliados neste trabalho. Foram comparadas as principais funcionalidades de cada um, suas vantagens, desvantagens e principalmente seu nível de suporte em relação as principais fases de pós-processamento de uma análise metagenômica atualmente. Um resumo desta comparação pode ser observados na Tabela 1. Este estudo comparativo nos permite visualizar as principais características e recursos de cada uma das ferramentas descritas no presente trabalho.

Entre os recursos compreendidos pela maioria das ferramentas disponíveis estão a avaliação e filtragem das sequências (item 2 da Tabela 1), que corresponde a um dos passos iniciais do processo de análise, onde as sequências muito pequenas e/ou com baixa qualidade de leitura são excluídas para a execução dos próximos passos. Algumas etapas críticas para a interpretação dos resultados também fazem parte da maioria das ferramentas analisadas, como a classificação de espécies (item 3), o pós-processamento destes resultados (item 5) e a análise estatística dos dados de classificação obtidos (item 6).

Tabela 1. Estudo comparativo entre os recurso disponibilizados por pipelines metagenômicos.

Etapa ou Recurso	Presença de Suporte						
	MEGAN	Mothur	Galaxy	RDP	RAST	Qiime	PANGEA
1. Suporte a vários formatos de arquivos (tecnologias de sequenciamento diferentes)							
2. Avaliação e filtragem das sequências por qualidade		■	■	■	■	■	■
3. Classificação de espécies contra uma base de dados	■		■	■	■	■	■
4. Comparação entre os métodos de classificação diferentes (consenso)							
5. Pós-processamento dos resultados de classificação		■	■	■	■	■	■
6. Análise estatística para organismos classificados		■	■	■		■	■
7. Análise estatística para organismos não classificados				■			■
8. Sumarização dos resultados			■	■	■		■
9. Análise filogenética	■	■	■	■	■	■	
10. Análise funcional	■		■		■	■	
11. Memória compartilhada	■	■	■				■
12. Memória distribuída (troca de mensagens)							
13. Versão local	■	■				■	■
14. Versão Web			■	■	■		

Entretanto, algumas fases também importantes para pós-processamento não estão presentes na maioria das ferramentas, como a análise estatística dos organismos não classificados (item 7) e análise funcional (item 10). E principalmente, podemos ver que existem três grandes necessidades, que correspondem a funcionalidades ou recursos não encontrados entre nenhum dos *pipelines* disponíveis atualmente (Tabela 1):

(1) Suporte a vários formatos de arquivos (item 1), provenientes de tecnologias de sequenciamento diferentes, tais como FASTA, FASTQ, QSEQ, entre outros. O formato de arquivos padrão aceito pelo pipelines

geralmente corresponde ao formato FASTA. Um maior suporte a formatos e sintaxes diferentes de arquivos diminuiria a intervenção do usuário nos passos iniciais das análises metagenômicas, não havendo mais a necessidade de realizar a conversão destes arquivos.

(2) Comparação entre métodos de classificação diferentes (item 4). Todos os *pipelines* revisados neste trabalho se baseiam em apenas um método de classificação de espécies, quando realizam a etapa de busca de sequências contra uma ou mais bases de dados. O método mais frequentemente utilizado é o MegaBLAST, seguido pelo RDP Classifier. Entretanto confiar em apenas um método de classificação pode prejudicar a qualidade e confiabilidade do processo de análise.

(3) Uso de memória distribuída (12). Entre os *pipelines* avaliados, não foram encontradas grandes otimizações para as suas etapas mais complexas e demoradas (demonstradas na figura 2). Um exemplo é o uso de otimizações de memória distribuída (como as bibliotecas MPI [26][27][66]) para subdividir as tarefas de análise entre nodos de um *cluster*. Esta otimização não foi encontrada em nenhuma das ferramentas revisadas neste trabalho.

Com base nestes motivos, podemos estabelecer os principais objetivos do trabalho.

3 PANGEA+

Nesta seção são descritas as otimizações propostas e desenvolvidas ao longo do presente trabalho.

3.1 Motivação e objetivos

Considerando as principais necessidades de otimizações para pós processamento em análises metagenômicas, os principais objetivos do presente trabalho consistem em:

(1) Identificar quais são as necessidades mais críticas, quanto a melhoramentos qualitativos (precisão, confiabilidade) e quantitativos (desempenho computacional) entre as ferramentas de análise existentes;

(2) Selecionar um *pipeline* entre os mais completos disponíveis;

(3) Desenvolver otimizações qualitativas e quantitativas que preencham as necessidades identificadas e incorporá-las no *pipeline* selecionado; e

(4) Validar as otimizações desenvolvidas comparando-as com outras ferramentas.

As presentes otimizações e avaliações de desempenho propostas são focadas em algumas das etapas mais complexas e demoradas de uma análise metagenômica: filtragem das sequencias de entrada, atribuição de classificação taxonômica e pós-processamento dos resultados da classificação. Estas, entre outras etapas, são realizadas pelos *pipelines* metagenômicos com diferentes níveis de automatização, praticidade, desempenho computacional e exatidão.

As otimizações qualitativas e quantitativas propostas foram implementadas e incorporadas no *pipeline* PANGEA, gerando uma nova versão denominada PANGEA+. Como mencionado na seção 2.6, PANGEA é um dos *pipelines* mais completos entre os avaliados neste estudo. Além disso, seu código é *open source*, implementado sob as linguagens Perl e

C++, de fácil acesso e adaptabilidade a diversos ambientes. Estes foram os principais motivos que levaram a seleção deste *pipeline* como base para desenvolver as otimizações propostas. Nas próximas subseções são discutidas as novas funcionalidades e otimizações que compõem a presente arquitetura desenvolvida. A Figura 3 mostra uma visão geral sobre as principais funções da presente arquitetura. Cada uma das funções demonstradas na Figura 3 serão detalhadas nas próximas seções.

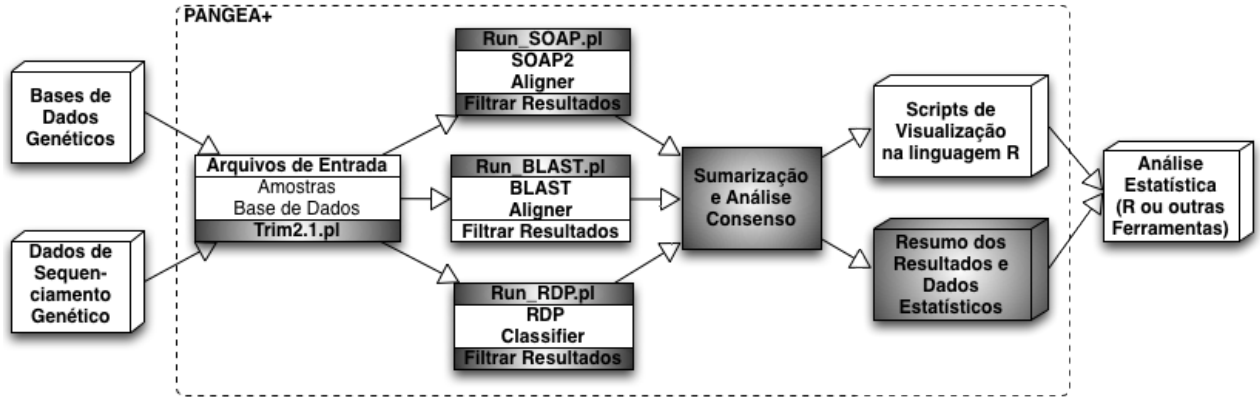

Figura 3. Visão geral da presente arquitetura.

3.2 Maior suporte a leitura e filtragem dos dados de entrada (Trim2)

O *pipeline* Pangea tem suporte apenas ao formato de arquivos FASTA como entrada. Um exemplo de arquivo no formato FASTA é apresentado na Tabela 2. Este suporte foi estendido no PANGEA+, para os formatos de arquivos provenientes de sequenciadores de nova geração: QSEQ e FASTQ. As alterações descritas nessa seção foram implementadas gerando uma nova versão do programa Trim2. O fluxograma para o novo Trim2 está descrito na Figura 4, onde os retângulos cinza representam os módulos introduzidos ou modificados na nova versão, seguidos da descrição de sua função. Os cilindros cinza representam os novos formatos de entrada suportados com a presente implementação. Os demais componentes em branco foram mantidos a partir do programa Trim2 original.

Tabela 2. Exemplo de arquivo no formato FASTA.

>gi|343806277|gb|JN037467.1| Escherichia coli strain ydiO gene GTCGGCAGATCCTGAAAGACTATCAGAACAAATAATCTGCAG GCGGCGCAGCTTCTTAACAAACTGCGCCGCCAGATTTATCCA ACAAGACTTACCGGTTGAGGAAATTC

Onde: “>” representa o início da linha com o cabeçalho de identificação da sequência (*label)*; e as linhas seguintes representam o código genético da sequência.

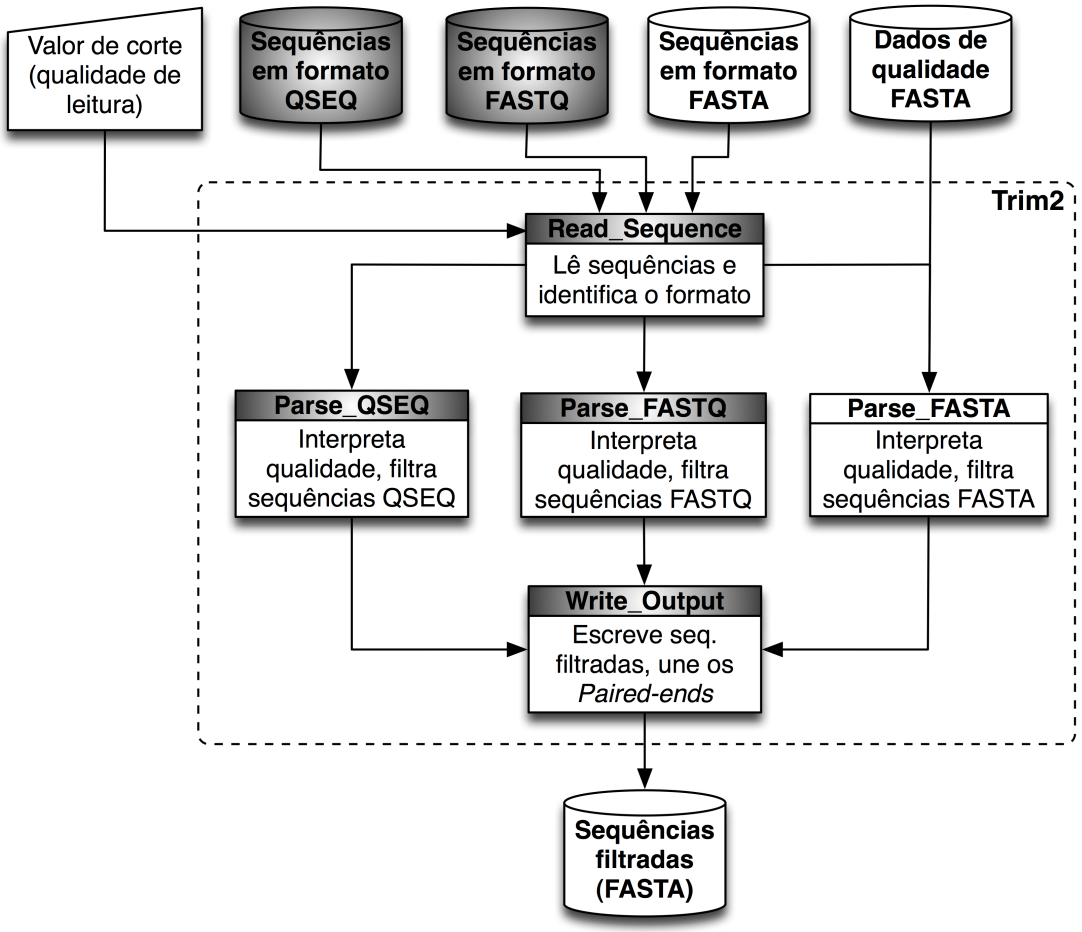

Figura 4. Fluxograma para o novo Trim2.

A saída gerada pelas máquinas de sequenciamento podem variar em muitos aspectos de formato e sintaxe, tornando os arquivos de sequências ilegíveis para o PANGEA. Um exemplo é o formato de sequenciamento em *Paired-ends* [34]. *Paired-ends* são sequências geradas pela tecnologia de sequenciamento Illumina, que são basicamente divididas em dois

fragmentos. A saída da máquina de sequenciamento é separada em dois arquivos, onde o primeiro arquivo (Sequência A) contém o início da sequência e segundo arquivo contém o fim da sequência (Sequência B), como no exemplo:

Sequência A LacunaSequência B

ATTTCGCGCTAA…_____GCGGTATACAT…

Cada um dos arquivos (Sequência A e Sequência B) geralmente são gerados no formato QSEQ, muito diferente e mais informativo do que o formato FASTA. Um exemplo de um destes arquivos pode ser observado na Tabela 3.

Tabela 3. Exemplo de arquivo no formato QSEQ.

HWUSI-EAS163FR132118147141501 G.GTGCCAGCAGCCGCGGTAATACAGAGGGTGCAAGCGTTAATCGGAAT TACTGGGCGTAAAGCGCGCGTAGGTGGTTCGTTAAGTTGGATGTGAAAT CCC aBa\aeddeegggggggggggggggc]ffWaccfcecfdeIT\[Yf`]ccdbdcdfdgdfadd_eQa aaa]aaM`^\\Q^S_\X_\\S^\\\YYL__\OP1

Onde cada entrada armazenada na primeira linha representa os dados sobre a máquina de sequenciamento e sua execução (identificador da máquina, número da canaleta, etc); nas linhas seguintes é definida a sequência genética (A, C, T, G ou “.” para sequências não identificadas). Por último, segue a informação de qualidade em um nível escalar, proporcional a um parâmetro de calibragem.

Esta mudança no formato de entrada causa problemas nas etapas de análise seguintes do PANGEA, uma vez que a distância e relação entre os pares dessas extremidades não são reconhecidas automaticamente pelo método de classificação de espécies utilizado pelo PANGEA (BLAST). Na presente versão do PANGEA+, esta limitação foi superada. A ferramenta de filtragem das foi modificada, incluindo suporte aos arquivos QSEQS e a opção de suporte a *Paired-ends*. Nesta otimização, o programa de filtragem (Trim2) reconhece cada par de sequências dos *Paired-ends* e os une em um único arquivo (função Wirte_Output), inserindo uma lacuna de bases

nucleotídicas não identificadas (representado por N, que significa a mesma probabilidade para as 4 bases nucleotídicas, A, C, T ou G). Junto a união das sequências, é aplicada uma função que converte os símbolos de qualidade para valores escalares (função Parse_QSEQ), e os converte para uma unidade de qualidade conhecida, no formato Phred [20][21]. Finalmente é gerado um arquivo no formato FASTA (função Write_Output) convertido a partir dos dois arquivos QSEQS filtrados por qualidade, com base um valor de corte fornecido pelo usuário. Um exemplo de saída da nova versão do programa Trim2 pode ser observada na Tabela 4.

Tabela 4. Exemplo de resultado da etapa de leitura, conversão e filtragem de arquivos no formato QSEQ.

>HWUSI-EAS163FR13211814714150SeqAB GCCGCGGTAATACAGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGT AAAGCGCGCGTAGGTGGTTCGTTAAGTTGGATGTGCCNNNNNNNNNNN NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN NNNNAGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCGC GTAGGTGGTTCGTTAAGTTGGATGTGA

Onde: “>” representa o início da linha com o cabeçalho de identificação da sequência (*label)*; e as linhas seguintes representam o código genético da sequência.

Além do formato QSEQ, outro formato de arquivo, conhecido como FASTQ é gerado a partir de algumas máquinas de sequenciamento de larga escala, como os sequenciadores Solexa. Estes apresentam uma formatação de sintaxe ainda mais diferente (exemplo descrito na Tabela 4). Estas mudanças no formato FASTQ torna o PANGEA incapaz de reconhecer estes arquivos, até mesmo no primeiro passo de pré-processamento que ocorre antes de executar o passo de classificação espécies. Esta limitação também foi superada na nova versão do PANGEA+, com a inclusão de suporte a arquivos FASTQ na função de leitura e filtragem de sequências (Parse_FASTQ). A saída gerada consiste em um arquivo no formato FASTA, já filtrado pelos dados de qualidade com base em um valor de corte fornecido pelo usuário. Assim como no formato QSEQ, para a filtragem por qualidade é utilizada uma função que converte os símbolos em valores escalares no formato conhecido como qualidade Phred [20][21].

Tabela 5. Exemplo de arquivo no formato FASTQ.

@EAS54_6_R1_2_1_413_324
CCCTTCTTGTCTTCAGCGTTTCTCC
+
;;3;;;;;;;;;;;;7;;;;;;;88

Onde: a primeira linha representa o identificador da sequência (como o *label* da sequência FASTA); a segunda linha representa a sequência genética; A terceira linha representa um identificador adicional da sequência (opcional); e a quarta linha codifica os valores de qualidade para a sequência da segunda linha, cada símbolo representando um valor de qualidade de ordem escalar com base em um fator de calibragem como referência.

3.3 Classificação de espécies

Nesta seção é descrito o fluxo de trabalho das otimizações introduzidas na presente arquitetura, que envolvem a etapa de classificação de espécies da análise metagenômica.

3.3.1 Inclusão de novos métodos de classificação (Classify)

A versão anterior do PANGEA possui suporte para apenas um algoritmo de classificação de espécies, conhecido como BLAST [9][28]. BLAST (*Best Local Alignment Search Tool*) é uma ferramenta para pesquisar o melhor alinhamento entre sequências a fim de identificar as regiões de semelhança entre elas. Geralmente as pesquisas de similaridade são realizadas contra uma base de dados de sequências classificadas. O algoritmo do BLAST envolve os seguintes passos: (1) remover regiões repetidas e de baixa complexidade das sequências de entrada; (2) subdividir a sequência de entrada em uma lista de palavras menores; (3) buscar por cada uma destas palavras na base de dados; (4) organizar uma árvore hierárquica de busca com base nas pontuações de similaridade entre as palavras da sequência e da base de dados; (5) estender as palavras,

recalculando o grau de similaridade; e (7) calcular e avaliar a significância desta similaridade. Após o alinhamento, a sequência classificada que apresentar a maior similaridade com a sequência de busca indicará que esta pertence ao seu grupo taxonômico. Os resultados são filtrados por testes de significância estatística.

BLAST tem demonstrado ser eficiente em muitos estudos [44][74][77]. No entanto, existem vários programas com métodos e resultados diferentes, além do BLAST (ex.: SOAP2 Aligner [48], RDP Classifier [13], Infernal Aligner [61], NAST [18], Bowtie [45], Velvet [81], MAQ [47], entre outros). Dependendo do tipo de base de dados e das sequências de entrada utilizadas, os resultados de cada método podem variarem nível de exatidão e eficiência. Além disso, cada um destes programas utiliza uma estratégia diferente para classificar a sequência de entrada. SOAP2, por exemplo, é baseado em índices de compactação para a indexação rápida de sequência de referência na memória principal [48]. É muito rápido, sendo capaz de classificar um milhão de leituras em três minutos (~30 vezes mais rápido do que BLAST). Entretanto, o SOAP2 é limitado a tamanhos pequenos de sequências (aproximadamente ~1000 bases nucleotídicas).

Outra ferramenta, conhecida como RDP Classifier, que inicialmente se baseou no Infernal Aligner [61], consiste de um método especial com a implementação gramáticas livres de contexto para traçar perfis estocásticos. Este método é conhecido como modelo de covariância [72][78]. Estes modelos se baseiam em um grupo de dados experimentais (*Training set*) disponibilizado pela própria base de dados da ferramenta. Sendo um algoritmo estocástico, pode ser mais rápido do que outros métodos, como a busca hierárquica feita pelo BLAST. No entanto, sua precisão pode ser comprometida pela sua natureza de ser uma aproximação baseada em um modelo. Os algoritmos citados são um exemplo de estratégias diferentes para classificar as espécies, os quais apresentam vantagens e desvantagens. Confiar em apenas um método pode levar a resultados não tão exatos, sendo que estes podem depender da eficiência da estratégia de classificação para cada tipo ou padrão de base de dados e da sequência de entrada.

Com o objetivo de superar essa limitação, os programas de classificação SOAP2 Aligner e RDP Classifier foram incluídos no PANGEA+. Esta mudança em relação ao anterior PANGEA, que apenas utilizava o BLAST, aumenta a abrangência da etapa de classificação e tem potencial de contribuir qualitativamente, na confiabilidade dos resultados. Além disso, no PANGEA+ uma nova versão do BLAST, conhecida como BLAST+, foi substituída pela versão anterior utilizada no PANGEA. Esta versão possui otimizações de menor complexidade que melhoraram a eficiência e diminuem o tempo de execução do programa [6]. Para automatizar a etapa de classificação, o *script* Classify foi desenvolvido para realizar a execução dos três métodos (BLAST+, SOAP2, RDP Aligner). O Classify executa a preparação da base de dados para os programas BLAST+ e SOAP2, utilizando as ferramentas formatdb [6] e 2bwt-builder [48]. Após a preparação, o executa os três métodos de classificação, especificando como entrada as sequências filtradas pela ferramenta Trim2 e as bases de dados preparadas. Para a execução do RDP Classifier, o seu próprio *Training set* é utilizado. O fluxograma para a etapa de classificação com o *script* Classify pode ser visualizado na Figura 5. Estão representados em cinza os componentes que foram introduzidos ou modificados no PANGEA+.

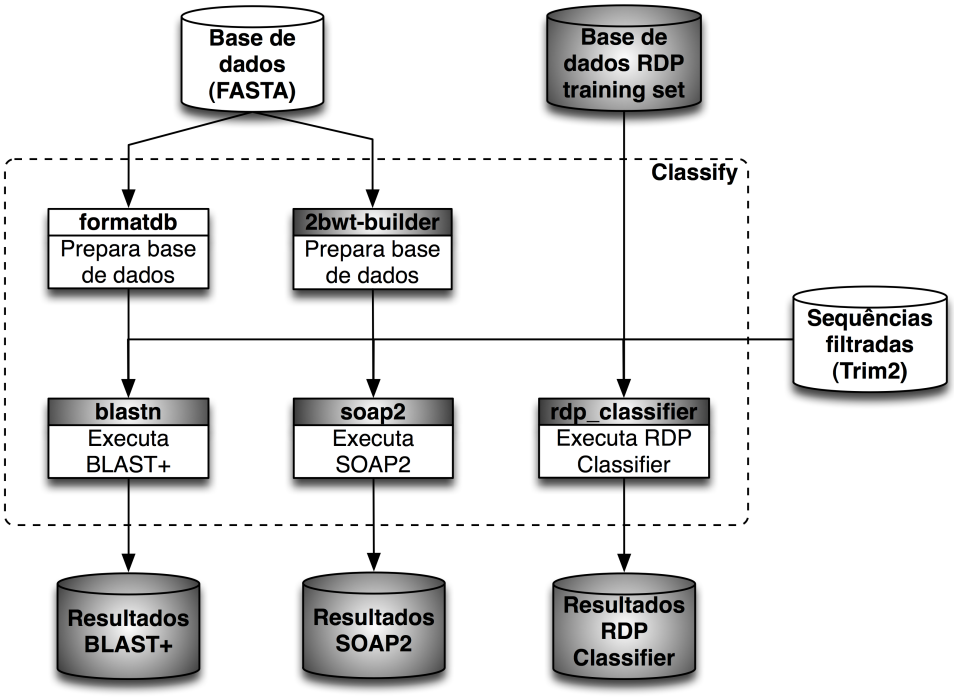

Figura 5. Fluxograma da etapa de classificação utilizando o *script* Classify.

Com este avanços no PANGEA+, o usuário é capaz de escolher e comparar três metodologias de classificação de espécies que utilizam abordagens diferentes. Se todas as estratégias (ou a maior parte) de classificação retornarem os mesmos resultados, estes podem expressar uma maior confiabilidade ou qualidade, mostrando um consenso entre eles. Por outro lado, se cada estratégia de classificação variar muito em seus resultados, o pesquisador pode considerar os resultados como inexatos. Desta maneira, apenas as saídas que são altamente similares ou idênticas entre métodos seriam consideradas como resultados mais confiáveis.

Como resultado, cada um dos métodos gera arquivos de texto em formato tabular. Estes arquivos contém as informações da sequência identificada e da qualidade ou grau de certeza desta classificação. Um exemplo dos resultados de cada um dos três métodos de classificação pode ser visto na Tabela 6. Em todos os resultados, “S000381745” representa o nome da sequência de entrada de exemplo e as demais informações representam os resultados da classificação e dados sobre sua qualidade. No exemplo da Tabela 6, foi realizada uma busca contra a base de dados NCBI.

Tabela 6. Exemplos de resultados para cada um dos métodos de classificação adotados no PANGEA+.

SOAP2
S000381745 ACGCTGGCGGCAGG… hhhhhhhhhhhhhhhhhhhh... 1 a 1437+ gi|343530241|gb|HQ204295.2| 413M 229C143G25C10G11A17T13C336T47A112GG12C 37G177CC240GCGC9
BLAST+
S000381745 gi|343530241|gb|HQ204295.2| 98.82 1438 16 1 1 438 1 1437 0.0 2562
S000381745 gi|309261195|gb|HQ246280.1| 97.93 1400 21 7 2 8 1422 1 1397 0.0 2418
S000381745 gi|270311616|gb|GU190190.1| 97.95 1270 22 4 5 5 1322 1 1268 0.0 2198
S000381745 gi|309261165|gb|HQ246250.1| 91.49 1398 86 32 2 8 1407 3 1385 0.0 1892
S000381745 gi|85001890|gb|DQ337072.1| 90.21 1389 101 34 6 7 1438 7 61446 0.0 1781
RDP Classifier
S000381745 Bacteria domain 1.0 "Proteobacteria" phylum 1.0 Gammaproteobacteria class 1.0 "Enterobacteriales" order 1.0 Enterobacteriaceae family 1.0 Enterobacter genus 0.94

No formato de saída do BLAST+ [6] e SOAP2 [48], a informação "gi|343530241|gb|HQ204295.2|" representa a sequência encontrada, com o código de identificação (TAXID ou GI) do NCBI/Genbank. Já no programa RDP Classifier, a informação já é gerada com os dados de classificação obtidos para seis níveis taxonômicos, desde domínio até gênero. Após a classificação de espécies, o próximo passo em uma análise metagenômica consiste na conversão dos dados de identificação encontrados para uma informação mais fácil de ser interpretada. As otimizações desenvolvidas para a próxima etapa estão descritas na seção seguinte.

3.3.2 Pós-processamento automático dos dados de classificação (NCBI-TaxCollector)

Como no exemplo descrito na Tabela 6, o resultado das classificações obtidas com o BLAST+ e SOAP2 devem ser convertidos para uma classificação taxonômica, similar ao formato de saída do RDP. Para poder interpretar esses resultados, o usuário deve procurar o número de identificação do resultado da classificação (ex.: "gi|343530241|gb|HQ204295.2|") em bases de dados taxonômicas, como NCBI [22], Greengenes [17] ou RDP *Taxonomy* [53]. Esse procedimento é conhecido como anexo ou atribuição taxonômica.

Atualmente, considerando que uma análise metagenômica pode ter centenas de milhares de sequências de entrada, a atribuição taxonômica manual torna-se impraticável. Com o objetivo de lidar com este problema, a ferramenta TaxCollector foi desenvolvida por Giongo et al. [24]. TaxCollector é capaz de unir informações taxonômicas sobre 16S rRNA dos bancos de dados RDP e Greengenes. A abordagem do TaxCollector envolve o carregamento de bancos de dados taxonômicos do NCBI (arquivos names.dmp e nodes.dmp provenientes da base de dados taxdump.tar.gz [35]) e sua conversão em estruturas de dicionário na linguagem Python. Os bancos de dados taxonômicos inteiros são carregados na memória principal (RAM) do computador (cerca de 4 GB). No entanto, considerando que bancos de dados são atualizados a cada quinze dias, e aumentaram seus

tamanhos, o carregamento de dados na memória RAM não parece ser a abordagem mais adequada.

Uma outra limitação está em os scripts do TaxCollector suportarem apenas os bancos de dados RDP e Greengenes, não lidando com a maior base de dados genética disponível, a base de dados NCBI nt / nr [22]. A inclusão deste suporte implicaria em uma maior complexidade do algoritmo, acrescentando uma nova base de dados a ser consultada. Esta base de dados, conhecida como GI vs TAXID (gi_taxid_nucl.dmp) está disponível na base de dados NCBI *Taxonomy* e contém números de identificação do GI, no GenBank [3], convertidas em seus correspondentes números TAXID, acompanhados bases de dados de árvores taxonômicas do NCBI (nodes.dmp) e nomes de classificação (names.dmp) [35]. Este carregamento de bancos de dados em memória RAM (cerca de 4 GB) e novas consultas múltiplas causam sobrecarga no TaxCollector e prejudicam o seu desempenho, tornando impossível executá-lo com os tamanhos de dados comumente encontrados em uma análise metagenômica. Considerando esta limitação no pós-processamento dos resultados do BLAST, propomos um novo algoritmo de menor complexidade e maior desempenho para a atribuição de níveis taxonômicos com suporte ao NCBI.

O presente algoritmo, denominado NCBI-TaxCollector, emprega uma estratégia de programação diferente a do TaxCollector proposto por Giongo *et al.* [24]. Na presente implementação é aplicada uma busca direta sem carregar os arquivos inteiros em memória. Esta busca foi projetada para maximizar o desempenho e permitir a realização de inúmeras pesquisas por segundo. Como dado de entrada, o NCBI-TaxCollector carrega um valor GI (Genbank ID) a partir dos resultados do BLAST+/SOAP2 e os arquivos de bases de dados do NCBI (gi_taxid_nucl.dmp, nodes.dmp e names.dmp). O arquivo gi_taxid_nucl.dmp mapeia os números GI e obtém as identificações correspondentes ao NCBI (Taxon ID). O arquivo nodes.dmp tem uma estrutura de nodos filho-pai, que é responsável por estruturar os níveis taxonômicos. O arquivo names.dmp contém uma lista com um ou mais nomes taxonômicos para cada Taxon ID do NCBI. Assim, o algoritmo consiste em quatro etapas (demonstrado na Figura 6):

(1) A função Parse_taxonomy faz a tradução do valor GI à sua identificação correspondente do NCBI (Taxon ID);

(3) Get_name encontra os nomes disponíveis para cada nível taxonômico;

(4) Get_uptaxa busca na lista de nodos o nível do nodo pai;

(5) As etapas 3 e 4 são repetidas até chegar no nodo pai correspondente ao nível taxonômico de Domínio;

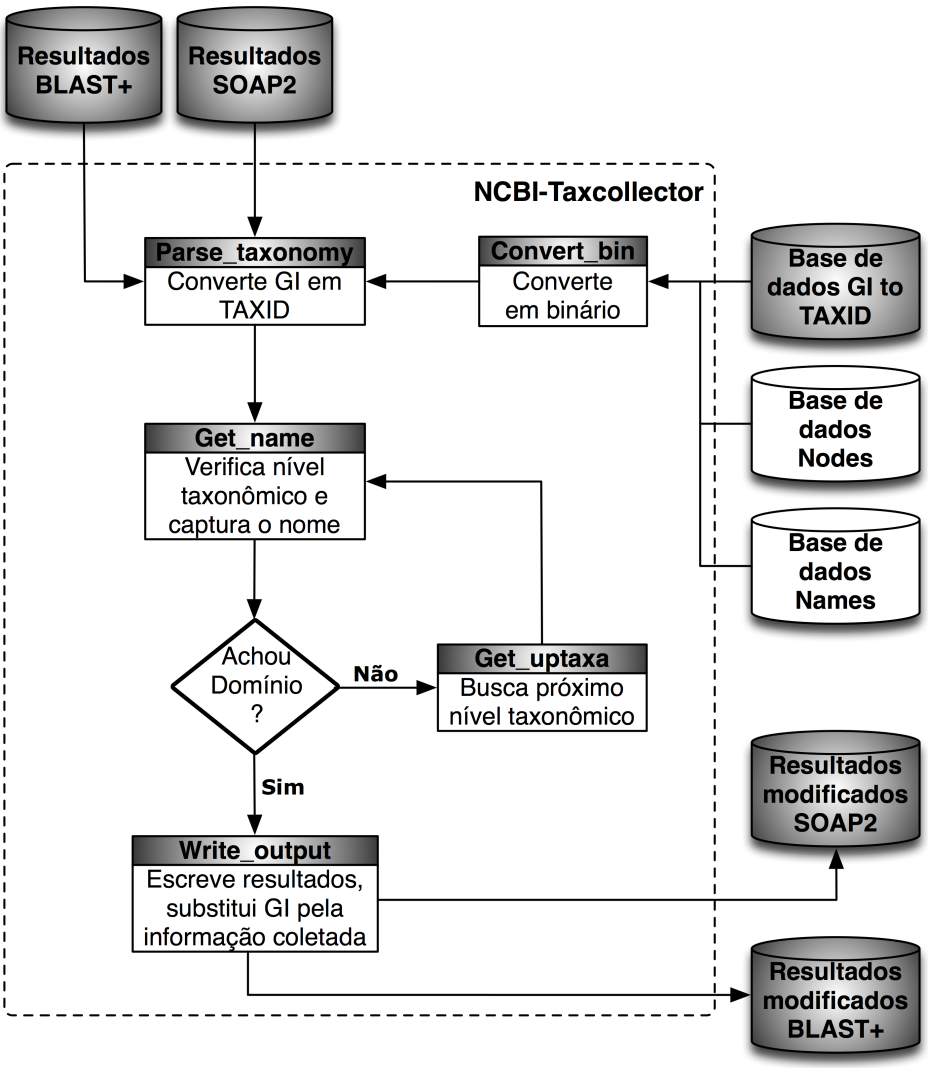

Figura 6. Algoritmo do NCBI-TaxCollector.

O NCBI-TaxCollector é executado como uma ferramenta de linha de comando e, por esta razão, é especialmente adequado para utilização em scripts. Antes da execução do programa, a função Convert_bin converte os bancos de dados do NCBI, originalmente no formato ASCII, para um formato binário. Este formato é otimizado para pesquisas rápidas. Esta conversão

não só acelera a busca, mas também reduz o tamanho do arquivo NCBI. Por exemplo, observamos uma redução de até 54% do tamanho total das bases em nossos experimentos.

Uma vez que o banco de dados otimizado é gerado, a tradução de um GI para um Taxon ID do NCBI é realizada com uma complexidade computacional muito pequena, conhecida como O(1) em notação algorítmica [4], isto é, é independente do número de entradas NCBI. Como o tamanho desses bancos de dados está aumentando constantemente, esta é uma característica chave para as ferramentas de atribuição taxonômica. A busca de nomes taxonômicos é realizada utilizando uma abordagem de pesquisa binária que tem uma maior complexidade computacional de O (log n), onde n é o número de entradas na base de dados. Como resultado, o programa reescreve a entrada substituindo a identificação da sequência classificada por um sumário de seus dados taxonômicos. Esse sumário consiste em seis níveis de classificação, que vão de Domínio até Espécie. A Tabela 7 descreve um exemplo de entrada e saída de NCBI-TaxCollector.

Tabela 7. Exemplos de entrada e saída de NCBI-TaxCollector.

Entrada (resultados do BLAST+ ou SOAP)
Query1 gi|309261160|gb|HQ246245.1| 81.87 1186 148 64 226 1375
Query2 gi|85001879|gb|DQ337061.1| 79.47 1554 166 122 9 1464 1
Query3 gi|319992851|emb|FR729081.1| 82.33 928 124 40 446 1352
Query4 gi|309261157|gb|HQ246242.1| 81.97 976 112 61 428 1371 401
Saída (Resultados do BLAST/SOAP modificados pelo NCBI-TaxCollector)
Query1 [0]Bacteria;[1]Proteobacteria;[2]Alphaproteobacteria;[3]Rhodospirillales; [4]Acetobacteraceae;[5]Roseomonas;[6]Roseomonas_sp._6A18S6; 81.87 1186 148 64 226 1375 128 1282 0.0 937
Query2 [0]Bacteria;[5]uncultured_bacterium;[6]uncultured_bacterium; 79.47 1554 166 122 9 1464 1 1499 0.0 961
Query3 [0]Bacteria;[1]Proteobacteria;[2]Deltaproteobacteria; [5]uncultured_delta_proteobacterium;[6]uncultured_delta_proteobacterium; 82.33 928 124 40 446 1352 1151023 0.0 769
Query4 [0]Bacteria;[1]Proteobacteria;[2]Gammaproteobacteria;[3]Pseudomonadales; [4]Pseudomonadaceae;[5]Pseudomonas;[6]Pseudomonas_sp._6A18S4; 81.97 976 112 61 428 1371 401 1344 0.0 769

3.4 Pós-processamento por geração de consenso (Consensus)

O uso de vários métodos de classificação de espécies incrementa a complexidade da análise dos resultados. Isso implica na comparação entre os resultados de cada um dos métodos com o objetivo de chegar a um resultado em comum. Se todos os métodos (ou a maior parte) de classificação retornarem os mesmos resultados, estes podem expressar uma maior confiabilidade ou qualidade, mostrando um consenso entre eles. Entretanto, até o momento nenhum *pipeline* entre os avaliados aplica um método automático para se chegar a esse consenso, pois estes se baseiam em apenas um método ou estratégia de classificação.

Para superar essa limitação, foi desenvolvido a ferramenta Consensus, que avalia os resultados de cada um dos métodos de classificação e gera como resultado um valor consenso entre estes. Considerando que o programa mais utilizado e considerado como um dos métodos mais confiáveis para a classificação é o BLAST, Consensus o considera como prioridade em sua avaliação. Sendo assim, o programa está dividido nas seguintes etapas (Figura 7): (1) Inicialmente, a função *Parse_RS* interpreta os resultados únicos obtidos da classificação pelo RDP e/ou SOAP; (2) Em seguida, a função *Consensus* lê cada um dos melhores resultados para cada uma das classificações obtidas pelo BLAST (40 *top hits*, funcionalidade anteriormente ausente no PANGEA) e compara essa classificação com os demais métodos; (3) quando os resultados apresentam classificações iguais, então esses resultados são armazenados (Resultado consenso). Do contrário, os resultados do BLAST são considerados como prioridade e o melhor resultado de cada *top hits* é armazenado.

Desta forma, a função Consensus percorre todos os resultados do BLAST, comparando cada um destes com os demais métodos de classificação utilizados. Essa comparação é feita para cada nível taxonômico, desde domínio até espécie. Em caso de empate, ou seja, dois ou mais resultados com o mesmo número de níveis taxonômicos iguais, o programa seleciona os resultados do BLAST com maior similaridade como prioridade. Se ainda houver empate na similaridade, o critério de seleção

segue para o menor valor do *e-value*, que representa o grau de probabilidade deste resultado ter sido encontrado por acaso ou coincidência. Se ainda há empate, então é selecionado o maior *bitscore*, que é uma pontuação que reflete a qualidade do alinhamento.

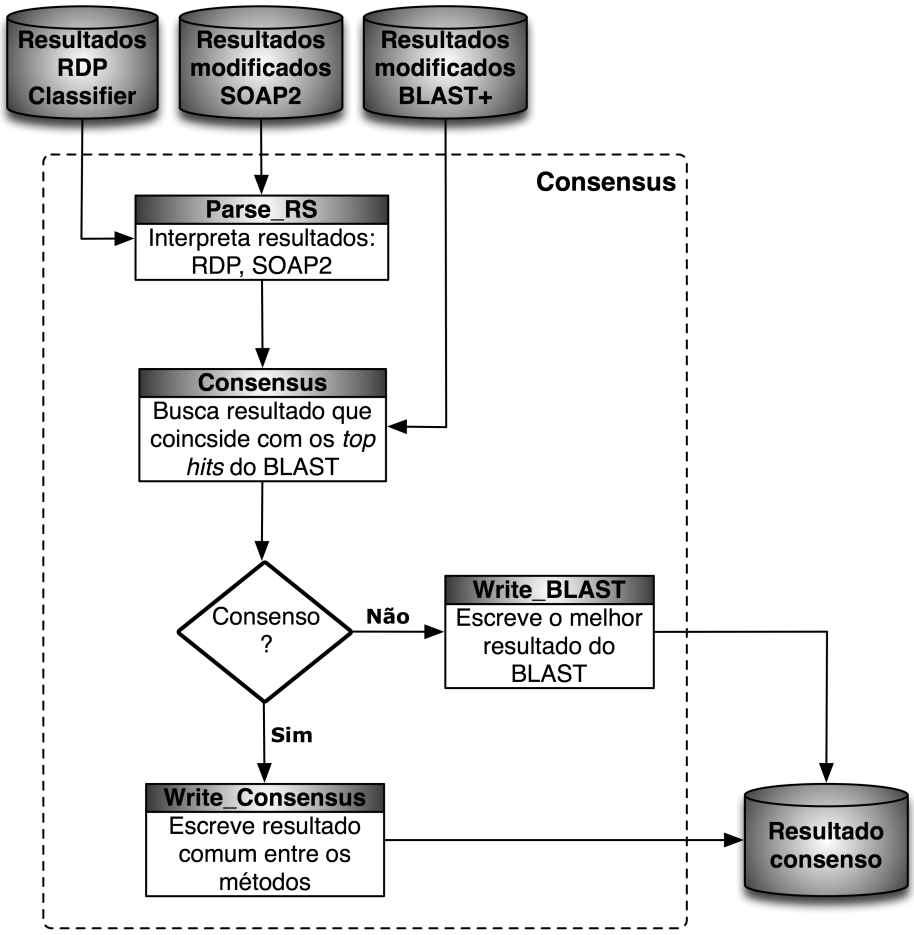

Figura 7. Filtragem dos resultados chegando a um resultado comum entre pelo menos 2 métodos de classificação, dando prioridade ao BLAST.

Sem a utilização da ferramenta Consensus, o usuário seria limitado a comparar manualmente cada um dos resultados. Considerando que atualmente os resultados podem apresentar centenas de milhares de linhas, seria impossível realizar essa tarefa manualmente para todos os resultados de classificação de espécies. Além disso, futuramente o programa Consensus pode ser expandido para mais métodos de classificação, sendo um algoritmo de comparação de *strings* de baixa complexidade. O programa foi desenvolvido em linguagem Perl, por ser uma linguagem adaptada para o

processamento e manipulação de arquivos de texto. Os resultados do Consensus são gerados com a mesma formatação dos resultados do NCBI-TaxCollector. Um exemplo de dados de entrada e de resultados da execução da ferramenta Consensus é descrito na Tabela 8. Os dados de saída da análise consenso são gerados no mesmo formato que os dados de saída do BLAST+. Uma linha de comentário é acrescentada abaixo de cada linha de resultado consenso: *#Matches found: X*, onde # representa o início da linha de comentário e X é o número de níveis taxonômicos (desde domínio até espécie) encontrados em comum entre o BLAST+ e o RDP Classifier ou SOAP2 Aligner. Entretanto, para o consenso entre BLAST+ e RDP Classifier, as comparações são feitas apenas até o nível de gênero, pois o RDP Classifier não possui suporte para a identificação de sequências a nível de espécie.

Tabela 8. Exemplo de dados de entrada para função Classify e seus resultados.

Resultados do BLAST (*top hits*) modificados pelo NCBI-TaxCollector
S000381745 [0]Bacteria;[5]bacterium_YX118S;[6]bacterium_YX118S; 98.82 1438 16 1 1 1438 1 1437 0.0 2562 S000381745 [0]Bacteria;[1]Proteobacteria;[2]Gammaproteobacteria;[3]Enterobacteriales; [4]Enterobacteriaceae;[5]Enterobacter;[6]Enterobacter_sp._7A18S4; 97.93 1400 21 7 28 1422 1 1397 0.0 2418 S000381745[0]Bacteria;[1]Proteobacteria;[2]Gammaproteobacteria;[3]Vibrionales; [4]Vibrionaceae;[5]Vibrio;[6]Vibrio_sp._6A18S2; 91.49 1398 8 6 32 28 1407 3 1385 0.01 892 . . . S000381745 [0]Bacteria;[5]uncultured_bacterium;[6]uncultured_bacterium; 83.07 1459 161 76 1 1415 26 1442 0.0 1247
Resultados do RDP
S000381745 Bacteriadomain 1.0 "Proteobacteria" phylum 1.0 Gammaproteobacteria class 1.0 "Enterobacteriales" order 1.0 Enterobacteriaceae family 1.0 Enterobacter genus 0.94
Resultados do SOAP2 modificados pelo NCBI-TaxCollector
S000381745 [0]Bacteria;[5]bacterium_YX118S;[6]bacterium_YX118S; 10413M 229C143G25C10G11A17T13C336T47A112GG12C37G177CC240GCGC9
Resultados do Consensus
S000381745 [0]Bacteria;[1]Proteobacteria;[2]Gammaproteobacteria;[3]Enterobacteriales; [4]Enterobacteriaceae;[5]Enterobacter;[6]Enterobacter_sp._7A18S4; 97.93 1400 21 7 28 1422 1 1397 0.0 2418 #Matches found: 6

Além da necessidade de otimizações qualitativas, como o Consensus e o NCBI-TaxCollector, uma demanda de otimização quantitativa foi identificada. Entre os três métodos de classificação utilizados no presente trabalho, os programas SOAP2 [48] e RDP Classifier [13] apresentam um tempo de execução muito menor do que o BLAST+ (com tempo sequencial de execução cerca de 20-30 vezes mais rápidos que o tempo sequencial do BLAST+). Além disso, nenhum dos *pipelines* fez otimizações para poder utilizar mais de um nodo computacional de um cluster, por exemplo, utilizando apenas a versão padrão atual (BLAST+ [6], implementado para utilizar multi-núcleos com Pthreads). Assim, foi identificada a necessidade de melhorar o desempenho do BLAST+ expandindo seu processamento para vários nodos de um cluster, na tentativa de melhorar consideravelmente seu desempenho. Na próxima seção será descrita a implementação de uma nova versão paralela do BLAST+, denominada MPI-blastn, para a classificação de espécies em análises metagenômicas.

3.5 Otimização de desempenho com MPI-blastn

Muitas otimizações de alto desempenho tem sido implementadas para o BLAST. Por exemplo, a versão BLAST+ possui otimizações de menor complexidade e suporte *multi-threading* [6]. O uso de memória compartilhada por *multi-threading* no BLAST+ possibilita ao usuário explorar vários núcleos de uma arquitetura de multiprocessador. Contudo, o BLAST+ ainda não é capaz de executar me mais de um nodo de um *cluster* de computadores em paralelo, limitando a escalabilidade desta ferramenta em tais ambientes.

Com o objetivo de lidar com essa limitação do BLAST, versões distribuídas deste programa foram desenvolvidas utilizando várias estratégias de programação. Um exemplo é a técnica MapReduce [16] utilizada no CloudBLAST [54], que combina máquinas virtuais e tecnologias de redes virtuais para distribuir a base de dados do NCBI e as sequências de entrada, executando o BLAST em paralelo através de uma estratégia de

programação mestre-escravo. Contudo, a implementação de MapReduce em *clusters* necessitam armazenamentos locais individuais em cada nodo, o que não é comum em c*lusters* de HPC atualmente. Como uma utra desvantagem, o MapReduce requer um formato de dados em particular, que não é completamente estruturado nos formatos de entrada e saída do BLAST. MapReduce também não é tão apropriado aos tipos de problemas com soluções de estilo *all-to-all*, como a busca de sequências por alinhamento [58].

Além do MapReduce, outras versões de BLAST paralelo com memória distribuída estão disponíveis atualmente, tais como mpiBLAST [50] e ScalaBLAST [62]. mpiBLAST melhorou o desempenho do alinhamento de sequências ampliando o número de nodos que podem ser utilizados para realizar esta tarefa computacionalmente intensiva. A estratégia de paralelismo do mpiBLAST é baseada no particionamento da base de dados de entrada em vários fragmentos, em número igual ao número de núcleos a serem utilizados [50]. Da mesma maneira, o ScalaBLAST particiona em fragmentos as sequências de entrada e da base de dados. Em ambos os casos, estes fragmentos são copiados individualmente para cada nodo correspondente e uma busca local é realizada [62]. Após estes passos, os resultados de cada processo são reunidos.

Contudo, o particionamento das entradas e o seguintes passos podem gerar *overhead* operacional, quando se trabalha com bases de dados muito grandes, tais como a *nt/nr* do NCBI, ou sequências de busca numerosas, tais como os dados obtidos de sequenciamento de nova geração. Ainda, uma busca paralela que particiona a base de dados precisa consolidar os segmento de sequências encontrados a partir dos fragmentos da base de dados para cada sequência de busca, requisitando um pós-processamento extra, se comparado com o BLAST sequencial. Nesta seção é descrita a nova estratégia de paralelização para o BLAST, focando no função de alinhamento de nucleotídeos (MPI-blastn)

A presente implementação, MPI-blastn, é baseada em uma versão recente do BLAST+ (2.2.25); Diferentemente do BLAST+, que pode somente utilizar individualmente máquinas multicore por memória compartilhada, a presente implementação explora o poder computacional de *clusters* de

máquinas multi-núcleo, para permitir o aumento de desempenho e escalabilidade.

Nossa abordagem basicamente consiste em duas etapas. Na primeira etapa, o algoritmo divide as sequências de entrada (*queries*) de forma igual entre o número de nodos disponíveis e coloca estas sub-*queries* em um armazenamento compartilhado, juntamente com a base de dados NCBI. Na segunda etapa. Cada nodo carrega sua sub-*query* e uma cópia da base de dados inteira para a memória local, e a execução paralela se inicia. A biblioteca MPI é utilizada para trocar mensagens entre os nodos do cluster através da estratégia de programação mestre-escravo e *threads* são utilizadas pelos escravos para explorar os núcleos de cada nodo. O processo mestre é responsável pela primeira etapa e todos os processos executam a segunda etapa. A figura 8 mostra o fluxo de trabalho para estas etapas.

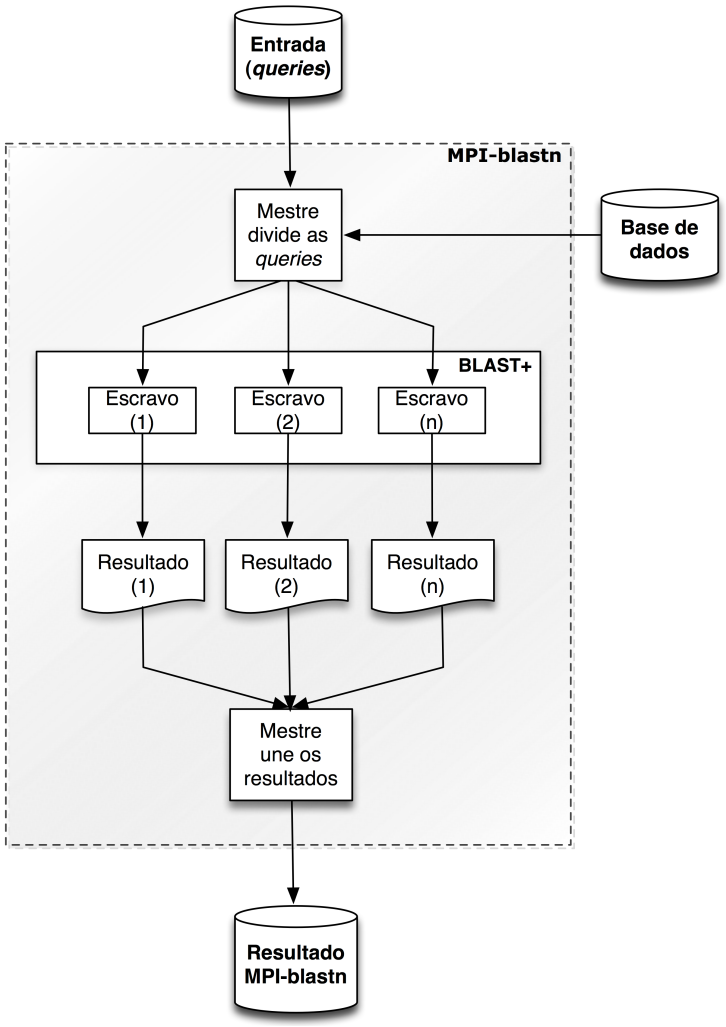

Figura 8. Fluxo de trabalho do programa MPI-blastn.

Quando o processo mestre inicia, cada processo escravo espera até que o mestre divida todas as *queries* de entrada. Depois desta divisão, o processo mestre envia a todos demais um sinal de início, indicando que o estágio de divisão de trabalho foi terminado. Após este passo, os processos carregam suas respectivas *queries* e a base de dados inteira a partir do armazenamento compartilhado e executam o alinhamento de suas respectivas *queries.* Quando o processamento termina, todos os escravos reportam ao mestre esta informação. Após realizar a busca de sequências por alinhamento, os dados de saída do MPI-blastn consistem em um grupo de arquivos de resultados gerados por cada processo. Por último, o processo mestre une todos os resultados em um único arquivo de saída.

Na presente implementação, as mensagens MPI não carregam de fato os dados de entrada, pois estes podem ser acessados diretamente do armazenamento compartilhado por todos os processos. Sendo assim, as mensagens trocadas ente o mestre e os escravos tem a função unicamente de sincronização entre estes. Esta decisão foi feita com o objetivo de simplificar a implementação, evitando mensagens muito grandes. Por exemplo, dependendo do tamanho do problema, as *queries* podem ter um tamanho de 8 MB e a base de dados pode ter um tamanho de 4 GB. Nossa alternativa causa o mínimo de impacto no desempenho porque o tempo necessário para dividir e unir as *queries* é muito pequeno comparado com o tempo total do processamento. No pior caso (pequenas cargas de trabalho por processo). Este I/O representa apenas 3% do tempo total de execução, como demonstrado em [62][68].

3.6 Sumarização dos resultados

Na versão anterior do PANGEA [23], a sumarização dos resultados, que consiste basicamente nas funções de contagem de sequências classificadas e análise estatísticas, utilizava como entrada apenas os dados de classificação de espécies obtidos através do BLAST. No PANEGA+, o usuário poderá executar estes passos de pós-processamento para os

resultados consenso além de poder executar para os resultados únicos do BLAST+. Para que isso seja possível, os dados de saída da função *Consensus* são gerados em no formato suportado pelas funções *Megaclust* e *Megaclustable*. Além disso, a função *Megaclust* foi modificada para identificar como comentário as linhas dos resultados consensos que correspondem ao número de níveis taxonômicos encontrados em comum entre os métodos de classificação (*Matches found).*

4 AVALIAÇÃO DA NOVA ARQUITETURA

A nova arquitetura PANGEA+ foi avaliada e comparada com demais *pipelines* metagenômicos em três principais fatores:

1) Avaliação qualitativa: A etapa de pré-processamento (Trim2) e da análise consenso (Consensus) foram validadas e comparadas com os resultados das mesmas etapas de outros *pipelines.*

2) Avaliação quantitativa (avaliação de desempenho): os programas desenvolvidos MPI-blastn e NCBI-TaxCollector foram comparados seus respectivos representantes de estado da arte.

3) Funcionalidades: As novas funcionalidades desenvolvidas foram comparadas com as dos demais *pipelines* metagenômicos existente, comparando o nível de abrangência de cada um.

Nesta seção são avaliadas as funções desenvolvidas no presente trabalho e introduzidas no PANGEA+, descrevendo os resultados obtidos para cada uma destas três etapas de avaliação de desempenho.

4.1 Avaliação qualitativa

Na presente seção são descritos os resultados qualitativos obtidos. São avaliadas as principais otimizações desenvolvidas ao longo do projeto, em nível de exatidão e qualidade dos resultados, para os programas Trim2 e Consensus.

4.1.1 Validação do suporte a Paired-ends (Trim2)

Para validar a nova versão do Trim2, foi feita a avaliação da sua nova funcionalidade principal, que consiste no suporte a *Paired-ends* [34]. A avaliação tem o objetivo de verificar se o resultado da busca de uma

sequência no seu formato normal e completo seria o mesmo que a busca para a mesma sequência no formato *Paired-end* (com uma lacuna de bases nucleotídicas não identificadas no meio da sequência).

Para realizar esta tarefa, foram extraídas aleatoriamente 50 sequências da base de dados NCBI [22]. As lacunas entre as extremidades de cada uma das sequência foram inseridas artificialmente, substituindo as bases nucleotídicas identificadas por bases não identificadas, representadas por N. Nas sequências em formato Paired-ends, o Trim2 insere automaticamente esta lacuna de tamanho definido pelo usuário, preenchida por Ns, unindo cada extremidade da sequência (*Single-end)*. O grupo de teste consiste em sequências de bactérias e eucariotos, provenientes dos genes 16S e 18S, que variam seu tamanho ente 1321 a 1546 bases nucleotídicas. Para esta simulação, o início das lacunas foi estabelecido aleatoriamente entre a posição 400 e 600 das sequências. O tamanho das lacunas varia entre 100 e 400 bases nucleotídicas.

A classificação de espécies foi executada com BLAST+ *online*, tanto para as sequências completas quanto para as na versão *Paired-ends,* e estes resultados foram comparados. A comparação foi feita com base nas espécies identificadas e na qualidade do alinhamento. Assim, os *Paired-ends* foram simulados de forma semelhante ao estudo realizado em [56], mas com o objetivo de validar esta nova funcionalidade do Trim2.

A comparação entre as sequências completas e as mesmas com as lacunas inseridas (*Paired-ends* simulados) demonstra que as lacunas afetam pouco os resultados de classificação de espécies. Entre as 50 sequências *type* de boa qualidade extraídas da base de dados RDP, o programa BLAST+ *online* encontrou os mesmos resultados em 90% dos casos, sem alterar as conclusões sobre classificações. Em 10% dos casos, ocorreram pequenas alterações onde o primeiro resultado (*top 1 hit*) passou para a 2-6 posição no ranque de resultados. Destas alterações 4% ocorreram em nível de espécie e 6% ocorreram em nível de gênero.

Ainda assim, essas pequenas alterações poderiam afetar uma pequena porção dos resultados dependendo da forma de execução do BLAST+, escolhida pelo usuário. Se o usuário seleciona apenas o primeiro resultado, por exemplo, esta mudança de ordenação modifica a classificação obtida.

Entretanto, se são considerados os *top* hits que compreendam essa margem de erro, os resultados de classificação que se deslocaram no ranque seriam considerados e não afetariam diretamente os resultados. Estes resultados indicam que deve-se considerar os *top hits* do BLAST+, em vez de considerar apenas o resultado único, é aconselhável para superar essa margem de erro ao classificar sequências no formato *Paired-ends*. Além disso, um consenso entre vários métodos seria uma solução em potencial ainda melhor, pois esta alternativa diminui a chance de desvios nos resultados. Os resultados obtidos desta solução, que consistem na avaliação qualitativa da etapa Consensus, são descritos na próxima seção.

4.1.2 Avaliação dos resultados consenso (Consensus)

Para validar a análise consenso (*Consensus*), foi avaliado o grau de seu melhoramento em comparação com os resultados do BLAST+ [6]. Após a execução da etapa *Consensus* no PANGEA+, os resultados foram comparados com os da versão anterior do PANGEA [23], que apenas utilizava o resultado único (*top 1 hit*) do BLAST [1]. Os dados utilizados para a presente validação consistem em duas bases de dados de sequências conhecidas, que foram isoladas e classificadas (*type*), disponíveis na base de dados RDP [53]. Entre estas sequências, foram selecionadas as com boa qualidade de leitura, resultando em dois grupos de teste: (1) 9178 sequências referentes ao gene 16S (bactérias), e (2) 373 sequências referentes ao gene 18S (eucariotos). Os grupos de teste 1 e 2 correspondem a 181966 linhas (14 MB de tamanho) e 7204 linhas (552 KB de tamanho), respectivamente.

O grupo de teste 1 corresponde a sequências com comprimento mínimo de 320 bases nucleotídicas, máximo de 1847, e médio de 1467. O grupo de teste 2 corresponde a sequências com comprimento mínimo de 949, máximo de 2210 e médio de 1426 bases nucleotídicas. Para acessar estas informações gerais sobre os dados de entrada, foi desenvolvido um

programa simples na linguagem Perl, *Fasta_statistics-0.1.pl,* que faz a contagem e as métricas de tamanho para arquivos no formato FASTA.

Como métrica de qualidade, foi utilizado o valor E-value, que é um parâmetro que descreve o número de vezes esperadas de se encontrar por acaso um dado resultado quando este é buscado contra uma base de dados com um tamanho determinado. Este valor diminui exponencialmente em função da qualidade do resultado da busca. Basicamente o E-value descreve o grau de ruídos nos resultados. Por exemplo, um E-value de valor 1 atribuído a um resultado de busca pode ser interpretado como se em uma base de dados pudesse ser observado um resultado igual, com uma pontuação de qualidade igual, simplesmente por acaso. Assim, quanto mais baixo o E-value, ou mais próximo de zero, mais significante será o resultado. O E-value é calculado a partir da seguinte equação:

$$E = \frac{N}{2^{S'}}$$

Onde E corresponde ao E-value; N representa o espaço de busca ou o número de possibilidades do resultado (N=m*n, onde m e n são os comprimentos da sequência de busca e da identificada); e S' representa a pontuação de qualidade do alinhamento associada ao E-value.

O cálculo do E-value leva em consideração o tamanho da sequência de busca, a pontuação de qualidade do alinhamento, e o tamanho das regiões do alinhamento de grande qualidade. Entretanto, um problema em confiar apenas no E-value é que sequências pequenas tendem a resultar em E-values altos. Estes valores altos tem sentido pois as sequências menores tem maior probabilidade de ocorrer na base de dados por acaso. Como uma alternativa para o uso do E-value, foram incluídos os valores de identidade e de *bitscore* na presente avaliação. O cálculo da identidade corresponde ao percentual de bases nucleotídicas iguais e alinhadas entre a sequência de busca e a identificada. O *bitscore* representa uma pontuação normalizada, expressa em bits, que faz uma estimativa do tamanho de espaço de busca necessário para encontrar uma pontuação tão boa ou melhor do que a encontrada por acaso. Em outras palavras, quanto melhor a qualidade do

alinhamento, maior será o *bitscore*. Esta métrica leva em consideração a qualidade do alinhamento, que consiste no grau de identidade, número de lacunas (*gaps*) e diferenças (*mismatches*) entre as sequências, e é definida com a seguinte fórmula:

$$S' = \frac{\lambda S - \ln(K)}{\ln(2)}$$

Onde S' representa o *bitscore*, lambda e K são parâmetros dependentes da matriz de pontuação e das penalidades para inserção de lacunas utilizadas para o alinhamento [1]. S é a pontuação bruta (*raw score*), calculada da seguinte forma:

$$S = \sum_{i=1}^{L} S_{r_{1,i} r_{2,i}}$$

Onde S corresponde a pontuação referente a matriz de substituição ou de penalidades para inserção de lacunas [1]; r_1 corresponde a sequência de busca (*query)*; r_2 é a sequência encontrada (*subject*); L é o comprimento das sequências alinhadas e i representa a posição da base nucleotídica.

Além dos destes indicativos de qualidade, também foram considerados os parâmetros de inserção de lacunas e incompatibilidades (*mismatches)* entre as sequências. Estes dois parâmetros são inversamente proporcionais a qualidade do alinhamento das sequências. Por exemplo, quanto mais lacunas inseridas ou quanto mais diferenças (incompatibilidades) são observados em um alinhamento, maior é a diferença entre as sequências comparadas. Por sua vez, quanto maior as diferenças entre as sequências de um alinhamento, pior será a qualidade dos resultados.

Assim, tendo estes parâmetros estabelecidos como base para a comparação dos resultados, pode ser calculada a correlação entre o número de resultados consenso encontrados e o grau de qualidade ou confiabilidade dos resultados. O número de resultados consenso é medido pelo valor do parâmetro *Matches found* de cada resultado, que indica quantos níveis taxonômicos em comum foram encontrados entre os métodos. E como mencionado, as métricas de qualidade dos resultados serão os valores de E-

value, identidade (%) e *bitscore*. Finalmente, o cálculo da correlação entre os resultados consenso e as métricas de qualidade fornece uma noção de o quanto os resultados consenso (*Matches found*) estão diretamente relacionados à qualidade e confiabilidade dos resultados da classificação de espécies.

Assim, para a avaliar a função Consensus, primeiramente a etapa *Classify* do PANGEA+ foi executada para os grupos de teste 1 (9178 sequências de bactérias) e 2 (373 sequências de eucariotos). Com a ferramenta *Classify* foram executados os três métodos de classificação: MPI-blastn (baseado no BLAST+), RDP Classifier e SOAP Aligner. A base de dados utilizada para a execução do MPI-blastn e SOAP2 Aligner foi a NCBI *nt/nr*. A base de dados foi pré-processada, filtrando as sequências referentes aos genes 16S e 18S. Para realizar esta tarefa foi desenvolvido um programa em Perl, *Fasta_filter*, que utiliza palavras-chave como entrada para gerar a base de dados filtrada.

Foram gerados no total 192.700 resultados pelo MPI-blastn, que variam entre os 20-40 melhores resultados (*top hits*), 9178 resultados pelo RDP Classifier, que correspondem a um resultado de classificação para cada sequência de entrada e 18 resultados do SOAP Aligner. Sendo que a maioria das sequências possui mais de 1000 bases nucleotídicas, a baixa quantidade de resultados do SOAP Aligner provavelmente se deve a característica de seu algoritmo que consiste em ter um desempenho satisfatório apenas para sequências menores (1000 bases de tamanho), como as obtidas pela tecnologia de sequenciamento Illumina.

Após a classificação, o NCBI-TaxCollector foi utilizado para o anexo de informações taxonômicas para os resultados do MPI-blastn e SOAP2 Aligner. Em seguida, a etapa *Consensus* foi executada. Nesta etapa, o MPI-blastn foi considerado como a prioridade de comparação, por ter se demonstrado como o método mais exato e mais utilizado em muitos estudos. Assim, os resultados consenso listados definem-se como os resultados iguais encontrados entre o MPI-blastn e o RDP Classifier, ou os resultados iguais encontrados entre o MPI-blastn e o SOAP2 Aligner. Ainda, antes de realizar a análise qualitativa as entradas redundantes, que correspondem a sequências pertencentes a mesma espécie, foram removidas dos grupos de

teste. Essa filtragem foi feita para eliminar repetições nos resultados que poderiam favorecer o método de classificação, caso este possua uma melhor exatidão para certos tipos de padrões entre as sequências analisadas. Após a filtragem de sequências redundantes, 5496 sequências foram mantidas no grupo de teste 1 e 338 sequências foram mantidas no grupo 2.

Os resultados consenso do grupo de teste 1, descritos na Tabela 9, demonstram que, até o nível taxonômico de classe, existe uma exatidão considerável entre os métodos de classificação (acima de 50%). Entretanto, para níveis taxonômicos mais derivados, como família, gênero e espécie, a quantidade de resultados corretos entre os métodos de classificação cai drasticamente, chegando a ~18% para o nível de gênero e 0.04% para o nível de espécie. Estes resultados demonstram uma divergência grande entre os métodos de classificação, que aumenta com a especificidade de cada resultado. E principalmente, os resultados da Tabela 9 demonstram que a etapa Consensus, introduzida no PANGEA+, foi capaz de melhorar consideravelmente a exatidão do MPI-blastn (BLAST+ executado em paralelo através da nova versão desenvolvida no presente trabalho). Estes resultados indicam que se basear em apenas um método de classificação pode levar a erros de identificação de espécies e demonstram que o uso de uma análise consenso melhora significativamente precisão dos resultados, se comparados com o uso de apenas um método de classificação de espécies.

Tabela 9. Resultados da etapa consenso para o grupo de teste 1, com o número de identificações corretas de cada método.

Nível	*Consensus*		MPI-blastn	
taxonômico	N. de acertos	(%)	N. de acertos	(%)
Super-reino	5496	100,0	5496	100,0
Filo	4870	88,61	3558	64,74
Classe	3805	69,23	2906	52,87
Ordem	2797	50,89	2428	44,18
Família	1790	32,57	1474	26,82
Genero	974	17,72	913	16,61
Espécie	2	0,04	1	0,02

Para o grupo de teste 2, a etapa Consensus foi capaz de elevar drasticamente a quantidade de acertos nos níveis taxonômicos de

classificação de super-reino e filo (Tabela 10). Entretanto, observa-se que para os demais níveis taxonômicos não houveram alterações. Isso ocorre devido a base de dados NCBI *nt/nr* possuir muitas entradas rotuladas como "*uncultured"* para sequências do gene 18S. Estas entradas numerosas, que não possuem detalhes de classificação por não serem sequências cultivadas ou *type*, preenchem praticamente todos os resultados (*top hits*) do MPI-blastn. Estes resultados acabam ocultando os dados mais detalhados de classificação taxonômica (sequências que possuem dados completos de super-reino até espécie). Esta limitação poderia ser superada filtrando a base de dados NCBI *nt/nr* ou aumentando o número de *top hits* a serem gerados pelo MPI-blastn.

Tabela 10. Resultados da etapa consenso para o grupo de teste 2, com o número de identificações corretas de cada método.

Nível taxonômico	*Consensus*		MPI-blastn	
	N. de acertos	(%)	N. de acertos	(%)
Super-reino	337	99,70	22	6,51
Filo	40	11,83	16	4,73
Classe	0	0	0	0,00
Ordem	0	0	0	0,00
Família	0	0	0	0,00
Genero	0	0	0	0,00
Espécie	0	0	0	0,00

Medindo em uma forma mais exata, medindo a qualidade dos resultados consenso para ambos grupos de teste, foi calculada a correlação ou dependência entre os parâmetros indicativos de qualidade (identidade, *bitscore,* entre outros) e a contagem de resultados em comum entre os métodos (parâmetro *Matches found)*, como mencionado na seção 4.1. Os resultados, descritos na Tabela 11, mostram que existe uma forte correlação (acima de 70%) entre o nível de identidade das sequências encontradas e a contagem de resultados em comum entre os métodos. Em outras palavras, estes resultados indicam que quanto maior for a contagem de resultados consenso, melhor é a qualidade dos resultados, em nível de identidade entre as sequências. Esses resultados são reforçados pela correlação positiva obtida com o *bitscore*, classificada de moderada a forte (30-70%), sendo também este um parâmetro indicativo de qualidade dos resultados. Assim, é

possível constatar que existe uma correlação fortemente positiva entre a qualidade e o consenso dos métodos de classificação de espécies.

Tabela 11. Estatística descritiva: correlação dos resultados consenso e média dos parâmetros qualitativos obtidos entre os métodos.

Parâmetros qualitativos	Correlação	Média	
		Consenso	MPI-blastn
Identidade	0,77	90,00	89,33
Incompatibilidades	-0,62	84,43	99,23
Inserção de lacunas	-0,54	35,48	43,00
Bitscore	0,68	1503,24	1709,08
E-value	0,00	0,00	0,00
Resultados consenso	---	3,6	2,1

Da mesma maneira, os parâmetros de incompatibilidades (*mismatches*) e inserção de lacunas demonstraram uma correlação negativa de moderada a forte (entre -30 e -70%) com o número de resultados consenso entre os métodos. Estes parâmetros, que são inversamente proporcionais a qualidade dos resultados (quanto menores, melhores são os resultados, e vice-versa), indicam novamente que existe uma relação entre a qualidade dos resultados e o número de resultados em comum entre os métodos de classificação de espécies. Essas informações novamente que é mais apropriado não confiar em apenas um único método de classificação nas análises metagenômicas e, em vez disso, utilizar um consenso entre métodos para aumentar a qualidade dos resultados.

Adicionalmente, os dados de estatística descritiva, também descritos na Tabela 11, mostram que houve uma diminuição no número de inserção de lacunas e de incompatibilidades nos resultados da etapa Consensus, se comparada com os resultados únicos do MPI-blastn. Também houve um pequeno aumento no nível de identidade entre as sequências. Entretanto, houve uma queda no valor do *bitscore*. Esta queda é esperada pois o grau de identidade foi a prioridade nos casos de empate durante a avaliação dos resultados consenso, como mencionado na seção 3.3, sendo a identidade muitas vezes um critério limitante para a seleção dos resultados em nível taxonômico. Não obstante, o algoritmo do programa *Consensus* pode ser adaptado para priorizar o *bitscore* no lugar do grau de identidade entre as

sequências, o que elevaria as média do resultado obtido para este parâmetro.

4.2 Avaliação quantitativa

Na presente seção são avaliadas as otimizações quantitativas desenvolvidas ao longo do projeto, para os programas MPI-blastn e NCBI-TaxCollector, discutindo os principais resultados obtidos. Os experimentos para a presente avaliação de desempenho foram realizados em dois ambientes:

1) *Cluster* Cerrado (16 nodos): cada nodo possui 2 processadores Intel(R) Xeon(R) E5645 2.40 GHz, cada processador com 6 núcleos, resultando em um total de 12 núcleos por nodo. Estes processadores possuem suporte a tecnologia *Hyper-Threading* (HT), que nos possibilita o uso de 24 núcleos por nodo. Os nodos são conectados por uma rede Infiniband. Os trocas de mensagens são realizadas pela biblioteca Open MPI [26]. O sistema operacional deste ambiente consiste na versão 10.04 do Linux Ubuntu. Cada nodo do *cluster* possui 24 GB de memória RAM, um único disco local e um armazenamento compartilhado entre todos os nodos. O *cluster* utiliza *network file system* (NFS) para armazenar os dados dos usuários. 16 nodos foram utilizados para os experimentos realizados com MPI-blastn e um nodo foi utilizado para NCBI-TaxCollector (sequencial).

2) *Cluster* Atlântica (10 nodos): cada nodo possui 2 processadores Intel(R) Xeon(R) Quad-Core E5520 2.27 GHZ, totalizando 16 núcleos por nodo com HT, 16 GB de memória RAM. Os nodos são conectados por uma rede Infiniband. Os trocas de mensagens são realizadas pela biblioteca Open MPI [26]. O sistema operacional deste ambiente consiste na versão 10.04 do Linux Ubuntu. Os nodos são interconectados por uma rede Gigabit Ethernet. Um nodo foi utilizado para realizar os experimentos de avaliação geral das funções da nova arquitetura desenvolvida.

Ambos os *clusters* estão localizados no Laboratório de Alto Desempenho (LAD-PUCRS). No caso do MPI-blastn, os resultados de desempenho foram comparados com o mpiBLAST apresentado por Lin *et al*. [50], que é um dos

representantes entre as implementações de BLAST paralelo. Para avaliar o desempenho do MPI-blastn, 10 repetições de cada execução paralela foram realizadas. Os testes foram variados em número de processos e em tamanho de entrada. Como dados de entrada foram utilizadas a base de dados *nt / nr* do NCBI e sequências aleatoriamente selecionadas da mesma base de dados [22]. Como métrica de desempenho foi calculado o fator de aceleração (*speedup)* a partir dos tempos médios das execuções paralelas e sequencial.

No caso do NCBI-TaxCollector, as otimizações de complexidade foram comparadas com o TaxCollector original [24], que faz parte do *pipeline* PANGEA [23]. Até então, esta versão proposta por Giongo *et al*. era a única ferramenta disponível para realizar a tarefa de anexo de informações taxonômicas. As métricas de comparação utilizadas foram tempo de execução e máximo uso de memória principal. Sendo que ambos NCBI-TaxCollector e TaxCollector são programas sequenciais, com diferenças apenas na complexidade de seus algoritmos, essas métricas comparativas foram extraídas de execuções sequenciais.

Também foram feitas avaliações de desempenho para os outros métodos de classificação de espécies: SOAP2 Aligner e RDP Classifier. Entretanto, considerando que os seus tempos de execução são muito pequenos (20-30 vezes menores que o BLAST+, para um mesmo dado de entrada), os testes de desempenho foram limitados a um nodo do cluster mencionado nesta seção. Assim, o maior foco da presente avaliação de desempenho se manteve nos grandes gargalos de desempenho computacional: BLAST+ e TaxCollector.

4.2.1 Desempenho do MPI-blastn

Com o objetivo de avaliar MPI-blastn sob diferentes cargas de trabalho, foram utilizados dois tamanhos de dados de entrada, que correspondem a sequências de 30,000 e 100,000 linhas. Estas sequências de entrada foram obtidas aleatoriamente a partir base de dados *nt* do NCBI [22]. Os testes com o BLAST paralelo foram realizados contra a base de dados *nt* original.

Foram realizadas 5 repetições para cada teste, que demonstraram um desvio padrão menor que 0.001. O primeiro experimento foi a comparação entre MPI-blastn e mpiBLAST em relação ao tempo sequencial do BLAST+ (executando em apenas 1 núcleo). Os resultados podem ser observados na Figura 9. Utilizando a entrada de 30,000 linhas, foi comparada a escalabilidade das duas implementações até 240 núcleos (10 nodos, em nosso caso). Apesar de nosso ambiente de teste ser um cluster de 16 nodos (384 núcleos), foi impossível executar mpiBLAST para mais de 244 núcleos, pois o programa apresenta uma mensagem de erro, indicando que não pode ser executado para ais de 244 núcleos.

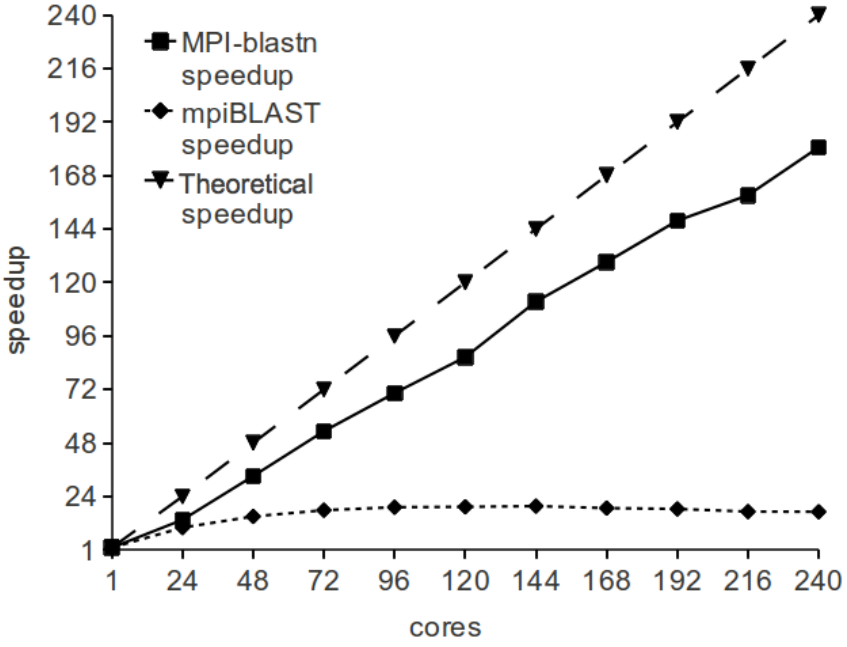

Figura 9.Comparação de desempenho entre MPI-blastn e mpiBLAST, utilizando 30,000 linhas de entrada em um *cluster* com 240 núcleos.

Nossas primeiras avaliações demonstram que a presente implementação escala muito melhor do que o mpiBLAST até mesmo para uma carga de trabalho pequena. Logo após 24 núcleos o *speedup* do mpiBLAST começa a estagnar e passa a declinar após 96 núcleos. O MPI-blastn, diferentemente do mpiBLAST, continua a escalar de forma linear quando comparado ao valor do *speedup* teórico, até 240 núcleos, para esta carga de trabalho.

Esta grande diferença de desempenho pode ser causada principalmente por duas razões: a versão do BLAST utilizada na implementação e a estratégia de paralelização mais otimizada. O MPI-blastn é baseado em uma versão recente do BLAST (BLAST+, versão 2.2.25), enquanto que o mpiBLAST é baseado em uma versão anterior do BLAST (2.0.9). BLAST+ demonstra um algoritmo reformulado e substancialmente mais eficiente do que as versões anteriores do BLAST [6]. Além disso, escolhendo uma aproximação menos complexa para particionar o problema, subdividindo apenas as sequências de entrada (*queries*) e fornecendo cópias completas da base de dados NCBI para os processos escravos, a presente implementação foi capaz de reduzir consideravelmente o tempo de execução do BLAST+ , de ~7,5 horas para menos de 3 minutos (Tabela 11). Este resultado representa uma execução até 186 vezes mais rápida para 240 núcleos computacionais, se comparado com o tempo sequencial, que representa uma eficiência de 77,5% em relação ao desempenho esperado (*speedup* teórico) para esta carga de trabalho (30,000 linhas).

Na Tabela 12 também são demonstrados os testes para todos os 16 nodos do *cluster* (384 núcleos), utilizando uma carga de trabalho maior (100,000 linhas). Com esta carga de trabalho foi obtido um desempenho ainda melhor, executando o MPI-blastn 408 vezes mais rápido que o tempo sequencial do BLAST+, reduzindo o tempo de execução de ~37 horas para 5,5 minutos. Estes resultados representam uma eficiência ainda maior se comparada com o *speedup* teórico (106%).

Na Figura 10 é avaliada a escalabilidade da presente implementação para os 16 nodos do *cluster* (384 núcleos) com uma carga de trabalho maior do que a anterior, que corresponde a 100,000 linhas de entrada. Observa-se que esta carga de trabalho mais pesada melhorou a eficiência dos núcleos devido a uma melhor distribuição dos tamanhos das tarefas por núcleo, diminuindo o tempo ocioso entre os processos e aumentando o desempenho.

Tabela 12. Tempo de execução (horas) para o mpiBLAST e MPI-blastn com diferentes tamanhos de entrada, em um cluster de 16 nodos (384 núcleos).

Input size (lines)	30,000		100,000
Number of cores	mpiBLAST	MPI-blastn	MPI-blastn
1	7.43	7.43	36.80
24	0.74	0.55	3.18
48	0.49	0.22	1.34
72	0.42	0.14	0.71
96	0.39	0.11	0.48
120	0.38	0.09	0.36
144	0.38	0.07	0.28
168	0.40	0.06	0.23
192	0.40	0.05	0.20
216	0.43	0.05	0.18
240	0.43	0.04	0.16
264	X	-	0.14
288	X	-	0.13
312	X	-	0.12
336	X	-	0.11
360	X	-	0.10
384	X	-	0.09

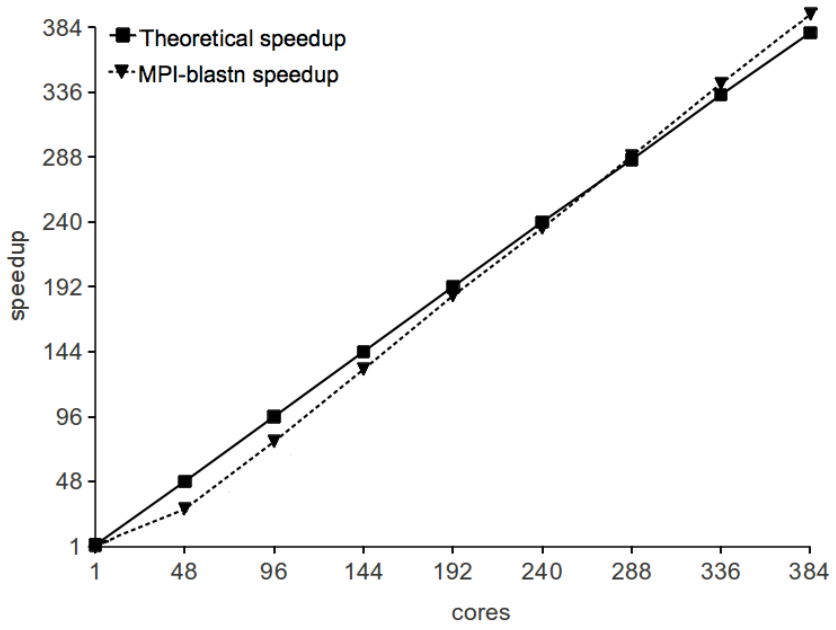

Figura 10. Escalabilidade do MPI-blastn utilizando 100,000 linhas de entrada em um cluster com 16 nodos (384 núcleos).

Como resultado, o MPI-blastn obteve um desempenho muito próximo do *speedup* teórico, até mesmo o ultrapassando na execução para 288 núcleos. Este comportamento superlinear de *speedup* pode ter sido causado por efeitos de uso de cache, quando o tamanho da carga de trabalho por processo é diminuída o suficiente para se adequar completamente ao tamanho da memória cache e se beneficia de uma latência de memória muito mais baixa.

Além da avaliação do mpiBLAST, a presente comparação foi estendida para uma outra versão paralela da ferramenta BLAST, conhecida como ScalaBLAST [62]. Contudo, aparentemente o ScalaBLAST não é compatível com o ambiente computacional de teste utilizado, porque o programa foi incapaz de lidar apropriadamente com a alocação de memória. Até mesmo a menor entrada, de 1000 linhas contra a base de dados NCBI, causou erros críticos de alocação e mapeamento de memória reportados nos relatórios de execução. Estes erros pararam prematuramente a execução do ScalaBLAST, impedindo a avaliação de seu desempenho. Até mesmo o autor do programa interagiu no presente trabalho, fornecendo suporte para resolver o problema, mas não foi possível o resolver até a elaboração deste trabalho.

Além do desempenho, foram observadas algumas vantagens no fluxo de trabalho do MPI-blastn. Por exemplo, no ScalaBLAST, ambas sequências de entrada e de saída são particionadas e a saída deve ser reunida pelo usuário após a execução, o que não ocorre no MPI-blastn. Uma outra vantagem nas funcionalidades do MPI-blastn é que o usuário não precisa se preocupar com operações de pré-configuração. No ScalaBLAST, contudo, o usuário precisa configurar um arquivo de parâmetros pré-execução (sb_param.in) que define as regras do fluxo de trabalho, tamanho dos grupos de tarefas, método de distribuição de carga de trabalho, distribuição do uso de memória, entre outros.

4.2.2 Desempenho do NCBI-TaxCollector

Com o objetivo de avaliar o desempenho do presente algoritmo, seu desempenho foi comparado com uma ferramenta similar. Atualmente, uma versão recente do TaxCollector (Giongo et al. [24]) é o único programa disponível para realizar a tarefa de anexo de informação taxonômica. Os arquivos de entrada utilizados para a presente avaliação de desempenho consistem nos resultados do MPI-blastn, em formato de texto tabular. Os resultados do MPI-blastn foram gerados a partir de sequências e bases de dados do NCBI, como descrito nas seções anteriores. O TaxCollector, que é compatível com a base de dados *nt* do NCBI, leva 5 minutos de tempo de execução para 4 linhas de resultados do MPI-blastn, e requer mais de 60 GB de memória RAM para realizar esta tarefa (aproximadamente 75 segundos por linha de entrada de resultados BLAST). Seguindo este exemplo de execução, um anexo de informações taxonômicas que levaria 2 dias com TaxCollector, agora pode ser realizado em menos de uma hora com o NCBI-TaxCollector. Contudo, não foi possível estender os testes com TaxCollector para entradas com maiores tamanhos devido a grande quantidade de memória necessária para executar o TaxCollector. Diferentemente desta ferramenta, o NCBI-TaxCollector leva menos de um segundo para ser executar a mesma entrada. Além disso, foi observado um consumo máximo de memória RAM menor que 0.5 GB (muito menor do que os 60 GB utilizados pelo TaxCollector), como demonstrado na Figura 11. Estes resultados mostram que o NCBI-TaxCollector é capaz de realizar anexos de informações taxonômicas 125 vezes mais rápido e precisando 120 vezes menos memória RAM do que o TaxCollector proposto por Giongo et *al.* [24].

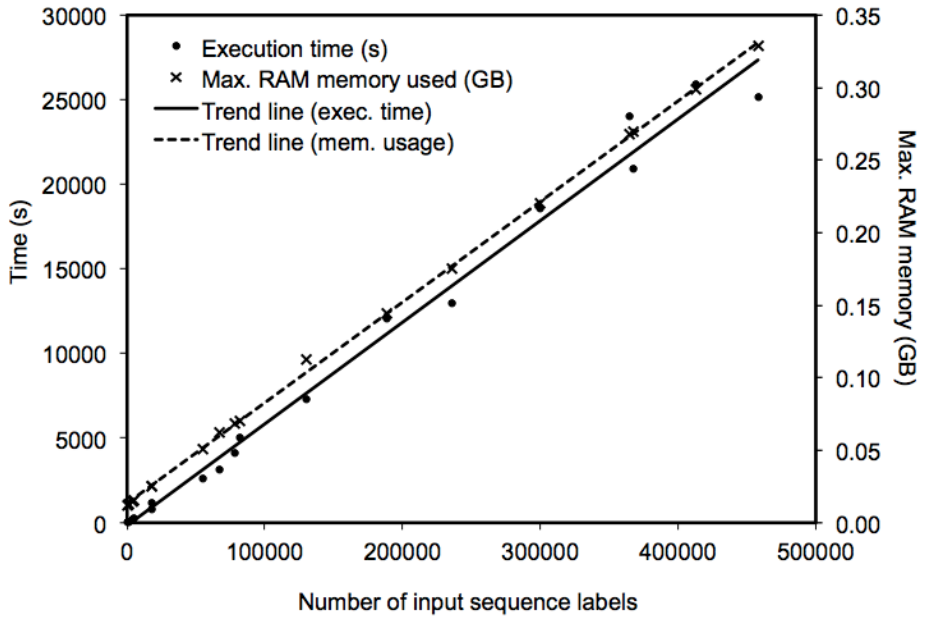

Figura 11. Desempenho do NCBI-TaxCollector: tempo de execução e uso de memória contra tamanho de entrada.

Estendendo a avaliação da presente ferramenta, foram aumentados os tamanhos das entradas (número de linhas de resultados do MPI-blastn) até 450.000 linhas. Este experimento foi realizado com o objetivo de analisar o comportamento do presente algoritmo para maiores tamanhos de entrada, em níveis de tempo de execução e uso de memória. Variando os tamanhos de entrada, foi observado que o tempo de execução e consumo de memória permanecem lineares, proporcionalmente ao tamanho dos arquivos de entrada. Estes resultados são demonstrados na Figura 10. A mesma avaliação foi impossível de ser realizada com o TaxCollector devido a grande quantidade de memória exigida e o tempo de execução infactível.

4.2.3 Desempenho das demais funções incluídas no PANGEA+

Além de avaliar o desempenho das funções de maior complexidade e tempo de execução, foram avaliados também os programas que correspondem as demais funções incluídas no PANGEA+. Estas funções são: SOAP2 Aligner, RDP Classifier, Trim2 e Consensus.

SOAP2 Aligner e RDP Classifier são ferramentas já existentes que foram incorporadas na presente arquitetura, sendo que a primeira possui suporte a multi-nodos (MPI) e a segunda possui suporte a multi-*threads* (Pthreads). Antes de selecionar o BLAST+ para realizar as otimizações de desempenho (resultando no MPI-blastn), foram avaliados o fator de aceleração e tempo de execução de cada um dos programas de alinhamento de sequências. Em um nodo do *cluster* Atlântica (16 núcleos com HT), foi avaliado primeiramente o fator de aceleração. Considerando este parâmetro de avaliação, observa-se um maior desempenho para a ferramenta MPI-blastn (Figura 12).

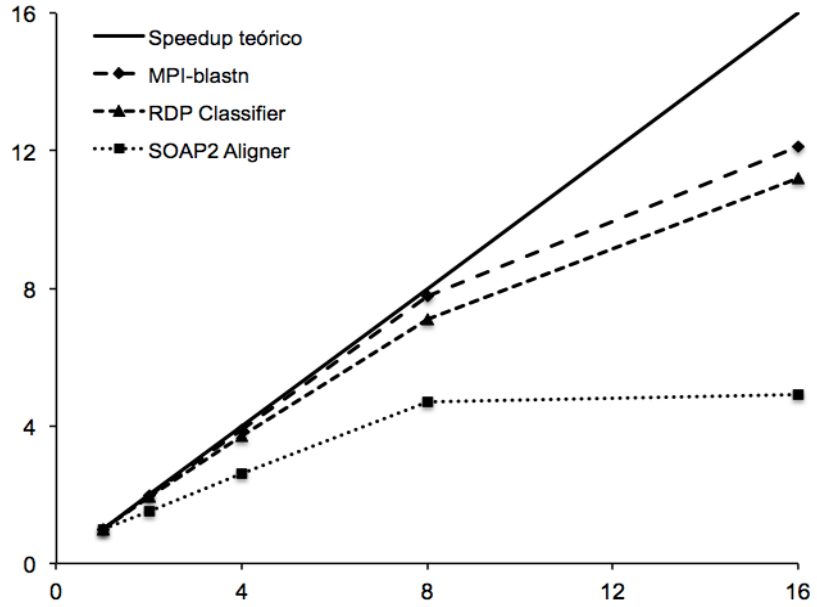

Figura 12. Desempenho alcançado entre os métodos de classificação.

Entretanto, se consideramos o tempo sequencial de cada uma destas ferramentas, os tempos de execução do SOAP2 e RDP Classifier somados chegam a ser 10 vezes menores do que o tempo sequencial de execução do MPI-blastn ou BLAST+ (a base sobre a qual a implementação paralela foi desenvolvida). Por exemplo, uma execução sequencial de alinhamento de sequências feito pelo SOAP2 Aligner e RDP Classifier levam 38 minutos no total, enquanto que o BLAST+ sequencial executa o mesmo em aproximadamente 6 horas e 20 minutos. Esse foi um dos principais motivos que levou a decisão de escolher o BLAST+ entre os demais programas como

o alvo da presente otimização de desempenho, além deste programa ser um dos mais utilizados e mais exatos para realizar a classificação de espécies.

Assim como os programas SOAP2 Aligner e RDP Classifier, as novas funções desenvolvidas, Trim2 e Consensus, possuem tempos de execução consideravelmente menores do que as demais etapas do PANGEA+. Então, a contribuição qualitativa das novas etapas (descritas na seção 5.1) são muito maiores e compensam o pequeno custo computacional acrescentado por elas. Em termos de tempo de execução, estas etapas representam menos de 10% da execução de todas as etapas do PANGEA+. Estes resultados são descritos na Tabela 13. Na Tabela 13, os tempos de execução da etapa de classificação de espécies no PANGEA+ correspondem aos programas MPI-blastn, RDP Classifier e SOAP2 Aligner, respectivamente. No caso do PANGEA, o tempo de referência para a etapa de classificação utilizado foi o tempo sequencial do BLAST+.

Tabela 13. Resultados de tempo de execução para os testes de validação do PANGEA+ (9178 sequências de entrada).

Principais fases de análise	Tempo de execução (h)	
	PANGEA	PANGEA+
Pré-processamento	0,11	0,14
Classificação de espécies	37,01	0,16 ; 2,1 ; 1,7
Pós-processamento	49,04	0,82
Consensus	-	0,02
Análise estatística	0,07	0,07
Total (h)	86,18	3,15

4.3 Funcionalidades

A avaliação das funcionalidades retoma o estudo comparativo entre cada um dos *pipelines* metagenômicos revisados neste trabalho. Cada uma das características ou funcionalidades novas identificadas do PANGEA+ serão comparadas com as demais ferramentas representantes do estado da arte. Serão comparadas as principais vantagens e desvantagens de cada com dos *pipelines* com os recursos da presente arquitetura. Com esta

finalidade, alguns dos *pipelines* foram instalados e testados, como o PANGEA [23] e MEGAN [32][33], e outros foram testados *online* por serem apenas ferramentas *web*, como o RDP Pipeline [12][13], RAST [25][55] e Galaxy [42]. No caso do *pipeline* Mothur [70], alguns problemas de compatibilidade com o sistema operacional do ambiente de teste impediram a execução completa de todas as etapas de análise. Neste caso a análise de funcionalidades foi guiada por referências e por tutoriais do programa.

As funcionalidades do PANGEA+ foram comparadas a destes *pipelines* que foram analisados. O resultado desta comparação, representado na Tabela 14, mostra que foram incluídas três grandes contribuições na presente arquitetura: (i) suporte a vários formatos de entrada (QSEQ, FASTQ e suporte a *Paired-ends*, além do formato FASTA já existente) com o novo Trim2; (ii) comparação entre métodos de classificação (MPI-blastn, RDP Classifier e SOAP2 Aligner) com o Consensus e NCBI-TaxCollector; e (iii) suporte a memória distribuída (MPI) para a etapa de classificação de espécies com o MPI-blastn.

Além das novas funcionalidades e recursos do PANGEA+, algumas outras não foram incluídas na presente arquitetura, como a análise filogenética (Tabela 14, item 9) e a análise funcional (item 10). A análise filogenética fornece informações sobre a relação evolutiva entre os organismos que correspondem pertencentes as sequências analisadas. Este é um recurso disponibilizado pela maioria dos *pipelines* e sua inclusão no PANGEA+ está sendo estudada para trabalhos futuros. A análise funcional, disponível nos *pipelines* MEGAN, Galaxy, RAST e Qiime, consiste na busca e inferência de possíveis funções enzimáticas para as sequências genéticas em análise. Esta busca é realizada por meio de consultas em bases de dados de funções enzimáticas e vias metabólicas, como a base de dados KEGG [57] utilizada pela MEGAN [33]. Esta funcionalidade pode ser adaptada ao PANGEA+ em trabalhos futuros, com otimizações de busca similares as desenvolvidas no NCBI-TaxCollector.

Tabela 14. Estudo comparativo entre os recursos atendidos por cada *pipeline* metagenômico avaliado ao realizar uma análise metagenômica.

Etapa ou Recurso	Presença de Suporte							
	MEGAN	Mothur	Galaxy	RDP	RAST	Qiime	PANGEA	PANGEA+
1. Suporte a vários formatos de arquivos (tecnologias diferentes)								
2. Avaliação e filtragem das sequências por qualidade								
3. Classificação de espécies contra uma base de dados								
4. Comparação entre métodos de classificação diferentes								
5. Pós-processamento dos resultados de classificação								
6. Análise estatística para organismos classificados								
7. Análise estatística para organismos não classificados								
8. Sumarização dos resultados								
9. Análise filogenética								
10. Análise funcional								
11. Memória compartilhada								
12. Memória distribuída (troca de mensagens)								
13. Versão local								
14. Versão Web								

5 CONCLUSÕES E TRABALHOS FUTUROS

Pipelines metagenômicos são ferramentas que otimizam o tempo de execução e a complexidade da análise de dados provenientes de tecnologias de sequenciamento em larga escala, tentando minimizar a intervenção humana neste processo. Essas ferramentas tornaram possíveis muitos estudos metagenômicos [36][37][40][41][73]. Entretanto, apesar das ferramentas computacionais terem acelerado os estudos metagenômicos, a quantidade de sequências geradas pelas tecnologias de última geração estão aumentando em uma velocidade ainda maior. O estudo comparativo realizado no presente trabalho revelou algumas das principais necessidades entre estas ferramentas, como o suporte a vários formatos de entrada, vários métodos de classificação e análise consenso. Além de novas funcionalidades, também foi verificada a necessidade de otimizações de desempenho computacional para as fases de maior complexidade e tempo de execução da análise metagenômica.

Para incluir estes novos recursos, um dos *pipelines* metagenômicos mais completos atualmente foi selecionado, o PANGEA. As novas funcionalidades e otimizações foram implementadas gerando uma nova arquitetura, o PANGEA+. Entre as otimizações qualitativas alcançadas estão as funções:

- Trim2: Suporte a vários formatos de entrada (QSEQ, FASTQ e suporte a *Paired-ends*, além do formato FASTA já existente);
- Classify: inclusão de novos métodos de classificação de espécies, RDP Classifier e SOAP2, além do MPI-blastn, expandindo e enriquecendo a análise metagenômica;
- Consensus: Comparação entre os resultados dos métodos de classificação, gerando resultados consenso que melhoram significativamente a qualidade dos resultados da etapa de classificação de espécies;
- NCBI-TaxCollector: inclusão de suporte a base de dados taxonômicos do NCBI, expandindo a classificação de espécies para o uso da base NCBI *nt/nr*, além do suporte ao RDP *database* já existente.

Entre as principais contribuições quantitativas do PANGEA+, estão:

- MPI-blastn: implementação de suporte a memória distribuída para a função de alinhamento de sequências do algoritmo BLAST+, ampliando o suporte da fase de classificação de espécies para vários nodos de um *cluster*;
- NCBI-TaxCollector: redução da complexidade do algoritmo de pós-processamento dos dados de classificação, diminuindo o tempo de execução da atribuição de dados taxonômicos em até 125 vezes e reduzindo o consumo de memória 120 vezes.

Resumidamente, em nível de funcionalidades e recursos, foram incluídas três grandes contribuições nas funcionalidades da presente arquitetura:

(i) suporte a vários formatos de entrada (QSEQ, FASTQ e suporte a *Paired-ends*, além do formato FASTA já existente) com o novo Trim2;
(ii) comparação entre métodos de classificação (incluindo MPI-blastn, RDP Classifier e SOAP2 Aligner) com o Consensus e NCBI-TaxCollector;
(iii) suporte a memória distribuída (MPI) para a etapa de classificação de espécies com o MPI-blastn.

Os resultados qualitativos da avaliação da nova arquitetura demonstram que o PANGEA+ melhorou significativamente a precisão dos resultados da etapa de classificação de espécies. Houve um aumento considerável na exatidão dos resultados de classificação, em todos os níveis taxonômicos, chegando a um melhoramento de até ~93% para o nível de super-reino.

Em nível quantitativo, a inclusão de suporte a memória distribuída e otimizações de menor complexidade contribuíram significativamente no desempenho da nova arquitetura, se comparada com o PANGEA. A classificação de espécies com MPI-blast pode ser executada 408 vezes

mais rápido em 384 núcleos, diminuindo o seu tempo de execução de ~37 horas para ~6 minutos. O anexo de informações taxonômicas com o NCBI-TaxCollector, que antes era executado em ~48 horas com o TaxCollector [24], agora pode ser executado em ~23 minutos, sendo 120 vezes mais rápido e consumindo 125 vezes menos memória.

Com o PANGEA+, os dados obtidos de tecnologias de sequenciamento em larga escala podem ser analisados de forma mais rápida e eficaz. Através desta ferramenta e com as otimizações desenvolvidas, foi diminuída a intervenção humana nas fases de leitura e filtragem das sequências, foi melhorada a qualidade dos resultados de classificação de espécies e foi minimizado o tempo de execução da análise. PANGEA+ foi desenvolvido em Perl, C e Java, é livre, de código-fonte aberto e possui suporte Linux e Windows, e está disponível para *download* na página do projeto Bioinfo-Tools: https://github.com/Bioinfo-Tools/PANGEA-plus.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Altschul, S. F.; Gish, W.; Miller, W.*; et al.* "Basic local alignment search tool". Journal of molecular biology, vol. 215, 1990, pp. 403-410.

[2] Bartram, A. K.; Lynch, M. D.; Stearns, J. C.*; et al.* "Generation of multi-million 16S rRNA gene libraries from complex microbial communities by assembling paired-end Illumina reads". Applied and environmental microbiology, vol. 2011, pp.

[3] Benson, D. A.; Karsch-Mizrachi, I.; Clark, K.*; et al.* "GenBank". Nucleic acids research, vol. 2011, pp.

[4] Black, P. E. "big-O notation". Dictionary of Algorithms and Data Structures, vol. 2007, pp.

[5] Brown, C. T.; Davis-Richardson, A. G.; Giongo, A.*; et al.* "Gut Microbiome Metagenomics Analysis Suggests a Functional Model for the Development of Autoimmunity for Type 1 Diabetes". PloS one, vol. 6, 2011, pp. e25792.

[6] Camacho, C.; Coulouris, G.; Avagyan, V.*; et al.* "BLAST+: architecture and applications". BMC bioinformatics, vol. 10, 2009, pp. 421.

[7] Caporaso, J. G.; Kuczynski, J.; Stombaugh, J.*; et al.* "QIIME allows analysis of high-throughput community sequencing data". Nat Methods, vol. 7, 2010, pp. 335-336.

[8] Carver, R. H. "Doing data analysis with spss version 18". Boston, MA: Wadsworth, 2011.

[9] Cha, I. E.; Rouchka, E. C. "Comparison of current BLAST software on nucleotide sequences". In: Parallel and Distributed Processing Symposium, 2005 Proceedings 19th IEEE International, Year, pp. 8 pp.

[10] Chao, A.; Shen, T. "SPADE (species prediction and diversity estimation)". Program and User's Guide Available at http://chao.stat.nthu.edu, vol. 2, 2010, pp.

[11] Chen, Y.; Dumont, M. G.; Neufeld, J. D.; Murrell, J. C. Towards “Focused” Metagenomics: A Case Study Combining DNA Stable-Isotope Probing, Multiple Displacement Amplification, and Metagenomics. Handbook of Molecular Microbial Ecology I: John Wiley & Sons, Inc.; 2011. p. 491-496.

[12] Cole, J. R.; Chai, B.; Farris, R. J.*; et al.* "The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis". Nucleic acids research, vol. 33, 2005, pp. D294-296.

[13] Cole, J. R.; Chai, B.; Farris, R. J.*; et al.* "The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data". Nucleic acids research, vol. 35, 2007, pp. D169-172.

[14] Cole, J. R.; Wang, Q.; Cardenas, E.*; et al.* "The Ribosomal Database Project: improved alignments and new tools for rRNA analysis". Nucleic acids research, vol. 37, 2009, pp. D141-145.

[15] Colwell, R. K. "EstimateS": Robert K. Colwell., 2000.

[16] Dean, J.; Ghemawat, S. "Mapreduce: Simplified data processing on large clusters". Communications of the Acm, vol. 51, 2008, pp. 107-113.

[17] Desantis, T. Z.; Hugenholtz, P.; Larsen, N.*; et al.* "Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB". Applied and environmental microbiology, vol. 72, 2006, pp. 5069-5072.

[18] Desantis, T. Z., Jr.; Hugenholtz, P.; Keller, K.*; et al.* "NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes". Nucleic acids research, vol. 34, 2006, pp. W394-399.

[19] Deshpande, A.; Pontaroli, A. C.; Chaluvadi, S. R.*; et al.* Plant Genetics for Study of the Roles of Root Exudates and Microbes in the Soil. In: Costa de Oliveira A, Varshney RK, editors. Root Genomics: Springer Berlin Heidelberg; 2011. p. 99-111.

[20] Ewing, B.; Green, P. "Base-calling of automated sequencer traces using phred. II. Error probabilities". Genome research, vol. 8, 1998, pp. 186-194.

[21] Ewing, B.; Hillier, L.; Wendl, M. C.; Green, P. "Base-calling of automated sequencer traces using phred. I. Accuracy assessment". Genome research, vol. 8, 1998, pp. 175-185.

[22] Geer, L. Y.; Marchler-Bauer, A.; Geer, R. C.*; et al.* "The NCBI BioSystems database". Nucleic acids research, vol. 38, 2010, pp. D492-496.

[23] Giongo, A.; Crabb, D. B.; Davis-Richardson, A. G.*; et al.* "PANGEA: pipeline for analysis of next generation amplicons". The ISME journal, vol. 4, 2010, pp. 852-861.

[24] Giongo, A.; Davis-Richardson, A. G.; Crabb, D. B.; Triplett, E. W. "TaxCollector: Modifying Current 16S rRNA Databases for the Rapid

Classification at Six Taxonomic Levels". Diversity, vol. 2, 2010, pp. 1015-1025.

[25] Glass, E. M.; Wilkening, J.; Wilke, A.*; et al.* "Using the metagenomics RAST server (MG-RAST) for analyzing shotgun metagenomes". Cold Spring Harbor protocols, vol. 2010, 2010, pp. pdb prot5368.

[26] Graham, R. L.; Woodall, T. S.; Squyres, J. M. "Open MPI: A flexible high performance MPI". Parallel Processing and Applied Mathematics, vol. 3911, 2006, pp. 228-239.

[27] Gropp, W.; Lusk, E.; Doss, N.; Skjellum, A. "A high-performance, portable implementation of the MPI message passing interface standard". Parallel Computing, vol. 22, 1996, pp. 789-828.

[28] Healy, M. D. "Using BLAST for performing sequence alignment". Current protocols in human genetics / editorial board, Jonathan L Haines [et al], vol. Chapter 6, 2007, pp. Unit 6 8.

[29] "HiSeq Systems - Illumina". Capturado em: http://www.illumina.com/systems/hiseq_systems.ilmn, 22/03, 2012.

[30] Huang, X.; Wang, J.; Aluru, S.*; et al.* "PCAP: a whole-genome assembly program". Genome research, vol. 13, 2003, pp. 2164-2170.

[31] Huffman, M. D. "An Improved Approximate 2-Sample Poisson Test". Applied Statistics-Journal of the Royal Statistical Society Series C, vol. 33, 1984, pp. 224-226.

[32] Huson, D. H.; Auch, A. F.; Qi, J.; Schuster, S. C. "MEGAN analysis of metagenomic data". Genome research, vol. 17, 2007, pp. 377-386.

[33] Huson, D. H. "MEGAN 4 - MEtaGenome ANalyzer - Algorithms in Bioinformatics". Capturado em: http://ab.inf.uni-tuebingen.de/software/megan/, 05/23, 2011.

[34] Inc., I. "Paired-End Sequencing - Achieve maximum coverage across the genome". Capturado em: http://www.illumina.com/technology/paired_end_sequencing_assay.ilmn, 2011.

[35] Information, N. C. F. B. "NCBI Taxonomy database". Capturado em: ftp://ftp.ncbi.nih.gov/pub/taxonomy/, 06/01/2012, 2012.

[36] Jaenicke, S.; Ander, C.; Bekel, T.*; et al.* "Comparative and joint analysis of two metagenomic datasets from a biogas fermenter obtained by 454-pyrosequencing". PloS one, vol. 6, 2011, pp. e14519.

[37] Jones, B. V.; Sun, F.; Marchesi, J. R. "Comparative metagenomic analysis of plasmid encoded functions in the human gut microbiome". BMC genomics, vol. 11, 2010, pp. 46.

[38] Kaeberlein, T.; Lewis, K.; Epstein, S. S. "Isolating "uncultivable" microorganisms in pure culture in a simulated natural environment". Science, vol. 296, 2002, pp. 1127-1129.

[39] Kahn, S. D. "On the Future of Genomic Data". Science, vol. 331, 2011, pp. 728-729.

[40] Kielak, A. M.; Van Veen, J. A.; Kowalchuk, G. A. "Comparative analysis of acidobacterial genomic fragments from terrestrial and aquatic metagenomic libraries, with emphasis on acidobacteria subdivision 6". Applied and environmental microbiology, vol. 76, 2010, pp. 6769-6777.

[41] Konstantinidis, K. T.; Braff, J.; Karl, D. M.; Delong, E. F. "Comparative metagenomic analysis of a microbial community residing at a depth of 4,000 meters at station ALOHA in the North Pacific subtropical gyre". Applied and environmental microbiology, vol. 75, 2009, pp. 5345-5355.

[42] Kosakovsky Pond, S.; Wadhawan, S.; Chiaromonte, F.*; et al.* "Windshield splatter analysis with the Galaxy metagenomic pipeline". Genome research, vol. 19, 2009, pp. 2144-2153.

[43] Kuczynski, J.; Stombaugh, J.; Walters, W. A.*; et al.* "Using QIIME to analyze 16S rRNA gene sequences from microbial communities". Current protocols in bioinformatics / editoral board, Andreas D Baxevanis [et al], vol. Chapter 10, 2011, pp. Unit 10 17.

[44] Kurokawa, K.; Itoh, T.; Kuwahara, T.*; et al.* "Comparative metagenomics revealed commonly enriched gene sets in human gut microbiomes". DNA research : an international journal for rapid publication of reports on genes and genomes, vol. 14, 2007, pp. 169-181.

[45] Langmead, B.; Trapnell, C.; Pop, M.; Salzberg, S. L. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome". Genome biology, vol. 10, 2009, pp. R25.

[46] Ledee, D. R.; Booton, G. C.; Awwad, M. H.*; et al.* "Advantages of using mitochondrial 16S rDNA sequences to classify clinical isolates of Acanthamoeba". Investigative Ophthalmology & Visual Science, vol. 44, 2003, pp. 1142-1149.

[47] Li, H.; Ruan, J.; Durbin, R. "Mapping short DNA sequencing reads and calling variants using mapping quality scores". Genome research, vol. 18, 2008, pp. 1851-1858.

[48] Li, R.; Yu, C.; Li, Y.*; et al.* "SOAP2: an improved ultrafast tool for short read alignment". Bioinformatics, vol. 25, 2009, pp. 1966-1967.

[49] Li, W.; Godzik, A. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences". Bioinformatics, vol. 22, 2006, pp. 1658-1659.

[50] Lin, H. S.; Ma, X. S.; Feng, W. C.; Samatova, N. F. "Coordinating Computation and I/O in Massively Parallel Sequence Search". Ieee Transactions on Parallel and Distributed Systems, vol. 22, 2011, pp. 529-543.

[51] Lozupone, C.; Knight, R. "UniFrac: a new phylogenetic method for comparing microbial communities". Applied and environmental microbiology, vol. 71, 2005, pp. 8228-8235.

[52] Lozupone, C.; Lladser, M. E.; Knights, D.*; et al.* "UniFrac: an effective distance metric for microbial community comparison". The ISME journal, vol. 5, 2011, pp. 169-172.

[53] Maidak, B. L.; Cole, J. R.; Lilburn, T. G.*; et al.* "The RDP-II (Ribosomal Database Project)". Nucleic acids research, vol. 29, 2001, pp. 173-174.

[54] Matsunaga, A.; Tsugawa, M.; Fortes, J. "CloudBLAST: Combining MapReduce and Virtualization on Distributed Resources for Bioinformatics Applications". In: eScience, 2008 eScience '08 IEEE Fourth International Conference on, Year, pp. 222-229.

[55] Meyer, F.; Paarmann, D.; D'souza, M.*; et al.* "The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes". BMC bioinformatics, vol. 9, 2008, pp. 386.

[56] Mitra, S.; Schubach, M.; Huson, D. H. "Short clones or long clones? A simulation study on the use of paired reads in metagenomics". BMC bioinformatics, vol. 11 Suppl 1, 2010, pp. S12.

[57] Mitra, S.; Rupek, P.; Richter, D. C.*; et al.* "Functional analysis of metagenomes and metatranscriptomes using SEED and KEGG". BMC bioinformatics, vol. 12 Suppl 1, 2011, pp. S21.

[58] Moretti, C.; Hoang, B.; Hollingsworth, K.*; et al.* "All-Pairs: An Abstraction for Data-Intensive Computing on Campus Grids". Parallel and Distributed Systems, IEEE Transactions on, vol. 21, 2010, pp. 33-46.

[59] Morgan, G. A. "IBM SPSS for introductory statistics : use and interpretation". New York: Routledge, 2011.

[60] Mount, D. W. "Using the Basic Local Alignment Search Tool (BLAST)". CSH protocols, vol. 2007, 2007, pp. pdb top17.

[61] Nawrocki, E. P.; Kolbe, D. L.; Eddy, S. R. "Infernal 1.0: inference of RNA alignments". Bioinformatics, vol. 25, 2009, pp. 1335-1337.

[62] Oehmen, C.; Nieplocha, J. "ScalaBLAST: A scalable implementation of BLAST for high-performance data-intensive bioinformatics analysis". Ieee Transactions on Parallel and Distributed Systems, vol. 17, 2006, pp. 740-749.

[63] Proctor, G. N. "Mathematics of microbial plasmid instability and subsequent differential growth of plasmid-free and plasmid-containing cells, relevant to the analysis of experimental colony number data". Plasmid, vol. 32, 1994, pp. 101-130.

[64] Pruesse, E.; Quast, C.; Knittel, K.*; et al.* "SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB". Nucleic acids research, vol. 35, 2007, pp. 7188-7196.

[65] R Project for Statistical Computing.; R Foundation for Statistical Computing.; Technische Universität Wien. Institut Für Statistik Wahrscheinlichkeitstheorie Und Versicherungsmathematik.*; et al.* R news the newsletter of the R Project. Wien: Technische Universität Wien, Institut für Statistik, Wahrscheinlichkeitstheorie, und Versicherungsmathematik,; 2001.

[66] Rezaei, S.; Monwar, M. M.; Bai, J. "Performance Comparison of MPI-Based Parallel Multiple Sequence Alignment Algorithm Using Single and Multiple Guide Trees". In: Cognitive Informatics, 2006 ICCI 2006 5th IEEE International Conference on, Year, pp. 595-600.

[67] Rudney, J. D.; Xie, H.; Rhodus, N. L.*; et al.* "A metaproteomic analysis of the human salivary microbiota by three-dimensional peptide fractionation and tandem mass spectrometry". Molecular oral microbiology, vol. 25, 2010, pp. 38-49.

[68] Sait, S. M.; Al-Mulhem, M.; Al-Shaikh, R. "Evaluating BLAST Runtime Using NAS-Based High Performance Clusters". In: Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on, Year, pp. 51-56.

[69] Schloss, P. D.; Handelsman, J. "Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness". Applied and environmental microbiology, vol. 71, 2005, pp. 1501-1506.

[70] Schloss, P. D.; Westcott, S. L.; Ryabin, T.*; et al.* "Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities". Applied and environmental microbiology, vol. 75, 2009, pp. 7537-7541.

[71] Shannon, C. E. "A mathematical theory of communication". SIGMOBILE Mob Comput Commun Rev, vol. 5, 2001, pp. 3-55.

[72] Smith, J. A. "RNA Search with Decision Trees and Partial Covariance Models". Computational Biology and Bioinformatics, IEEE/ACM Transactions on, vol. 6, 2009, pp. 517-527.

[73] Steward, G. F.; Preston, C. M. "Analysis of a viral metagenomic library from 200 m depth in Monterey Bay, California constructed by direct shotgun cloning". Virology journal, vol. 8, 2011, pp. 287.

[74] Tringe, S. G.; Rubin, E. M. "Metagenomics: DNA sequencing of environmental samples". Nature reviews Genetics, vol. 6, 2005, pp. 805-814.

[75] Tringe, S. G.; Hugenholtz, P. "A renaissance for the pioneering 16S rRNA gene". Current opinion in microbiology, vol. 11, 2008, pp. 442-446.

[76] Urich, T.; Lanzen, A.; Qi, J.*; et al.* "Simultaneous assessment of soil microbial community structure and function through analysis of the meta-transcriptome". PloS one, vol. 3, 2008, pp. e2527.

[77] Van De Vossenberg, J.; Woebken, D.; Maalcke, W. J.*; et al.* "The metagenome of the marine anammox bacterium 'Candidatus Scalindua profunda' illustrates the versatility of this globally important nitrogen cycle bacterium". Environmental microbiology, vol. 2012, pp.

[78] Wenbo, J.; Wiese, K. C. "Combined covariance model for non-coding RNA gene finding". In: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2011 IEEE Symposium on, Year, pp. 1-5.

[79] Wuyts, J.; Van De Peer, Y.; Winkelmans, T.; De Wachter, R. "The European database on small subunit ribosomal RNA". Nucleic acids research, vol. 30, 2002, pp. 183-185.

[80] Wuyts, J.; Perriere, G.; Van De Peer, Y. "The European ribosomal RNA database". Nucleic acids research, vol. 32, 2004, pp. D101-103.

[81] Zerbino, D. R.; Birney, E. "Velvet: algorithms for de novo short read assembly using de Bruijn graphs". Genome research, vol. 18, 2008, pp. 821-829.