UNIVERSIDADE FEDERAL DE CIÊNCIAS DA SAÚDE DE PORTO ALEGRE

PROGRAMA DE PÓS-GRADUAÇÃO EM

CIÊNCIAS DA SAÚDE

Gustavo Henrique Cervi

Pipeline Metagenômico Com o Uso de Algoritmo de Compressão Lossy e Matching Heurístico Não-Determinístico

Porto Alegre

2022

Gustavo Henrique Cervi

Pipeline Metagenômico Com o Uso de Algoritmo de Compressão Lossy e Matching Heurístico Não-Determinístico

Tese submetida ao Programa de Pós-Graduação em Ciências e Saúde da Fundação Universidade Federal de Ciências da Saúde de Porto Alegre como requisito para a obtenção do grau de Doutor.

Orientadora: Profa. Dra. Claudia Elizabeth Thompson

Coorientadora: Profa. Dra. Cecília Dias Flores

Porto Alegre

2022

Catalogação na Publicação

Cervi, Gustavo Henrique
Pipeline Metagenômico Com o Uso de Algoritmo de Compressão Lossy e Matching Heurístico Não-Determinístico / Gustavo Henrique Cervi. -- 2022.
94 f. : 30 cm.

Tese (doutorado) -- Universidade Federal de Ciências da Saúde de Porto Alegre, Programa de Pós-Graduação em Ciências da Saúde, 2022.

Orientador(a): Claudia Elizabeth Thompson ; coorientador(a): Cecilia Dias Flores.

1. Metagenômica. 2. Diagnóstico. 3. Computação. I. Título.

AGRADECIMENTO

Este autor agradece a todos os envolvidos neste projeto, em especial:

A minha orientadora Claudia Thompson pelo acompanhamento.

A minha coorientadora Cecília Flores pelo apoio de sempre.

Aos colegas das disciplinas pelo companheirismo nos estudos e nos trabalhos que tivemos juntos.

A todos professores e funcionários da universidade pelo empenho nas atividades.

Aos amigos que direta ou indiretamente apoiaram esta jornada.

Aos meus pais que estão comigo desde sempre.

A minha esposa Fátima Cervi pelo apoio incondicional.

RESUMO

Introdução: o processamento de dados metagenômicos é um grande desafio para a genética e bioinformática. De forma geral, o grande volume de dados combinado com a natureza das mutações pode impactar fortemente no desempenho das aplicações de alinhamento de sequências de nucleotídeos. Nos últimos anos, o estudo metagenômico evoluiu para o diagnóstico de agentes etiológicos, especialmente em casos de infecções de difícil descoberta e tratamento. Sabe-se que quanto antes houver o diagnóstico do agente infectante, maiores são as chances de desfecho positivo para o paciente. O processo metagenômico faz uso intenso da computação e o avanço das técnicas computacionais trazem benefícios práticos no tratamento das infecções. **Objetivo:** esta tese propõe um mecanismo de redução de banco de dados, com perda (*lossy*), de forma que o volume de nucleotídeos seja otimizado, sem prejudicar a sensibilidade da busca pelos organismos de interesse. Esta compactação é utilizada para acelerar o processo de combinação de sequências (alinhamento) e produz resultados mais sensíveis em menor espaço de tempo. **Metodologia:** a técnica explora a característica natural do DNA/RNA onde um ou mais nucleotídeos modificados (mutações), removidos e adicionados (*indels*) não significam, obrigatoriamente, um identificador biológico/genético divergente de sua base de referência. Esta característica, aliada ao alfabeto reduzido de quatro letras (A, C, G e T), é peça-chave para a construção da técnica computacional proposta. Apesar de poucas letras, a combinação entre os quatro nucleotídeos é o código-fonte de todo ser vivo, possuindo milhões de combinações. Sequências de DNA podem conter milhares ou até milhões de nucleotídeos sendo que o DNA humano, por exemplo, possui mais de 3 bilhões de bases. A técnica proposta consiste na construção de uma espécie de onda. Os nucleotídeos de mesma base modulam a frequência e produzem sequências de mesmo período. **Resultados:** ao final do processo de redução da base, os experimentos mostram que há importante compactação na massa de dados (80% em alguns casos) e, por consequência, melhor performance como um todo. Esta redução significa que os processos computacionais envolvidos com os algoritmos de alinhamento utilizarão menor tempo de CPU (menos instruções) e também menos memória RAM, permitindo que mais dados possam ser computados no mesmo intervalo de tempo. Em experimentos comparativos com a ferramenta Blast, no

alinhamento de um sequenciamento metagenômico (*run*), o resultado mostra uma performance acelerada em 10x, podendo ainda ampliar centenas de vezes se considerar outras estratégias como tabelas hash, já utilizadas por outras ferramentas de alinhamento metagenômico.

Palavras-chave: metagenômica, bioinformática, diagnóstico, algoritmo

ABSTRACT

Introduction*: metagenomic data processing is a major challenge for genetics and bioinformatics. The large volume of data combined with the nature of mutations can strongly impact the performance of nucleotide sequence alignment applications. In recent years, the metagenomic study has evolved towards the diagnosis of etiologic agents, especially in cases of infections that are difficult to discover and treat. It is known that the earlier the diagnosis of the infecting agent is made, the greater the chances of a positive outcome for the patient. The metagenomic process makes intensive use of computing and the advancement of computational techniques brings practical benefits in the treatment of infections.* ***Objective****: this thesis proposes a lossy database reduction mechanism, optimizing the volume of nucleotides, without impairing the sensitivity of the search for organisms of interest. This compression is used to speed up the process of combining sequences (alignment) and produces more sensitive results in a shorter amount of time.* ***Methodology****: the technique explores the natural characteristic of DNA/RNA where one or more modified nucleotides (mutations), removed and added (indels) do not necessarily mean a biological/genetic identifier that diverges from its reference base. This feature, combined with the reduced alphabet of four letters (A, C, G and T), is a key element for the construction of the proposed computational technique. Despite having few letters, the combination between the four nucleotides is the source code of every living being, having millions of combinations. DNA sequences can contain thousands or even millions of nucleotides and human DNA, for example, has more than 3 billion bases. The proposed technique consists in the construction of a kind of wave. Nucleotides with the same base modulate frequency and produce sequences of the same period.* ***Results****: at the end of the database reduction process, the experiments show that there is important compression in the data mass (80% in*

some cases) and, consequently, better performance as a whole. This reduction means that the computational processes involved with the alignment algorithms will use less CPU time (fewer instructions) and also less RAM memory, allowing more data to be computed in the same time interval. In comparative experiments with the Blast tool, in the alignment of a metagenomic sequencing (run), the result shows an accelerated performance by 10x, and can still be magnified hundreds of times if we consider other strategies such as hash tables, already used by other metagenomic alignment tools.

Key words: metagenomics, bioinformatics, diagnostics, algorithm

LISTA DE FIGURAS

Fig. 1 – Publicações por ano, segundo o PubMed, pesquisando-se pelo termo “metagenomics” pode demonstrar um crescente interesse da comunidade científica pelo tema. Extraído em meados de 2022. 14

Fig. 2 – Evolução do banco de dados Genbank, mantido pelo NCBI, pode-se observar o aumento na curva de depósitos de dados genômicos. Este aumento pode representar a crescente dificuldade do tratamento de dados. 15

Fig. 3 – Amostras de LCR em tubos. A apresentação do líquido cristalino sugere estado hígido. Imagem ilustrativa.....16

Fig. 4 – Exemplo ilustrativo do sequenciador Illumina MiSeq. Este sequenciador é utilizado em diversas publicações por ser relativamente acessível a comunidade científica.....17

Fig. 5 – Exemplo ilustrativo do kit de reagente do fabricante Illumina. A utilização do kit faz parte do protocolo de sequenciamento, conforme o fabricante.....18

Fig. 6 – Gramática de um arquivo FASTQ. Exemplo ilustrativo demonstrando a construção do arquivo conforme sua especificação formal.....20

Fig. 7 – Exemplo de arquivo FASTQ formatado conforme especificação da gramática.....20

Fig. 9 – Dois tipos básicos de mutações: nas transições ocorre a troca das bases de mesma classe (purina ou pirimidina) enquando nas transversões ocorre troca de bases de classes diferentes.....24

Fig. 10 – Alinhamento entre sequências de DNA. No exemplo ilustrativo pode-se observar os matches (|), indels (-) e os mismatches (ausência de correspondente |).. 24

Fig. 11 – Exemplo de dendrograma filogenético. A representação é formada por uma árvore contendo galhos e folhas, os galhos representam as relações de “parentalidade” e “herança”.....25

Fig. 12 – Programação dinâmica. Na ilustração pode-se observar o uso de uma matriz “mn” onde o caminho grifado representa a menor distância entre as sequências em alinhamento.....28

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
AWS	*Amazon Web Services*
BLAST	*Basic Local Alignment Search Tool*
DDBJ	*DNA Data Bank of Japan*
DNA	Ácido desoxirribonucleico
DVE	Derivação Ventricular Externa
EMBL	*European Molecular Biology Laboratory*
FASTA	*Fast All*
FASTQ	*Fast (All) with Quality*
FTP	*File Transfer Protocol*
FPGA	*Field-Programmable Gate Array*
GCP	*Google Cloud Platform*
GPU	*Graphics Processing Unit*
HTTP	*Hypertext Transfer Protocol*
LCR	Líquido cefalorraquidiano
NCBI	*National Center for Biotechnology Information*
NGS	*Next Generation Sequencing*
PCR	*Polymerase Chain Reaction*
RNA	Ácido ribonucleico
SNC	Sistema Nervoso Central
SRA	*Sequence Read Archive*
WGS	*Whole Shotgun Sequencing*

SUMÁRIO

1 INTRODUÇÃO....11
2 REFERENCIAL TEÓRICO....13
2.1 METAGENÔMICA....13
2.2 PROCESSO METAGENÔMICO....15
2.2.1 Processo bioquímico....16
2.2.2 Processo computacional....19
2.2.2.1 Preparo do arquivo....19
2.2.2.2 Aparação e remoção de adaptadores....21
2.2.2.3 Remoção de sequências idênticas....21
2.2.2.4 Remoção do DNA do hospedeiro....22
2.2.2.5 Alinhamento e combinação....22
2.2.2.6 Resultado do processamento....24
2.2.2.7 Bases de dados....25
2.2.2.8 Algoritmos....27
2.2.2.9 Softwares de alinhamento e anotação....29
2.2.2.10 ....Inteligência artificial 32
2.3 BENCHMARK E RELATO DE EXPERIMENTAÇÃO....34
3 OBJETIVOS....37
3.1 GERAIS....37
3.2 ESPECÍFICOS....37
4 ARTIGOS CIENTÍFICOS E CONTRIBUIÇÕES....38
CAPÍTULO DE LIVRO INTERNACIONAL....52
5 CONCLUSÃO....63
REFERÊNCIAS BIBLIOGRÁFICAS....65
ANEXO I – COMITÊ DE ÉTICA EM PESQUISA....71
ANEXO II – REGISTRO JUNTO AO INPI....72
ANEXO III – INSTRUÇÕES – PERIÓDICO DATABASE....74
ANEXO IV – EXPERIMENTOS E DADOS....87

1 INTRODUÇÃO

A etiologia de infecções do Sistema Nervoso Central (SNC) representa um grande desafio clínico, já que em mais de 50% dos casos não é possível a realização de um diagnóstico etiológico, de acordo com dados americanos (Glaser *et al*., 2006; Dugue 2022). Em nível mundial, essas infecções são geralmente endêmicas, classificadas como meningites ou encefalites, e ocorrem em frequências alarmantes em vários países. Tais infecções, em geral, levam a enormes gastos no sistema de saúde pública em diversos locais no mundo. Como exemplo, nos Estados Unidos, há cerca de 20.000 hospitalizações por ano em razão de encefalites, resultando em 200.000 dias de internação e um custo hospitalar de dois bilhões de dólares por ano (Kiyani *et al*., 2020).

As infecções do SNC são comumente confundidas com outras doenças e há uma diversidade de agentes etiológicos que não são facilmente identificáveis. Há estimativas de que a identificação do agente etiológico ocorra em somente 25% dos casos, mesmo considerando excelentes facilidades laboratoriais (Rotbart, 2000, Dugue 2022). Essas infecções são emergências neurológicas e possuem alta morbidade e mortalidade. Consequentemente, é fundamental que sejam realizados estudos epidemiológicos e desenvolvidos novos métodos diagnósticos e estratégias terapêuticas a fim de se diminuir os custos para o sistema de saúde no Brasil.

As infecções do SNC adquiridas na comunidade têm um desfecho desfavorável em cerca de 30% dos casos, incluindo sequelas severas ou até mesmo a morte em razão da dificuldade de identificação do agente etiológico e, consequente, tratamento inadequado (Erdem *et al*., 2017). Além desse tipo de infecção, há infecções do SNC ainda mais difíceis de tratar. Entre elas, encontram-se aquelas que ocorrem em pacientes com Derivação Ventricular Externa (DVE) ou outros implantes neurocirúrgicos, que possuem taxas de infecção de até 15% (Darouiche *et al*., 2004), pacientes imunocomprometidos, pacientes com espondilodiscite, abscesso espinhal epidural ou osteomielite vertebral e aqueles com infecções do SNC pós-operatórias (com uma incidência de até 10%, McClelland III and Hall, 2007). Essas infecções severas resultam em altas taxas de mortalidade e bactérias resistentes a antibióticos estão frequentemente relacionadas a esses

casos (van de Beek *et al*., 2016; Tsioutis *et al*., 2017). No entanto, os guidelines clínicos disponíveis focam no tratamento empírico de infecções causadas por microorganismos suscetíveis aos antibióticos mais comumente utilizados. Isso reforça a importância do diagnóstico efetivo e do desenvolvimento de novas estratégias terapêuticas.

Ventriculites, meningites, sepse ou abscessos cerebrais são infecções severas que podem ocorrer depois da colocação de Derivações Ventriculares Externas (DVEs) (Beer *et al*., 2008; Tsioutis *et al*., 2017). Cabe ressaltar que todos os implantes médicos estão sujeitos a colonização por micro-organismos e infecção (Bryers, 2008; Treter e Macedo, 2011; Busscher *et al*., 2012), sendo que o risco de infecção está associado ao déficit imunológico na interface implante-hospedeiro e leva a uma reduzida habilidade de eliminar os micro-organismos próximos ao biomaterial (Rochford *et al*., 2012). Assim, como essas infecções estão relacionadas a biofilmes bacterianos, o diagnóstico e tratamento se tornam difíceis, levando a altos custos hospitalares, novas intervenções cirúrgicas e altas taxas de mortalidade e morbidade (Conen *et al*., 2017).

O aumento da prevalência de pacientes imunocomprometidos tem levado a um aumento significativo na proporção de pacientes com infecções de difícil diagnóstico e tratamento do SNC. Como consequência da dificuldade na identificação dos agentes etiológicos, protocolos clínicos ineficientes e inadequados são administrados aos pacientes. No Brasil, os guidelines clínicos para tratamento de meningites e encefalites incluem a administração intravenosa de vancomicina, meropenem, fluconazol e aciclovir. Assim, considerando que é necessário um tratamento de 21 dias, o custo hospitalar somente considerando os medicamentos é de cerca de 90 mil reais, de acordo com dados do Hospital Cristo Redentor do Grupo Hospitalar Conceição (Porto Alegre, Rio Grande do Sul).

As novas tecnologias de sequenciamento e recursos computacionais de alto desempenho têm o potencial de contribuir para o entendimento e identificação dos agentes etiológicos das infecções de difícil diagnóstico e tratamento do SNC. Em 2016, um projeto piloto desenvolvido por um grupo de médicos e pesquisadores da Johns Hopkins University utilizou sequenciamento de metagenoma e um novo pipeline de análise computacional para detectar a presença de micro-organismos

patogênicos em biópsias de cérebro e medula de 10 pacientes com suspeita de infecção neurológica (Salzberg *et al*., 2016). Eles mostraram que essa estratégia é uma alternativa efetiva em relação aos métodos tradicionalmente utilizados para identificação de agentes etiológicos de infecções do SNC.

Ao decorrer dos anos, diversos autores publicaram soluções computacionais explorando os mais diversos ramos da ciência da computação. Vacek (2011) trabalhou em processadores específicos de múltiplos núcleos. Mahram *et al* (2012) e Wu *et al* (2019), entre outros, construíram alternativas utilizando chips de portas lógicas programáveis “Field-Programmable Gate Array” (FPGA) e obtiveram importantes ganhos de performance. Liu *et al* (2013), Kobus *et al* (2017), entre outros, trabalharam em soluções que exploram os processadores gráficos “Graphics Processing Unit” (GPU) e, com auxílio de fabricantes como a Nvidia (cuda cores), obtiveram ganhos expressivos de performance.

Apesar de todos os esforços empregados por vários pesquisadores, o crescente volume de dados, advindos de novas tecnologias da engenharia biomolecular, trazem novos desafios na luta contra o tempo – especialmente quando se trata de um paciente precisando de um diagnóstico. Esta tese tem como objetivo a construção de um mecanismo de aceleração da etapa de alinhamento de sequências de DNA em um *pipeline* metagenômico experimental para diagnóstico de infecções.

2 REFERENCIAL TEÓRICO

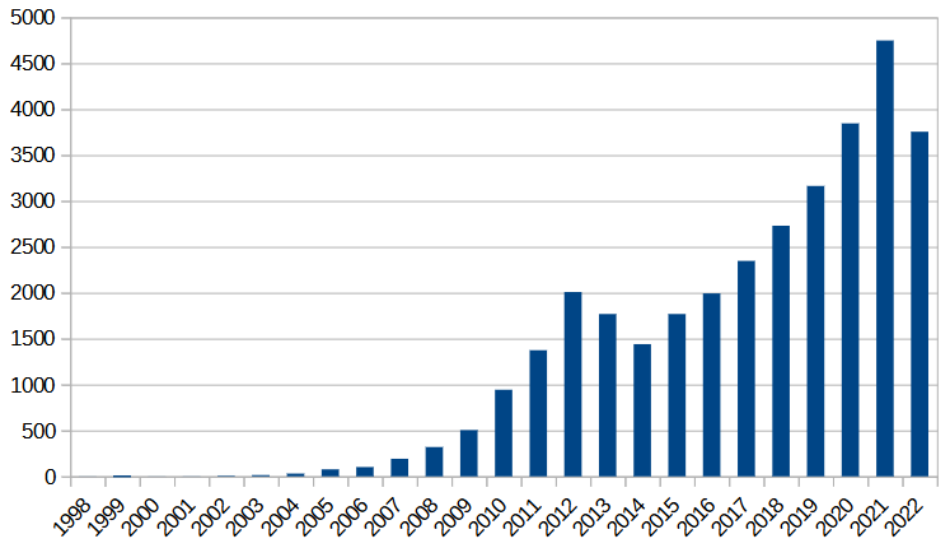
2.1 METAGENÔMICA

A etimologia da palavra metagenômica remete ao prefixo grego “meta” que significa “transcendente”, ou seja, aquilo que excede os limites normais, como define Council *et al* (2007): "*Like genomics itself, metagenomics is both a set of research techniques, comprising many related approaches and methods, and a research field. In Greek, meta means “transcendent*”. Hunter *et al*. (2013) explica que o termo metagenoma, significando "*the collective genomes of [...] microflora*", foi cunhado por Handelsman *et al* (1998) quando o uso da técnica ainda era prejudicada pela falta de recursos laboratoriais e computacionais. Garrido-Cardenas (2017) define metagenômica como uma técnica ou conjunto de técnicas com o objetivo de

determinar a população microbial que se encontra em determinado ambiente. Hugenholtz *apud* Kunin (2008) considera que a metagenômica é uma derivação da genômica microbial convencional, porém ultrapassando a exigência da obtenção de uma cultura pura para o sequenciamento. A evolução da definição de metagenômica, através dos anos, foi refinando na medida que novas tecnologias foram desenvolvidas, porém o cerne permanece o mesmo, podendo ser definido como uma técnica que permite a identificação de agentes biológicos através da coleta de material genético de um ambiente, processado em laboratório e posteriormente computado utilizando algoritmos de análise de dados.

A história da metagenômica inicia em 1676 quando Leeuwenhoek reportou suas observações sobre a microbiota oral, passando pelo isolamento de culturas por Robert Koch em 1888. Fred Sanger desenvolveu o sequenciamento de DNA em 1977, Kary Mullis desenvolveu a técnica PCR em 1980. Giovannoni *et al* executou o primeiro estudo utilizando rRNA 16S em 1990 e em 1998 Handelsman *et al* propõe o termo metagenômica, conforme artigo "*The Road to Metagenomic*" de Escobar-Zepeda *et al* (2015). Na medida que o desenvolvimento da tecnologia NGS (*Next Generation Sequencing*) foi desenvolvida, no início da década de 2000, pode-se observar uma curva acentuada na quantidade de publicações relacionadas a metagenômica (Fig. 1).

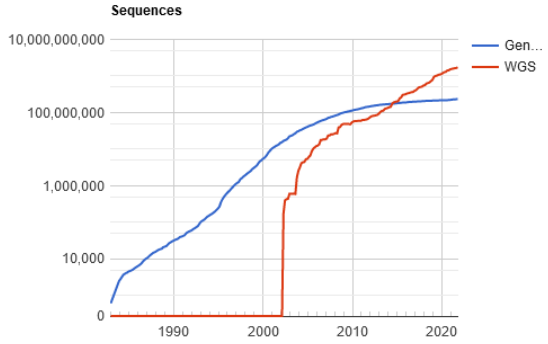
Fig. 1 – Publicações por ano, segundo o PubMed, pesquisando-se pelo termo "metagenomics" pode demonstrar um crescente interesse da comunidade científica pelo tema. Extraído em meados de 2022.



Fonte: PubMed, 2021.

Acompanhando a evolução das tecnologias de sequenciamento, os bancos de dados de sequências genéticas também cresceram de forma acentuada, como se pode ver no gráfico extraído da página de estatísticas do NCBI em 2021, na Fig. 2.

Fig. 2 – Evolução do banco de dados Genbank, mantido pelo NCBI, pode-se observar o aumento na curva de depósitos de dados genômicos. Este aumento pode representar a crescente dificuldade do tratamento de dados.



Fonte: página de estatísticas do Genbank/NCBI.

Atualmente existem diversos fabricantes de máquinas sequenciadoras de DNA, entre os principais encontram-se as empresas Roche, Illumina, Life Technologies, Beckman Coulter, Pacific Biosciences, MGI/BGI e Oxford Nanopore Technologies. Estas empresas investem em pesquisa e desenvolvimento de tecnologias e cada fabricante possui sua metodologia com resultados distintos, em alguns casos mais lentos e precisos e em outros mais rápidos e menos acurados. Xiao *et al* (2016) publicou um artigo com uma comparação de performance de cada tecnologia, destacando as métricas: tamanho da leitura, leituras por execução, tamanho do conjunto de dados gerado, tempo de execução, qualidade do resultado, custo por sequenciamento e custo da máquina de sequenciamento.

2.2 PROCESSO METAGENÔMICO

A exploração da tecnologia de sequenciamento massivamente paralelo possibilitou o avanço de diversas frentes relacionadas às ciências da vida, estudos de materiais orgânicos e diagnóstico de doenças causadas por infecção de agentes biológicos (Schlaberg, 2017). Para ilustrar o processo de sequenciamento, pode-se

observar a técnica do sequenciamento por síntese utilizado pela empresa Illumina no equipamento HiSeq 2000.

2.2.1 Processo bioquímico

Os processos iniciam na etapa da colheita de material biológico de um ambiente, no caso de um diagnóstico de neuroinfecção, pode-se utilizar, como exemplo, uma amostra de líquido cefalorraquidiano (LCR) (Fig. 3), também chamado de líquor ou fluido cerebroespinhal. Este líquido, em seu estado hígido, é cristalino e estéril, encontrado na medula espinhal e no espaço subaracnóideo. Um adulto normal possui entre 125ml e 150ml de LCR e as funções naturais deste líquido estão relacionadas a proteção mecânica do cérebro, atuando como um amortecedor de impactos, e como meio de transporte de nutrientes e descarte de tecido cerebral (Wright, 2012).

Fig. 3 – Amostras de LCR em tubos. A apresentação do líquido cristalino sugere estado hígido. Imagem ilustrativa.

Fonte: James Heilman, MD. 2021. (Creative Commons)

Em eventos de infecção de sistema nervoso central, o LCR apresenta alterações específicas como a xantocromia (tonalidade amarelada), podendo significar meningite viral, varicela-zóster, enterovírus, infecções por fungos cryptococcus e histoplasma, além de outros patógenos como doença de Lyme,

sífilis, sarampo, John Cunningham e Whipple (Wright, 2012). A colheita deste material é feita através de punção lombar, entre o disco das vértebras L3 e L4, com o paciente em decúbito lateral (Wright, 2012). Em seguida, o material deve ser enviado ao laboratório para a extração de material genético. As técnicas de extração constituem na “quebra” do núcleo da célula através de processos químicos, térmicos e mecânicos. Existem “kits” laboratoriais específicos para este fim, produzidos por diversos fabricantes (Gutiérres-Lucas, 2014). O material genético extraído das células pode conter DNA de diversos agentes biológicos, além do próprio paciente em estudo. Ao final do preparo da amostra, o material é inserido no recipiente da máquina e posteriormente inicia-se o processo de sequenciamento. Equipamentos ilustrados na Fig. 4 e na Fig. 5.

Fig. 4 – Exemplo ilustrativo do sequenciador Illumina MiSeq. Este sequenciador é utilizado em diversas publicações por ser relativamente acessível a comunidade científica.

Fonte: o fabricante.

Fig. 5 – Exemplo ilustrativo do kit de reagente do fabricante Illumina. A utilização do kit faz parte do protocolo de sequenciamento, conforme o fabricante.

Fonte: o fabricante.

O processo empregado pelo equipamento HiSeq, aqui utilizado como exemplo, é denominado sequenciamento por síntese (Illumina, 2021). O DNA é "quebrado" em pequenas partes e posteriormente sintetizadas com nucleotídeos eletricamente carregados. A cada nova ligação, sensores captam as cargas liberadas pelos nucleotídeos e a máquina armazena a sequência gerada. Esta técnica permite a análise chamada de *Whole Genome Shotgun Sequencing* ou WGS. Os parâmetros de performance deste sequenciador é publicado pelo fabricante e é caracterizado por (Illumina, 2021):

- Tamanho do *read*: o processo de sequenciamento massivamente paralelo é performado "quebrando" o DNA em pequenas partes que são "lidas" pela máquina através de um processo físico-químico, cada leitura é denominada "*read*". Este modelo trabalha com leituras de 25 a 300 bases por *read*. O processo inteiro é chamado de "*run*".
- Tempo total de sequenciamento: pode variar, neste modelo, entre ~5,5h até ~56h, dependendo do tamanho dos *reads* e do kit de reagente utilizado.
- Saída (*output*): é a quantidade de dados que a máquina produz ao final do processo, podendo variar entre 750 Mb (milhões de bases) até 15 Gb (bilhões de bases).
- Escore de qualidade: é a probabilidade de uma base estar incorreta, é calculada através da fórmula $Q = -10\log_{10}(e)$. Um escore acima de 30 significa que uma em 1.000 bases pode estar incorreta, ou 99,9% das leituras estão

corretas.

Esta máquina produz dados com escore superior a Q30 para todos os kits de reagentes (Illumina, 2021).

2.2.2 Processo computacional

O processo computacional para a obtenção dos agentes biológicos encontrados na amostra da pesquisa possui várias etapas, entre elas pode-se se destacar as seguintes:

1. Preparo do arquivo;
2. Remoção de *reads* de baixa qualidade;
3. Recorte de sequências (*trim*);
4. Remoção de *reads* idênticos (duplicados);
5. Remoção de DNA do hospedeiro (no caso do LCR, DNA humano);
6. Alinhamento e comparação com bancos de dados de referência;
7. Relatório final.

2.2.2.1 Preparo do arquivo

Os sequenciadores genéticos produzem arquivos computacionais contendo os *reads* obtidos no processo de leitura. Muitas vezes o tamanho do arquivo ultrapassa 30 GB (bilhões de bytes). O formato do arquivo resultante depende do fabricante, mas pode-se considerar que existe um padrão na comunidade científica chamado FASTQ e, de modo geral, os fabricantes fornecem ferramentas para a conversão dos seus formatos específicos para este padrão.

O formato FASTQ é um arquivo comum, do tipo texto ASCII[1], que possui quatro linhas por *read*: 1) identificação e descrição do read; 2) sequência de nucleotídeos; 3) identificação e descrição do read (opcional) e 4) dados de qualidade. A Fig. 6 apresenta a gramática padrão para o formato de arquivo FASTQ enquanto a Fig. 7 mostra um pequeno fragmento do arquivo, com as quatro linhas por *read* (FASTQ, 2021).

1: *American Standard Code for Information: tabela utilizada para codificar caracteres alfanuméricos e especiais em números inteiros, utilizado para interoperabilidade de dados entre diferentes arquiteturas.*

Fig. 6 – Gramática de um arquivo FASTQ. Exemplo ilustrativo demonstrando a construção do arquivo conforme sua especificação formal.

```
<FASTQ>     :=    <block>+
<block>     :=    @<seqname>\n<seq>\n+[<seqname>]\n<qual>\n
<seqname>   :=    [A-Za-z0-9_.:-]+
<seq>       :=    [A-Za-z\n\.~]+
<qual>      :=    [!-~\n]+
```

Fonte: FASTQ Format (2021).

Fig. 7 – Exemplo de arquivo FASTQ formatado conforme especificação da gramática.

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

Fonte: os autores.

A quarta linha de cada bloco contém caracteres que representam o escore de qualidade Q, também chamado de “phred quality”. Um arquivo resultante de um sequenciamento pode conter centenas de milhares de blocos, cada um representando um *read* (FASTQ, 2021). O escore de qualidade Phred pode ser calculado segundo a fórmula:

$$Q = -10 \log_{10} P \qquad \textit{ou} \qquad P = 10 \frac{-q}{10}$$

Apesar de não ser uma regra, escores inferiores a Q30 (99,9% de probabilidade de acerto) podem ser considerados de baixa qualidade e, dependendo da análise e do estudo, podem ser descartados para evitar ruídos nos processos seguintes (FASTQ, 2021).

Os motivos que levam a uma determinada leitura ter alto ou baixo escore de qualidade são variados e geralmente relacionados a sobreposição de captações entre diferentes nucleotídeos, neste caso a máquina considerara a possibilidade de não ser uma leitura confiável e reduz o escore do *read*.

Dependendo do fabricante e da tecnologia utilizada cada *read* pode conter até 300 bases (ou mais de 1000 bases em arquiteturas específicas). Eventualmente determinadas tecnologias podem resultar em leituras mais precisas que outras.

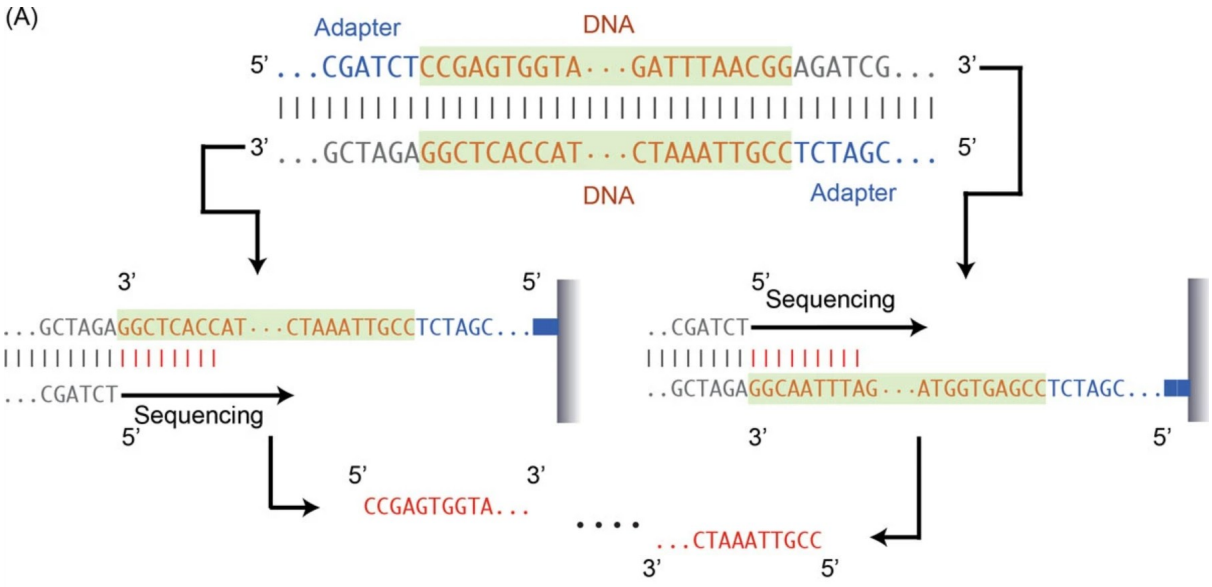
A etapa de eliminação de *reads* de baixa qualidade é determinística e o pesquisador pode variar as métricas limitadoras (recorte das sequências e filtro de escore Q). Este processo possui complexidade O(n) e pode agilizar de forma

importante a etapa de alinhamento – que muitas vezes têm complexidade O(mn) ou até maior. A filtragem de dados de baixa qualidade é um processo trivial onde os *reads* contendo leituras consideradas ruins são removidas do arquivo. A classificação de qualidade é feita pela própria máquina sequenciadora e cabe ao pesquisador definir seus parâmetros. Estes dados podem ser obtidos na quarta linha de cada bloco do arquivo FASTQ.

2.2.2.2 Aparação e remoção de adaptadores

O processo de apara (*trim*) e remoção de adaptadores permite que trechos de nucleotídeos que não fazem parte do DNA sequenciado sejam removidos (Li *et al*, 2015). Estes trechos são inseridos no processo de síntese dos *reads* e podem gerar ruídos no alinhamento de bases com as referências. Conforme se pode observar na Fig. 8, os adaptadores são sequenciados junto com o DNA contido nas amostras, no entanto, além de não representarem o organismo sequenciado, podem influenciar negativamente nos resultados dos alinhamentos.

Fig. 8 – Exemplo ilustrativo de uso de adaptadores no sequenciamento de DNA. Adaptadores são peças importantes para iniciar o processo de síntese.



Fonte: Li *et al* (2015)

2.2.2.3 Remoção de sequências idênticas

A natureza do sequenciamento *shotgun* inevitavelmente resultará em leituras idênticas e duplicadas, isso ocorre porque é esperado que os trechos de DNA que

foram quebrados sejam duplicados nas etapas bioquímicas do preparo da amostra. Estes *reads* não contribuem na pesquisa, devendo ser eliminados para não gerar custo computacional desnecessário, uma vez que o processo – neste ponto – é determinístico, ou seja, leituras idênticas produzem resultados idênticos (Cogo, 2021). Para fins de comparação, este processo possui complexidade O(n). A estratégia computacional normalmente adotada é a inserção das sequências em uma lista “hash”, naturalmente contendo o *constraint* “*unique*” em memória. Outra possibilidade é a inserção de chaves em um dicionário, eliminando os *reads* idênticos.

2.2.2.4 Remoção do DNA do hospedeiro

A remoção de DNA do hospedeiro[2] aumenta a performance do processo, sendo provável que nos resultados se encontre DNA do paciente. Estes *reads* também são desnecessários nas etapas seguintes. Em contrapartida, é possível que um agente infeccioso contenha *reads* idênticos a um genoma de referência humano, neste caso a remoção do DNA do hospedeiro poderia resultar em um erro do tipo 2 (falso negativo). A remoção de DNA humano é feita de forma computacional, alinhando os *reads* com um genoma de referência[3] ou então em etapa bioquímica pré-sequenciamento (Shi, 2022).

2.2.2.5 Alinhamento e combinação

Ao término da etapa de preparo dos dados, filtragem de reads de baixa qualidade e remoção de sequências, os dados estão prontos para a etapa de alinhamento. Este processo é relativamente complexo, dada a natureza do DNA sequências computacionalmente diferentes podem ser biologicamente idênticas. O estudo das mutações genéticas é uma área complexa da genética molecular e possui várias causas que podem ser espontâneas (ex: tautomeria, desaminação, oxidação e alquilação) devido a erros de replicação/reparação ou até devido a indução química ou radioativa (Montelone, 1998).

Para esta tese, considera-se as características estruturais das mutações, ou

---

2: *O estudo metagenômico pode ser utilizado para diversos fins como análise de solo, pesquisas relacionadas à microbiologia de petróleo e controle de agentes infecciosos em ambientes de saúde, entre outros. O caso em tela trata de um diagnóstico em humanos.*

3: *Os genomas de referência são curados pelas principais bases de dados, como exemplo se pode citar o Genome Reference Consortium Human Build 38 patch release 14 (GRCh38.p14).*

seja, o ponto de interesse é: quais efeitos estruturais uma mutação pode interferir em uma sequência de nucleotídeos. Considerando este motivo, pode-se dividir em duas categorias: mutações de grande escala (cromossomos) e de pequena escala (nucleotídeos). O objetivo de combinar pequenas sequências determina, por corolário, que o foco deste estudo seja as pequenas mutações. As mutações de pequena escala dividem-se entre: pontuais, inserção e deleção (Rahman, 2017) e podem ser listadas como:

- Transição: ocorre a troca de uma base purina (A/G) por outra base purina ou uma base pirimídica (C/T) por outra base pirimídica;
- Transversão: ocorre a troca de uma base purina (A/G) por uma pirimídica (C/T);
- Inserções: um ou mais nucleotídeos são inseridos na sequência original;
- Deleções: um ou mais nucleotídeos são removidos da sequência original.

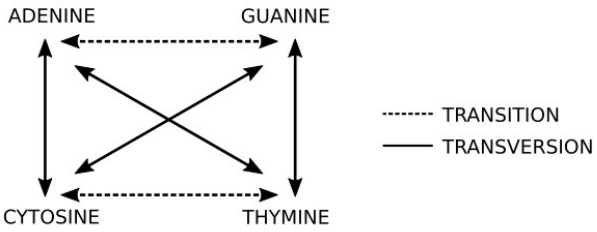
Transições e transversões podem ser consideradas como “silenciosas” quando a troca do nucleotídeo resulta na codificação do mesmo aminoácido. Se a troca do nucleotídeo resulta em um aminoácido diferente, diz-se que há uma mutação “missense”. Caso a mutação resulta em um códon de parada (*stop codon*), então chama-se mutação “sem sentido”.

As inserções e deleções (chamadas de “indels”) podem ser causadas por erros durante a replicação das sequências e resultam em um deslocamento (*shift)* e eventual alteração do produto genético resultante.

Apesar destas mutações possuírem relativa importância nas características genéticas do sujeito de pesquisa, elas não significam necessariamente uma diferença determinante entre o objeto de pesquisa e suas sequências de referência. Mutações são naturais e comuns e o dois agentes etiológicos idênticos podem ter pequenas alterações em comparação com os agentes de referência (refseqs[4]). A Fig. 9 ilustra as diferenças entre transições e transversões.

4: *Refseqs são sequências de referência depositadas nos bancos de dados. Podem ser parciais (genes) ou completos (genomas). São utilizados para comparação com dados obtidos de sequenciadores.*

Fig. 9 – Dois tipos básicos de mutações: nas transições ocorre a troca das bases de mesma classe (purina ou pirimidina) enquando nas transversões ocorre troca de bases de classes diferentes.



Fonte: os autores.

Sequências de DNA, mesmo com pequenas diferenças, podem ser geneticamente iguais. O alinhamento da Fig. 10 mostra uma sequência informada (*Query*) e uma sequência de DNA encontrada no banco de dados de referências. Este experimento foi feito utilizando o NCBI/Blast e apontou 100% de cobertura, ou seja, todos os nucleotídeos tiveram alguma combinação, mesmo com dois indels[5], ainda que tenha 95.08% de identidade, além de mutação. O algoritmo Blast ainda resultou na probabilidade deste *match* estar errado como 4e-16.

Fig. 10 – Alinhamento entre sequências de DNA. No exemplo ilustrativo pode-se observar os matches (|), indels (-) e os mismatches (ausência de correspondente |).

```
Query  1   ATTAAAGGTATATACCTTCCCAGGTAACCAACCAACCA-CTTTCGGATCTCTTGTAGATC  59
           ||||||||| ||||||||||||||||||||||||||| ||||| ||||||||||||||||
Sbjct  1   ATTAAAGGTTTATACCTTCCCAGGTAACCAACCAACCAACTTTC-GATCTCTTGTAGATC  59
```

Fonte: NCBI Blast, SARS-COV-2.

2.2.2.6 Resultado do processamento

Ao final do processamento das sequências e seus alinhamentos, um relatório é gerado. As métricas que indicam a possibilidade de uma combinação (*match*) são:

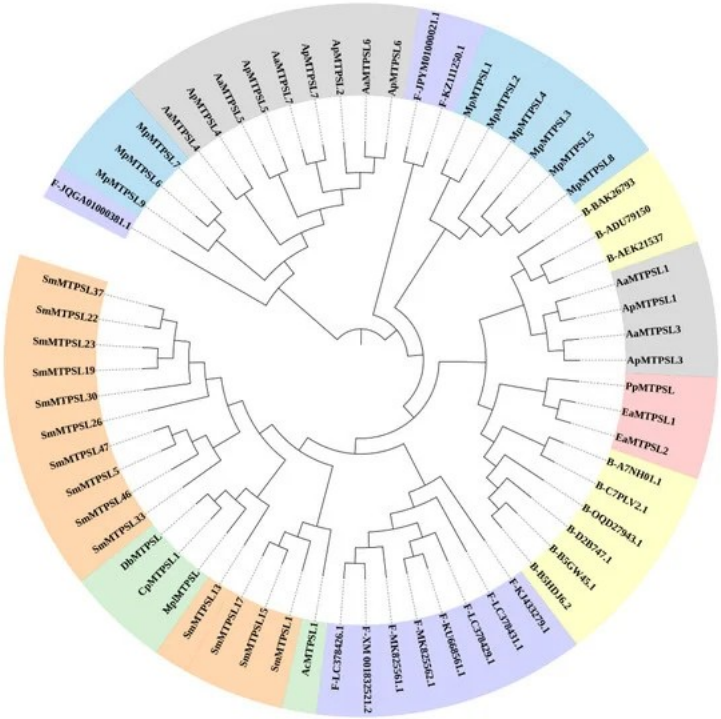
- Cobertura: indica quanto da sequência utilizada para pesquisa é encontrada no banco de referência. Mesmo com indels e mutações, uma sequência que é encontrada no banco é considerada como 100% de cobertura;
- Identidade: é o valor percentual que indica o quanto idêntica são as sequências, ou seja, quantos nucleotídeos combinam exatamente;
- Valor "E": também chamado de "*expected value",* é um número que representa a possibilidade do *match* ser ao acaso, quanto menor o valor, mais

5: *Inserções e deleções*

significante é o resultado.

Pipelines metagenômicos podem incluir dendrogramas filogenéticos (Fig. 11) na etapa de definição taxonômica dos sujeitos envolvidos, este produto gráfico também faz parte da análise dos resultados estatísticos, porém mais comumente voltado ao estudo evolutivo.

Fig. 11 – Exemplo de dendrograma filogenético. A representação é formada por uma árvore contendo galhos e folhas, os galhos representam as relações de "parentalidade" e "herança".



Fonte: Zhao et al (2021)

2.2.2.7 Bases de dados

Entre as bases de dados mais complexas disponíveis, pode-se citar três provedores majoritários na comunidade científica mundial: DDBJ (Japão) (DDBJ, 2021); EMBL (Comunidade Europeia) (EMBL, 2021) e GenBank (Estados Unidos) (NCBI, 2021). Estas bases possuem dezenas de terabytes de dados de sequências genéticas (nucleotídeos) e, com exceção da base japonesa, são de livre acesso público, permitindo a qualquer pesquisador ou interessado consultar, descarregar e utilizar os dados como melhor entender. O formato dos dados armazenados nas bases é o SRA (*Sequence Read Archive*) (FILE FORMAT GUIDE, 2021). Este formato permite o armazenamento compactado de sequências nucleotídeos

alinhados ou não, incluindo os metadados do estudo. Alternativamente é possível extrair os dados para o formato FASTQ e efetuar as computações nesta estrutura. Uma breve comparação sobre o tamanho (bases) destes provedores pode-se observar na Tabela 1.

Tabela 1: tamanho das principais bases de dados de nucleotídeos em dezembro de 2021. Exemplo ilustrativo do volume de dados armazenados nas bases de dados.

Banco de dados	Bases (nucleotídeos)
DDBJ (Japão)	16.670.849.721.017
GenBank (EUA)	15.975.309.037.332
EMBL (Europa)	30.079.100.000.000

Fonte: os autores.

O acesso aos dados pode ser feito via serviço de FTP[6], HTTP[7] ou através de serviços de nuvem (Amazon e Google). A base de dados GenBank, utilizada neste estudo, ainda possui um arquivo detalhado de mais de 220 mil dados taxonômicos de vários seres vivos. Esses dados podem ser genomas completos ou apenas parciais, ambos podem ser utilizados para comparação e alinhamento com dados obtidos pelas máquinas sequenciadoras. É possível fazer o download completo de todos os genomas em um único arquivo ou então obter os dados separadamente, relativos a cada ser vivo de interesse.

No caso específico do NCBI é possível fazer o download dos genomas de referências e dos *accessions*[8] através de API (*Application Programming Interface*) própria ou então utilizando serviços de nuvem[9] como AWS (Amazon) e GCP (Google). Os dados são abertos e podem ser obtidos independente de cadastros ou permissões. Exemplo de URL disponibilizada pela AWS:

https://sra-pub-run-odp.s3.amazonaws.com/sra/SRR15372564/SRR15372564

O identificador SRR15372564 é a referência do estudo no banco de dados NCBI. As três primeiras letras significam o tipo de *accession* (Biosamples, 2022).

6: *File Transfer Protocol: protocolo utilizado para transferência de arquivos em redes.*

7: *Hypertext Transfer Protocol: protocolo utilizado por vários serviços sendo o exemplo mais notável a página web.*

8: *Projetos postados por pesquisadores já processados e anotados.*

9: *Algumas bases possuem espelhos de dados nos principais provedores de nuvem, facilitando a distribuição de dados.*

A primeira letra tem os seguintes significados:

- S: significa que é um arquivo da base de dados NCBI SRA
- E: indica que é um arquivo do banco de dados DBI
- D: significa que é um arquivo do banco de dados DDBJ

A terceira letra especifica o tipo de dado representado:

- R: significa que se trata de um sequenciamento (run)
- X: se trata de um experimento
- S: se trata de uma amostra
- P: se trata de um projeto ou estudo

No caso do *accession* SRR15372564, trata-se de um sequenciamento depositado na base NCBI.

Identificadores iniciando com "SAM" significam que se trata de uma amostra, sendo que a letra seguinte identifica qual banco de dados foi depositado: E para EMBL-EBI, N para NCBI e D para DDBJ.

A lista com os *assemblies* é disponibilizada diretamente do FTP do NCBI e pode ser obtida diretamente do endereço ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/.

2.2.2.8 Algoritmos

A distância Levenshtein (Levenshtein, 1966), também chamada de "*edit distance"* é um algoritmo baseado em programação dinâmica onde duas cadeias de caracteres (*strings*) são alinhadas em uma matriz e cada letra combinante ganha um escore, incrementado na medida que se avança pela *string*, iniciando com valor "um" e aumentando ou diminuindo conforme a distância do caractere combinante. Procedimento conforme ilustrado na Fig. 12.

Fig. 12 – Programação dinâmica. Na ilustração pode-se observar o uso de uma matriz "mn" onde o caminho grifado representa a menor distância entre as sequências em alinhamento.

			1	2	3	4	5	6	7	8	9
		-	G	A	A	C	G	T	A	G	T
	-	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
1	G	-2	1	-1	-3	-5	-7	-9	-11	-13	-15
2	A	-4	-1	2	0	-2	-4	-6	-8	-10	-12
3	A	-6	-3	0	3	1	-1	-3	-5	-7	-9
4	C	-8	-5	-2	1	4	2	0	-2	-4	-6
5	G	-10	-7	-4	-1	2	5	3	1	-1	-3
6	A	-12	-9	-6	-3	0	3	4	4	2	0
7	G	-14	-11	-8	-5	-2	1	4	3	5	3
8	T	-16	-13	-10	-7	-4	-1	2	1	3	6
			↑	↑	↑	↑	↑	↑	←	↑	↑
			G	A	A	C	G	T	A	G	T
			G	A	A	C	G	-	A	G	T

Fonte: os autores

Este algoritmo possui complexidade O(mn) e não considera as questões da natureza do DNA, porém serviu de base para outros algoritmos que implementam tais questões e formam um conjunto de alternativas.

Assim como o algoritmo de Levenshtein, a proposta de Temple Smith e Michael Waterman (Smith *et al*, 1981) também utiliza a programação dinâmica como base de seu trabalho. As diferenças estão na forma com que os escores são calculados, considerando uma matriz de substituição e um escore para as lacunas. Este algoritmo é caracterizado pela possibilidade de obtenção de um alinhamento local, ou seja, a sequência de interesse não precisa ter o comprimento da sequência de referência. Esta é uma das principais diferenças do proposto por Saul Needleman e Christian Wunsch (Needleman *et al*, 1970). No caso do algoritmo de Needleman-Wunsch, o alinhamento é global e exige sequências de mesmo tamanho para o procedimento – que é feito por programação dinâmica. Esta técnica também é chamada de alinhamento ideal e pode ser explorada dividindo as sequências em pequenas partes e analisando separadamente.

Uma forma diferente de calcular a similaridade de sequências foi baseada no trabalho de Michael Burrows e David Wheeler (Burrows, 1994). Estes pesquisadores desenvolveram um algoritmo de compressão de dados em 1994 (apesar de existirem trabalhos não publicados por Wheeler em 1983) muito utilizado até hoje nos compactadores de dados baseados em bzip2[10]. Esta técnica permite a compactação sem perda (*lossless*) e comparação de resultados agilizando a análise

10: *Compactador de arquivos (sem perdas) utilizado em diversos softwares, muito popular no sistema operacional Unix, Linux e derivados.*

de combinação de sequências. A etapa de compactação possui complexidade O(n).

Estes algoritmos podem incluir a quebra de sequências em pequenas partes chamadas de *k-mers* e posterior busca simples ou então aceleração através de tabelas *hash* utilizando grande quantidade de memória RAM para mapeamento de dados, facilitando a etapa de alinhamento.

Entre os algoritmos de compressão específicos para dados genéticos, pode-se destacar:

- LibCSAM: possui compactação sem perda (*lossless*) dos dados de qualidade contidos nos arquivos FASTQ (Cánovas, 2014);
- LZ77 (baseado): possui compactação sem perda (*lossless*) dos dados de nucleotídeos e tem como foco acelerar o processo de armazenamento (Du, 2020);
- CoLoRd: compactação sem perdas (*lossless*) de dados de nucleotídeos e compactação com perdas (lossy) de dados de qualidade (Kokot, 2021);
- FCLQC: compactação sem perdas (*lossless*) de dados de qualidade com foco em multiprocessamento (Cho, 2021);
- LZW (baseado): compactação sem perdas (*lossless*) de dados de nucleotídeos com foco em múltiplos dicionários indexados (Keerthy, 2017);
- ProjectDNACompression: compactação diferencial de dados de nucleotídeos, armazenando uma sequência “original” e a diferença para as sequências semelhantes (Brandon, 2009).

2.2.2.9 Softwares de alinhamento e anotação

Entre os softwares de alinhamento a anotação, como exemplo no estudo de Verce (2021), estão as ferramentas BLAST (Altschul et al., 1990), Kraken (Wood et al., 2019), Kaiju (Menzel et al., 2016), e Centrifuge (Kim et al., 2016). Cada software possui suas características que vão desde o algoritmo utilizado, seu consumo de recursos computacionais e o resultado produzido. Estes quatro exemplos de ferramentas são uma amostra das diversas soluções propostas por pesquisadores da área. Fabricantes de sequenciadores e empresas especializadas em diagnóstico metagenômico possuem suas próprias soluções, possivelmente baseadas em alguma destas citadas – ou muito semelhante.

Conforme o manual do Blast (BLAST, 2021), este software é “*provavelmente*

a ferramenta de busca por similaridades mais popular existente". Foi introduzida em 1989 pelo *The National Center for Biotechnology Information (NCBI)*, seu nome é uma sigla que significa *Basic Local Alignment Search Tool*. Apesar de ter sido publicada em 1989, a ferramenta recebeu melhorias e avanços durante os anos e é, até a presente data (2022), mantida e atualizada pelo NCBI, onde inclusive possui uma versão online disponível para pesquisas gratuitas. A estratégia de alinhamento do Blast é baseada em heurísticas combinadas com programação dinâmica. O processo de utilização é relativamente simples, após a instalação da ferramenta é necessário formar o banco de dados de referência (download do próprio NCBI) e então é possível executar os comandos conforme necessário. O resultado do processo do alinhamento é um relatório que pode ser interpretado em forma de dendrograma contendo as espécies identificadas no processo de busca. O software pode ser executado nos sistemas operacionais Linux e Windows através de linha de comando (terminal). A sua utilização (sintaxe) pode ser consultada nos manuais fornecidos pelo desenvolvedor (NCBI) e os experimentos feitos com o Blast constam nos anexos deste trabalho.

Com o avanço dos métodos computacionais, outras ferramentas surgiram com performance superior ao experimentado no Blast. A ferramenta Kraken (Wood, 2014) foi inicialmente publicada em 2014 e, já em sua primeira versão, os autores fizeram o uso de *k-mers* exatos e estimaram uma performance próxima a 1000 vezes mais rápida que o Blast, obtendo alinhamentos semelhantes. Esta ferramenta representa um grande avanço pelo motivo de utilizar os recursos computacionais, aliados as estratégias biológicas e explorando a possibilidade de se utilizar LCAs (*Lowest Common Ancestors)*, traçando uma espécie de "rota" entre a raiz da árvore taxonômica em direção as folhas (*root-to-leaf*). A sua instalação é relativamente simples, sendo obtida diretamente do seu repositório no GitHub e compilada localmente. É utilizada no sistema operacional Linux, através de linha de comando (terminal). Para a formação de sua base de dados existem duas possibilidades: construção local ou download de bases pré-compiladas. Em experimentos feitos neste trabalho, em um servidor Linux com 24 núcleos de processamento (Xeon E5-2620) com 256 GB de RAM, a construção da base de dados levou mais de 50 horas de processamento, sem considerar o download dos arquivos. A utilização de uma

base pré-compilada não exige este custo computacional, no entanto pode estar desatualizada, dependendo de onde se obtém os arquivos. Os experimentos, junto com os comandos utilizados estão nos anexos deste trabalho.

A ferramenta Centrifuge (Kim et al., 2016) teve sua primeira versão publicada em 2016 pelo Centro de Biologia Computacional da Universidade Johns Hopkins. A estratégia deste software é a utilização dos algoritmos de indexação baseadas em Burrows-Wheeler (BWT) e Ferragina-Manzini (FM), permitindo que os índices de interesse fossem reduzidos em uma base de referência limitada. Um dos grandes avanços desta ferramenta foi a possibilidade do uso de menores recursos computacionais (inclusive computadores pessoais *desktop*), que, pelo fato de não ser baseado em *k-mer*, possui uma base reduzida. A ferramenta pode ser executada no sistema operacional Linux, em linha de comando (terminal). Pode ser obtido diretamente do seu repositório do GitHub, onde teve sua ultima atualização (*release*) em agosto de 2021, e compilado localmente. Após a compilação é necessário fazer download de uma base pré-compilada ou então a compilação local de uma base de referência. A base pré-compilada de referência, disponível no site do centro de biologia computacional, teve sua ultima atualização em abril de 2018 e possui o tamanho de 64gb. Em seu site, o departamento ainda informa que devido a pandemia de SARS-CoV-2, complementos adicionais, específicos para o estudo do vírus, foram disponibilizados. Os comandos necessários para a obtenção e execução da ferramenta, junto com os exemplos de relatórios estão nos anexos deste trabalho.

Kaiju (Menzel et al., 2016) é uma ferramenta de alinhamento a anotação taxonômica utilizada para estudo metagenômico que, apesar de utilizar metodologia semelhante ao Centrifuge (Burrows-Wheeler), alinha sequências de aminoácidos em vez de nucleotídeos. Segundo sua documentação disponível, pode ser utilizado com a base de dados “nr” (*non-redudant protein database – NCBI)* e, para as comparações, transforma os *reads* sequenciados em aminoácidos. Seu grande diferencial é o baixo uso de recursos computacionais, assim como Centrifuge, permitindo o uso em computadores *desktop, c*onsiderando que um *subset* de 4.821 genomas microbiais pode ser armazenado em apenas 10 GB de RAM. O software é obtido diretamente de seu repositório no GitHub ou através de arquivo compactado

(*release*). Pode ser executado em ambiente Linux, por linha de comando. Após a obtenção do programa, deve-se compilar o código-fonte e construir o banco de dados. Assim como outros produtos, possui dados pré-compilados disponíveis em seu repositório. Até a escrita deste trabalho, a data de atualização das bases era abril de 2022. Os arquivos completos de referência possuem entre 60 GB e 70 GB. Os experimentos, comandos necessários para a instalação, compilação e execução; e os relatórios estão listados nos anexos deste trabalho.

2.2.2.10 Inteligência artificial

A inteligência artificial aplicada a metagenômica é uma área de pesquisa computacional para o processamento de larga escala de dados genéticos. Methieu (2022) descreve o uso de inteligência artificial no alinhamento de sequências metagenômicas como promissor e cita exemplos de utilização como DeepMicrobes, Woods e Vervier. Existem várias técnicas de inteligência artificial aplicáveis aos dados metagenômicos. Como exemplo se pode citar *Naive Bayes* (NB), *Support Vector Machine* (SVM) e *Deep Learning* (entre outras) (Methieu, 2022). Tonkovik *et al* (2020) realizou um estudo que divide as técnicas em "*k-mer based*" e "*non k-mer based*". Dentro das técnicas baseadas em *k-mer*, explora-se os conceitos de *deep forest*, no entanto o autor considera que este modelo já está ultrapassado por técnicas tradicionais (como as técnicas utilizadas, por exemplo, no software Kraken). Uma abordagem alternativa é a utilização de *casacade deep forest*, que, segundo o autor, possui performance semelhante a técnica de *deep learning*. Um dos problemas no uso de aprendizado de máquina na análise metagenômica, segundo Methieu (2022), é a velocidade com que os dados são atualizados nas bases de referência. A necessidade de se criar modelos específicos para cada banco é um fator negativo se comparado às técnicas existentes como Kraken, Diamond, Blast, entre outros. Liang *et al* (2020) utilizou *deep learning* para a criação da ferramenta DeepMicrobes e obteve resultados promissores, apesar de, em termos gerais, a preocupação com a performance temporal ser mais acadêmica do que produtiva, sendo o estudo mais focado em precisão e acurácia do que no tempo de execução propriamente estipulado.

Yang *et al* (2020) fez uma revisão de literatura voltada ao estudo do aprendizado de máquina como mineração de dados genéticos. Entre os

experimentos estudados estão as abordagens que utilizam árvores randômicas, SVM e *deep learning*. O autor considera, em 2020, que as atenções estão nos métodos heurísticos, talvez pela natureza “black box” do aprendizado de máquina, como diz:

> *The black-box nature of machine learning models brings new challenges to biological applications. It is usually very difficult to interpret the output of a given model from a biological point of view, which limits the application of the model.*

Yeang *et al* (2020) ainda conclui que, ainda que as técnicas de aprendizado de máquina estão no cerne dos estudos de *big data*, existem alguns problemas relacionados e este guarda-chuva, como exemplo:

> *1. There are still efficiency challenges when processing large-scale DNA sequence data;*
>
> *2. For different biological needs, suitable DNA sequence data mining algorithms should be designed according to the corresponding background knowledge and sequence characteristics;*
>
> *3. How to extract the sequence characteristics of DNA sequences and how to design an effective similarity measure to measure sequence similarity is very important;*

Apesar das abordagens computacionais que utilizam a inteligência artificial serem interessantes e promissoras, o foco desta tese é a criação de um algoritmo com perda, utilizando a natureza genética do DNA. Não obstante, a utilização futura de métodos de detecção baseados em inteligência artificial pode ser um grande impulsionador neste e em outros estudos. Durante os estudos bibliográficos desta tese, um experimento utilizando NetML foi executado como ferramenta de sugestão de agentes infecciosos em função da suspeita clínica advinda do profissional/pesquisador. Os materiais de ensaio estão nos anexos deste manuscrito.

2.3 BENCHMARK E RELATO DE EXPERIMENTAÇÃO

A estratégia de construção do *benchmark* foi baseada nos ensaios obtidos e analisados em publicações já existentes na base de dados NCBI. Para esta finalidade foram obtidos mais de 100 *accessions* (listados nos anexos). Cada *accession* possui um ou mais arquivos do tipo fa/fasta/fastq - que devem ser convertidos para o formato fastq, quando necessário. Para o *download* dos arquivos o software desenvolvido faz a conexão direta com o banco de dados de referências e os recebe sob demanda, armazenando localmente para uso posterior. Para a conversão dos arquivos “fa”, utilizou-se as ferramentas do próprio NCBI.

As sequências de referência utilizadas foram também obtidas dos bancos de dados *mainstream*. O acesso é público e existem mais de 200.000 genomas disponíveis, os *scripts* de *download* também estão disponíveis nos anexos.

Em posse do material de análise, a próxima etapa é executar os passos do estudo na ferramenta desenvolvida neste trabalho. Em tempo, esperou-se responder as seguintes questões: a) Encontrou-se o agente infeccioso conforme estudo publicado? b) Quanto tempo se levou para a obtenção deste resultado?

Nos primeiros experimentos não foi possível encontrar nenhuma combinação que corroborava com a sua respectiva publicação (*accession*), no entanto se tratava de um ajuste de estratégia de compactação. Considerando que os adaptadores, quando não removidos, podem interferir no alinhamento, a remoção deles se fez necessária, seja por alinhamento ou por redução de comprimento dos *reads*. A segunda opção foi adotada, uma vez que traria maior sensibilidade e poderia resolver o problema dos adaptadores, desta forma cada *read* pode ser comparado a uma espécie de *k-mer* (relativamente grande).

Feitos os ajustes, os primeiros resultados foram obtidos. Como exemplo, no *accession* SRR12665147, que trata de um caso de sequenciamento metagenômico de fluído cérebro-espinhal (CSF), foi encontrado alinhamentos com o agente infeccioso *Neisseria Meningitidis*, da mesma forma que o estudo publicado no experimento SRX9145862.

Fig. 13 - Exemplo de alinhamento experimental do accession SRR12665147 com uma base de referência filtrada pelo usuário/pesquisador

```
Total reads: 5.701
Total reads len: 132.949
Total refseqs: 5
Total refseqs len: 3.919.721
=== DONE ===
Elapsed: 00:00:07.3451015
```

ACCESSION	ORGANISM	MATCHES	STATUS
GCF_002845425.1@ASM284542v1	NEISSERIA MENINGITIDIS	393	DONE
GCF_900073015.1@MBR1	MYCOLICIBACTERIUM BRUMAE	0	DONE
GCF_000668415.1@Myco_tube_UT0045_V1	MYCOBACTERIUM TUBERCULOSIS UT0045	0	DONE
GCF_000003925.1@ASM392v1	BACILLUS MYCOIDES DSM 2048	0	DONE
GCF_000009885.1@ASM988v1	KLEBSIELLA PNEUMONIAE SUBSP. PNEUMONIAE NTUH-K2044	0	DONE

Fonte: os autores

Alguns genomas de referência, estranhos ao estudo, foram adicionados neste experimento a fim de se detectar erros do tipo 1, no entanto os mesmos não foram detectados. É perfeitamente possível, e esperado, por corolário, que mais de um organismo tenha a mesma combinação parcial de nucleotídeos, no entanto o volume de dados e a quantidade de combinações pode ser um fator excludente, por este motivo é informado ao pesquisador a quantidade de combinações e a visualização posterior das sequências.

Este ensaio foi repetido dezenas de vezes com outros *accessions* (listados nos anexos), com resultados semelhantes aos encontrados nos estudos publicados nas bases de referência. Obteve-se a indicação de que se chegou na resposta da pergunta “a”, ou seja, é possível encontrar os agentes infecciosos.

O tempo de processamento é um fator determinante na metagenômica, como visto na seção 2.2, diversos autores citam este como um dos maiores problemas ao se obter um diagnóstico, porém a medida de tempo é relativa e deve-se tratar com cautela. Na construção deste algoritmo, uma das preocupações foi trazer um pouco do conhecimento de processamento de sinais para o alinhamento genômico. Como exemplo ilustrativo, a transmissão de um vídeo em uma rede limitada pode ter uma compactação com perda significante, sob pena de não se obter um fluxo de dados constante, fazendo com que o vídeo “trave” com frequência. A redução da qualidade do vídeo (perda) nem sempre significa na descaracterização do mesmo, em outras palavras, a pessoa que está assistindo determinado vídeo, mesmo em qualidade baixa, ainda consegue identificar o seu conteúdo. Eventualmente, em uma rede mais

rápida, a qualidade pode ser aumentada e o conteúdo ter mais detalhamento. A construção do algoritmo traz este mesmo conceito, podendo ser extremamente permissivo e, eventualmente falhar em erros do tipo 2, ou então ser deveras exclusivo, com um risco menor, porém existente, de erros do tipo 1.

Observa-se que as comparações de performance de ferramentas de alinhamento são geralmente pareadas com o “padrão ouro”, que é, neste momento, a ferramenta Blast (conteúdo explorado no capítulo de livro publicado, copiado nos anexos deste trabalho), no entanto foram feitas duas comparações, uma direta e outra indireta. Na comparação direta, foram consideradas três ferramentas existentes de alinhamento de nucleotídeos (Blast, Kraken, Centrifuge e Kaiju). É importante ressaltar que a comparação direta entre as ferramentas pode não ser, exatamente, cientificamente justa; pelo motivo de que as finalidades de cada ferramenta são diferentes, trazendo um viés de comparação importante ao estudo, porém, para este estudo, há de se focar apenas no resultado comum das soluções estudadas, conforme tabela adicionada no artigo em anexo.

3 OBJETIVOS

3.1 GERAIS

Desenvolver um *pipeline* experimental e uma ferramenta gráfica para estudo de um algoritmo de compactação de sequências de DNA com a finalidade de acelerar o processo de alinhamento e posterior anotação taxonômica.

3.2 ESPECÍFICOS

- Revisar a bibliografia existente sobre a metagenômica e bioinformática;
- Revisar o estado da arte das ferramentas de alinhamento e pipelines existentes;
- Desenvolver uma ferramenta para a análise de sequências de DNA;
- Realizar experimentos com diferentes técnicas de aceleração de alinhamento;
- Validar os resultados em comparação com os dados curados nas bases.

4 ARTIGOS CIENTÍFICOS E CONTRIBUIÇÕES

Periódico: *Database: The Journal of Biological Databases and Curation*

Impact Factor: *4.642 (Clarivate)*

Qualis: *Interdisciplinar: A2, Medicina I: B1.*

Indexação: *Journal Citation Reports / Science Edition; MEDLINE/PubMed; PubMed Central; Asian Science Citation Index; Chemical Abstracts; Directory of Open Access Journals (DOAJ); The Standard Periodical Directory*

The MetaGens Algorithm for Metagenomic Database Lossy Compression and Subject Alignment

Gustavo Henrique Cervi[1], Cecília Dias Flores[1], Claudia Elizabeth Thompson[1]

[1]Department of Health Sciences, Federal University of Health Sciences (UFCSPA), Porto Alegre, RS, Brazil
gustavohc@gmail.com

Abstract. Introduction: the advancement of genetic sequencing techniques led to the production of a large volume of data. The extraction of genetic material from a sample is one of the early steps of the metagenomic study. With the evolution of the processes, the analysis of the sequenced data allowed the discovery of etiological agents and, by corollary, the diagnosis of infections. One of the biggest challenges of the technique is the huge volume of data generated with each new technology developed. **Objective:** to introduce an algorithm that may reduce the data volume, allowing faster DNA matching with the reference databases. **Methods:** using techniques like lossy compression and substitution matrix, it is possible to match nucleotide sequences without losing the subject. This lossy compression explores the nature of DNA mutations, insertions, and deletions and the possibility that different sequences are the same subject. **Results:** the algorithm can reduce the overall size of the database to 15% of the original size, depending on parameters, it may reduce up to 5% of the original size. The match algorithm, althrought is the same as the other platforms, it is more sensible because it ignores the transitions and transversions, resulting in a faster way to get the diagnostic results. The first experiment results in an increase in speed 10 times faster than Blast while maintaining high sensitivity. This performance gain can be extended by combining other techniques already used in other works, such as hash tables.

Keywords: Metagenomic, Diagnosis, Computing Methodologies, High-Throughput Nucleotide Sequencing

Introduction

Metagenomics is the processing of genetic material extracted from an environment (soil, fluids, water and others) through the DNA/RNA sequencing of the collected biological material [1]. A huge volume of data is generated from modern sequencing machines, which can go from gigabytes to terabytes of data [2]. Data analysis is a major challenge that involves a great demand for computer equipment and is carried out through specialized software that analyze the data and produce results, including statistical data [3]. While the genomic analysis is focused on a single biological subject, metagenomics collects genetic information from all biological subjects present in the sample [9]. The DNA is measured in atoms and molecules (molecular biology), thus the amount of genetic material from a single sample is too small to be sequenced. The genetic material obtained

from samples must be duplicated and amplified. The PCR method is widely used and consists of thermal cycles that denatures (breaks) the DNA allowing the duplication and amplification processes [4,5,8,9]. In the next phase, a machine identifies the nucleotide sequences.

The NGS (Next Generation Sequencing) [10] works in a massive parallel way, obtaining gigabytes of DNA data per run (chemical process cycle). Until the mid-2000s, when the first NGS sequencers appeared, it was hard to obtain data. Today, it is hard to analyze the huge amount of data" [9].

Once biological agents can be detected from a fluid sample, the metagenomic technique eventually evolved into what is called clinical metagenomics [4]. This study modality has the purpose of diagnosing diseases through the analysis of samples, in the search for etiological agents responsible for diseases. A special chapter of this history is to use metagenomics as a tool for the diagnosis of infectious diseases [4,5,8]. As an example, central nervous system (CNS) infections are commonly confused with other diseases and there are a variety of etiologic agents that are not easily identifiable. There are estimates that the identification of the etiologic agent occurs in only 25% of cases, even considering excellent laboratory facilities [6]. These infections are neurological emergencies and have high morbidity and mortality. Consequently, it is essential that epidemiological studies be carried out and new diagnostic methods and therapeutic strategies developed in order to reduce costs for the health system. Community-acquired CNS infections have an unfavorable outcome in about 30% of cases, including severe sequelae or even death due to difficulty in identifying the etiologic agent and, consequently, inadequate treatment [7].

Metagenomics is extensive and this paper aims to present a novel algorithm to optimize the database and process data to obtain faster results.

Pipeline

In general terms, a metagenomic pipeline can be divided into four steps: (i) filtering data, (ii) aligning with databases, (iii) filtering results, and (iv) statistical reports (Fig. 1). Despite the fact the DNA vocabulary is very reduced, containing only four letters, the combination of these letters are the "source code" of all living matter, which varies from a simple thousand-base bacteria to a multi-billion-base animal genome [9].

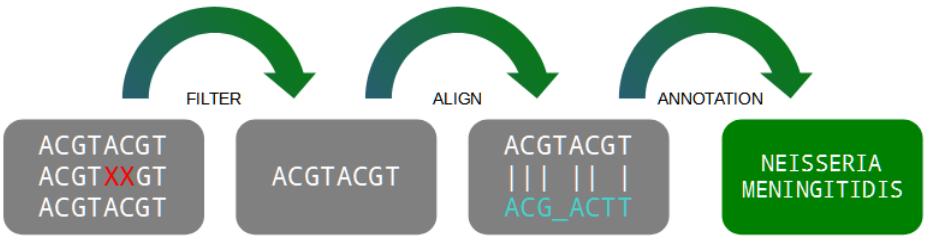


Fig. 1 - a traditional pipeline has several steps chained together, each containing its specific function. In the drawing above, three steps are presented: filtering, aligning and annotating. Source: the authors.

The computational processing behind this amount of data is a challenging problem [11], especially when the researcher/physician is running against the time, for example, waiting for the diagnosis of a disease. In

a computational perspective, the main time consuming problem relies on the huge volume of data to be filtered, organized and compared. Once the genetic sequencer finishes the sequencing process, all that data must be processed. The NGS (Next Generation Sequencing) technology produces a large number of "short reads", which are sequences up to 300 nucleotide bases (adenosine, guanine, thymine or cytosine). They are represented as a string of data like "ACGGATCGATTCGATTG…". The comparison of these sequences with the reference database results in a possible diagnosis with the identification of the specific etiological agent (virus, bacteria and/or fungus). These reference databases are commonly accessible from public sites like GenBank (NCBI, USA), EMBL Bank (EMBL-EBI, Europe) and DDBJ Bank (DDBJ, Japan), which easily overpass the terabyte of data. Fig. 2 shows the evolution of the GenBank database through the years.

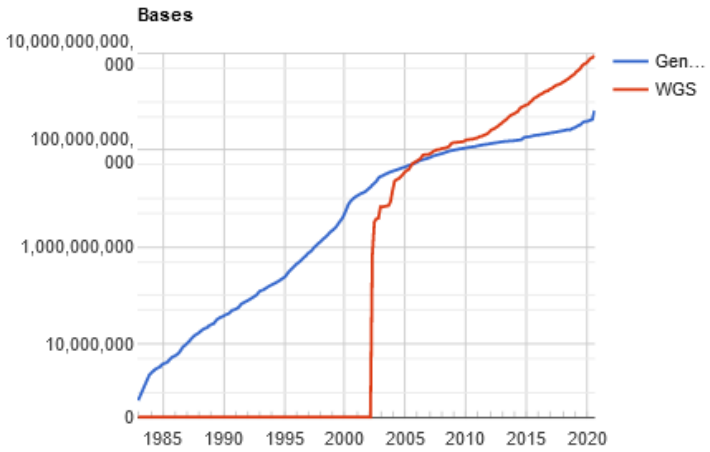


Fig. 2 - Evolution of the Genbank database, maintained by the NCBI, an increase in the curve of genomic data deposits can be observed. This increase may represent the increasing difficulty of data processing. Source: NBCI statistics webpage [12]

Quality control

The first stage after sequencing is the "quality control". The sequencer produces a large amount of data, but not all with the same quality. This step is important to remove "low quality" data [13,14]. In summary, the first generation and some second generation sequencers collect information through a fluorescent agent bound to the nucleotide [10], using a very precise wavelength laser, capturing the light emitted by the molecule to infer the sequence. The resulting signal may be biased or not deterministic. The sequencer calculates the "quality" of the read based on the light intensity and writes the score in the result file - each sequenced nucleotide has its specific quality score [15]. Fig. 3 and Fig. 4 show chromatograms with low (multiple peaks per base) and good quality data, respectively, obtained by a first generation sequencer [14].

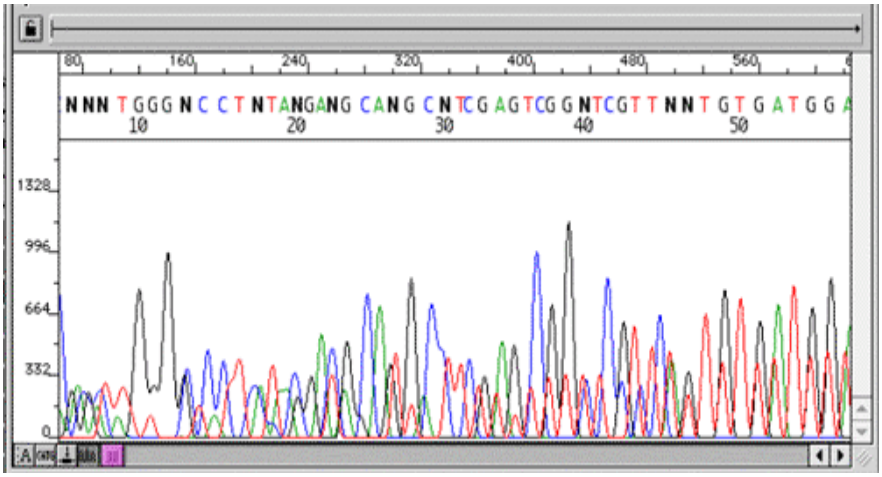


Fig. 3 - Chromatogram with multiple peaks per base - low quality data. Source: Roswell Park Comprehensive Cancer Center [16]

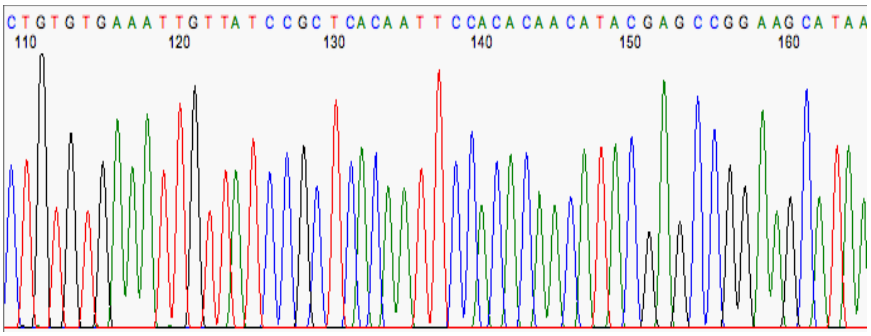


Fig. 4 - Chromatogram indicating good quality of data sequence. Source: U-M Biomedical Research Core Facilities [17]

Amplification

All first and second generation sequencers require a DNA amplification step, where the DNA is duplicated into multi million copies, amplifying the signal to be detected by the machine. After sequencing, duplicate reads are removed resulting in data reduction [18].

Host removal

This step is also important to speed up the analysis and reduce the risk of bias. Since the sample is obtained from a living host (human), it is likely to have the host DNA present in the data after sequencing. This stage is performed by searching for the host DNA, in the result file, through comparison with the reference genome available in public databases. Once the host reads are identified, they have to be removed. In case of human samples, a reference human genome is used, preferentially the most recent and curated available data.

Searching through Reference Databases

This step is the most computing expensive task. The sequencer yields a huge amount (>100 million) of short reads (up to 300 nucleotides bases in current NGS technology) written in a text file, whose format is

commonly the FASTQ type [15] (same from the original FASTA file format but with quality information). Each read is represented by one DNA string like "ACGATCGATTCGGA(…)" and it must be compared to reference datasets (terabytes of genomes from all sorts of living organisms, available on public organized databases). The first guess is O(m+n) like Knuth-Morris-Pratt or O(m)+Ω(n/m) like Boyer-Moore algorithms could be applied to solve this problem. However, these algorithms cannot produce efficient results from a biological viewpoint. In order to be able to compare the new sequence to all sequences committed in databases, it is necessary to perform sequence alignment. The DNA is not a rigid and static sequence, it is submitted to evolutionary forces such as mutation, selection, genetic drift, and migration. Considering the mutational aspect, the DNA substitutions can be classified as (i) transitions: when involve bases with similar shape, interchanges of two-ring purines (A ⥋ G) or one-ring pyrimidines (C ⥋ T) and (ii) transversions: when involve substitutions of one-ring and two-ring DNA bases, interchanges of purine for pyrimidine bases and vice-versa. Fig. 5 indicates the possible transitions and transversions.

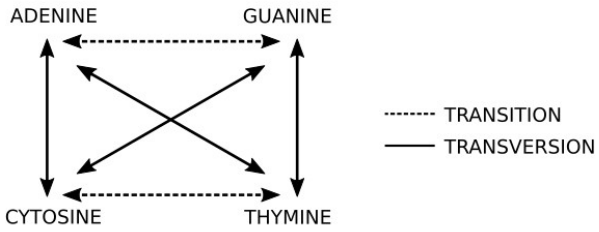


Fig. 5 - Two basic types of mutations: in transitions there is an exchange of bases of the same class (purine or pyrimidine) while in transversions there is an exchange of bases of different classes. Source: the authors, based on [19]

When comparing two sequences to obtain an alignment, the main objective is to identify the positional homology, i.e., identify sites with a common ancestry in the alignment. It may be necessary to include gaps (indels, corresponding to deletion in sequence 1 and insertion in sequence 2) to better accommodate one sequence in relation to another. In this sense, the sequence "ACGATCGAT" may be biologically equivalent to the sequence "ACGCTCGGAT" (one mutation and one indel), i.e., they may be homologous. Homology is a biological concept that indicates two sequences share a common ancestry. Common algorithms used to align sequences in genomic research are Levenshtein Distance [20], Smith-Waterman [21], Needleman-Wunsch [22], Burrows-Wheeler [23] plus hashing and its derivatives. Blast, which uses a heuristic method based on Smith-Waterman, is the most commonly used software to perform local alignment. It allows identifying subject sequences in a database that are similar to a query sequence. Fig. 6 shows a local alignment obtained by a Blast search, with the indication of mismatches (blue arrow and lack of | symbol), indels (green arrow and gaps) and matches (red arrow and | symbol).

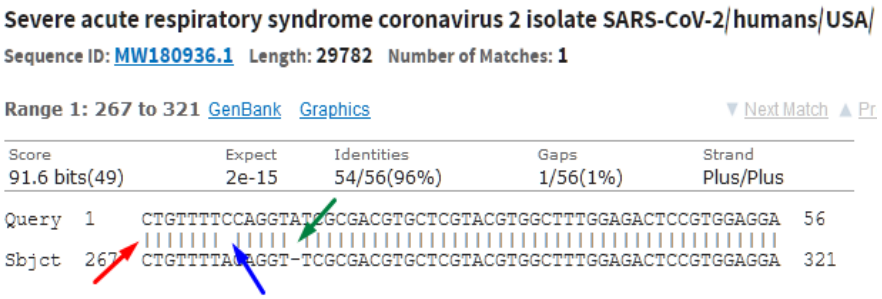


Fig. 6 - In the alignment of genetic sequences, mismatches (blue), indels (green) and matches (red) can occur. The occurrence of these events does not necessarily mean that they are different subjects. Source: the authors (sample on NCBI Blast)

Dynamic Programming

Dynamic programming applied to bioinformatics (eg. Levenshtein Distance, Smith-Waterman and Needleman-Wunsch) has complexity in order of O(mn) in the worst case, but it is possible to improve as demonstrated by Berghel and Roach [24]. It is very time-consuming in terms of computational processing, although it is possible to parallelize the task since an input does not depend on other data in the same processing stage. It is not rare that a software solution has more than one combination of algorithms. For example, in case of seed-and-extend algorithms, it is very common for a software aligner to use the Burrows-Wheeler algorithm to reduce the size and hash tables to find the seed portions. Dynamic programming applied to sequence alignment can be explained using a 2-dimensional matrix where two sequences are compared and there are three main steps: (i) matrix initialization; (ii) matrix fill (scoring), and (iii) traceback (alignment). Match, penalty-gap and mismatch values are defined according to a score [25]. During the matrix fill, for each cell, all possibilities are evaluated and received a value: (i) in diagonal: match or mismatch; (ii) gap in sequence y, and (iii) gap in sequence x. The traceback step determines the actual alignment(s) that result in the maximum score. In Fig. 7, the maximum alignment score for the two sequences is 11 and the best alignment is shown in red.

		1	2	3	4	5	6	7	8	9
		-	G	A	A	C	G	T	A	G	T
	-	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
1	G	-2	1	-1	-3	-5	-7	-9	-11	-13	-15
2	A	-4	-1	2	0	-2	-4	-6	-8	-10	-12
3	A	-6	-3	0	3	1	-1	-3	-5	-7	-9
4	C	-8	-5	-2	1	4	2	0	-2	-4	-6
5	G	-10	-7	-4	-1	2	5	3	1	-1	-3
6	A	-12	-9	-6	-3	0	3	4	4	2	0
7	G	-14	-11	-8	-5	-2	1	4	3	5	3
8	T	-16	-13	-10	-7	-4	-1	2	1	3	6
			↑	↑	↑	↑	↑	↑	←	↑	↑
			G	A	A	C	G	T	A	G	T
			G	A	A	C	G	-	A	G	T

Fig. 7 - Dynamic programming. In the illustration, it is possible to observe the use of a matrix "mn" where the highlighted path represents the shortest distance between the sequences in alignment. Source: the authors. Reference:[26]

Proposed algorithm

To be feasible, the nucleotide matching algorithm must accomplish the nature of the DNA strand. In other words, the matching string must be straight enough to pair the sequence and be flexible enough to address the mutations and indels. The strategy behind this proposed algorithm is to divide and conquer, removing the most obvious non-matches from the database, limiting the fine search to the relevant subjects. To achieve this, the algorithm computes the sequence in order to create identities of the reads, counting the distance between nucleotides A to next A and C to next C, simulating a wave where the same nucleotide interval is interpreted as a computed frequency (Fig. 8). The cycle count obtained from the interval is stored in a database and used to perform the preliminary filter.

```
     READ: ACGTCGATCGTGCTGATTCGGATCGGATCGGATCGAT
   DIST A:  12345 12345678 12345 1234 1234 123
 RESULT A:      5        8     5    4    4   3   = 585443

     READ: ACGTCGATCGTGCTGATTCGGATCGGATCGGATCGAT
   DIST C:   12 123 123 12345 1234 1234 1234
 RESULT C:    2   3   3     5    4    4    4     = 2335444
```

Fig. 8 - Hypothetical reads. The bases in red are used as a distance marker (green line). The result, in blue, makes up the sequence identity. Source: the authors.

The algorithm locates the first nucleotide A in the sequence, then counts the number of nucleotides for the next A, and then repeats with the next A until the end of the read (Fig. 9). The same process is performed with C nucleotide. The result of this computation is a set of numeric values containing the nucleotide distances, which is, in a computational approach, more efficient to evaluate than alphabetic values such as "ACGT". This algorithm reduces the size of the data by a fraction of the original. Mismatches on nucleotides G and T are irrelevant to the final result, although it can be calculated too, using the same method. Another important decision is that the repeating zones may be irrelevant because distances longer than nine are ignored (low priority) and distances equal to zero are ignored too. These identities are used to match the reads to the reference genome in a highly sensitive way, discarding the unmatched reads due to the probability of a high e-value in a subsequent alignment. With the set of identities in a database, the next step is to filter and find possible matches with the reference genome, which must go through the same identity processing. The resulting data set is the block that will be used for the actual alignment of sequences, using consolidated techniques such as the Blast algorithm.

Fig. 9 - illustrative example of a possible wave and its frequency. Reference sample read from SARS-COV2 from ERR4329467 [27]

This approach may have flaws because of indels, where nucleotides can be inserted or deleted in the

strain (Fig. 10). In this case, the distance between the same nucleotides will vary. The solution is to create a range of tolerance using a substitution matrix where a distance 5 can be represented as a range 5 (more or less 2) represented by the letter "Q". Another approach relies on slicing the result code to reduce the final size of the comparison (Figure 10).

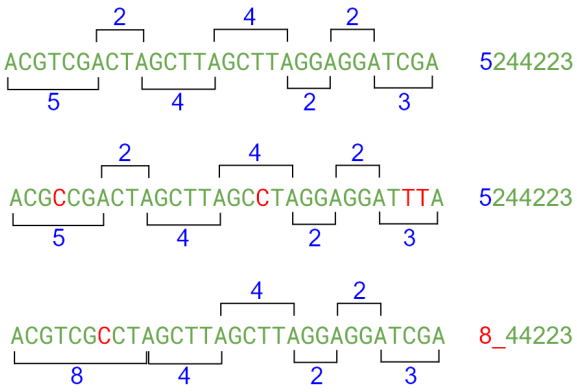


Fig. 10 - the indel problem. Source: the authors.

After the range code substitution, the resulting sequence is "normalized" creating a kind of wave softness allowing the matching of sequences with small gaps or insertions (Fig. 11).

```
    READ: ACGTCGATCGTGCTGATTCGGATCGGATCGGATCGAT
  DIST A:  12345 12345678 12345 1234 1234 123
RESULT A:      5        8     5    4    4   3   = 585443
RESULT A:      5        8     5    4    4   3   = RSRRRR

    READ: ACGTCGATCGTGCTGATTCGGATCGGATCGGATCGAT
  DIST C:   12 123 123 12345 1234 1234 1234
RESULT C:    2   3   3     5    4    4    4     = 2335444
RESULT C:    2   3   3     5    4    4    4     = QRRRRRR
```

Fig. 11 - final result. Source: the authors.

The exact strategy to handle gaps is the substitution (partial) table:

From n (or equal)	To n (or equal)	Identifier
0	2	1
3	4	2
17	18	9
..	..	..
19	20	A
..	..	..
39	99999	Z

- handle gaps substitution matrix. Source: the authors.

Also the repetitions can be compressed using the sample (partial) table:

Repetition	Identifier
1111111111	a
..	..
11	i
2222222222	j
..	..
22	r
12	s
21	t

- naive compression substitution matrix. Source: the authors.

Sample of Neisseria Meningitidis before processing:

AAAAAATGCTCCTGTTTTCTCGTTTAGAATAAAGAAACAGGAGCGTT
TTGCGTTTTCAGACGGCATTTGAAAACCAATGCTGTCTGAAAGACAG
AATCCGTGAAAACTCCCCACGCAGGTATTATCCCGATCGGGTGTAAA

The same sequence after processing:

34233i1t3i351q316i52i

Experiments

For the experiments, a tool with a graphical interface was built as an application for Windows (it can be ported to other operating systems). The purpose is to allow the researcher to build experiments with minimal computer knowledge. This tool allows the user to organize their research projects (diagnoses) into patients and runs. It is also possible to automatically download ready-made experiments directly from the NCBI and manipulate the algorithm parameters to vary its sensitivity and specificity. The Fig. 12 shows a screenshot of the run's quality control panel, allowing the user to manipulate the controls and apply different filters.

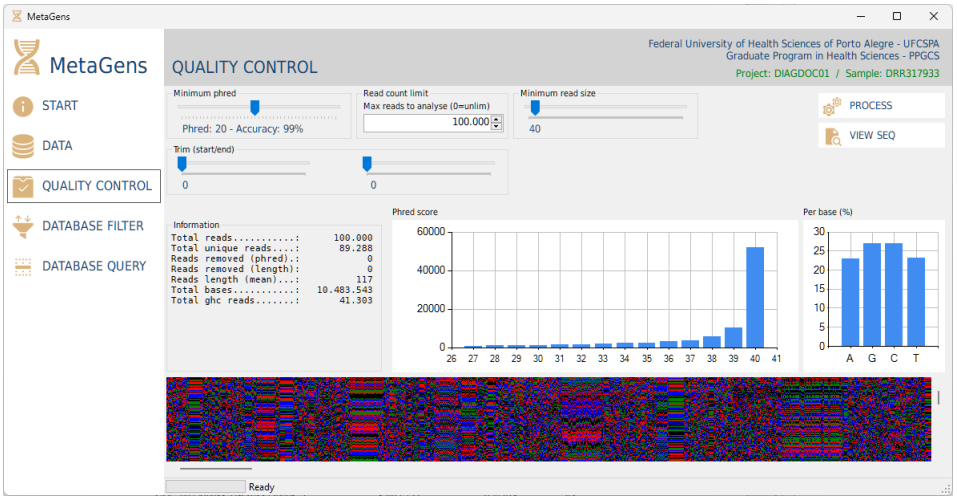


Fig. 12 - On this screen you can calibrate the software, allowing the specification of quality control. Source: the auhors

Another important part of the tool is the selection of refseqs for building the diagnostic project. The researcher can filter references among the more than 220,000 records available in the NCBI database and further speed up the alignment process. This step is totally visual and the download of refseqs is also automated, directly

from one of the available NCBI mirrors (cloud included). The Fig. 13 shows what the reference selection panel looks like.

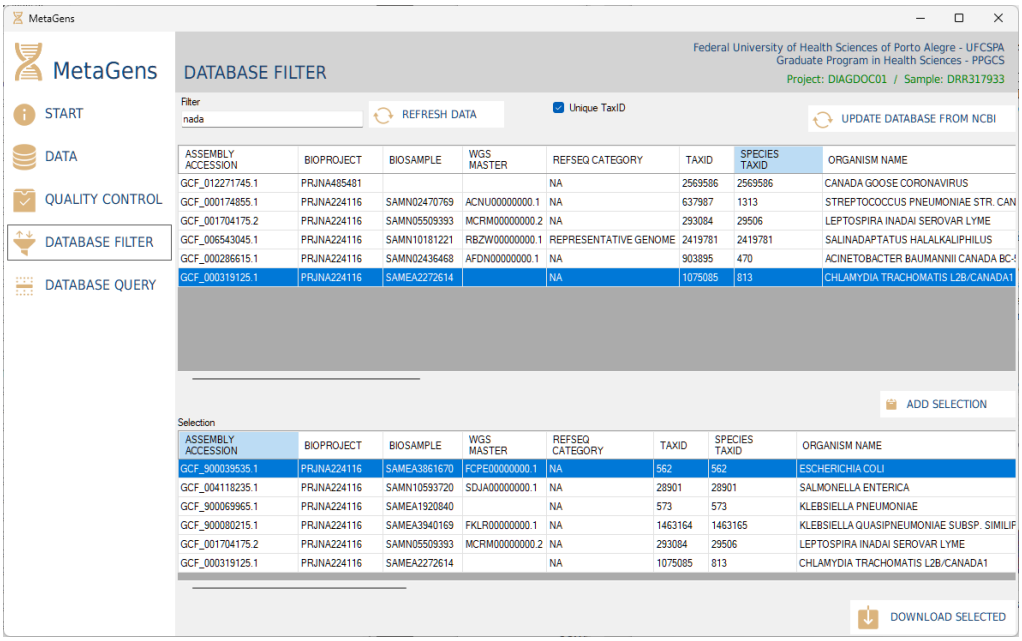


Fig. 13 - When filtering the reference sequences, it is possible to reduce the size of the database in order to reduce the analysis time. Source: the authors.

After filtering and selecting refseqs, the next step is processing the sequences and aligning the reads with the selected refseqs. It is at this stage that the massive processing is computed. All CPU cores (CPU-Threads included) are used in order to make the most of the available equipment. Fig. 14 shows the alignment control panel and some results found.

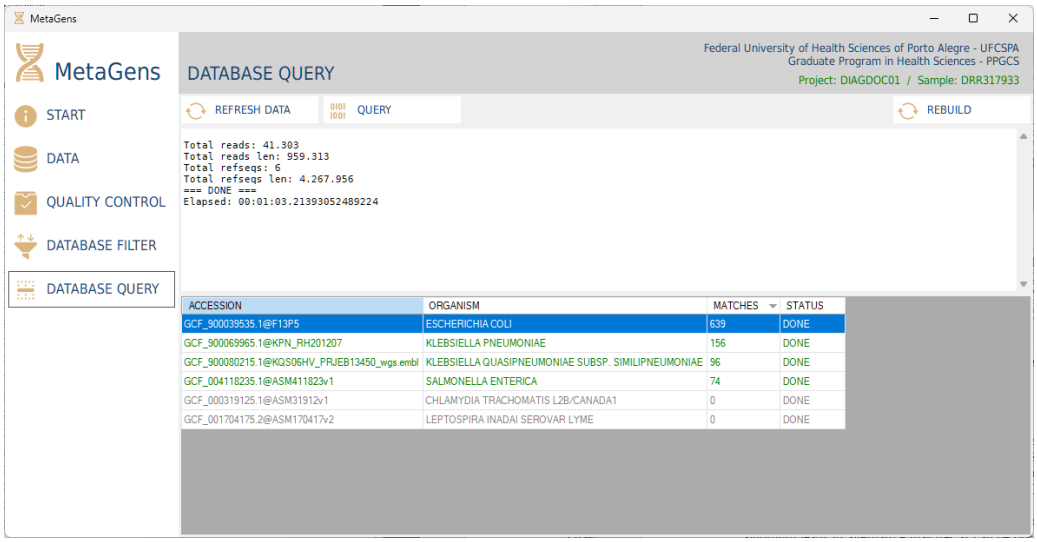


Fig. 14 - On the query screen, it is possible to follow the progress of the alignment process. The table shows the subjects under analysis and their matches. Source: the authors.

Discussion and results

The first impact of the algorithm (O(n)) is the data size reduction (~80%) in comparison with the original nucleotide sequence. A sample read of the SARS-COV2 containing 480 nucleotides yields a 79 character identity. The overall reduced size increases the main performance of the massive matching process. The algorithm is ~10x faster than naive search (up to O(mn), both implemented in C#, using dotnet framework). As an example, the Neisseria Meningitidis strain 433_NMEN (ASM106340v1) contains 2.397.512 bytes (nucleotides) before processing and 373.439 bytes after compression. The high sensitivity nature of the algorithm leads to alternative matches as can be observed on the accession SRR12665177. The overall coverage leads to Neisseria Meningitidis (100%) although other subjects can be matched too (Klebsiella Quasipneumoniae - 98% coverage). The table 1 shows an example of the length of the files before and after the process.

File	Original size (bytes)	Reduced Size (bytes)	Shrink %
GCF_000003925.1@ASM392v1	5.631.514	749.257	87
GCF_000006945.2@ASM694v2	5.013.482	797.157	84
GCF_900087615.2@WHOM	2.255.268	352.126	84
GCF_006334535.1@ASM633453v1	2.159.924	331.985	85
GCF_000002825.2@ASM282v1	54.436.962	6.471.680	88
GCF_004115315.1@ASM411531v1	6.507.834	1.034.216	84
GCF_003290055.1@ASM329005v1	5.893.322	970.815	84

Table 1 - comparison between before and after processing files. Source: the authors

For the experiments a GUI was implemented in the dotnet framework and published at *https://github.com/ghc4/metagens*. Since the algorithm is based on lossy compression, the data reduction may be translated as loss of information, but it is not. Both data from the sequencer machine and the REFSEQ is processed using the same technique, so the comparison is between the same "genomic language" - translated to a kind of wave. The final comparison is between the same metrics, and the match is preserved. Once the distance between the nucleotides can vary due to insertions and deletions, a substitution range can be defined by the user, making the indels irrelevant. The transitions and transversions may be irrelevants too because the algorithm is not analyzing all nucleotides between the "waves", only the "wavelength" is important to this approach. In comparison with dynamic programming, this technique is very permissive, detecting a great variety of subjects due to the nature of the matching strategy. It is possible, depending on how wide the parameters are, to reach false positives, however this algorithm - at this stage - may be more useful as a pre-filter, allowing faster dynamic programming (with smaller databases) or even a faster Blast.

In a direct comparison with the alignment algorithm used in Blast, the run SRR12665147 (taken from the NCBI biosample SAMN16133045) aligned with the subject *Neisseria Meningitidis* (GCF_008330805.1_ASM833080v1), resulted in over a million matches in 609.64 seconds of processing. On the same equipment, the proposed algorithm resulted in 23,622 matches in 49 seconds. The reduction in the number of matches occurs because the database was compressed and similar sequences were grouped, also almost all reads had some duplicates in the Blast analysis, streamlining the process as a whole. It is observed that the processing time has been reduced by more than 90%. Despite the comparison using the same hardware (i7-6700/64GB RAM), the Blast implementation is made using C/C++ while this algorithm prototype was written in C#, so an even higher gain is expected in the analysis of the results with better performance implementations (C/C++, Rust, etc). In another direct comparison, the same run (SRR12665147) aligned with the genome of

Neisseria Sicca (GCF_017753665.1_ASM1775366v1) took 175.47 seconds to process 69,235 matches while the proposed algorithm took 27.74 seconds. Despite the *Neisseria Sicca* genome being of almost same size as the Neisseria Meningiditis genome, the number of matches directly influences the Blast processing time, but does not have a significant impact on the processing of the proposed algorithm since what weighs more, in this case, is the complexity of nucleotide sequences. It is important to emphasize that this study considers the matches to be relevant for the purposes of diagnosing infection, and in this case the important thing is to identify the etiological agent and not necessarily its phylogeny.

Classifier	***N.Mening.* (matches)**	***N.Mening.* (time)**	***N.Sicca* (matches)**	***N.Sicca* (time)**
Kaiju	226,054	1,589 seconds	490	1,589 seconds
Kraken2	458,504	80 seconds	1,035	80 seconds
BLAST	Over one million	609 seconds	69,235	175 seconds
Centrifuge	245,417	89 seconds	95	89 seconds
MetaGens	12,461 (compressed)	30 seconds	430 (compressed)	27 seconds

Table 2: benchmark comparison on the alignment of the SRR12665147 with *Neisseria Meningiditis* and *Neisseria Sicca*.

Another important information is that the number of matches does not necessarily mean greater sensitivity or accuracy, since the reference database and the accession are compressed, the number of matches is expected to be smaller than the same uncompressed data, the reason is that the lossy compression will eventually generate equal sequences that will be discarded during the quality control process.

The metagenomics aided diagnosis of diseases is an important ally in several specific cases. Once the etiologic agent is identified, treatment can be accelerated and the patient's chances of improvement are increased. In some cases, as in infectious diseases of the CNS, the time to diagnosis is decisive in the outcome of the clinical case and, the faster it is, the greater the chances of cure. The combination of several matching acceleration algorithms can be the key to the efficiency in the search for etiological agents in a large mass of genetic data.

In the last few years, clinical metagenomics has jumped from ~70 publications on PubMed in 2010 to ~540 publications in 2019, probably as a result of the advances on computational methods and development of new sequencing technologies. While the mainstream factories are spreading genetic sequencers through the biotech laboratories, some companies are developing pocket sequencers [29] at a cost of ~U$ 4.500,00 that produce up to 30Gb of data. In the near future, it may allow the self diagnosis of some diseases with effective confidence, thus it will result in a massive amount of data to process, turning the analysis even more challenging.

Conclusion

In some preliminary experiments, with use of functional massive parallel processing, it was possible to observe an interesting gain of performance in comparison with the very same structure running standard algorithms, using the reduced mass of data, although it is not yet parametrized or precisely measured because of the different metrics involved. Through the years, the researchers are evolving their techniques to speed up the analysis process and produce results earlier [30,31,32]. Computer technology is also evolving, increasing speed and capacities in new processor generation. However, the genomic databases are also growing in an exponential

way. Consequently, it is necessary for a faster solution able to deal with large amounts of data comparisons, enabling the use of clinical metagenomics as an important weapon against infections of difficult diagnosis and treatment.

Acknowledgements

This work is funded by grant number 440084/2020-2 from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - Brazilian Ministry of Science and Technology) and Amazon Web Service (AWS - Cloud Credits for Research).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. Chen K, Pachter L (2005) Bioinformatics for Whole-Genome Shotgun Sequencing of Microbial Communities. PLOS Comput Biol 1:e24. https://doi.org/10.1371/journal.pcbi.0010024
2. (2009) Metagenomics versus Moore's law. Nat Methods 6:623–623. https://doi.org/10.1038/nmeth0909-623
3. Kakirde KS, Parsley LC, Liles MR (2010) Size Does Matter: Application-driven Approaches for Soil Metagenomics. Soil Biol Biochem 42:1911–1923. https://doi.org/10.1016/j.soilbio.2010.07.021
4. Chiu CY, Miller SA (2019) Clinical metagenomics. Nat Rev Genet 20:341–355. https://doi.org/10.1038/s41576-019-0113-7
5. Dekker JP (2018) Metagenomics for Clinical Infectious Disease Diagnostics Steps Closer to Reality. J Clin Microbiol 56:. https://doi.org/10.1128/JCM.00850-18
6. Rotbart HA. Viral meningitis. Semin Neurol 20: 277-292, 2000. doi: 10.1055/s-2000-9427
7. Erdem H, Inan A, Guven E, Hargreaves S, et al. The burden and epidemiology of community-acquired central nervous system infections: a multinational study. Eur J Clin Microbiol Infect Dis. Apr 10, 2017. doi: 10.1007/s10096-017-2973-0
8. Pallen MJ (2014) Diagnostic metagenomics: potential applications to bacterial, viral and parasitic infections. Parasitology 141:1856–1862. https://doi.org/10.1017/S0031182014000134
9. Compeau P (2015) BIOINFORMATICS ALGORITHMS,VOL.I, 2nd Edition. Active Learning Publishers, La Jolla, CA
10. Benefits of SBS Technology | Robust sequencing data quality. https://www.illumina.com/science/technology/next-generation-sequencing/sequencing-technology/sbs-benefits.html. Accessed 26 Oct 2020
11. Council NR (2007) The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet
12. GenBank and WGS Statistics. https://www.ncbi.nlm.nih.gov/genbank/statistics/. Accessed 26 Oct 2020
13. Cook DA, Hatala R, Brydges R, et al (2011) Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. Jama 306:978–988
14. Sequencing Quality Scores. https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/quality-scores.html. Accessed 26 Oct 2020
15. FASTQ files explained. https://support.illumina.com/bulletins/2016/04/fastq-files-explained.html. Accessed 26 Oct 2020
16. Troubleshooting Your Data | Roswell Park Comprehensive Cancer Center. https://www.roswellpark.org/shared-resources/genomics/services-and-fees/sanger-sequencing/troubleshooting-your-data. Accessed 26 Oct 2020
17. Interpretation of Sequencing Chromatograms | Sanger Sequencing/Fragment Analysis FAQs. In: U-M Biomed. Res. Core Facil. https://brcf.medicine.umich.edu/cores/advanced-genomics/faqs/sanger-sequencing-faqs/interpretation-of-sequencing-chromatograms/. Accessed 26 Oct 2020
18. Porta A, Enners E (2012) Determining Annealing Temperatures for Polymerase Chain Reaction
19. Shewaramani S (2015) Effects of aerobic and anaerobic environments on bacterial mutation rates and mutation spectra assessed by whole genome analyses : a thesis presented in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Genetics at Massey University, Palmerston

North, New Zealand. Thesis, Massey University
20. Levenshtein VI (1966) Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Sov Phys Dokl 10:707
21. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197. https://doi.org/10.1016/0022-2836(81)90087-5
22. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453. https://doi.org/10.1016/0022-2836(70)90057-4
23. Burrows M, Wheeler DJ (1994) A Block-sorting Lossless Data Compression Algorithm. Digital, Systems Research Center
24. Hal Berghel and David Roach, An Extension of Ukkonen's Enhanced Dynamic Programming ASM Algorithm. http://berghel.net/publications/asm/asm.php. Accessed 26 Oct 2020
25. Carroll H, Clement M, Ridge P, Snell Q (2006) Effects of Gap Open and Gap Extension Penalties. In: undefined. /paper/Effects-of-Gap-Open-and-Gap-Extension-Penalties-Carroll-Clement/ce0915ad115793154e0e3c9e4db76ac932248d76. Accessed 27 Oct 2020
26. Eddy SR (2004) What is dynamic programming? Nat Biotechnol 22:909–910. https://doi.org/10.1038/nbt0704-909
27. Biosample SAMEA7049302. From https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=ERR4329467. Access on june 2020.
28. Thomas, Sean & Martinez, L & Westenberger, Scott & Sturm, Nancy. (2007). A population study of the minicircles in Trypanosoma cruzi: Predicting guide RNAs in the absence of empirical RNA editing. BMC genomics. 8. 133. 10.1186/1471-2164-8-133.
29. MinION. In: Oxf. Nanopore Technol. http://nanoporetech.com/products/minion. Accessed 27 Oct 2020
30. Mishra P, Bhoi N (2020) Genomic signal processing of microarrays for cancer gene expression and identification using cluster-fuzzy adaptive networking. Soft Comput. https://doi.org/10.1007/s00500-020-05068-3
31. Quaid MAK, Jalal A (2020) Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm. Multimed Tools Appl 79:6061–6083. https://doi.org/10.1007/s11042-019-08463-7
32. Chattopadhyay A, Menon V (2020) Fast simulation of Grover's quantum search on classical computer. ArXiv200504635 Quant-Ph
33. Simon, H. Y., Siddle, K. J., Park, D. J., & Sabeti, P. C. (2019). Benchmarking metagenomics tools for taxonomic classification. Cell, 178(4), 779-794.

CAPÍTULO DE LIVRO INTERNACIONAL

Cervi, G.H., Flores, C.D., Thompson, C.E. (2022). Metagenomic Analysis: A Pathway Toward Efficiency Using High-Performance Computing. In: Yang, XS., Sherratt, S., Dey, N., Joshi, A. (eds) Proceedings of Sixth International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems, vol 236. Springer, Singapore. https://doi.org/10.1007/978-981-16-2380-6_49

Metagenomic Analysis: A Pathway Towards Efficiency Using High Performance Computing

Gustavo Henrique Cervi[1], Cecília Dias Flores[1], Claudia Elizabeth Thompson[1]

[1]Department of Health Sciences, Federal University of Health Sciences (UFCSPA), Porto Alegre, RS, Brazil
gustavohc@gmail.com

Abstract. Clinical metagenomics is a technique that allows the search for an infectious agent in a biological tissue/fluid sample. Over the past few years, this technique has been refined and the volume of data increases in a logarithmic proportion. DNA sequencers generate gigabytes of data and these data must be paired with bases that exceed one terabyte. The molecular nature of DNA does not always allow the use of traditional exact string search algorithms, since biologically analogous sequences are not syntactically identical without exception, which makes it difficult to analyze matching sequences. In the case of clinical diagnosis, for specific situations, the sooner the diagnosis is available to the physician, the greater the chance of the patient's recovery, so the data processing must be made as soon as possible, maintaining the necessary detail and coverage. This paper describes some current techniques of computational processing and acceleration of the search for genomic data and possible alternative computational paths to streamline the process of metagenomic diagnosis.

Keywords: Metagenomic, Diagnosis, Computing Methodologies, High-Throughput Nucleotide Sequencing

1 Introduction

Metagenomics refers to the study of the genetic material collected from environmental biological samples, such as soil, water, fluids, and others, processed at the lab (the "wet" part) and sequenced by a machine (genetic sequencer) [1]. A huge amount of data, which may vary from some gigabytes to a terabyte data, is obtained by this process called sequencing [2]. The "dry" analysis is made in silico (computational environment), using specific softwares that analyse data and produce results, including statistical data [3]. In the last years, the metagenomic analysis has been applied as clinical metagenomics [4], where the environmental samples were collected from human tissues and fluids, with the main objective of looking for infectious and parasitic biological agents. A special chapter of this history is to use metagenomics as a tool for the diagnosis of infectious diseases [4–6]. While the genomic analysis is focused on a single biological subject, metagenomics collects genetic information from all biological subjects present in the sample [7]. Since the DNA is measured in atoms and molecules (molecular biology), the amount of genetic material from a single sample is small to be sequenced, thus the genetic material obtained from samples must be duplicated and amplified through a technique called PCR. This method is widely used and consists of thermal cycles that denatures (breaks) the DNA allowing the duplication and amplification processes [4, 7]. In the next phase, a machine identifies the nucleotide sequences.

In the early days of the genetic sequencers, this was a time consuming job because the process occurred in a serial way [7]. Nowadays, the NGS (Next Generation Sequencing) [8] works in a massive parallel way, obtaining gigabytes of DNA data per run (chemical process cycle). Until the mid-2000s, when the first NGS sequencers appeared, it was hard to obtain data. Today, it is hard to analyse the huge amount of data" [7].

The field of metagenomics is extensive and this paper aims to discuss the computational steps, which are commonly performed using software "pipelines" [3]. In these "pipelines", the raw data "enters" the first software (normally a filter) and passes through a sequence of other softwares and scripts, each one with a specific function (filter, organizer, alignment, matching, statistics, etc) [7]. The term "pipeline" may lead to an incorrect interpretation where an abstract stream of data enters the "pipe" and, while data is flowing in, there is data flowing out. It is not that simple, although it is possible to process data while the sequencer machine is still producing results.

2 Pipeline

In general terms, a metagenomic pipeline can be divided into four steps: (i) filtering data, (ii) aligning with databases, (iii) filtering results, and (iv) statistical reports. Despite the fact the DNA vocabulary is very reduced, containing only four letters, the combination of these letters are the "source code" of all living matter, which varies from a simple thousand-base bacteria to a multi-billion-base animal genome [7]. The computational processing behind this amount of data is a challenging problem [9], especially when the researcher/physician is running against the time, for example, waiting for the diagnosis of a disease. In a computational perspective, the main time consuming problem relies on the huge volume of data to be filtered, organized and compared. Once the genetic sequencer finishes the sequencing process, all that data must be processed. The NGS (Next Generation Sequencing) technology produces a large amount of "short reads", which are sequences up to 300 nucleotide bases (adenosine, guanine, thymine or cytosine). They are represented as a string of data like "ACGGATCGATTCGATTG…". The comparison of these sequences with the reference database results in a possible diagnosis with the identification of the specific etiological agent (virus, bacteria and/or fungus). These reference databases are commonly accessible from public sites like GenBank (NCBI, USA), EMBL Bank (EMBL-EBI, Europe) and DDBJ Bank (DDBJ, Japan), which easily overpass the terabyte of data. Figure 1 shows the evolution of the GenBank database through the years.

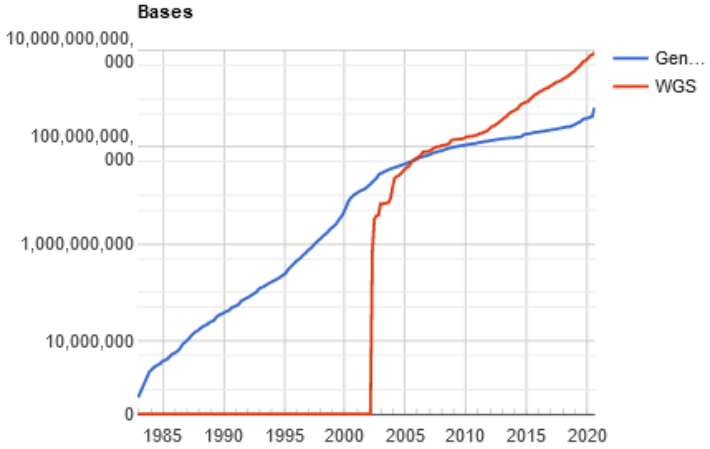


Figure 1: GenBank and WGS statistics - amount of data over the year. Source: NBCI statistics webpage [10]

3 Quality control

The first stage after sequencing is the "quality control". The sequencer produces a large amount of data, but not all with the same quality. This step is important to remove "low quality" data [11, 12]. In summary, the first generation and some second generation sequencers collect information through a fluorescent agent bound to the nucleotide [8], using a very precise wavelength laser, capturing the light emitted by the molecule to infer the sequence. The resulting signal may be biased or not deterministic. The sequencer calculates the "quality" of the read based on the light intensity and writes the score in the result file - each sequenced nucleotide has its specific quality score [13]. Figures 2 and 3 show chromatograms with low (multiple peaks per base) and good quality

data, respectively, obtained by a first generation sequencer [12].

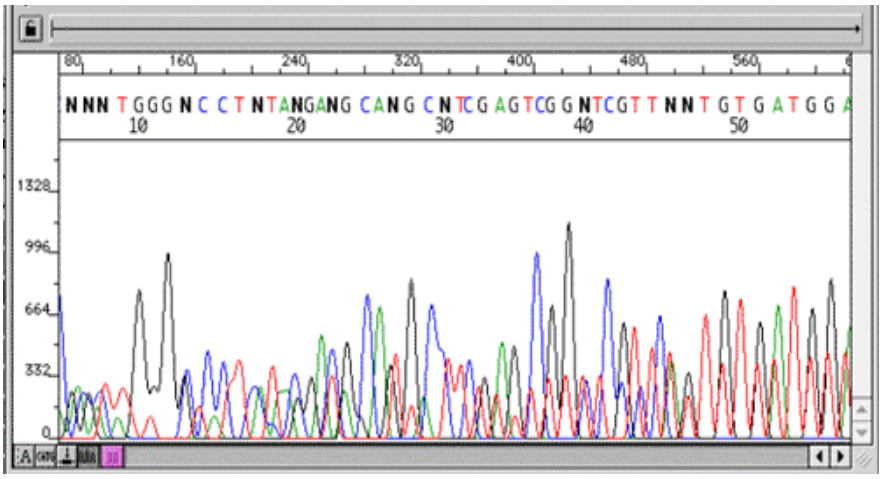


Figure 2: chromatogram with multiple peaks per base - low quality data. Source: Roswell Park Comprehensive Cancer Center [14]

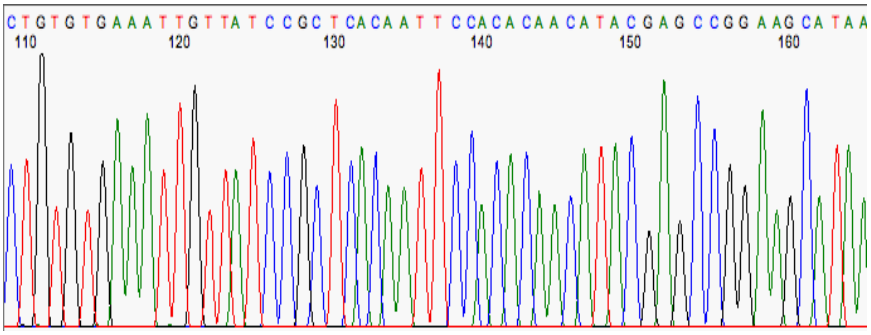


Figure 3: chromatogram indicating good quality of data sequence. Source: U-M Biomedical Research Core Facilities [15]

4 Amplification

All first and second generation sequencers require a DNA amplification step, where the DNA is duplicated into multi million copies, amplifying the signal to be detected by the machine. After sequencing, duplicate reads are removed resulting in data reduction [16].

5 Host removal

This step is also important to speed up the analysis and reduce the risk of bias. Since the sample is obtained from a living host (human), it is likely to have the host DNA present in data after sequencing. This stage is performed by searching for the host DNA, in the result file, through comparison with the reference genome available in public databases. Once the host reads are identified, they have to be removed. In case of human samples, a reference human genome is used, preferentially the most recent and curated available data.

6 Searching through Reference Databases

This step is the most computing expensive task. The sequencer yields a huge amount (>100 million) of short reads (up to 300 nucleotides bases in current NGS technology) written in a text file, whose format is commonly the FASTQ type [13] (same from the original FASTA file format but with quality information). Each read is represented by one DNA string like "ACGATCGATTCGGA(…)" and it must be compared to reference datasets (terabytes of genomes from all sorts of living organisms, available on public organized databases). The first guess is O(m+n) like Knuth-Morris-Pratt or O(m)+Ω(n/m) like Boyer-Moore algorithms could be applied to solve this problem. However, these algorithms cannot produce efficient results from a biological viewpoint. In order to be able to compare the new sequence to all sequences committed in databases, it is necessary to perform

sequence alignment. The DNA is not a rigid and static sequence, it is submitted to evolutionary forces such as mutation, selection, genetic drift, and migration. Considering the mutational aspect, the DNA substitutions can be classified as (i) transitions: when involve bases with similar shape, interchanges of two-ring purines (A ⥋ G) or one-ring pyrimidines (C ⥋ T) and (ii) transversions: when involve substitutions of one-ring and two-ring DNA bases, interchanges of purine for pyrimidine bases and vice-versa. Figure 4 indicates the possible transitions and transversions.

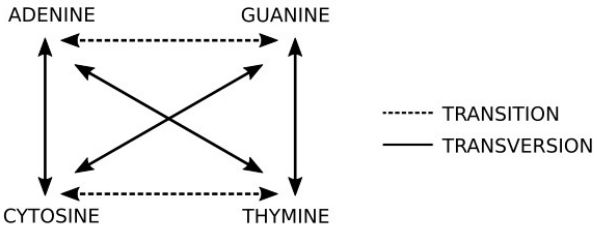


Figure 4: transitions vs transversions in a DNA sequence. Source: the authors, based on [17]

When comparing two sequences to obtain an alignment, the main objective is to identify the positional homology, i.e., identify sites with a common ancestry in the alignment. It may be necessary to include gaps (indels, corresponding to deletion in sequence 1 and insertion in sequence 2) to better accommodate one sequence in relation to another. In this sense, the sequence "ACGATCGAT" may be biologically equivalent to the sequence "ACGCTCGGAT" (one mutation and one indel), i.e., they may be homologous. Homology is a biological concept that indicates two sequences share a common ancestry. Common algorithms used to align sequences in genomic research are Levenshtein Distance [18], Smith-Waterman [19], Needleman-Wunsch [20], Burrows-Wheeler [21] plus hashing and its derivatives. Blast, which uses a heuristic method based on Smith-Waterman, is the most commonly used software to perform local alignment. It allows identifying subject sequences in a database that are similar to a query sequence. Figure 5 shows a local alignment obtained by a Blast search, with the indication of mismatches (blue arrow and lack of | symbol), indels (green arrow and gaps) and matches (red arrow and | symbol).

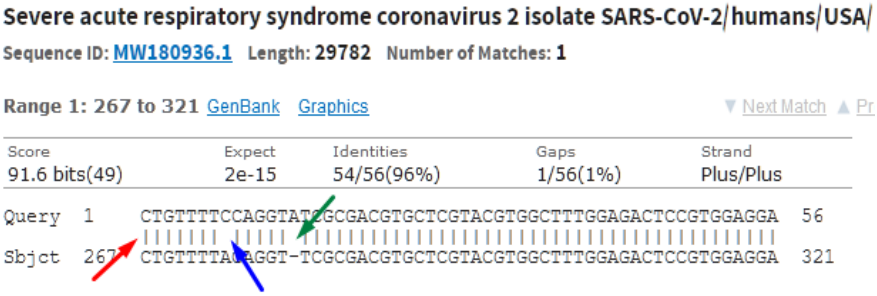


Figure 5: NCBI Blast output example. Source: the authors (sample on NCBI Blast)

6.1 Dynamic Programming

Dynamic programming applied to bioinformatics (eg. Levenshtein Distance, Smith-Waterman and Needleman-Wunsch) has complexity in order of O(mn) in the worst case, but it is possible to improve as demonstrated by Berghel and Roach [22]. It is very time-consuming in terms of computational processing, although it is possible to parallelize the task since an input does not depend on other data in the same processing stage. It is not rare that a software solution has more than one combination of algorithms. For example, in case of seed-and-extend algorithms, it is very common for a software aligner using the Burrows-Wheeler algorithm to reduce the size and hash tables to find the seed portions. Dynamic programming applied to sequence alignment can be explained using a 2-dimensional matrix where two sequences are compared and there are three main steps: (i) matrix initialization; (ii) matrix fill (scoring), and (iii) traceback (alignment). Match, penalty-gap and mismatch values are defined according to a score [23]. During the matrix fill, for each cell, all possibilities are evaluated and received a value: (i) in diagonal: match or mismatch; (ii) gap in sequence y, and (iii) gap in sequence x. The traceback step determines the actual alignment(s) that result in the maximum score. In Figure 6, the maximum alignment score for the two sequences is 11 and the best alignment is shown in red.

			1	2	3	4	5	6	7	8	9
		-	G	A	A	C	G	T	A	G	T
	-	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
1	G	-2	1	-1	-3	-5	-7	-9	-11	-13	-15
2	A	-4	-1	2	0	-2	-4	-6	-8	-10	-12
3	A	-6	-3	0	3	1	-1	-3	-5	-7	-9
4	C	-8	-5	-2	1	4	2	0	-2	-4	-6
5	G	-10	-7	-4	-1	2	5	3	1	-1	-3
6	A	-12	-9	-6	-3	0	3	4	4	2	0
7	G	-14	-11	-8	-5	-2	1	4	3	5	3
8	T	-16	-13	-10	-7	-4	-1	2	1	3	6
			↑	↑	↑	↑	↑	↑	←	↑	↑
			G	A	A	C	G	T	A	G	T
			G	A	A	C	G	-	A	G	T

Figure 6: example of dynamic program applied to sequence alignment. Source: the authors. Reference: [24]

6.2 Accelerated Alternatives

Through the years, several alternative methods were designed to accelerate the process of alignment and matching of biological sequences. Some of them use common hardware pieces like a standard x86 home computer containing graphics cards with GPU processors and others using specific hardware pieces. In summary, there are four main approaches: ASIC, ASIP, FPGA, and GPU.

The ASIC approach. The Application Specific Integrated Circuits (ASIC) may be the lowest level of the computational data integration where the circuit is designed to be the most specific as possible [6]. In the 90's, when the MP3 was widespread, to play a single sound a common home computer had to be used with full capacities to run the MP3 algorithm and decode the sound. The industry ran as fast as it could to design an integrated chip and the result was the flood of the pocket MP3 players, costing no more than a few US dollars. As an example, this approach was observed again with the boom of the cryptic coins. The first coins were mined with a simple home computer. As the complexity of the mining was growing, a simple computer was not enough to mine a coin (the electric cost was higher than the value of the minerated coin). The solution: specific computer hardwares called "rigs" to process the coin with the maximum efficiency. An actual example of ASIC used in metagenomics is the Nanopore [25] for the sequencing and DNASSWA [26] and [27] for sequence alignment. It is the most computational efficient way to standardized algorithms. The weak part relies on the cost per device and the time to develop the solution. The ASIC must be engineered by specialist (at a high cost) and produced by a capable industry, at a huge cost, sometimes bypassing a million US Dollars to build the masks and necessary artifacts. This cost can vary and the manufacturers do not actually publish their specific prices, but it is possible to ask for an estimation if you have the project and parameters well defined [28]. This approach, while known by the best results, due to the high prices involved is financially not viable for academic research purposes although it is viable in large scale consolidated products. The figure 7 shows a layout diagram of one ASIC chip in 40nm technology.

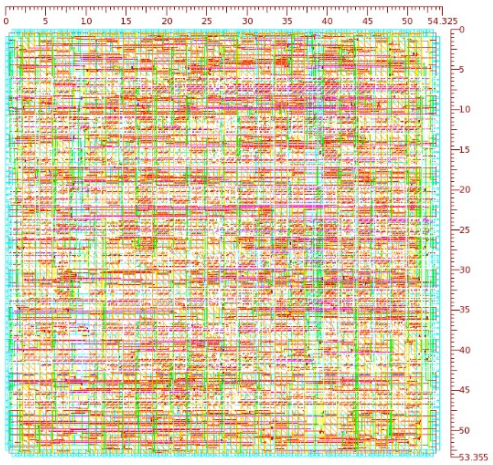


Figure 7: layout of the TSMC 40nm. Source: Darwin [29]

The ASIP approach. The General Purpose Processor (GPP) is an architecture that allows the solution of a vast range of problems [30], from calculations and signal analysis to neural networks. The instruction set is developed with general processing in mind. This architecture can be useful to the vast majority of the computations but may be not highly efficient when specific tasks are required. One of the most known tasks the GPP has struggled through time is the signal processing in multimedia computation such as lossy data compression (audio and video) and other tasks that require streaming processing. In 1997, Intel released the Pentium MMX processor containing 47 new instructions [30] focused on multimedia and bringing a new approach to the Intel x86 family. This was the first time the brand used SIMD (Single Instruction Multiple Data) in their products of the x86 line. In some cases, these instructions increased the performance 23 times in comparison to the equivalent task using the standard instructions, as shown by [31]. A simple task that required 1.881 cycles to perform was reduced to 81 cycles using the specific SIMD instruction. This evolution was a convergence between two technologies: the GPP and the ASIP (Application Specific Instruction-set Processor), also known as co-processor. The ASIP approach has the flexibility of the GPP and the performance of an ASIC [32] but has exactly the same financial issues of the ASIC. It costs about the same as an ASIC to be built and it may be prohibitive for academic and research purposes. Projects that implement this solution in the genomics are actually using FPGA to synthesize the ASIP, one example is this project [33].

The FPGA approach. The Field Programmable Gate Array (FPGA) is an integrated circuit that contains an array of programmable (configurable) logic blocks. These logic units (LU) can be configured as logic gates (AND, OR, XOR), flip-flops and memory elements [34], as seen on figure 8.

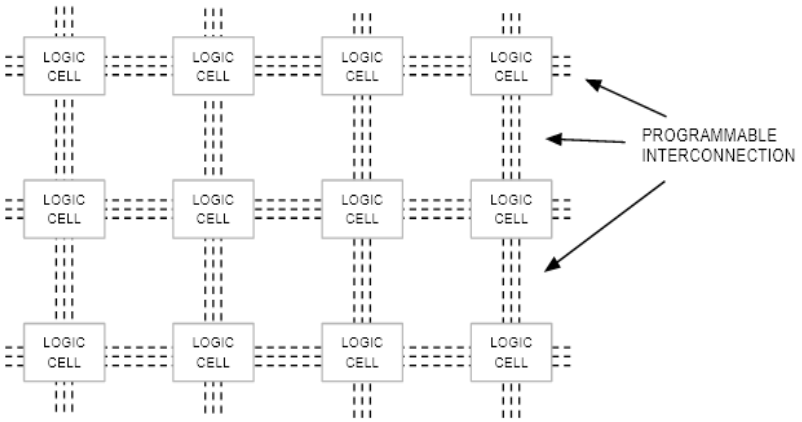


Figure 8: detail of a logic unit in a PFGA. Source: the authors.

The FPGA approach is the most viable and feasible to academic researchers due to low cost and high availability of the required hardware. A simple research board can cost as low as 50 US Dollars. The manufacturers like [35] and [36] have libraries that help the developer to explore its functionalities. A simple FPGA implementation can speed up at 10x over the original GATK pipeline [37]. Metagenomic projects that use this approach can be listed as [37–39], yielding results in order of 81 times faster than software and 32 times more cost efficient [39].

The GPU approach. In the late 1970's, the early computer systems had adopted auxiliary processors to handle the video signal and display elements, which led to what is called today as GPU. The term GPU stands for Graphics Processor Unit and through the time was optimized to the level of a high performance massive parallel processing unit, allowing a large amount of calculations in parallel [40]. The direct benefit is the advance and performance of the computer display graphics (2D and 3D). As the technology advanced, other computations took advantage of the GPU as a way to optimize results such as matrix calculations and vector mathematics [41]. One of the optimized functions was the Smith-Waterman algorithm implementation in CUDASW++ [42] with use of the CUDA technology [43]. With the spread of the GPU through general computing machines (home computers included), the access to this technology was straightforward. The issue related to this approach is that the required GPU to make real world biological computations requires an upper level machinery that is not commonly available for general computing. Additionally, a single board can cost more than an array of standard computers. The figure 10 shows a process flow of a CUDA core.

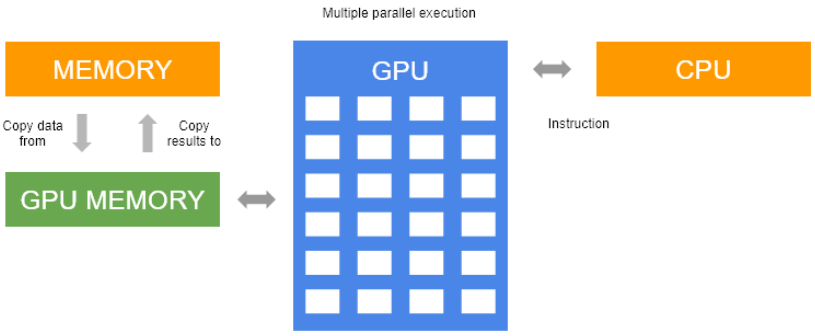


Figure 10: CUDA core parallel process. Source: the authors.

The use of GPUs in metagenomic analysis can yield a ~30 times faster performance in comparison with the same algorithm using only software implementation [44].

Software optimizations. The most common approach to this problem is software optimization [45–50], using all sorts of algorithms that may have advantages (speed and precision) to the process. These optimizations include hash tables [45], heuristics [46], reference comparison [47], and other mixed techniques [48–50]. Once the General Purpose Processor (GPP) is not optimized to solve this kind of problem, the computing time using the software optimization without a massive cluster may be longer than the patient can wait. The most common example of software is Blast (Basic Local Alignment Search Tool) [46], sponsored by the NCBI [36] and released at first in 1990. Blast can be simplified as a kind of search and match tool that receives a given string and tries to match with reference genomes looking for similarities that may conclude that the subject is the etiological agent. The algorithm consists of various phases and the most computational consuming is the "seed-and-extend", where the tool establishes short sequences called k-mer, finds the matches and extends them. This method may speed up the process 50 times in comparison with the string distancing Smith-Waterman algorithm [51]. Over the years, other solutions were developed based on this approach. Some of these projects use hardware acceleration and present a gain in performance of 20 times in comparison with computer (PC) based Blast [52]. Other solutions using GPU accelerations gained up to 10 times in performance when compared with standard PC software [53, 54]. Other approaches explore different algorithms, such as the HSBLAST [55], which claims to have the same result of the MEGABLAST (NCBI) with up to 20 times the performance, using the Burrow-Wheeler Transform (BWT) - the same algorithm used in other software such BWA [56] and Bowtie2 [57]. Table 1 illustrates some technologies and their implementations.

Table 1: technologies and their implementations. Source: the authors.

Technology	Application	Performance	Feasibility
ASIC	DNASSWA [26], SWASAD[58], Darwin [29]	Very high	Hard
ASIP	SIMD [59], GMAP [60], SSW [61]	Medium	High
FPGA	INTEL [35], Falcon [36], Survey [62]	Very high	Medium
GPU	Nvidia [41, 63], CUDASW [42]	High	Medium
Software	Kraken [45], DIAMOND [48], Kaiju [47], HSBlast [55]	Vary	High

7 Research

In the last years, clinical metagenomics has jumped from ~70 publications on PubMed in 2010 to ~540 publications in 2019, probably as a result of the advances on computational methods and development of new

sequencing technologies. While the mainstream factories are spreading genetic sequencers through the biotech laboratories, some companies are developing pocket sequencers [64] at a cost of ~U$ 4.500,00 that produce up to 30Gb of data. In the near future, it may allow the self diagnosis of some diseases with effective confidence. Other researchers are working on wearable devices like watches, rings, earrings, and glasses. It will result in a massive amount of data to process, turning the analysis even more challenging. The traditional algorithms and the commonly used hardware, as in the cryptocoins case, may be not enough for the probe job. New technologies like fuzzy pattern matching [65], signal analysis [66], and even quantum computing [67] may be a game changer that will bring metagenomics research to the masses.

8 Conclusion

As observed through the years, the researchers are evolving their techniques to speed up the analysis process and produce results earlier. The computer technology is also evolving, increasing speed, and capacities in new processors generation. However, the genomic databases are also growing in an exponential way. Consequently, a faster solution is necessary to deal with large amounts of data comparisons, enabling the use of clinical metagenomics as an important weapon against infections of difficult diagnosis and treatment.

Acknowledgements

This work is funded by grant number 440084/2020-2 from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - Brazilian Ministry of Science and Technology) and Amazon Web Service (AWS - Cloud Credits for Research).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. Chen K, Pachter L (2005) Bioinformatics for Whole-Genome Shotgun Sequencing of Microbial Communities. PLOS Comput Biol 1:e24. https://doi.org/10.1371/journal.pcbi.0010024
2. (2009) Metagenomics versus Moore's law. Nat Methods 6:623–623. https://doi.org/10.1038/nmeth0909-623
3. Kakirde KS, Parsley LC, Liles MR (2010) Size Does Matter: Application-driven Approaches for Soil Metagenomics. Soil Biol Biochem 42:1911–1923. https://doi.org/10.1016/j.soilbio.2010.07.021
4. Chiu CY, Miller SA (2019) Clinical metagenomics. Nat Rev Genet 20:341–355. https://doi.org/10.1038/s41576-019-0113-7
5. Dekker JP (2018) Metagenomics for Clinical Infectious Disease Diagnostics Steps Closer to Reality. J Clin Microbiol 56:. https://doi.org/10.1128/JCM.00850-18
6. Pallen MJ (2014) Diagnostic metagenomics: potential applications to bacterial, viral and parasitic infections. Parasitology 141:1856–1862. https://doi.org/10.1017/S0031182014000134
7. Compeau P (2015) BIOINFORMATICS ALGORITHMS,VOL.I, 2nd Edition. Active Learning Publishers, La Jolla, CA
8. Benefits of SBS Technology | Robust sequencing data quality. https://www.illumina.com/science/technology/next-generation-sequencing/sequencing-technology/sbs-benefits.html. Accessed 26 Oct 2020
9. Council NR (2007) The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet
10. GenBank and WGS Statistics. https://www.ncbi.nlm.nih.gov/genbank/statistics/. Accessed 26 Oct 2020
11. Cook DA, Hatala R, Brydges R, *et al* (2011) Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. Jama 306:978–988
12. Sequencing Quality Scores. https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/quality-scores.html. Accessed 26 Oct 2020
13. FASTQ files explained. https://support.illumina.com/bulletins/2016/04/fastq-files-explained.html. Accessed 26 Oct 2020
14. Troubleshooting Your Data | Roswell Park Comprehensive Cancer Center. https://www.roswellpark.org/shared-resources/genomics/services-and-fees/sanger-sequencing/

troubleshooting-your-data. Accessed 26 Oct 2020
15. Interpretation of Sequencing Chromatograms | Sanger Sequencing/Fragment Analysis FAQs. In: U-M Biomed. Res. Core Facil. https://brcf.medicine.umich.edu/cores/advanced-genomics/faqs/sanger-sequencing-faqs/interpretation-of-sequencing-chromatograms/. Accessed 26 Oct 2020
16. Porta A, Enners E (2012) Determining Annealing Temperatures for Polymerase Chain Reaction
17. Shewaramani S (2015) Effects of aerobic and anaerobic environments on bacterial mutation rates and mutation spectra assessed by whole genome analyses : a thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Genetics at Massey University, Palmerston North, New Zealand. Thesis, Massey University
18. Levenshtein VI (1966) Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Sov Phys Dokl 10:707
19. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197. https://doi.org/10.1016/0022-2836(81)90087-5
20. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453. https://doi.org/10.1016/0022-2836(70)90057-4
21. Burrows M, Wheeler DJ (1994) A Block-sorting Lossless Data Compression Algorithm. Digital, Systems Research Center
22. Hal Berghel and David Roach, An Extension of Ukkonen\'s Enhanced Dynamic Programming ASM Algorithm. http://berghel.net/publications/asm/asm.php. Accessed 26 Oct 2020
23. Carroll H, Clement M, Ridge P, Snell Q (2006) Effects of Gap Open and Gap Extension Penalties. In: undefined. /paper/Effects-of-Gap-Open-and-Gap-Extension-Penalties-Carroll-Clement/ce0915ad115793154e0e3c9e4db76ac932248d76. Accessed 27 Oct 2020
24. Eddy SR (2004) What is dynamic programming? Nat Biotechnol 22:909–910. https://doi.org/10.1038/nbt0704-909
25. How it works. In: Oxf. Nanopore Technol. http://nanoporetech.com/how-it-works. Accessed 26 Oct 2020
26. An ASIC application for DNA sequencing by Smith-Waterman algorithm (DNASSWA) - UQ eSpace. https://espace.library.uq.edu.au/view/UQ:295057. Accessed 26 Oct 2020
27. Halim AK, Majid ZA, Mansor MA, *et al* (2010) Design and Analysis of 8-bit Smith Waterman based DNA Sequence Alignment Accelerator's Core on ASIC Design Flow. In: 2010 Fourth UKSim European Symposium on Computer Modeling and Simulation. pp 126–131
28. PeopleVine S via ASICs. https://www.sigenics.com/page/asics-c. Accessed 26 Oct 2020
29. Turakhia Y, Zheng KJ, Bejerano G, Dally WJ (2017) Darwin: A Hardware-acceleration Framework for Genomic Sequence Alignment. bioRxiv 092171. https://doi.org/10.1101/092171
30. Saltzer JH, Kaashoek MF (2009) Principles of Computer System Design: An Introduction. Morgan Kaufmann
31. Conte G, Tommesani S, Zanichelli F (2000) The long and winding road to high-performance image processing with MMX/SSE. In: Proceedings Fifth IEEE International Workshop on Computer Architectures for Machine Perception. pp 302–310
32. Shahabuddin S, Janhunen J, Juntti M, *et al* (2014) Design of a transport triggered vector processor for turbo decoding. Analog Integr Circuits Signal Process 78:611–622. https://doi.org/10.1007/s10470-013-0183-y
33. Vacek G (2011) Hybrid-Core Computing for High-Throughput Bioinformatics. J Biomol Tech JBT 22:S37–S38
34. FPGA Architecture for the Challenge. https://www.eecg.utoronto.ca/~vaughn/challenge/fpga_arch.html. Accessed 26 Oct 2020
35. FPGA Genomics - FPGA for Life Science Applications - Intel® FPGA. In: Intel. https://www.intel.com/content/www/br/pt/healthcare-it/products/programmable/applications/life-science.html. Accessed 26 Oct 2020
36. Falcon Accelerated Genomics Pipelines. In: Xilinx. https://www.xilinx.com/products/acceleration-solutions/1-zzroc0.html. Accessed 26 Oct 2020
37. Mahram A, Herbordt MC (2012) FMSA: FPGA-Accelerated ClustalW-Based Multiple Sequence Alignment through Pipelined Prefiltering. In: 2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines. pp 177–183
38. Jacob A, Lancaster J, Buhler J, Chamberlain RD (2007) FPGA-accelerated seed generation in Mercury BLASTP. In: 15th Annual IEEE Symposium on Field-Programmable Custom

Computing Machines (FCCM 2007). pp 95–106
39. Wu L, Bruns-Smith D, Nothaft FA, *et al* (2019) FPGA Accelerated INDEL Realignment in the Cloud. In: 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA). pp 277–290
40. GPU History: Hitachi ARTC HD63484 | IEEE Computer Society. https://www.computer.org/publications/tech-news/chasing-pixels/gpu-history-hitachi-artc-hd63484/. Accessed 26 Oct 2020
41. (2019) nVidia CUDA Bioinformatics: BarraCUDA. In: BioCentric. https://www.biocentric.nl/biocentric/nvidia-cuda-bioinformatics-barracuda/. Accessed 26 Oct 2020
42. Liu Y, Wirawan A, Schmidt B (2013) CUDASW++ 3.0: accelerating Smith-Waterman protein database search by coupling CPU and GPU SIMD instructions. BMC Bioinformatics 14:117. https://doi.org/10.1186/1471-2105-14-117
43. High Performance Computing Products and Solutions. In: NVIDIA. https://www.nvidia.com/en-us/high-performance-computing/. Accessed 26 Oct 2020
44. Kobus R, Hundt C, Müller A, Schmidt B (2017) Accelerating metagenomic read classification on CUDA-enabled GPUs. BMC Bioinformatics 18:11. https://doi.org/10.1186/s12859-016-1434-6
45. Wood DE, Lu J, Langmead B (2019) Improved metagenomic analysis with Kraken 2. Genome Biol 20:257. https://doi.org/10.1186/s13059-019-1891-0
46. BLAST: Basic Local Alignment Search Tool. https://blast.ncbi.nlm.nih.gov/Blast.cgi. Accessed 26 Oct 2020
47. Menzel P, Ng KL, Krogh A (2016) Fast and sensitive taxonomic classification for metagenomics with Kaiju. Nat Commun 7:11257. https://doi.org/10.1038/ncomms11257
48. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods 12:59–60. https://doi.org/10.1038/nmeth.3176
49. Bağcı C, Beier S, Górska A, Huson DH (2019) Introduction to the Analysis of Environmental Sequences: Metagenomics with MEGAN. In: Anisimova M (ed) Evolutionary Genomics: Statistical and Computational Methods. Springer, New York, NY, pp 591–604
50. Steinegger M, Söding J (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol 35:1026–1028. https://doi.org/10.1038/nbt.3988
51. Oehmen C, Nieplocha J (2006) ScalaBLAST: A Scalable Implementation of BLAST for High-Performance Data-Intensive Bioinformatics Analysis. IEEE Trans Parallel Distrib Syst 17:740–749. https://doi.org/10.1109/TPDS.2006.112
52. Herbordt MC, Model J, Sukhwani B, *et al* (2007) Single pass streaming BLAST on FPGAs. Parallel Comput 33:741–756. https://doi.org/10.1016/j.parco.2007.09.003
53. Vouzis PD, Sahinidis NV (2011) GPU-BLAST: using graphics processors to accelerate protein sequence alignment. Bioinformatics 27:182–188. https://doi.org/10.1093/bioinformatics/btq644
54. Liu W, Schmidt B, Muller-Wittig W (2011) CUDA-BLASTP: Accelerating BLASTP on CUDA-Enabled Graphics Hardware. IEEE/ACM Trans Comput Biol Bioinform 8:1678–1684. https://doi.org/10.1109/TCBB.2011.33
55. Chen Y, Ye W, Zhang Y, Xu Y (2015) High speed BLASTN: an accelerated MegaBLAST search tool. Nucleic Acids Res 43:7762–7768. https://doi.org/10.1093/nar/gkv784
56. Fast and accurate short read alignment with Burrows–Wheeler transform | Bioinformatics | Oxford Academic. https://academic.oup.com/bioinformatics/article/25/14/1754/225615. Accessed 26 Oct 2020
57. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–359. https://doi.org/10.1038/nmeth.1923
58. Han T, Parameswaran S (2002) Swasad: An Asic Design For High Speed Dna Sequence Matching. In: Proceedings of the 2002 Asia and South Pacific Design Automation Conference. IEEE Computer Society, USA, p 541
59. Jacob A, Paprzycki M, Ganzha M, Sanyal S (2008) Applying SIMD Approach to Whole Genome Comparison on Commodity Hardware. In: Wyrzykowski R, Dongarra J, Karczewski K, Wasniewski J (eds) Parallel Processing and Applied Mathematics. Springer, Berlin, Heidelberg, pp 1220–1229
60. GMAP and GSNAP for Genomic Sequence Alignment: Enhancements to Speed, Accuracy, and Functionality. https://doi.org/10.1007/978-1-4939-3578-9_15
61. Zhao M, Lee W-P, Garrison EP, Marth GT (2013) SSW Library: An SIMD Smith-Waterman

C/C++ Library for Use in Genomic Applications. PLOS ONE 8:e82138. https://doi.org/10.1371/journal.pone.0082138
62. Salamat S, Rosing T (2020) FPGA Acceleration of Sequence Alignment: A Survey. ArXiv200202394 Cs Q-Bio
63. (2020) NVIDIA Clara Parabricks. In: NVIDIA Dev. https://developer.nvidia.com/clara-parabricks. Accessed 27 Oct 2020
64. MinION. In: Oxf. Nanopore Technol. http://nanoporetech.com/products/minion. Accessed 27 Oct 2020
65. Mishra P, Bhoi N (2020) Genomic signal processing of microarrays for cancer gene expression and identification using cluster-fuzzy adaptive networking. Soft Comput. https://doi.org/10.1007/s00500-020-05068-3
66. Quaid MAK, Jalal A (2020) Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm. Multimed Tools Appl 79:6061–6083. https://doi.org/10.1007/s11042-019-08463-7
67. Chattopadhyay A, Menon V (2020) Fast simulation of Grover's quantum search on classical computer. ArXiv200504635 Quant-Ph
68. Biosample SAMEA7049302. From https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=ERR4329467. Access on june 2020.
69. Thomas, Sean & Martinez, L & Westenberger, Scott & Sturm, Nancy. (2007). A population study of the minicircles in Trypanosoma cruzi: Predicting guide RNAs in the absence of empirical RNA editing. BMC genomics. 8. 133. 10.1186/1471-2164-8-133.
70. Sound Spectrogram. https://commons.wikimedia.org/wiki/File:Spectrogram-19thC.png. Access on june 2020.
71. Stanford-Clark, A., & Truong, H. L. (2013). Mqtt for sensor networks (mqtt-sn) protocol specification. International business machines (IBM) Corporation version, 1(2).

5 CONCLUSÃO

O desafio do diagnóstico metagenômico ainda é grande, pesquisadores do mundo todo, de várias áreas das ciências, desde a engenharia da bioquímica molecular até os cálculos estatísticos resultantes, estão trabalhando em melhorias das técnicas. Atualmente é possível empregar o método metagenômico em ambientes sensíveis a patógenos, na área de energia e na medicina – espera-se que no futuro seja possível a utilização para medicina personalizada, diagnóstico e tratamento.

Este trabalho não tem a finalidade de ser uma solução definitiva para todos os problemas relacionados a precisão e computação de dados metagenômicos, mas espera colaborar como uma alternativa adicional na construção de ferramentas melhores e mais ágeis no diagnóstico de doenças através do sequenciamento de amostras de ambiente.

Os resultados apresentados no artigo direcionam a um resultado positivo no sentido de se identificar o agente infeccioso em um tempo semelhante ou menor que as ferramentas em uso atualmente. Foram respondidas as questões norteadoras: é possível encontrar o agente infeccioso utilizando o algoritmo proposto e o tempo utilizado é relativamente baixo, em comparação com ferramentas de referência, quando a massa de dados é adequada a proposta de pesquisa.

Diversas melhorias ainda podem ser implementadas, a compressão "*lossy*" pode ser ampliada e melhorada com o uso de técnicas mais específicas como a substituição de padrões e técnicas semelhantes aos compactadores "*lossless*" como o bzip2 (que se baseou no trabalho de Burrows-Wheeler), além de validações e parâmetros retroalimentados. O emprego de técnicas de inteligência artificial também é promissor e, mesmo que ainda existam desafios importantes a serem vencidos, as técnicas e os equipamentos estão evoluindo no sentido que se possa vencer as barreiras existentes.

Enfim, como qualquer trabalho sob a luz da inovação, os resultados podem transitar em ambos os sentidos do vetor do conhecimento, porém qualquer sentido encontrado já seria uma resposta de pesquisa.

Este autor acredita que o presente trabalho possa servir como um pequeno

passo de uma caminhada maior já existente, espera também que esta contribuição seja fonte de inspiração de trabalhos futuros de outros pesquisadores, nos quais pretende seguir colaborando com pesquisas subsequentes e correlatas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALTSCHUL, S. F., GISH, W., MILLER, W., MYERS, E. W., AND LIPMAN, D. J. (1990). Basic local alignment search tool. **J. Mol. Biol**. 215, 403–410. doi: 10.1016/S0022-2836(05)80360-2

BEER R, LACKNER P, PFAUSLER B, *et al*. Nosocomial ventriculitis and meningitis in neurocritical care patients. **J Neurol** 255: 1617-1624, 2008. doi: 10.1007/s00415-008-0059-8. 2008.

BIOSAMPLES. BioSamples < **EMBL-EBI**. Retrieved from https://www.ebi.ac.uk/biosamples/docs/faq. 2022.

BLAST Command Line Applications User Manual [Internet]. Bethesda (MD): **National Center for Biotechnology Information (US)**; 2008-. Introduction. 2008 Jun 23 [Updated 2021 Jan 7]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK279670/

BRANDON, M. C., WALLACE, D. C., & BALDI, P. Data structures and compression algorithms for genomic sequence data. **Bioinformatics**, 25(14), 1731–1738. doi: 10.1093/bioinformatics/btp319. 2009.

BROUWER MC, TUNKEL AR, VAN DE BEEK D. Epidemiology, diagnosis, and antimicrobial treatment of acute bacterial meningitis. **Clin Microbiol** Rev 23(3): 467–492, 2010.

BRYERS JD. Medical biofilms. **Biotechnol Bioeng** 100: 1-18. doi: 10.1002/bit.21838. 2008.

BURROWS M, WHEELER DJ A Block-sorting Lossless Data Compression Algorithm. **Digital Systems Research Center.** 1994.

BUSSCHER HJ, VAN DER MEI HC, Subbiahdoss G, Jutte PC, van den Dungen JJ, Zaat SA, Schultz MJ, Grainger DW. Biomaterial-associated infection: locating the finish line in the race for the surface. **Sci Transl Med** 4(153): 153rv10. doi: 10.1126/scitranslmed.3004528. 2012.

CALISTRI A, PALÙ G. **Unbiased Next-Generation Sequencing and New Pathogen Discovery : Undeniable Advantages and Still-Existing Drawbacks**, 60: 889–891, 2018.

CÁNOVAS, R., MOFFAT, A., & TURPIN, A. Lossy compression of quality scores in genomic data. **Bioinformatics**, 30(15), 2130–2136. doi: 10.1093/bioinformatics/btu183. 2014.

CHO, M., & NO, A. FCLQC: fast and concurrent lossless quality scores compressor. **BMC Bioinf**., 22(1), 1–14. doi: 10.1186/s12859-021-04516-7. 2021.

COGO, V., J. Paulo and A. Bessani, GenoDedup: Similarity-Based Deduplication and

Delta-Encoding for Genome Sequencing Data. **IEEE Transactions on Computers**, vol. 70, no. 5, pp. 669-681, 1 May 2021, doi: 10.1109/TC.2020.2994774. 2021.

CONEN A, FUX CA, VAJKOCZY P, TRAMPUZ A. Management of infections associated with neurosurgical implated devices. **Expert Rev Anti Infect Ther** 15: 241-255, 2017. doi: dx.doi.org/10.1080/14787210.2017.1267563. 2017.

COUNCIL, N. R. *et al*. The new science of metagenomics: Revealing the secrets of our microbial planet. [S. l.]: **National Academies Press**, 2007.

DAROUICHE RO. Treatment of infections associated with surgical implants. **N Engl J Med** 350: 1422-1429, 2004.

DEURENBERG RH, BATHOORN E, CHLEBOWICZ MA, *et al*. Application of next generation sequencing in clinical microbiology and infection prevention. **J Biotechnol** 243: 16-24, 2017.

DU, S., LI, J., & BIAN, N. A compression method for DNA. **PLoS One**, 15(11), e0238220. doi: 10.1371/journal.pone.0238220. 2020.

DUGUE, Rachelle et al. **Time to Confirmed Neuro-infection Diagnoses: Characterization of Current Trends and Implications for Diagnostic Testing Approaches** (S38. 006). 2022.

ERDEM H, INAN A, GUVEN E, HARGREAVES S, *et al*. The burden and epidemiology of community-acquired central nervous system infections: a multinational study. **Eur J Clin Microbiol Infect Dis**. Apr 10, 2017. doi: 10.1007/s10096-017-2973-0. 2017.

ESCOBAR-ZEPEDA, A.; VERA-PONCE DE LEÓN, A.; SANCHEZ-FLORES, A. The Road to Metagenomics: From Microbiology to DNA Sequencing Technologies and Bioinformatics. **Frontiers in Genetics**, [s. l.], v. 6, 2015. Disponível em: Acesso at: 1 Dec. 2021.

FASTQ FORMAT. [S. l.], [s. d.]. Disponível em: http://maq.sourceforge.net/FASTQ.shtml. Acesso at: 1 Dec. 2021.

GARRIDO-CARDENAS, J. A.; MANZANO-AGUGLIARO, F. The metagenomics worldwide research. **Current Genetics**, [s. l.], v. 63, n. 5, p. 819–829, 2017. Disponível em: Acesso at: 30 Nov. 2021.

GLASER CA, HONARMAND S, ANDERSON LJ, *et al*. Beyond viruses: clinical profiles and etiologies associated with encephalitis. **Clin Infect Dis** 43: 1565–1577. doi: doi.org/10.1086/509330. 2006.

GLASER, AN. High-Yield Biostatistics, Epidemiology, and Public Health. **LWW**, 2013.

GUTIÉRREZ-LUCAS, L. R. *et al*. Strategies for the Extraction, Purification and Amplification of Metagenomic DNA from Soil Growing Sugarcane. **Advances in Biological Chemistry**, [s. l.], v. 04, n. 04, p. 281–289, 2014.

HANDELSMAN, J. *et al*. Molecular biological access to the chemistry of unknown soil

microbes: a new frontier for natural products. **Chemistry & Biology**, [s. l.], v. 5, n. 10, p. R245–R249, 1998.

HANDELSMAN, J. **Metagenomics: Application of Genomics to Uncultured Microorganisms. Microbiology and Molecular Biology Reviews**, 69(1): 195–195, 2005.

HUNTER, S. *et al*. EBI metagenomics—a new resource for the analysis and archiving of metagenomic data. **Nucleic Acids Research**, [s. l.], v. 42, n. D1, p. D600–D606, 2013.

ILLUMINA. MISEQ SYSTEM. Disponível em: https://www.illumina.com/systems/sequencing-platforms/miseq.html. Acesso em 2021.

KEERTHY, A. S. **Lempel-Ziv-Welch Compression of DNA Sequence Data with Indexed Multiple Dictionaries**. 2017.

KHETSURIANI N, HOLMAN RC, ANDERSON LJ. Burden of encephalitis-associated hospitalizations in the United States, 1988-1997. **Clin Inf Dis** 35: 175-182. doi: doi.org/10.1086/341301. 2002.

KIM D, SONG L, BREITWIESER FP, AND SALZBERG SL. Centrifuge: rapid and sensitive classification of metagenomic sequences. **Genome Research.** 2016

KIYANI, Musa et al. The longitudinal health economic impact of viral encephalitis in the United States. **Journal of Medical Microbiology**, v. 69, n. 2, p. 270, 2020.

KOBUS R, HUNDT C, MÜLLER A, SCHMIDT B. Accelerating metagenomic read classification on CUDA-enabled GPUs. **BMC Bioinformatics** 18:11. https://doi.org/10.1186/s12859-016-1434-6. 2017.

KOKOT, M., GUDYŚ, A., Li, H., & DEOROWICZ, S. CoLoRd: Compressing long reads. **BioRxiv**, 2021.07.17.452767. 2021.

KUNIN, V. *et al*. A **Bioinformatician's Guide to Metagenomics. Microbiology and Molecular Biology Reviews**, [s. l.], v. 72, n. 4, p. 557–578, 2008

LEVENSHTEIN VI Binary Codes Capable of Correcting Deletions, Insertions and Reversals. **Sov Phys Dokl**. 1966.

LI, YL., WENG, JC., HSIAO, CC. *et al*. PEAT: an intelligent and efficient paired-end sequencing adapter trimming algorithm. **BMC Bioinformatics** 16, S2 (2015). https://doi.org/10.1186/1471-2105-16-S1-S2. 2015.

LIANG, Qiaoxing et al. DeepMicrobes: taxonomic classification for metagenomics with deep learning. **NAR Genomics and Bioinformatics**, v. 2, n. 1, p. lqaa009, 2020.

LIU Y, WIRAWAN A, SCHMIDT B. CUDASW++ 3.0: accelerating Smith-Waterman protein database search by coupling CPU and GPU SIMD instructions. **BMC Bioinformatics** 14:117. https://doi.org/10.1186/1471-2105-14-117. 2013.

MADDEN, Thomas. The BLAST sequence analysis tool. In: The NCBI Handbook [Internet]. 2nd edition. **National Center for Biotechnology Information** (US), 2013.

MAHRAM A, HERBORDT MC: FPGA-Accelerated ClustalW-Based Multiple Sequence Alignment through Pipelined Prefiltering. In: 2012 **IEEE 20th International Symposium on Field-Programmable Custom Computing Machines**. Pp 177–183. 2012.

MATHIEU, Alban et al. Machine Learning and Deep Learning Applications in Metagenomic Taxonomy and Functional Annotation. **Frontiers in Microbiology**, v. 13, 2022.

MCCLELLAND III S, HALL WA. Postoperative central nervous system infection: incidence and associated factors in 2111 neurosurgical procedures. **Clin Infect Dis** 45: 55-59, 2007.

MENZEL, P., NG, K. L., AND KROGH, A. (2016). Fast and sensitive taxonomic classification for metagenomics with Kaiju. **Nat.Commun.** 7:11257. doi: 10.1038/ncomms11257

MONTELONE, B. Mutation, Mutagens, and DNA Repair. BIOL400 Supplement. **Kansas State University**. 1998.

NEEDLEMAN SB, WUNSCH CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. **J Mol Biol** 48:443–453. https://doi.org/10.1016/0022-2836(70)90057-4. 1970.

NCBI. **National Center for Biotechnology Information**. Retrieved from https://www.ncbi.nlm.nih.gov. 2022.

OASIS. **MQTT Specification Version 5.0**. Retrieved June, v. 22, p. 2020. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html Acesso em 2022.

RAHMAN, N. The clinical impact of DNA sequence changes. **TGMI**, 27 Jan. 2017.

ROCHFORD ET, RICHARDS RG, MORIARTY TF. Influence of material on the development of device-associated infections. **Clin Microbiol Infect** 18(12): 1162-7. doi: 10.1111/j.1469-0691.2012.04002.x. 2021.

ROTBART HA. Viral meningitis. **Semin Neurol** 20: 277-292. doi: 10.1055/s-2000-9427. 2000.

SALZBERG SL, BREITWIESER FP, KUMAR A, *et al*. Next-generation sequencing in neuropathologic diagnosis of infections of the nervous system. **Neurol Neuroimmunol Neuroinflamm** 3: e251. doi: dx.doi.org/10.1212/NXI.0000000000000251. 2016.

SANGER F, NICKLEN S, COULSON AR. DNA sequencing with chain-terminating inhibitors. **Proceedings of the National Academy of Sciences**, 74(12): 5463–5467, 1977.

SCHLABERG, R. *et al*. Validation of Metagenomic Next-Generation Sequencing

Tests for Universal Pathogen Detection. **Archives of Pathology & Laboratory Medicine**, [s. l.], v. 141, n. 6, p. 776–786, 2017.

SCHMIDT B, HILDEBRANDT A. Next-generation sequencing: big data meets high performance computing. **Drug Discov Today** 22: 712-717, 2017.

SHI, Y., WANG, G., LAU, H. C.-H., & YU, J. Metagenomic Sequencing for Microbial DNA in Human Samples: Emerging Technological Advances. **Int. J. Mol. Sci.**, 35216302. Retrieved from https://pubmed.ncbi.nlm.nih.gov/35216302. 2022.

SMITH TF, WATERMAN MS, Identification of common molecular subsequences. **J Mol Biol** 147:195–197. https://doi.org/10.1016/0022-2836(81)90087-5. 1981.

TANENBAUM, Andrew S., STEEM, V. a. p. d. M. Distributed Systems: Principles and Paradigms: United States Edition. **Pearson**. 2001.

TONKOVIC, Petar et al. Literature on applied machine learning in metagenomic classification: a scoping review. **Biology**, v. 9, n. 12, p. 453, 2020.

TRETER J, MACEDO A.J. Catheters: a suitable surface for biofilm formation. In: MENDEZ-VILAS, A. (Ed.) **Science against Microbial Pathogens: Communicating Current Research and Technological Advances**. p. 835-842. 2011.

TSIOUTIS C, KARAGEORGOS SA, STRATAKOU S, *et al*. Clinical characteristics, microbiology and outcomes of external ventricular drainage-associated infections: the importance of active treatment. **J Clin Neuroscience** 42: 54-58, 2017. doi: doi.org/10.1016/j.jocn.2017.03.023. 2017.

VACEK G. Hybrid-Core Computing for High-Throughput Bioinformatics. **J Biomol Tech JBT** 22:S37–S38. 2011.

VAN DE BEEK D, CABELLOS C, DZUPOVA O, *et al*. ESCMID guideline: diagnosis and treatment of acute bacterial meningitis. **Clin Microbiol Infect** 22: S37-62, 2016. doi: dx.doi.org/10.1016/j.cmi.2016.01.007. 2016.

VERCE, Marko et al. A combined metagenomics and metatranscriptomics approach to unravel costa rican cocoa box fermentation processes reveals yet unreported microbial species and functionalities. **Frontiers in microbiology**, v. 12, p. 641185, 2021.

WOOD DE, SALZBERG SL. Kraken: ultrafast metagenomic sequence classification using exact alignments. **Genome Biol** 15: R46, 2014.

WRIGHT, B. L. C.; LAI, J. T. F.; SINCLAIR, A. J. Cerebrospinal fluid and lumbar puncture: a practical review. **Journal of Neurology**, [s. l.], v. 259, n. 8, p. 1530–1545, 2012. Disponível em: Acesso at: 1 Dec. 2021.

WU L, BRUNS-SMITH D, NOTHAFT FA, et al. FPGA Accelerated INDEL Realignment in the Cloud. In: 2019 **IEEE International Symposium on High Performance Computer Architecture (HPCA)**. Pp 277–290. 2019.

XIAO, W. *et al*. Challenges, Solutions, and Quality Metrics of Personal Genome

Assembly in Advancing Precision Medicine. **Pharmaceutics**, [s. l.], v. 8, n. 2, p. 15, 2016.

YANG, Aimin et al. Review on the application of machine learning algorithms in the sequence data mining of DNA. **Frontiers in Bioengineering and Biotechnology**, v. 8, p. 1032, 2020.

ZHAO, Y.; HU, T.; LIU, R.; HAO, Z.; LIANG, G.; Li, G. Biochemical Characterization and Function of Eight Microbial Type Terpene Synthases from Lycophyte Selaginella moellendorffii. **Int. J. Mol. Sci**. 2021, 22, 605. https://doi.org/10.3390/ijms22020605. 2021.

ANEXO I – COMITÊ DE ÉTICA EM PESQUISA

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA EDUCAÇÃO

UFCSPA
UNIVERSIDADE FEDERAL DE CIÊNCIAS DA SAÚDE DE PORTO ALEGRE
COMISSÃO DE PESQUISA

Atestado

Atestamos que o projeto de pesquisa intitulado *"Diagnóstico de infecções do sistema nervoso central: desenvolvimento de pipeline computacional para análise de dados de sequenciamento"* está registrado na Comissão de Pesquisa da Universidade Federal de Ciências da Saúde de Porto Alegre com o número 075/2019, sob responsabilidade de Claudia Elizabeth Thompson.

Salientamos que este registro **não autoriza o pesquisador a coletar ou analisar dados oriundos de sujeitos de pesquisa**.

Salientamos ainda que este registro **não garante a concessão de recursos financeiros por parte da UFCSPA a este projeto de pesquisa**.

Porto Alegre, 17 de fevereiro de 2020.

Henrique Silveira
Secretário da Comissão de Pesquisa
UFCSPA

Henrique M. Silveira
Assistente em Administração
SIAPE 1933135

ANEXO II – REGISTRO JUNTO AO INPI

INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL

29409191952868794

Pedido de Registro de Programa de Computador - RPC

Dados do Titular

Titular 1 de 1

Nome ou Razão Social:	GUSTAVO HENRIQUE CERVI
Tipo de Pessoa:	Pessoa Fisica
CPF/CNPJ:	78220033053
Nacionalidade:	Brasileira
Qualificação Física:	Professor do ensino superior
Endereço:	AV. PEDRO ADAMS FILHO, 3968, 2305
Cidade:	Novo Hamburgo
Estado:	RS
CEP:	93410118
País:	Brasil
Telefone:	
Fax:	
Email:	gustavohc@gmail.com

Dados do Programa

Data de Publicação:	01/01/2022
Data de Criação:	01/01/2022

- **§ 2º do art. 2º da Lei 9.609/98:** "Fica assegurada a tutela dos direitos relativos a programa de computador pelo prazo de cinquenta anos contados a partir de 1º de janeiro do ano subsequente ao da sua publicação ou, na ausência desta, da sua criação"

Título:	MetaGens
Algorítimo hash:	SHA-256 - Secure Hash Algorithm
Resumo digital hash:	2445775f52ad9843307b8ab728cd5ec433a60d3e73433eb5e54c6c31904c8898

§1º e Incisos VI e VII do §2º do Art. 2º da Instrução Normativa: O titular é o responsável único pela transformação, em resumo digital hash, dos trechos do programa de computador e demais dados considerados suficientes para identificação e caracterização, que serão motivo do registro. O titular

terá a inteira responsabilidade pela guarda da informação sigilosa definida no inciso III, § 1º, art. 3º da Lei 9.609 de 19 de fevereiro de 1998.

Linguagem: C#

Campo de Aplicação: SD07-MEDICINA (ALOPÁTICA, HEMEOPÁTICA, PREVENTIVA, TROPICAL, NUCLEAR, MEDICINA DO TRABALHO, LEGAL, DE URGÊNCIA)

SD08-ESPECIALIDADES MÉDICAS (CARDIOLOGIA, ENDOCRINOLOGIA, EPIDEMIOLOGIA, GINECOLOGIA, OFTALMOLOGIA, PSIQUIATRIA, PATOLOGIA, DERMATOLOGIA, RADIOLOGIA, ETC; MEDICINA NÃO-CONVENCIONAL: NATUROPÁTICA, CASEIRA, ACUPUNTURA, DO-IN, ETC)

Tipo de Programa: TC02 - PESQUISA OPERACIONAL

Dados do Autor

Autor 1 de 1

Nome: GUSTAVO HENRIQUE CERVI

CPF: 78220033053

Nacionalidade: Brasileira

Qualificação Física: Professor do ensino superior

Endereço: Avenida Pedro Adams Filho, 3968, 2305

Cidade: Novo Hamburgo

Estado: RS

CEP: 93410-118

País: BRASIL

Telefone: (51) 991 227232

Fax:

Email: gustavohc@gmail.com

Declaração de Veracidade - DV

Nome:declaracaoVeracidadeAssinado.pdf

ANEXO III – INSTRUÇÕES – PERIÓDICO DATABASE

Instructions to Authors

Manuscripts must be submitted online. Once you have prepared your manuscript according to the instructions below please visit the online submission web site. Instructions on how to submit your manuscript online can be found by following this link: Instructions on how to submit your manuscript online.

Aims

The journal will publish:

Detailed descriptions of databases, and database tools, in the broad arena of biology - authors are strongly encouraged to include a biological discovery or a 'testable' hypothesis in their papers.

Shorter papers describing significant updates to established databases.

Objective reviews of complementary and ancillary databases and database tools.

User tutorials for database tools.

Methodology and technical notes on database development.

Improvements to automated prediction and annotation for biomedical datasets.

Detailed descriptions of the state and updating of the annotation of genomes.

Descriptions of the development and content of ontologies of relevance to the biomedical community and the tools specific to the use of these ontologies.

Articles relevant to the annotation process such as standards for curation, annotation best practices, annotation methodologies, the use of automated and semi-automated methods for annotation and the measures for annotation consistency.

Perspective manuscripts describing novel approaches, technologies, standards, or methods for database and/or annotation explorations including comparisons of state-of-the-art approaches. Perspectives can also include opinion pieces addressing challenges or debates in the field of biological databases and

biocuration.

Brief invited conference reports on topics related to the scope of the journal.

Descriptions of databases and biocuration application and activities in all areas of biology are welcome, including biological chemistry, genomics, proteomics, glycomics, molecular biology, biomedicine, physiology, ecology, botany, zoology, and taxonomy.

Open Access Licence

DATABASE is a fully open access journal, and all articles are published in the journal under an open access licence immediately upon publication. You will need to pay an open access charge to publish under an open access licence.

If the corresponding author is based in one of the countries included in our Developing Countries Initiative, your article will be eligible for a full waiver of the open access charge.

OUP has a growing number of Read and Publish agreements with institutions and consortia which provide funding for open access publishing. This means authors from participating institutions can publish open access, and the institution may pay the charge. Find out if your institution is participating.

Please note that you may be eligible for a discount to the open access charge based on society membership. Authors may be asked to prove eligibility for the member discount.

Authorship

The authorship of the paper should be confined to those who have made a significant contribution to the design and execution of the work described.

'Umbrella' groups and authorship:

Many large collaborative studies (frequently resistance surveys) are organized under

a group name which represents all the participants. All articles must have at least one named individual as author. Authors who wish to acknowledge the umbrella group from which the data originate should first list the author(s) of the article and follow this with 'on behalf of the GROUP NAME'. If necessary the names of the participants may be listed in the Acknowledgements section.

Peer Review Process

Manuscripts are initially considered by the Editor-in-Chief, sometimes with advice of the Associate Editors and DATABASE Editorial Board members. This is usually completed within a week. Manuscripts that are not triaged for rejection are sent out for peer review to at least 2-3 independent peer reviewers. Peer reviewers remain anonymous to the authors at all times, unless a peer reviewer suggests that their name is included in the review. The choice of peer reviewers is made by the Editor-in-Chief or an Associate Editor with strong consideration to suggestions and conflicts provided by the author(s) during the submission process.

Previous publication

Submission of a manuscript to DATABASE implies that it reports unpublished work, that it is not under consideration for publication elsewhere and that, if accepted, it will not be published elsewhere in the same form, either in English or in any other language, without the consent of the publisher. Authors should provide the references of similar work that they have already published, or which is currently under consideration by another journal. If the work has previously been presented at a conference, authors should provide details in the covering letter. The journal will consider publication of work that has previously been presented as either a short abstract or poster at a conference, but not as a full paper. If previously published tables, illustrations or one or more blocks of more than 200 words of text are to be included, then the copyright holder's written permission must be obtained. Include copies of any such permission letters with your paper.

Information on the journal’s preprint and self-archiving policy.

Plagiarism

Manuscripts submitted may be screened with iThenticate anti-plagiarism software in an attempt to detect and prevent plagiarism. Any manuscript may be screened, especially if there is reason to suppose part or all of the text has been previously published. Prior to final acceptance any manuscript that has not already been screened may be put through iThenticate. Please see more information about iThenticate.

Conflict of interest

At the point of submission, Database policy requires that each author reveal any financial interests or connections, direct or indirect, or other situations that might raise the question of bias in the work reported or the conclusions, implications, or opinions stated - including pertinent commercial or other sources of funding for the individual author(s) or for the associated department(s) or organization(s), personal relationships, or direct academic competition. When considering whether you should declare a conflicting interest or connection please consider the conflict of interest test: Is there any arrangement that would embarrass you or any of your co-authors if it was to emerge after publication and you had not declared it?

As an integral part of the online submission process, Corresponding authors are required to confirm whether they or their co-authors have any conflicts of interest to declare, and to provide details of these. If the Corresponding author is unable to confirm this information on behalf of all co-authors, the authors in question will then be required to submit a completed conflict of interest form to the Editorial Office. It is the Corresponding author’s responsibility to ensure that all authors adhere to this policy.

If the manuscript is published, Conflict of Interest information will be communicated in a statement in the published pap

Availability of databases

In line with the expectations and standards of the community, authors are expected to ensure their databases and/or online resources remain available for at least 2

years following publication of their paper in DATABASE.

Databases must be freely available to all via the web, not require any login or registration and not be password-protected.

Deposition of sequence and structural data

Sequence information, co-ordinates used to create molecular models described in a manuscript, and structural data must be submitted in electronic form, prior to acceptance, to the appropriate database for release no later than the date of publication of the corresponding article in DATABASE. Deposition numbers and/or accession numbers provided by the database should be included in the manuscript and entered into the relevant boxes during online submission or communicated to the Editor handling the manuscript as soon as received. In cases where there may be no appropriate database, authors must make their data available on request. Atomic co-ordinates may be included in the publication as supplementary material. Manuscripts will not be published until DATABASE is in receipt of the deposition number.

For papers reporting novel nucleic acid sequences

Nucleic acid sequence information must be deposited with one of the three major collaborative databases (EMBL/GenBank/DDBJ). For sequences obtained from a public or private web site, it is the author's responsibility to ensure that any sequence used within the manuscript is deposited before publication. It is necessary to submit sequences to one database only since data are exchanged between EMBL, GenBank and DDBJ on a daily basis. New sequence names and their accession numbers should be listed at the beginning of the Methods section to aid searches by readers. In order to allow new methods of data search, DATABASE encourages authors to cite GenBank accession numbers when referring to established sequences within their manuscript.

For papers reporting novel three-dimensional structures

Atomic co-ordinates and the related experimental data (structure factor amplitudes/intensities and/or NMR restraints) must be deposited with a database.

Authors must agree to release the atomic coordinates and experimental data when the associated article is published.
The Cambridge Crystallographic Data Centre (CCDC) is appropriate for deposition of data on nucleosides, nucleotides and other small molecules.
A member site of the Worldwide Protein Data Bank: RCSB PDB, MSD-EBI), PDBj , or BMRB is appropriate for deposition of data on proteins determined by X-ray crystallography and for all macromolecules determined by NMR methods.
The Nucleic Acid Database (NDB) is appropriate for atomic co-ordinate and structure factor data for crystal structures of nucleic acids.

For papers reporting novel protein sequences
Protein sequences, which have been determined by direct sequencing of the protein, must be submitted to UniProt (i.e. TrEMBL, Swiss-Prot and PIR) using the interactive submission tool SPIN. Please note that they do not provide accession numbers, IN ADVANCE, for protein sequences that are the result of translation of nucleic acid sequences. These translations will forwarded automatically from the nucleotide sequence databases (EMBL/GebBank/DDBJ) and assigned UniProt accession numbers on incorporation into UniProt. Results from characterization experiments should also be submitted to UniProt: for novel sequences, these should be included with the sequence submission. Existing UniProt entries should also be updated. This can include information such as function, subcellular location, subunit, etc.

For papers reporting new ChIP-Seq data
New ChIP-Seq data must be deposited in GEO, with accession numbers at or before acceptance for publication.

Microarray data
All authors must comply with the 'Minimal Information About a Microarray Experiment' (MIAME) guidelines published by the Microarray Gene Expression Data Society, which can be found at here. NAR also requires submission of microarray data to the GEO or ArrayExpress) databases, with accession numbers at or before acceptance for publication.

Quantitative PCR

Authors are encouraged to follow the 'Minimal Information for Publication of Quantitative Real-Time PCR Experiments' (MIQE) guidelines, if appropriate. The guidelines are published by the Real-Time PCR Data Markup Language Consortium.

Material Disclaimer

The opinions expressed in Database are those of the authors and contributors, and do not necessarily reflect those of the editors, the editorial board, Oxford University Press or the organization to which the authors are affiliated.

General

Papers must be clearly written in English. Papers should be submitted in Word, although we do allow Latex files if necessary.

Authors should avoid the use of language or slang which is not in keeping with the academic and professional style of the Journal. Authors should not also seek to use the Journal as a vehicle for marketing any specific product or service.

Authors should follow the conventions of the CSE Style Manual (Council of Science Editors, Reston, VA, 2006). Chemical Abstracts and its indices should be followed for chemical names. For biochemical terminology the recommendations issued by the IUPAC-IUB Commission on Biochemical Nomenclature, as given in Biochemical Nomenclature and Related Documents, published in 1992 by the Biochemical Society, UK should be followed. For enzymes, the recommended name assigned by the IUPAC-IUB Committee on Biochemical Nomenclature, 1978, as given in Enzyme Nomenclature, published by Academic Press, New York, 1992 should be used. Wherever possible, the recommended SI units should be used. Genotypes should be italicised. Phenotypes should not be italicised. For bacterial agents nomenclature Demerec.

Section headings within the manuscript (e.g. Methods, Discussion, Future directions)

are at the authors' discretion.

Please list the Database URL clearly on the line beneath the abstract, in the format: 'Database URL: .....'
References

These should be cited in the text by sequential number only, in order of appearance, and listed numerically in the References section. Online references should be cited as in example 5, below. Please see examples 6 and 7 for papers that have been published online in more than one version. The initial version of a paper published in this way can be cited by the Digital Object Identifier (doi) but, if available, the reference should also include the citation of the final version. Authors should check all references carefully, and in particular ensure that all references in the Reference section are cited in the text. Note that multiple references or page spans under one number are not allowed. Personal communications, unpublished results, manuscripts submitted or in preparation, statistical packages, computer programs and web sites should be cited in the text only, NOT included in the References section.

All references must be cited in the text and should be denoted using numbers in parentheses before the punctuation., e.g. (1, 3–5). At the start of a sentence the authors can be named, e.g. Shadforth *et al*. (15)...

Style in the References section should be as follows. Journal names should be abbreviated in the style of Chemical Abstracts. NOTE THAT FULL TITLES OF JOURNAL ARTICLES MUST BE PROVIDED.

1. Schmitt,E., Panvert,M., Blanquet,S. and Mechulam,Y. (1995) Transition state stabilisation by the 'high' motif of class I aminoacyl-tRNA synthetases: the case of Escherichia coli methionyl-tRNA synthetase. Nucleic Acids Res., 23, 4793-4798.

2. Huynh,T.V., Young,R.A. and Davies,R.W. (1988) Constructing and screening cDNA libraries in lambdagt10 and lambdagt11. In Glover,D.M. (ed.), DNA Cloning - A

Practical Approach. IRL Press, Oxford, Vol. I, pp. 49-78.

3. Maniatis,T., Fritsch,E.F. and Sambrook,J. (1982) Molecular Cloning: A Laboratory Manual. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

4. Burnett,R.C. (1993) EMBL accession no. X52486.

5. Capaldi,S., Getts,R.C. and Jayasena,S.D. (2000) Signal amplification through nucleotide extension and excision on a dendritic DNA platform. Nucleic Acids Res., 28, e21.

6. Qiao,D., Chen,W., Stratagoules,E. and Martinez,J. (March 10, 2000) Bile acid-induced activation of activator protein-1 requires both extracellular signal-regulated kinase and protein kinase C signaling. J. Biol. Chem., 10.1074/jbc.M908890199

7. Qiao,D., Chen,W., Stratagoules,E. and Martinez,J. (2000) Bile acid-induced activation of activator protein-1 requires both extracellular signal-regulated kinase and protein kinase C signaling. J. Biol. Chem., 275, 15090-15098. First published on May 19, 2000, 10.1074/jbc.M908890199

8. Bernhagen,J., Elkine,B., Geiger,G., Tovar,G. and Vitzthum,F. (1999) Patent DE-198198889.2-44; PCT/WO/EP/99/03047.

If there are four or more authors, then use the first three followed by *et al*.
Papers in preparation or submitted for publication should not be in the reference list.

Authors are asked to ensure the references to named people and/or organisations are accurate and without libellous implications.
Crossref Funding Data Registry

In order to meet your funding requirements authors are required to name their funding sources, or state if there are none, during the submission process. Further

information on this process and the CHORUS initiative.

Details of all funding sources for the work in question should be given in a separate section entitled 'Funding'. This should appear before the 'Acknowledgements' section.

The following rules should be followed:

The sentence should begin: ‘This work was supported by …’

The full official funding agency name should be given, i.e. ‘the National Cancer Institute at the National Institutes of Health’ or simply 'National Institutes of Health' not ‘NCI' (one of the 27 subinstitutions) or 'NCI at NIH’ (full RIN-approved list of UK funding agencies)

Grant numbers should be complete and accurate and provided in brackets as follows: ‘[grant number ABX CDXXXXXX]’

Multiple grant numbers should be separated by a comma as follows: ‘[grant numbers ABX CDXXXXXX, EFX GHXXXXXX]’

Agencies should be separated by a semi-colon (plus ‘and’ before the last funding agency)

Where individuals need to be specified for certain sources of funding the following text should be added after the relevant agency or grant number 'to [author initials]'

An example is given here: ‘This work was supported by the National Institutes of Health [P50 CA098252 and CA118790 to R.B.S.R.]
and the Alcohol & Education Research Council [HFY GR667789].

Oxford Journals will deposit all NIH-funded articles in PubMed? Central. See our

Author's Resources page for details. Authors must ensure that manuscripts are clearly indicated as NIH-funded using the guidelines above.

Figures

You are required to submit high-resolution images, preferably with your initial submission but no later than revision stage. Electronic images (figures and schemes) must be at a minimum resolution of 600 d.p.i. for line drawings (black and white) and 300 d.p.i. for colour or greyscale. Colour figures must be supplied in CMYK not RGB colours. Please ensure that the prepared electronic image files print at a legible size (with lettering of at least 2 mm).

A number of different file formats are acceptable, including: PowerPoint (.ppt), Tagged Image File Format (.tif), Encapsulated PostScript (.eps), Joint Photographic Experts Group (.jpg), Graphics Interchange Format (.gif), Adobe Illustrator (.ai) (please save your files in Illustrator's EPS format), Portable Network Graphics (.png), Microsoft Word (.doc), Rich Text Format (.rtf), and Excel (.xls) but not Portable Document Format (PDF).

Please ensure that the figure is clearly labelled with its figure number.

Third-Party Content in Open Access papers

If you will be publishing your paper under an Open Access licence but it contains material for which you do not have Open Access re-use permissions, please state this clearly by supplying the following credit line alongside the material:

Title of content

Author, Original publication, year of original publication, by permission of [rights holder]

This image/content is not covered by the terms of the Creative Commons licence of this publication. For permission to reuse, please contact the rights holder.

Internet screen dumps

Internet screen dumps should be provided electronically as BITMAP, with a minimum acceptable resolution of 300 dpi. Their approximate final positions should be indicated in the margin of the text. Authors should be aware that graphics supplied with low resolution are not guaranteed to reproduce well and should be avoided whenever possible.

Tables

Tables should be submitted in electronic form, preferably in MS Word or Excel. Tables should be referred to in the text and numbered consecutively. They should be supplied separately from the main body of the text, with their approximate final positions indicated in the text. Each column should have a short heading and, where appropriate, the units should be stated. Table legends should describe the content and should be understood independently from the text. Data columns should be right-hand aligned, or aligned by decimal place, where appropriate; data should be sorted where possible. Footnotes should be included on the same pages as the tables themselves and should be used to explain any abbreviations used in the table and denote them by letter. Footnotes should also be used to quote sources.

Proofs

All manuscripts will undergo some editorial modification, so it is important to check proofs carefully. PDF page proofs will be sent via e-mail to the corresponding author for checking. To avoid delays in publication, proofs should be checked and returned within 48 hours. Corrections should be returned by annotated PDF, e-mail or fax. Extensive changes to the text may be charged to the author.

Preprint policy

Authors retain the right to make an Author’s Original Version (preprint) available through various channels, and this does not prevent submission to the journal. For further information see our Online Licensing, Copyright and Permissions policies. If accepted, the authors are required to update the status of any preprint, including your published paper’s DOI, as described on our Author Self-Archiving policy page.

ORCID

DATABASE requires submitting authors to provide an ORCID iD at submission to the journal. More information on ORCID and the benefits of using an ORCID iD is available. If you do not already have an ORCID iD, you can register for free via the ORCID website.

ANEXO IV – EXPERIMENTOS E DADOS

Listagem de bases de dados de referência

ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/assembly_summary_refseq.txt

Extração de SRR

```
fastq-dump.exe --split-files .\SRR12665147
```

Exemplo de classificação utilizada no NCBI

```
Unidentified reads: 3.63%
Identified reads: 96.37%
    cellular organisms: 96.17%
        Bacteria: 54.36%
            Proteobacteria: 52.43%
                Betaproteobacteria: 39.41%
                    Neisseriales: 39.38%
                        Neisseriaceae: 39.38%
                            Neisseria: 31.47%
                            Bergeriella: < 0.01% (2 Kbp)
                    Burkholderiales: < 0.01% (7 Kbp)
                Gammaproteobacteria: < 0.01% (5 Kbp)
            Terrabacteria group: 0.01%
        Eukaryota: 40.94%
            Homininae: 36.71%
                Homo sapiens: 22.33%
    Viruses: 0.19%

Strong signals
SuperKingdom      Organism          Rank        %%    Kbp         Coverage
Bacteria Neisseria meningitidis     species     54.7  133,301     61.8
Bacteria Neisseria gonorrhoeae      species     0.2   528         0.2
Bacteria Neisseria lactamica        species     0.2   494         0.2
Viruses  Torque teno virus          species     0.1   208
Viruses  Anelloviridae sp.          species     0.1   184
Viruses  SEN virus                  species     0.0   27
Viruses  Torque teno midi virus 9   species     0.0   16
Viruses  Small anellovirus          KR-BD-0191  0.0   14
Viruses  Small anellovirus          KR-BD-0282  0.0   9
```

EXPERIMENTOS

URL para baixar os arquivos SRR

```
https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR12665147
```

URLs para baixar os arquivos fasta

```
https://www.ncbi.nlm.nih.gov/genome/
https://www.ncbi.nlm.nih.gov/genome/172?genome_assembly_id=678164
```

Sed para converter o arquivo fasta ou fastq pra fa

```
sed -n '1~4s/^@/>/p;2~4p' INFILE.fasta > OUTFILE.fa
```

CLI Blast com report para ver os matches

```
.\blastn.exe -subject GCF_008330805.1_ASM833080v1_genomic.fna -query .\
SRR12665147.fa -out res.txt
```

CLI Blast sem report para testar o tempo de cpu sem io

```
.\blastn.exe -subject GCF_008330805.1_ASM833080v1_genomic.fna -query .\
SRR12665147.fa > $null
```

Comandos para medição do tempo de execução

```
Measure-Command { .\blastn.exe -subject
GCF_008330805.1_ASM833080v1_genomic.fna -query .\SRR12665147.fa > $null }
```

Exemplo de medição BLAST

```
Measure-Command { .\blastn.exe -subject
GCF_008330805.1_ASM833080v1_genomic.fna -query .\SRR12665147.fa > $null }
Days              : 0
Hours             : 0
Minutes           : 10
Seconds           : 9
Milliseconds      : 642
Ticks             : 6096423671
TotalDays         : 0,00705604591550926
TotalHours        : 0,169345101972222
TotalMinutes      : 10,1607061183333
TotalSeconds      : 609,6423671
TotalMilliseconds : 609642,3671
```

Comandos para medição do tempo de execução

```
Measure-Command { .\blastn.exe -subject
GCF_008330805.1_ASM833080v1_genomic.fna -query .\SRR12665147.fa > $null }
```

Exemplo de experimento e medição CENTRIFUGE

```
git clone https://github.com/infphilo/centrifuge
cd centrifuge/
make
make install prefix=/usr/local
wget
https://genome-idx.s3.amazonaws.com/centrifuge/p_compressed_2018_4_15.tar.g
z
tar -xzvf p_compressed_2018_4_15.tar.gz
date; centrifuge -x p_compressed -U SRR12665147.fastq --report-file evol1-
report.txt -S evol1-results.txt; date
centrifuge-kreport -x p_compressed evol1-results.txt > evol1-kreport.txt
```

Exemplo de report do CENTRIFUGE

```
 42.65 225438 225438 U 0      unclassified
 57.35 303183 0      - 1      root
 57.35 303183 334    - 131567   cellular organisms
 57.23 302535 594    D 2          Bacteria
 56.25 297328 368    P 1224         Proteobacteria
 53.18 281120 45     C 28216          Betaproteobacteria
 53.08 280594 8      O 206351           Neisseriales
 53.07 280554 122    F 481                Neisseriaceae
 53.04 280405 29199  G 482                  Neisseria
 46.43 245417 245417 S 487                    Neisseria meningitidis
  0.73 3844   3844   S 486                    Neisseria lactamica
  0.31 1663   1663   S 485                    Neisseria gonorrhoeae
  0.02 95     95     S 490                    Neisseria sicca
  0.02 81     81     S 488                    Neisseria mucosa
  0.01 67     67     S 495                    Neisseria elongata
```

Exemplo de report do MetaGens

```
Neisseria Meningitidis
Total reads: 227.638
Total reads len: 5.133.738
Total refseqs: 1
Total refseqs len: 346.911
=== DONE ===
Elapsed: 00:00:29.4572728

Neisseria Sicca
Total reads: 176.101
Total reads len: 3.745.223
Total refseqs: 1
Total refseqs len: 390.380
=== DONE ===
Elapsed: 00:00:27.7421608
```

Exemplo de experimento e medição KRAKEN

```
wget https://github.com/DerrickWood/kraken2/archive/refs/tags/v2.1.2.tar.gz
tar -xzvf v2.1.2.tar.gz
cd kraken2-2.1.2/
./install_kraken2.sh ../kraken2
kraken2-build --standard --db defdb
date; ./kraken2 --db fdb --report k2_report.txt --output k2_output.txt
SRR12665147.fastq; date
```

Exemplo de report do KRAKEN

```
 96.71  1022463 14873   R      1      root
```

```
 95.00  1004328 166     R1      131567    cellular organisms
 53.07  561029  4227    D       2           Bacteria
 52.65  556596  1085    P       1224          Proteobacteria
 52.47  554737  54      C       28216           Betaproteobacteria
 52.46  554602  70      O       206351            Neisseriales
 52.45  554527  1619    F       481                 Neisseriaceae
 52.29  552849  71337   G       482                   Neisseria
 43.37  458504  440991  S       487                     Neisseria meningitidis
  0.71  7461    88      S1      135720                    Neisseria meningitidis serogroup C
  0.44  4662    4662    S2      374833                      Neisseria meningitidis 053442
  0.24  2546    2546    S2      604162                      Neisseria meningitidis 8013
  0.02  165     165     S2      272831                      Neisseria meningitidis FAM18
  0.30  3136    3136    S1      662598                    Neisseria meningitidis alpha14
  0.27  2856    2856    S1      935588                    Neisseria meningitidis M01-240355
  0.09  904     904     S1      935590                    Neisseria meningitidis M0579
  0.08  863     0       S1      491                       Neisseria meningitidis serogroup B
  0.06  604     604     S2      630588                      Neisseria meningitidis alpha710
  0.02  241     241     S2      909420                      Neisseria meningitidis H44/76
  0.00  18      18      S2      122586                      Neisseria meningitidis MC58
  0.07  776     0       S1      65699                     Neisseria meningitidis serogroup A
  0.07  776     776     S2      122587                      Neisseria meningitidis Z2491
  0.06  586     586     S1      1386087                   Neisseria meningitidis LNP21362
  0.03  325     325     S1      935599                    Neisseria meningitidis G2136
  0.03  268     268     S1      935591                    Neisseria meningitidis M01-240149
  0.02  185     185     S1      942513                    Neisseria meningitidis WUE 2594
  0.01  107     107     S1      935593                    Neisseria meningitidis M04-240196
  0.00  23      23      S1      935589                    Neisseria meningitidis NZ-05/33
  0.00  23      23      S1      1095685                   Neisseria meningitidis M7124
  0.96  10169   7792    S       486                     Neisseria lactamica
  0.22  2377    2377    S1      489653                    Neisseria lactamica 020-06
  0.42  4452    4344    S       485                     Neisseria gonorrhoeae
  0.01  71      71      S1      1247414                   Neisseria gonorrhoeae NG-k51.05
  0.00  20      20      S1      242231                    Neisseria gonorrhoeae FA 1090
  0.00  15      15      S1      528354                    Neisseria gonorrhoeae MS11
  0.00  2       2       S1      521006                    Neisseria gonorrhoeae NCCP11945
  0.26  2737    2737    S       483                     Neisseria cinerea
  0.13  1340    1340    S       484                     Neisseria flavescens
  0.10  1035    503     S       490                     Neisseria sicca
  0.05  532     532     S1      547045                    Neisseria sicca ATCC 29256
  0.07  789     789     S       28449                   Neisseria subflava
```

Exemplo de experimento e medição KAIJU

```
git clone https://github.com/bioinformatics-centre/kaiju.git
cd kaiju/src
make
mkdir ref
cd ref
wget --no-check-certificate
https://kaiju.binf.ku.dk/database/kaiju_db_refseq_2022-03-23.tgz
tar -xzvf kaiju_db_refseq_2022-03-23.tgz
date; ../kaiju -t nodes.dmp -f kaiju_db_refseq.fmi -i SRR12665147.fastq -o
kaiju.out; date
../kaiju2table -t nodes.dmp -n names.dmp -r species -o kaiju_summary.tsv
kaiju.out
```

Exemplo de report do KAIJU

```
file            percent         reads   taxon_id  taxon_name
kaiju.out       21.381481       226054  487       Neisseria meningitidis
kaiju.out       0.599011        6333    486       Neisseria lactamica
kaiju.out       0.472739        4998    485       Neisseria gonorrhoeae
kaiju.out       0.246017        2601    489       Neisseria polysaccharea
kaiju.out       0.142635        1508    483       Neisseria cinerea
kaiju.out       0.141973        1501    1491      Clostridium botulinum
```

```
kaiju.out       0.055995        592     488       Neisseria mucosa
kaiju.out       0.046347        490     490       Neisseria sicca
kaiju.out       0.041712        441     28449     Neisseria subflava
kaiju.out       0.025916        274     484       Neisseria flavescens
kaiju.out       0.020431        216     727       Haemophilus influenzae
kaiju.out       0.014188        150     504       Kingella kingae
```

Lista de accessions utilizada nos experimentos

SRR14596733	SRR14596772	SRR14596811	SRR12826677
SRR14596734	SRR14596773	SRR14596812	SRR12826678
SRR14596735	SRR14596774	SRR14596813	SRR12826679
SRR14596736	SRR14596775	SRR14596814	SRR12826680
SRR14596737	SRR14596776	SRR14596815	SRR12826681
SRR14596738	SRR14596777	SRR14596816	SRR12826682
SRR14596739	SRR14596778	SRR14596817	SRR12826683
SRR14596740	SRR14596779	SRR14596818	SRR12826684
SRR14596741	SRR14596780	SRR14596819	SRR12826685
SRR14596742	SRR14596781	SRR14596820	SRR12826686
SRR14596743	SRR14596782	SRR14596821	SRR12826687
SRR14596744	SRR14596783	SRR14596822	SRR12826688
SRR14596745	SRR14596784	SRR14596823	SRR12826689
SRR14596746	SRR14596785	SRR14596824	SRR12826690
SRR14596747	SRR14596786	SRR14596825	SRR12826691
SRR14596748	SRR14596787	SRR14596826	SRR12826692
SRR14596749	SRR14596788	SRR14596827	SRR12826693
SRR14596750	SRR14596789	SRR14596828	SRR12826694
SRR14596751	SRR14596790	SRR14596829	SRR12826695
SRR14596752	SRR14596791	SRR14596830	SRR12826696
SRR14596753	SRR14596792	SRR14596831	SRR12826697
SRR14596754	SRR14596793	SRR14596832	SRR12826702
SRR14596755	SRR14596794	ERR5087973	SRR12826710
SRR14596756	SRR14596795	SRR12826655	SRR12826711
SRR14596757	SRR14596796	SRR12826662	SRR12826712
SRR14596758	SRR14596797	SRR12826663	SRR12826713
SRR14596759	SRR14596798	SRR12826664	SRR12826714
SRR14596760	SRR14596799	SRR12826665	SRR12826715
SRR14596761	SRR14596800	SRR12826666	SRR12826716
SRR14596762	SRR14596801	SRR12826667	SRR12826717
SRR14596763	SRR14596802	SRR12826668	SRR12826718
SRR14596764	SRR14596803	SRR12826669	SRR12826719
SRR14596765	SRR14596804	SRR12826670	SRR12826720
SRR14596766	SRR14596805	SRR12826671	SRR12826721
SRR14596767	SRR14596806	SRR12826672	SRR12826722
SRR14596768	SRR14596807	SRR12826673	SRR12826723
SRR14596769	SRR14596808	SRR12826674	SRR12826724
SRR14596770	SRR14596809	SRR12826675	SRR12826725
SRR14596771	SRR14596810	SRR12826676	SRR12826726

SRR12826727
SRR12826730
SRR12826731
SRR12826732
SRR12826733
SRR12826734
SRR12826735
SRR12826736
SRR12826737
SRR12826738
SRR12826739
SRR12826742
SRR12826743
SRR12826744
SRR12826745
SRR12826746
SRR12826747
SRR12826748
SRR12826749
SRR12826750
SRR12826751
SRR12826752
SRR12826753
SRR12826754
SRR12826755
SRR12826756
SRR12826757
SRR12826758
SRR12826759
SRR12826760
SRR12826761
SRR12826762
SRR12826763
SRR12826764
SRR12826765
SRR12826766
SRR12826767
SRR12826768
SRR12826769
SRR12826770
SRR13266646
SRR13266659
SRR13266661
SRR14596833
SRR12826771
SRR12826772
SRR12826773
SRR12826774
SRR12826775
SRR12826776
SRR12826777
SRR12826778
SRR12826779
SRR12826780
SRR12826782
SRR12826783
SRR12826784
SRR12826785
SRR12826786
SRR12826787
SRR12826788
SRR12826789
SRR12826790
SRR12826791
SRR12826792
SRR12826793
SRR12826794
SRR12826795
SRR12826796
SRR12826797
SRR12826798
SRR12826799
SRR12826800
SRR12826801
SRR12826802
SRR12826803
SRR12826804
SRR12826805
SRR12826806
SRR12826807
SRR12826808
SRR12826809
SRR12826810
SRR12826811
SRR12826812
SRR12826813
SRR12826814
SRR12826815
SRR12826816
SRR12826817
SRR12826818
SRR12826819
SRR12826820
SRR12826821
SRR12826822
SRR12826823
SRR12826824
SRR12826825
SRR12826826
SRR12826827
SRR12826828
SRR12826829
SRR12826830
SRR12826831
SRR12826832
SRR12826833
SRR12826834
SRR12826835
SRR12826836
SRR12826837
SRR12826838
SRR12826839
SRR12826840
SRR12826841
SRR12826842
SRR12826843
SRR12826844
SRR12826845
SRR12826846
SRR12826847
SRR12826848
SRR12826849
SRR12826850
SRR12826851
SRR12826852
SRR12826853
SRR12826854
SRR12826855
SRR12826856
SRR12826857
SRR12826858
SRR12826859
SRR12826860
SRR12826861
SRR12826862
SRR12826863
SRR12826864
SRR12826865
SRR12826866
SRR12826867
SRR12826871
SRR12826872
SRR12826877
SRR12826878
SRR12826395
SRR12826396
SRR12826397
SRR12826398
SRR12826399
SRR12826400
SRR12826401
SRR12826402
SRR12826403
SRR12826404
SRR12826405
SRR12826406
SRR12826407
SRR12826412
SRR12826423
SRR12826424
SRR12826431
SRR12826432
SRR12826433
SRR12826434
SRR12826435
SRR12826436
SRR12826437
SRR12826438
SRR12826439
SRR12826440
SRR12826441
SRR12826442
SRR12826443
SRR12826444
SRR12826445
SRR12826446
SRR12826447
SRR12826451
SRR12826452
SRR12826453
SRR12826454
SRR12826455
SRR12826456
SRR12826457
SRR12826458
SRR12826459
SRR12826460
SRR12826461
SRR12826462
SRR12826463
SRR12826464
SRR12826465
SRR12826466
SRR12826467
SRR12826468
SRR12826469

SRR12826470
SRR12826471
SRR12826472
SRR12826473
SRR12826474
SRR12826475
SRR12826476
SRR12826477
SRR12826478
SRR12826479
SRR12826480
SRR12826481
SRR12826484
SRR12826485
SRR12826486
SRR12826487
SRR12826488
SRR12826489
SRR12826491
SRR12826492
SRR12826493
SRR12826494
SRR12826495
SRR12826879
SRR12826880
SRR12826881
SRR12826882
SRR12826883
SRR12826884
SRR12826885
SRR12826886
SRR12826887
SRR12826888
SRR12826889
SRR12826890
SRR12826891
SRR12826893
SRR12826894
SRR12826895
SRR12826896
SRR12826897
SRR12826898
SRR12826899
SRR12826900
SRR12826901
SRR12826902
SRR12826903
SRR12826904
SRR12826496
SRR12826497
SRR12826498
SRR12826499
SRR12826500
SRR12826501
SRR12826502
SRR12826507
SRR12826508
SRR12826509
SRR12826510
SRR12826511
SRR12826512
SRR12826513
SRR12826514
SRR12826515
SRR12826518
SRR12826519
SRR12826520
SRR12826521
SRR12826522
SRR12826523
SRR12826526
SRR12826527
SRR12826528
SRR12826529
SRR12826530
SRR12826531
SRR12826532
SRR12826533
SRR12826534
SRR12826535
SRR12826536
SRR12826537
SRR12826538
SRR12826539
SRR12826540
SRR12826541
SRR12826542
SRR12826543
SRR12826544
SRR12826545
SRR12826546
SRR12826547
SRR12826549
SRR12826550
SRR12826551
SRR12826552
SRR12826553
SRR12826554
SRR12826555
SRR12826556
SRR12826557
SRR12826558
SRR12826559
SRR12826560
SRR12826561
SRR12826562
SRR12826563
SRR12826564
SRR12826565
SRR12826566
SRR12826567
SRR12826568
SRR12826569
SRR12826570
SRR12826571
SRR12826572
SRR12826573
SRR12826574
SRR12826575
SRR12826576
SRR12826577
SRR12826578
SRR12826579
SRR12826580
SRR12826581
SRR12826582
SRR12826583
SRR12826584
SRR12826585
SRR12826586
SRR12826587
SRR12826588
SRR12826589
SRR12826590
SRR12826591
SRR12826592
SRR12826593
SRR12826594
SRR12826596
SRR12826597
SRR12826598
SRR12826599
SRR12826600
SRR12826601
SRR12826602
SRR12826603
SRR12826604
SRR12826605
SRR12665142
SRR12665143
SRR12665144
SRR12665145
SRR12665146
SRR12665147
SRR12665148
SRR12665149
SRR12665150
SRR12665151
SRR12665152
SRR12665153
SRR12665154
SRR12665155
SRR12665156
SRR12665157
SRR12665158
SRR12665159
SRR12665160
SRR12665161
SRR12665162
SRR12665163
SRR12665164
SRR12665165
SRR12665166
SRR12665167
SRR12665168
SRR12665169
SRR12665170
SRR12665171
SRR12665172
SRR12665173
SRR12665174
SRR12665175
SRR12665176
SRR12665177
SRR12665178
SRR12665179
SRR12665180
SRR12665181
SRR12665182
SRR12665183
SRR12665184
SRR12665185
SRR12665186
SRR12665187
SRR12665188
SRR12665189

SRR12665190
SRR12665191
SRR12665192
SRR12665193
SRR12665194
SRR12665195
SRR12665196
SRR12665197
SRR12665215
SRR12665218
SRR12665221
SRR12665224
SRR12826606
SRR12826607
SRR12826608
SRR12826609
SRR12826610
SRR12826611
SRR12826612
SRR12826613
SRR12826614
SRR12826615
SRR12826616
SRR12826617
SRR12826618
SRR12826619
SRR12826620
SRR12826621
SRR12826622
SRR12826623
SRR12826624
SRR12826625
SRR12826626
SRR12826627
SRR12826628
SRR12826629
SRR12826630
SRR12826631
SRR12826632
SRR12826633
SRR12826635
SRR12826636
SRR12826637
SRR12826638
SRR12826639
SRR12826643
SRR12826644
SRR12826645
SRR12826646
SRR12826647
SRR12826648
SRR12826649
ERR1749798
SRR10069985
SRR10425388
SRR10425389
SRR10425429
SRR10425431
SRR10425434
SRR10425435
SRR10425442
SRR10425444
SRR10425445
SRR10425476
SRR10425478
SRR10425480
SRR10425481
SRR10425488
SRR12345890
SRR12345901
SRR12447977
SRR12447978
SRR12447979
SRR12447980
SRR12447981
SRR12447982
SRR12447983
SRR12447984
SRR12447985
SRR12447986
SRR12447987
SRR12447988
SRR12447989
SRR12447990
SRR12447991
SRR12447992
SRR12447993
SRR12447994
SRR12447995
SRR12447996
SRR12447997
SRR12447998
SRR12447999
SRR12448000
SRR12448001
SRR12448002
SRR12448003
SRR12448004
SRR12448005
SRR12448006
SRR12448007
SRR12448008
SRR12448009
SRR12665198
SRR12665199
SRR12665200
SRR12665201
SRR12665202
SRR12665203
SRR12665204
SRR12665205
SRR12665206
SRR12665207
SRR12665208
SRR12665209
SRR12665210
SRR12665211
SRR12665212
SRR12665213
SRR12665214
SRR12665216
SRR12665217
SRR12665219
SRR12665220
SRR12665222
SRR12665223
SRR12665225
SRR7722337
SRR7722339
SRR7722341
SRR7722343
SRR7722344
SRR7722345
SRR7722347
SRR7722349
SRR7722350
SRR7722351
SRR8074909
SRR8074911
SRR8074914
SRR8074918
SRR8074920
SRR8074921
SRR8074922
SRR8074923
SRR8074925
SRR8074926
SRR8074927
SRR8074928
SRR8074929
SRR8486103
SRR8486104
ERR1749797
SRR3223108
SRR3223109
SRR3223110
SRR3223111
SRR3223112
SRR3223139
SRR3223181
SRR3223182
SRR3223183
SRR3223184
SRR3223185
SRR3223186
SRR3223195
SRR3223196
SRR3223197
SRR3223198
SRR3223199
SRR3223759
SRR3581665
SRR3581666
SRR4217202
SRR4217203
SRR4217204
SRR4217205
SRR4217206
SRR4217207
SRR4217208
SRR4217209
SRR4217210
SRR4217211
SRR4217212
SRR4217213
SRR4217214
SRR4217215
SRR4217216
SRR4217217
SRR4217218
SRR4217219
SRR4217220
SRR4217221
SRR4217222
SRR4217223
SRR4217224

SRR4217225
SRR4217226
SRR4217227
SRR4217228
SRR4217229
SRR4217230
SRR4217231
SRR4217232
SRR4217233
SRR4217234
SRR4217235
SRR4217236
SRR4217237
SRR4217238
SRR4217239
SRR4217240
SRR4217241
SRR4217242
SRR4217243
SRR4217244
SRR4217245
SRR4217246
SRR4217247
SRR4217248
SRR4217249
SRR4217250
SRR4217251
SRR4217252
SRR4217253
SRR4217254
SRR4217255
SRR4217256
SRR4217257
SRR4217258
SRR4217259
SRR4217260
SRR4217261
SRR4217262
SRR4217263
SRR4217264
SRR4217265
SRR4217266
SRR4217267
SRR4217268
SRR4217269
SRR4217270
SRR4217271
SRR4217272
SRR4217273
SRR4217274
SRR4217275
SRR4217276
SRR4217277
SRR4217278
SRR4217279
SRR4217280
SRR1014837
SRR1014838
SRR1014839
SRR1014840
SRR1014841
SRR1014842
SRR1014843
SRR1014844
SRR1014845
SRR1014846
SRR1014847
SRR1014848
SRR1014849
SRR1014850
SRR1014851
SRR1014852
SRR1014853
SRR1014854
SRR1014855
SRR1014856
SRR1014857
SRR1014858
SRR1014859
SRR1014860
SRR1014861
SRR1014862
SRR1014863
SRR1014864
SRR1014865
SRR1014866
SRR1014867
SRR1014868
SRR1014869
SRR1014870
SRR1014871
SRR1014872
SRR1014873
SRR1014874
SRR1014875
SRR1014876
SRR1014877
SRR1014878
SRR1014879
SRR1014880
SRR1014881
SRR1014882
SRR1014883
SRR1014884
SRR1014885
SRR1014886
SRR1014887
SRR1014888
SRR1014889
SRR1014890
SRR1106129
SRR1145844
SRR1145845
SRR1145846
SRR1145847
SRR1930101
SRR2583948
SRR2584687
SRR2584688
SRR2584689
SRR2584690
SRR2584691
SRR2584692
SRR3198230
SRR3206996
SRR3211955
SRR3211962
SRR3211969
SRR3217251
SRR3217255
SRR3217285
SRR3217894
SRR3218221
SRR3218222
SRR3218224
SRR3218379
SRR3218380
SRR3222505
SRR3222506
SRR3222507
SRR3222508
SRR3222561
SRR3222836
SRR3222837
SRR3222838
SRR3222839
SRR3223078
SRR3223080
SRR3223081
SRR3223082
SRR3223083
SRR3223084
SRR3223104
SRR3223105
SRR3223106
SRR3223107
SRR1014825
SRR1014826
SRR1014827
SRR1014828
SRR1014829
SRR1014830
SRR1014831
SRR1014832
SRR1014833
SRR1014834
SRR1014835
SRR1014836

Script de obtenção de accessions

```
cat accessions.txt | while read X; do
  if [ ! -f $X.html ]; then
    wget "https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=$X" -O $X.html
  fi
done
```

Script de processamento de taxa

```
import glob
for file in glob.glob("*.html"):
    print(file)
    try:
        s = open(file, "r").read()
        s = s.split("oTaxAnalysisData = ")[1].split(";")[0]
        open(file.replace("html", "json"), "w+").write(s)
    except Exception as exc:
        print(str(exc))
```

Script de processamento de folhas para experimento de NETML

```
# vim: tabstop=4 shiftwidth=4 softtabstop=4 expandtab:
import json
import glob
res = {}
taxa = {}
taxa["0"] = "root"
for file in glob.glob("*.json"):
    try:
        d = json.load(open(file, "r"))
        for i in d:
            if type(i) != type({}): continue
            taxa[i["n"]] = i["d"]["name"]
        nodesPais = []
        for i in d:
            if type(i) != type({}): continue
            if i["p"] not in nodesPais:
                nodesPais.append(i["p"])
        nodesFolhas = []
        tree = {}
        def procPai(node, pai, detalhe):
            if pai not in nodesPais:
                node[pai] = detalhe
                #nodesFolhas[detalhe["name"]] = pai
                if detalhe["name"] not in nodesFolhas and
float(detalhe["percent"])>=0.05:
                    nodesFolhas.append(pai)
                return
            node[pai] = {}
            for i in d:
                if type(i) != type({}): continue
                if i["p"] == pai:
                    procPai(node[pai], i["n"], i["d"])
```

```
        procPai(tree, "tx2", "root")
        for i in nodesFolhas:
            if i not in res:
                res[i] = ""
            for j in nodesFolhas:
                if i != j:
                    if j not in res[i]:
                        res[i] += " " + j
    except Exception as exc:
        pass
        #print(str(exc))
for i in sorted(res):
    print(taxa[i] + "," + res[i])
```

Script para processamento de massa de dados para NETML

```
# vim: tabstop=4 shiftwidth=4 softtabstop=4 expandtab:
import json
import glob
for file in glob.glob("*.json"):
    try:
        d = json.load(open(file, "r"))
        taxa = {}
        taxa["0"] = "root"
        for i in d:
            if type(i) != type({}): continue
            taxa[i["n"]] = i["d"]["name"]
        nodesPais = []
        for i in d:
            if type(i) != type({}): continue
            if i["p"] not in nodesPais:
                nodesPais.append(i["p"])
        nodesFolhas = {}
        tree = {}
        def procPai(node, pai, detalhe):
            if pai not in nodesPais:
                node[pai] = detalhe
                nodesFolhas[detalhe["name"]] = pai
                return
            node[pai] = {}
            for i in d:
                if type(i) != type({}): continue
                if i["p"] == pai:
                    procPai(node[pai], i["n"], i["d"])
        procPai(tree, "tx2", "root")
        paths = []
        def procPaths(node, path):
            if 'name' in node:
                #path += " " + node["name"]
                paths.append(path)
                return
            for i in node:
                path += " " + taxa[i]
                procPaths(node[i], path)
        procPaths(tree, "")
```

```
        res = []
        for i in nodesFolhas:
            for j in paths:
                if not j.endswith(i):
                    res.append(i+","+j)
        #import pprint
        #pprint.pprint(tree)
        for i in res:
            print(i)
    except:
        pass
```

Exemplo de JSON extraído com os scripts de processamento de massa de dados


```
[{"n":"tx131567", "p":"0", "d":{"name":"cellular
organisms","total_count":"4455","percent":"78.21"}},
{"n":"tx2759", "p":"tx131567", "d":
{"name":"Eukaryota","total_count":"4455","percent":"78.21"}},
{"n":"tx33154", "p":"tx2759", "d":
{"name":"Opisthokonta","total_count":"4143","percent":"72.74"}},
{"n":"tx4751", "p":"tx33154", "d":
{"name":"Fungi","total_count":"3875","percent":"68.03"}},
{"n":"tx451864", "p":"tx4751", "d":
{"name":"Dikarya","total_count":"3816","percent":"66.99"}},
{"n":"tx4890", "p":"tx451864", "d":
{"name":"Ascomycota","total_count":"819","percent":"14.38"}},
{"n":"tx716545", "p":"tx4890", "d":
{"name":"saccharomyceta","total_count":"819","percent":"14.38"}},
{"n":"tx147538", "p":"tx716545", "d":
{"name":"Pezizomycotina","total_count":"757","percent":"13.29"}},
{"n":"tx716546", "p":"tx147538", "d":
{"name":"leotiomyceta","total_count":"757","percent":"13.29"}},
{"n":"tx715962", "p":"tx716546", "d":
{"name":"dothideomyceta","total_count":"556","percent":"9.76"}},
{"n":"tx147541", "p":"tx715962", "d":
{"name":"Dothideomycetes","total_count":"556","percent":"9.76"}},
{"n":"tx451867", "p":"tx147541", "d":
{"name":"Dothideomycetidae","total_count":"146","percent":"2.56"}},
{"n":"tx134362", "p":"tx451867", "d":
{"name":"Capnodiales","total_count":"145","percent":"2.55"}},
{"n":"tx452563", "p":"tx134362", "d":
{"name":"Cladosporiaceae","total_count":"145","percent":"2.55"}},
{"n":"tx5498", "p":"tx452563", "c":0, "d":
{"name":"Cladosporium","total_count":"145","percent":"2.55"}},
{"n":"tx451868", "p":"tx147541", "d":
{"name":"Pleosporomycetidae","total_count":"408","percent":"7.16"}},
{"n":"tx92860", "p":"tx451868", "d":
{"name":"Pleosporales","total_count":"408","percent":"7.16"}},
{"n":"tx715340", "p":"tx92860", "c":0, "d":
{"name":"Pleosporineae","total_count":"408","percent":"7.16"}},
{"n":"tx715989", "p":"tx716546", "d":
{"name":"sordariomyceta","total_count":"59","percent":"1.04", "kbp":"36"}},
{"n":"tx147550", "p":"tx715989", "d":
{"name":"Sordariomycetes","total_count":"59","percent":"1.04",
"kbp":"36"}},
```

```
{"n":"tx222543", "p":"tx147550", "d":
{"name":"Hypocreomycetidae","total_count":"58","percent":"1.02",
"kbp":"35"}},
{"n":"tx5125", "p":"tx222543", "d":
{"name":"Hypocreales","total_count":"58","percent":"1.02", "kbp":"35"}},
{"n":"tx110618", "p":"tx5125", "d":
{"name":"Nectriaceae","total_count":"58","percent":"1.02", "kbp":"35"}},
{"n":"tx5506", "p":"tx110618", "d":
{"name":"Fusarium","total_count":"58","percent":"1.02", "kbp":"35"}},
{"n":"tx450425", "p":"tx5506", "d":{"name":"Fusarium incarnatum-equiseti
species complex","total_count":"4","percent":"0.07", "kbp":"2"}},
{"n":"tx61235", "p":"tx450425", "c":0, "d":{"name":"Fusarium
equiseti","total_count":"4","percent":"0.07", "kbp":"2"}},
{"n":"tx5204", "p":"tx451864", "d":
{"name":"Basidiomycota","total_count":"2573","percent":"45.17"}},
{"n":"tx5302", "p":"tx5204", "d":
{"name":"Agaricomycotina","total_count":"1186","percent":"20.82"}},
{"n":"tx155616", "p":"tx5302", "d":
{"name":"Tremellomycetes","total_count":"1179","percent":"20.7"}},
{"n":"tx90886", "p":"tx155616", "d":
{"name":"Filobasidiales","total_count":"1179","percent":"20.7"}},
{"n":"tx5408", "p":"tx90886", "d":
{"name":"Filobasidiaceae","total_count":"1179","percent":"20.7"}},
{"n":"tx5209", "p":"tx5408", "d":
{"name":"Filobasidium","total_count":"1179","percent":"20.7"}},
{"n":"tx104409", "p":"tx5209", "c":0, "d":{"name":"Filobasidium
magnum","total_count":"1104","percent":"19.38"}},
{"n":"tx155619", "p":"tx5302", "d":
{"name":"Agaricomycetes","total_count":"2","percent":"0.04", "kbp":"1"}},
{"n":"tx355688", "p":"tx155619", "d":{"name":"Agaricomycetes incertae
sedis","total_count":"2","percent":"0.04", "kbp":"1"}},
{"n":"tx452338", "p":"tx355688", "d":
{"name":"Corticiales","total_count":"2","percent":"0.04", "kbp":"1"}},
{"n":"tx5304", "p":"tx452338", "d":
{"name":"Corticiaceae","total_count":"2","percent":"0.04", "kbp":"1"}},
{"n":"tx391237", "p":"tx5304", "d":
{"name":"Peniophorella","total_count":"2","percent":"0.04", "kbp":"1"}},
{"n":"tx391082", "p":"tx391237", "c":0, "d":{"name":"Peniophorella
odontiiformis","total_count":"2","percent":"0.04", "kbp":"1"}},
{"n":"tx452284", "p":"tx5204", "d":
{"name":"Ustilaginomycotina","total_count":"1384","percent":"24.3"}},
{"n":"tx1538075", "p":"tx452284", "d":
{"name":"Malasseziomycetes","total_count":"1383","percent":"24.28"}},
{"n":"tx162474", "p":"tx1538075", "d":
{"name":"Malasseziales","total_count":"1383","percent":"24.28"}},
{"n":"tx742845", "p":"tx162474", "d":
{"name":"Malasseziaceae","total_count":"1383","percent":"24.28"}},
{"n":"tx55193", "p":"tx742845", "d":
{"name":"Malassezia","total_count":"1383","percent":"24.28"}},
{"n":"tx76775", "p":"tx55193", "c":0, "d":{"name":"Malassezia
restricta","total_count":"1383","percent":"24.28"}},
0]
```


Excerto do código de matching experimentado (contúdo completo no github)

```
foreach (var seq in seqsGhc) {
```

```
if (running == false) break;
var name = Statics.Data.DatabaseFilterSelection.AsEnumerable().Where(x =>
x.Field<string>("ASSEMBLY ACCESSION") == seq.Key.Split('@')[0]).Select(x =>
x.Field<string>("ORGANISM NAME")).FirstOrDefault();
if (name is null) name = "--";
Invoke((MethodInvoker)delegate {
      dt.Rows.Add(seq.Key, name, "0", "COMPUTING");
      Application.DoEvents();
});
if (kmerSize > 0) {
      l = Statics.Data.reads.AsParallel()
            .Select(x => x)
            .Where(x => seq.Value.Contains(
                     x.Value.Substring(
                              ((x.Value.Length / 2) - kmerSize / 2) < 1 ?
0 : (x.Value.Length / 2) - (kmerSize / 2)
                           , ((x.Value.Length / 2) - kmerSize / 2) < 1 ?
x.Value.Length : kmerSize/2
                           )
                  ))
            .Select(x => x).ToArray();
} else {
      l = Statics.Data.reads.AsParallel().Select(x => x).Where(x =>
seq.Value.Contains(x.Value)).Select(x => x).ToArray();
}
if (!seqsMatch.ContainsKey(name)) {
      seqsMatch[name] = new ConcurrentDictionary<string, string>();
}
if (l.Count() > 0) {
      Invoke((MethodInvoker)delegate {
            for (int i = 0; i < dt.Rows.Count; i++) {
                  if (dt.Rows[i][0].ToString() == seq.Key) {
                        dt.Rows[i][2] = l.Count();
                        dt.Rows[i][3] = "DONE";
                  }
            }
            Application.DoEvents();
      });
      foreach (var kv in l) {
            seqsMatch[name][kv.Key] = kv.Value;
      }
} else {
      Invoke((MethodInvoker)delegate {
            for (int i = 0; i < dt.Rows.Count; i++) {
                  if (dt.Rows[i][0].ToString() == seq.Key) {
                        dt.Rows[i][2] = 0;
                        dt.Rows[i][3] = "DONE";
                  }
            }
            Application.DoEvents();
      });
}
}
sw.Stop();
Invoke((MethodInvoker)delegate {
textBoxLog.AppendText(Environment.NewLine + "=== DONE ===");
textBoxLog.AppendText(Environment.NewLine + "Elapsed: " +
sw.Elapsed.ToString());
Application.DoEvents();
```

```
EnableAllButtons();
});
```