UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
PROGRAMA INTERUNIDADES DE PÓS-GRADUAÇÃO
EM BIOINFORMÁTICA

JOSÉ DENEY ALVES DE ARAÚJO

INTEGRAÇÃO DE BASES DE DADOS ADMINISTRATIVOS COM FERRAMENTAS GENÔMICAS

São Paulo
2021

JOSÉ DENEY ALVES DE ARAÚJO

INTEGRAÇÃO DE BASES DE DADOS ADMINISTRATIVOS COM FERRAMENTAS GENÔMICAS

Tese apresentada ao Programa Interunidades de Pós-Graduação em Bioinformática da Universidade de São Paulo (USP), como parte dos requisitos para obtenção do título de doutor em Bioinformática.

Orientador: Dr. HELDER TAKASHI IMOTO NAKAYA

São Paulo

2021

FICHA CATALOGRÁFICA

Araújo, José Deney Alves de
A663 Integração de bases de dados administrativos com ferramentas genômicas / José Deney Alves de Araújo, orientador [Prof.] Dr. Helder Takashi Imoto Nakaya.
São Paulo : 2021.
47 p.

Tese (Doutorado) - Universidade de São Paulo
Orientador: [Prof.] Dr. Helder Takashi Imoto Nakaya
Programa Interunidades de Pós-Graduação em Bioinformática
Área de concentração: Bioinformática

1. Codificação em DNA. 2. Record linkage. 3. Ferramentas genômica. 4. Epidemiologia. 5. BLAST. I. Nakaya, Helder Takashi Imoto, orientador II. Universidade de São Paulo. III. Instituto de Matemática e Estatística. IV. Título

CDD: 572.8

Elaborada pelo Serviço de Informação e Biblioteca Carlos Benjamin de Lyra do IME-USP, pela Bibliotecária Maria Lucia Ribeiro CRB-8/2766

À minha mãe Deuza *(in memorian)*, que mesmo não tendo a oportunidade de acesso à educação, teve o empenho em me educar sempre em primeiro lugar.
Aqui estão os resultados dos seus esforços.
Com muita gratidão!

Agradecimentos

Ao meu Pai Arleison e minha Mãe Deuza *(in memorian)*. Minha mãe sempre me dizia: “você vai ter a oportunidade de estudo que nunca tive”. Essa oportunidade me dada foi árdua, juntos muitas vezes transportavam cargas pesadas nos ombros para me manter nos estudos. Minha base de sustentação diária!

Meu irmão Denis e minhas irmãs: Deuzenilde, Deuzenir, Damízia e Débora que sempre me apoiaram e a resposta em comum sempre foi: o que você decidir, estamos juntos. Minha gratidão!

Ao meu orientador Helder Nakaya, pela oportunidade e a incrível liberdade que nos dá à pesquisa. Mentor fundamental para impulsionar ideias malucas e incríveis como a proposta aqui. Obrigado por acreditar e compartilhar sua visão de mundo!

Aos meus amigos e colegas do CSBL: Allan, Alysson, Amanda, André Nicolau, Bruna, Durão, Edson, Fred, Ícaro, Jeevan, Leandro, Mauro, Rodrigo, Felipe, Thomaz, Tiago, Vanessa, Vitor, Vivi e Youvika. Um especial agradecimento ao Thiago e Pati pelas correções e sugestões.

Ao André Guilherme (com sua mega experiência em genômica) e Juan Silva (doutorando) que fazem parte deste trabalho. Sem a força que me deram em longas horas de discussão este trabalho não teria avançado na mesma proporção.

Aos ex-CSBLers: César, Pedro, Diógenes, Mindy, Mary, Naty, Lucas, Fernando, Gustavo e Fábio pelos rolês e contribuição no início da ideia do trabalho.

Ao meu amigo/irmão Willian Lira, pelas trocas de ideias e força tarefa nos primeiros protótipos gerados.

À população de São Paulo, que por meio da contribuição dos impostos estaduais, tive a oportunidade de ter acesso à Universidade de São Paulo (USP), que fornece uma estrutura completa de ensino e pesquisa.

Finalmente, um agradecimento mais que especial a minha noiva Joana Azevedo por toda paciência e amor dedicado, inclusive em muitos feriados e fins de semanas sacrificados à programação. Você faz parte dessa caminhada!

“A tarefa não é apenas ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo viu.”

(Arthur Schopenhauer)

RESUMO

Araújo, JDA. **Integração de Bases de Dados Administrativos com Ferramentas Genômicas.** 2021. Tese (Doutorado em Ciências) - Universidade de São Paulo, São Paulo, 2021.

A pesquisa em saúde pública frequentemente requer a integração de informações de diferentes fontes de dados. As metodologias de *record linkage* (RL) utilizam os campos de identificação de cada registro para vincular indivíduos de diferentes bancos de dados. No entanto, erros nos registros e o alto custo computacional tornam o RL um grande desafio para integrar grandes bancos de dados administrativos. Apresentamos Tucuxi-BLAST, uma ferramenta versátil para RL que utiliza uma abordagem de codificação e análise *in silico* de DNA para grandes bancos de dados administrativos. Pela reproposição de ferramentas genômicas, fomos capazes de integrar três bases de dados de saúde brasileiras com alta sensibilidade e especificidade e rastrear indivíduos em vários bancos de dados epidemiológicos. Comparado com cinco ferramentas de RL existentes, nosso método obteve a mais alta precisão e velocidade. Além disso, a etapa de validação independente usando 300 milhões de registros simulados, mostrou um consumo de memória RAM de apenas ~4GB e 23h de processamento em um *desktop* comum, sem necessidade do uso de plataformas de processamento de alto desempenho. Tucuxi-BLAST pode melhorar a pesquisa médica baseada em dados e fornece uma maneira rápida e precisa de integrar informações individuais em vários bancos de dados administrativos.

Palavras-chave: Codificação em DNA, *record linkage*, ferramentas genômica, epidemiologia, BLAST

ABSTRACT

Araújo, JDA. **Record Linkage of Administrative Databases with Genomic Tools.** 2021. Thesis (Doctorate in Sciences) - University of São Paulo, São Paulo, 2021.

Public health research frequently requires integrating information from different data sources. Record linkage (RL) methodologies utilize the identification fields of each record to link individuals from different databases. However, errors in the records and the high computational costs involved make RL a major challenge for linking large administrative databases. We present Tucuxi-BLAST, a versatile tool for RL that utilizes a DNA-encoded approach to analyze massive administrative databases. By repurposing genomic tools, we were able to integrate three Brazilian health databases with great sensitivity and specificity, and to perform subject tracing across multiple epidemiological databases. Compared to five existing RL tools, our method obtained the highest accuracy and speed. Furthermore, in an independent validation step using 300 million simulated records. On a desktop, the RAM memory consumption was only ~4GB and 23 hours of processing. Tucuxi-BLAST can improve data-driven medical research and provides a rapid and accurate way for linking individual information across several administrative databases.

Keywords: DNA-encoded, *record linkage*, genomic tools, epidemiological, BLAST

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

AIDS	*Acquired Immunodeficiency Syndrome*
BLAST	*Basic Local Alignment Search Tool*
CIDACS	Centro de Integração de Dados e Conhecimentos para Saúde
CNES	Cadastro Nacional de Estabelecimentos de Saúde
CSBL	*Computational Systems Biology Laboratory*
DATASUS	Departamento de Informática do SUS
DDBJ	*DNA Data Bank of Japan*
DNA	*Deoxyribonucleic acid*
EBI	*European Bioinformatics Institute*
HIV	*Human Immunodeficiency Virus*
HPC	*High Performance Computing*
MEN	*Meningite*
MS	Ministério da Saúde
NCBI	*National Center for Biotechnology Information*
RL	*Record Linkage*
RNA	*Ribonucleic Acid*
SIM	Sistema de Informações sobre Mortalidade
SINAN	Sistema de Informação de Agravos de Notificação
SRA	*Sequence Read Archive*
SUS	Sistema Único de Saúde
SVS	Secretaria de Vigilância em Saúde
TB	Tuberculose
USP	Universidade de São Paulo

SUMÁRIO

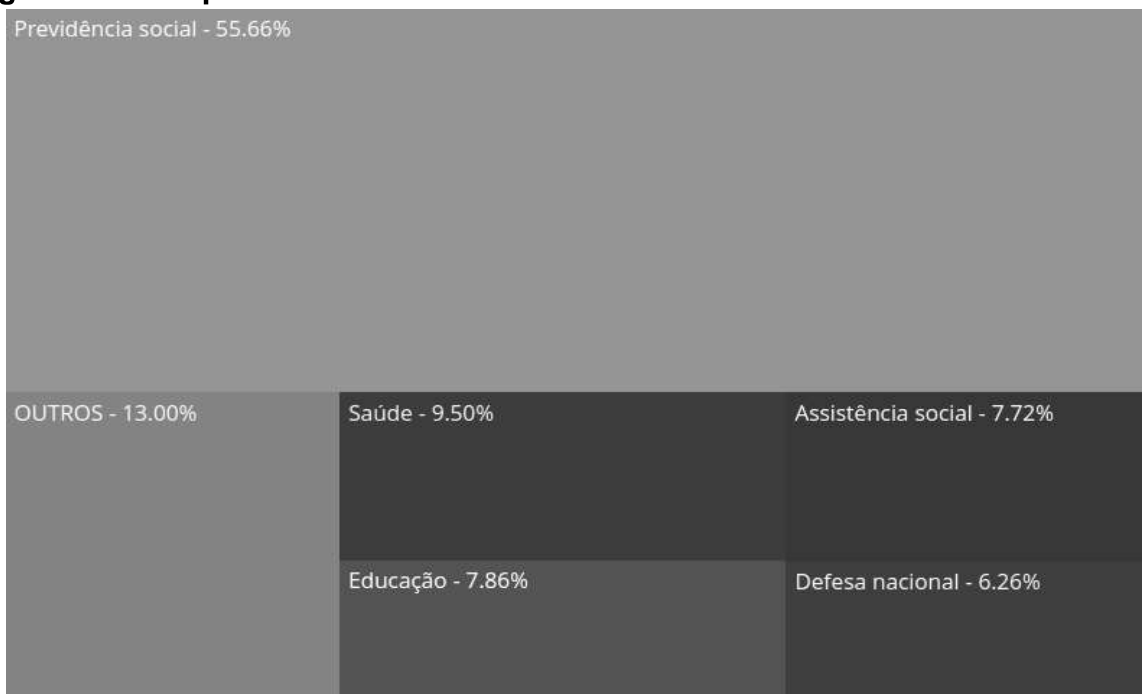
1. INTRODUÇÃO

1.1. SAÚDE PÚBLICA

O atendimento de saúde pública do Brasil é realizado pelo Sistema Único de Saúde (SUS), e está agregado ao Ministério da Saúde (MS). Este é formado pelas três instâncias do poder: União, Estados e Municípios conforme determinado pela Constituição Federal de 1988 (CF-88). O SUS garante o acesso universal à saúde, englobando desde o atendimento da atenção médica primária, até procedimentos médicos de alta complexidade em hospitais, como transplante de órgãos.

O MS é o terceiro Órgão Superior do Governo Federal com maior destino de verbas públicas (http://bit.ly/despesas_2019) (Figura 1). Somente no ano de 2019, antes da crise mundial de COVID-19, os recursos destinados a este segmento foi de 114,18 bilhões de reais segundo o Portal da Transparência (http://bit.ly/minsaude2019). Estes recursos são distribuídos principalmente em: assistência hospitalar e ambulatorial (49,94%); atenção básica (23,12%); suporte profilático e terapêutico (9,61%); administração geral (6,97%); e vigilância epidemiológica (5,49%). A vigilância epidemiológica é utilizada pelo MS para auxiliar na tomada de decisões de gestão de recursos financeiros. Além disso, através do monitoramento é possível identificar áreas (ou doenças) que necessitem mais recursos, estratégias, ou que causam um maior impacto na saúde da população brasileira.

Figura 1: Principais destinos das verbas do Governo Federal



Fonte: Portal da Transparência <http://bit.ly/despesas-saude2019>. Acesso em: 27 abr. 2021.

1.1.1. Monitoramento e tomadas de decisão

Muitas características podem levar ao aparecimento e persistência de doenças/patógenos. Essas incluem fatores socioeconômicos como nível de escolaridade, cultura, e alimentação, assim como fatores ambientais como clima, por exemplo. Diversos trabalhos propuseram o uso dessas variáveis na determinação de políticas públicas voltadas para a melhoria da qualidade de vida das populações (FANG et al., 2019; REIS-SANTOS et al., 2019). O monitoramento da saúde é realizado através da Secretaria de Vigilância em Saúde (SVS), estabelecido por legislação específica (BRASIL, 2017). Dezenas de bancos de dados comportam as informações epidemiológicas de todas as unidades federativas do país. Com base nas informações coletadas nesses inúmeros bancos de dados é possível desenvolver estratégias para identificar situações que resultam em risco de agravos à saúde e tomar decisões estratégicas no controle de surtos de doenças em todas as regiões do país.

1.1.2. Bancos de dados administrativos/epidemiológicos

O SUS possui uma rede de bancos de dados que auxiliam na gestão da saúde no Brasil. Estes bancos de dados agregam informações de pacientes

acometidos com diferentes doenças. Essas informações incluem dados clínicos, escolaridade, moradia, unidade de atendimento, e tratamento aplicado. Os sistemas do SUS são gerenciados pelo Departamento de Informática do Sistema Único de Saúde (DATASUS). O DATASUS é responsável por prover os sistemas de informação e suporte de informática, necessários ao processo de planejamento, operação e controle de segurança. Atualmente, o DATASUS possui mais de 200 sistemas que auxiliam o MS na gestão da saúde brasileira. Dentre estes, o Cadastro Nacional de Estabelecimentos de Saúde (CNES) possui todas as informações sobre a estrutura instalada para atendimento à população no país, como leitos, profissionais e equipamentos disponíveis, tanto do segmento privado conveniado ao SUS quanto do segmento público (DATASUS, 2021).

Entre os principais bancos de dados, destacamos o Sistema de Informação de Agravos de Notificação (SINAN). O SINAN recebe informações que constam na lista nacional de doenças de notificação compulsória, a qual foi definida por meio da Portaria de Consolidação nº 4, de 28 de Setembro de 2017, anexo V - Capítulo I. No SINAN são mantidos 54 bancos de dados de notificação de doenças e agravos como, por exemplo, meningite, tuberculose, febre amarela, malária e HIV. A coleta de informações específicas de cada doença possibilita a realização do diagnóstico de um determinado evento na população, fornecendo subsídios para rastreamento das causas dos agravos de notificação. Isso contribui para a compreensão das condições epidemiológicas de todas as regiões do país (SINAN, 2021). Apesar da quantidade, as bases de dados do SINAN não são integradas, ou seja, o sistema não é capaz de associar automaticamente registros do mesmo indivíduo oriundo de múltiplos bancos de dados que o compõem.

A diversidade de dados de saúde armazenada em bases de dados possibilita a análise de combinação de fatores, o que pode auxiliar até mesmo na compreensão e previsão de eventos futuros, tais como, novos surtos de doenças, local, grupo e possível impacto causado por este surto (SÁNCHEZ-GONZÁLEZ et al., 2018; ZHANG et al., 2019). Ao mesmo tempo que a grande quantidade de dados coletados favorece análises e melhora o monitoramento de eventos epidemiológicos, esta também exige maior estrutura computacional de processamento e armazenamento.

1.2. *BIG DATA*

O aumento exponencial em nossa capacidade de coletar e armazenar dados fomentou o desenvolvimento de uma área de pesquisa denominada "*big data*". O termo *big data*, foi utilizado pela primeira vez em 1997, para se referir a grandes conjuntos de dados que não cabiam na memória RAM ou disco de armazenamento (COX; ELLSWORTH, 1997). Atualmente, o conceito de *big data* ganhou um uso mais amplo. De modo geral, *big data* refere-se sempre a grandes volumes de dados, os quais podem possuir dados estruturados, não estruturados e semiestruturados (SHARMA; AGARWAL; MAHAPATRA, 2020). Com seu alto nível de complexidade, a análise de *big data* requer tecnologias poderosas e algoritmos de processamento de dados avançados. Portanto, dado o volume atual de informações, as ferramentas estatísticas tradicionais já não conseguem ser tão eficientes no caso de aplicações envolvendo *big data* (OUSSOUS et al., 2017).

Em todas as áreas vem crescendo o volume de dados e houve uma verdadeira explosão de informações, a partir da internet das coisas (IoT - *Internet of Things*), com o volume de armazenamento saltando de 10 *zettabytes* (ZB) em 2016 para 40 ZB em 2019. Um ZB equivale a um trilhão de *gigabytes* (REINSEL; GANTZ; RYDNING, 2018). Neste contexto, a quantidade de novas informações sobre saúde não é diferente. Extrair informações das bases de dados de saúde pode fornecer evidências para o controle de surtos de doenças infecciosas. A análise integrada rápida desses dados pode economizar milhões de reais aos cofres públicos, principalmente em regiões emergentes.

No entanto, o crescimento exponencial no volume de dados traz desafios. As limitações dos *softwares* tradicionais formam barreiras para pesquisadores analisarem *big data*. Esse problema decorre do grande volume e diversidade dos dados (heterogeneidade), o que causa demora excessiva na condução de análises (KUO et al., 2019). Além disso, apesar da riqueza em potencial existente em bancos de dados extraídos de atendimentos de saúde, um levantamento realizado pela *Dimensional Insight* demonstrou que 56% dos hospitais e consultórios médicos nos Estados Unidos ainda não possuem governança de *big data* ou planos de análise de longo prazo adequados (DIMENSIONAL INSIGHT, 2017).

1.3. *RECORD LINKAGE*

A maneira de conseguir conectar diferentes bancos de dados é denominada *Record Linkage* (RL). Existem duas principais abordagens que são aplicadas para RL entre bases de dados. A primeira é o *linkage* determinístico que é o processo de vincular informações por uma ou mais chaves únicas nas bases de dados. Assim o RL ocorre caso a chave seja idêntica entre os registros (SAYERS et al., 2016). No entanto, nem sempre chaves únicas estão disponíveis para todos os registros ou são comuns entre dois bancos. Além disso, existe a manipulação humana no preenchimento dos formulários que podem agregar erros na chave, o que impossibilita a correspondência entre registros.

A segunda abordagem de RL, é conhecida como *linkage* probabilístico, que por sua vez, tenta vincular dois ou mais campos como se fossem múltiplas chaves não-exclusivas. O pareamento de múltiplos campos compartilhados entres os registros é analisada de forma probabilística e a correspondência entre eles é estabelecida. Em um cenário ideal, seria necessário comparar todos os registros no banco de dados. O principal problema é quando há necessidade de comparar grandes bases de dados. Se for considerado o pareamento de todas as possibilidades, computacionalmente se torna inviável (MICHELSON; KNOBLOCK, 2006). Uma solução comum para reduzir o custo computacional é conhecida como *blocking* (NEWCOMBE, 1967). Essa técnica consiste em segmentar os registros em subconjuntos menores com uma ou mais características idênticas para serem considerados um *match* verdadeiro. Esta abordagem pode acelerar a integração das bases de dados, entretanto, considerando as bases de dados de registros epidemiológicos, isso diminui a eficiência do processo de integração, pois limita a quantidade de comparações (SAYERS et al., 2016). Além disso, esta abordagem requer pré-processamento excessivo e etapas de criação de *blocking* que precisam ser pré-definidos e estabelecidos pesos específicos para cada critério (ALI et al., 2019). Deve-se considerar ainda, que devido a quantidade de erros/divergências entre as bases de dados pode afetar diretamente no resultado do RL quando se utiliza o método de *blocking*.

1.4. FERRAMENTAS DE RECORD LINKAGE

Existem inúmeras ferramentas disponíveis para RL, descrevemos aqui cinco ferramentas *open source* de fácil acesso e com suporte e atualização ativa. São elas:

1) Python Record Linkage Toolkit (DE BRUIN, 2019): escrito em Python, esta ferramenta permite além de *record linkage*, um módulo de desduplicação de dados. O pacote tem finalidade acadêmica e é ideal para pequenos e médios conjuntos de dados. O código está disponível em https://github.com/J535D165/recordlinkage.

2) csvdedupe (GREGG; EDER, 2019): escrito em Python, utiliza *fuzzy matching*, possui módulo de desduplicação de dados. Faz parte da suíte de serviços chamada Dedupe.io. A biblioteca faz uso do aprendizado ativo para combinar pares de registros. O Dedupe.io oferece gratuitamente o programa csvdedupe, disponível em https://github.com/dedupeio/csvdedupe.

3) RecordLinkage (R) (SARIYAR; BORG, 2010): escrito em linguagem R, possui funções para *record linkage* e desduplicação de dados. Possui classificação tanto supervisionada, quanto não-supervisionada. O programa está disponível no repositório do R-Cran em: https://cran.r-project.org/web/packages/RecordLinkage/index.html).

4) fastLink (ENAMORADO; FIFIELD; IMAI, 2019): também escrito em linguagem R, permite *missing data*. É implementado o modelo probabilístico de Fellegi-Sunter o qual usa o algoritmo Expectation-Maximization. Está disponível em: https://github.com/kosukeimai/fastLink.

5) CIDACS-RL (BARBOSA et al., 2020): desenvolvido em linguagem Java pelo Centro de Integração de Dados e Conhecimento para Saúde - Fundação Oswaldo Cruz - Bahia. Utiliza um algoritmo de *linkage* determinístico iterativo e *fully match* baseado em uma combinação de busca de indexação e algoritmos de pontuação (fornecidos pelo Apache Lucene). O sistema foi otimizado para trabalhar com baixos recursos computacionais e roda com processamento distribuído/paralelo com utilização do Apache Spark. Está disponível em: https://github.com/gcgbarbosa/cidacs-rl-v1.

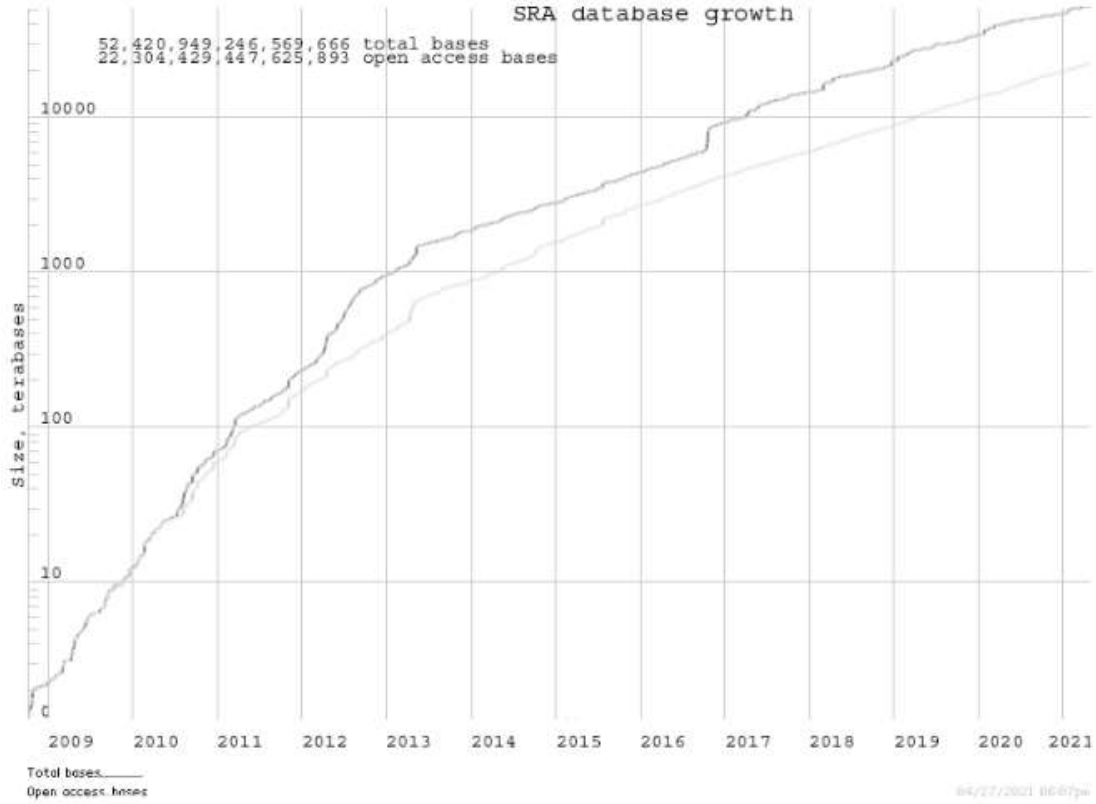
1.5. CIÊNCIAS ÔMICAS

A bioinformática sempre precisou lidar com enormes quantidades de dados. Desde o Projeto Genoma Humano (1990-2003), os dados ômicos já exigiam algoritmos e processamentos de grande massa de dados, que envolviam milhões de sequências de DNA. Somente o primeiro genoma humano gerou 3,2 bilhões de pares de bases (LANDER et al., 2001). Com o avanço dos sequenciadores de segunda geração, houve um crescimento exponencial de geração de dados biológicos. Esse avanço permitiu o sequenciamento de 2.504 genomas humanos de 26 populações distribuídas em todos os continentes em menos de 10 anos (1000 GENOMES PROJECT CONSORTIUM et al., 2015).

As ciências ômicas lidam com a totalidade das moléculas de uma mesma classe de organismo. Dentre as áreas ômicas, três em especial são responsáveis pela geração e disponibilização de uma gigantesca quantidade de dados: 1) genômica para dados de DNA; 2) transcriptômica (RNA); e 3) proteômica (proteínas). Por exemplo, uma única corrida de sequenciamento de RNA pode gerar até 20 bilhões de sequências com 300 pares de bases (Illumina NovaSeq™ 6000 System). Muitos desses dados estão disponíveis em bancos de dados online. Por exemplo, o *Sequence Read Archive* (SRA) armazena e disponibiliza dados não-processados de experimentos provenientes de diferentes tecnologias de sequenciamento de RNA. Estas tecnologias incluem as plataformas Illumina, 454, IonTorrent, Complete Genomics, PacBio e OxfordNanopores (SRA, 2021).

O SRA armazena uma grande quantidade de dados públicos e faz parte de uma parceria internacional entre o *National Center for Biotechnology Information* (NCBI), *European Bioinformatics Institute* (EBI), e o *DNA Database of Japan* (DDBJ). Todos os dados enviados para qualquer uma das três organizações são compartilhados entre elas (LEINONEN et al., 2011). Juntos, estes dados totalizam aproximadamente 18.400 *terabytes* de dados ômicos (Figura 2).

Figura 2: Dados de sequenciamentos depositados no SRA

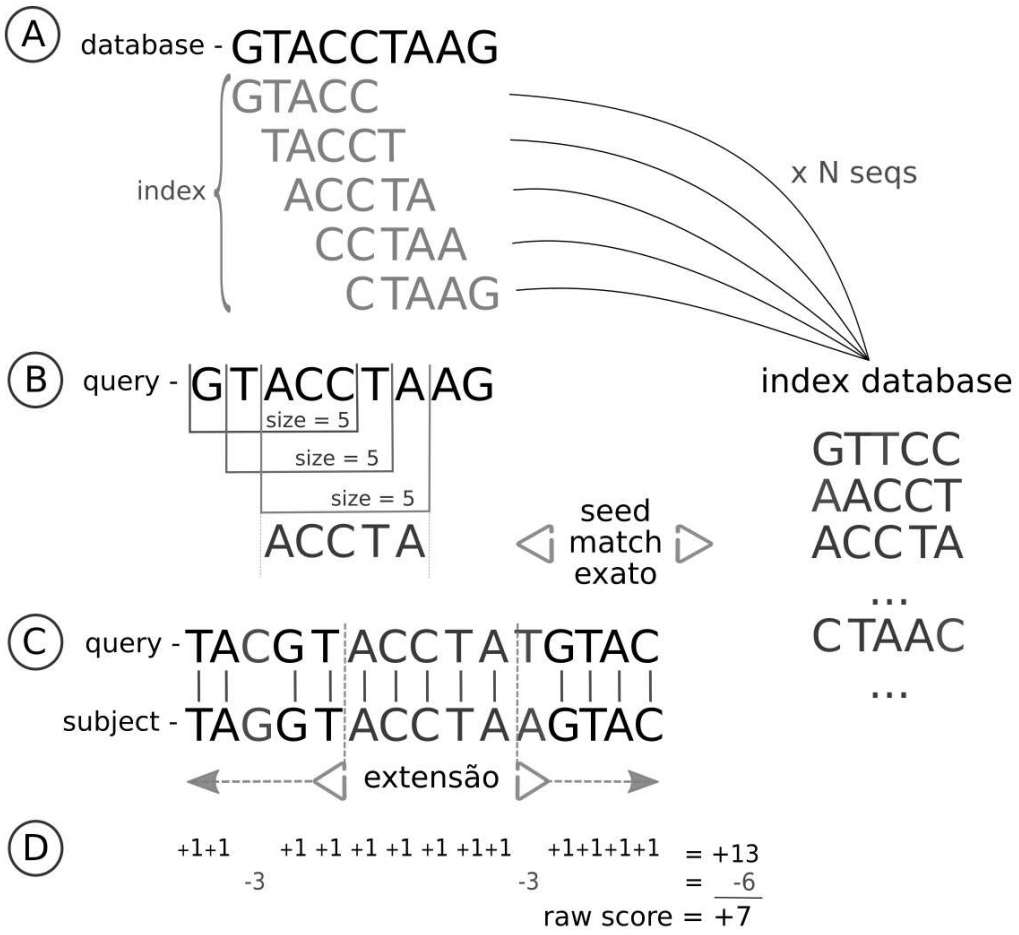


A linha azul representa a quantidade total de dados gerados, enquanto a linha amarela representa os dados que já se tornaram públicos na plataforma. Fonte: SRA, 2021 <https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi>. Acesso em: 27 abr. 2021.

1.5.1. Algoritmo BLAST

Recentemente listado entre os dez códigos da computação que transformaram a ciência (PERKEL, 2021), o BLAST (ALTSCHUL et al., 1990), é um algoritmo de alinhamento local de sequências. Essa ferramenta é capaz de comparar sequências de nucleotídeos ou aminoácidos contra bilhões de outras diferentes em um banco de dados, e retornar aquela com a maior similaridade com a sequência submetida, calculando um *score* para cada correspondência. O princípio de funcionamento do BLAST pode ser resumido em quatro etapas principais: 1) criação de index da base de dados de referência; 2) *seeds* correspondentes entre a *query* (consulta) e o *subject* (index criado da referência); 3) extensão do alinhamento em ambas as direções; e 4) pontuação entre os pares de alinhamentos das sequências (Figura 3).

Figura 3: Princípio simplificado de funcionamento do algoritmo BLAST



É realizado a criação de index da base de dados de referência (A). A *query* é lida base a base seguindo o tamanho predefinido, neste exemplo (*size*=5) (B). A partir da *seed* encontrada na *query* e index com *match* exato, começa a extensão do alinhamento (C) e finalmente é calculada a pontuação das métricas (D).

1.5.1.1. Estatísticas de alinhamento entre as sequências e terminologia do BLAST

Um alinhamento local sem *gaps* (sem lacunas), consiste em um par de segmentos de comprimento exato entre duas sequências comparadas (Figura 3B). Uma modificação do algoritmo Smith-Waterman (SMITH; WATERMAN, 1981) busca todos os pares de segmentos entre as *queries* e os *subjects*. Estes são chamados de pares de segmentos de alta pontuação (*high-scoring segment pairs*, HSPs). O melhor alinhamento entre a *query* e o *subject* é conhecido como *hit* (ou ainda, *top hit*).

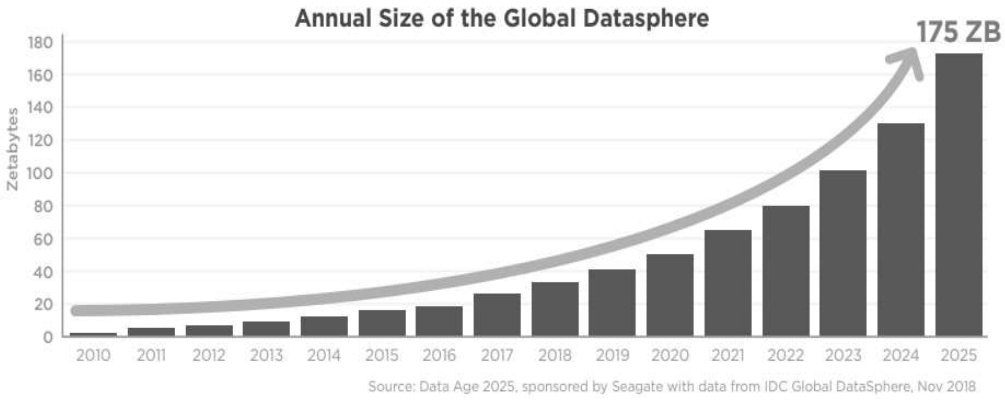
Os valores de pontuação do algoritmo BLAST são fundamentais para avaliar a similaridade e o grau de confiabilidade entre duas sequências alinhadas. Existem duas principais métricas que permitem avaliar as correspondências entre os alinhamentos realizados pelo algoritmo BLAST: o *E-value* e o *Bit-score*. O primeiro estima o número de alinhamentos ao caso esperados entre duas sequências. O cálculo deste valor leva em conta tanto o tamanho do alinhamento e a composição de bases das sequências isca e alvo, quanto também o tamanho do banco de sequências que está sendo pesquisado (ALTSCHUL et al., 1990). Quanto maior o número de sequências no banco, maior é o número esperado de alinhamentos que aconteçam por similaridade espúria entre as duas sequências, e maior será o valor de *E-value*. Desse modo, um *E-value* de 2, significa que esse alinhamento é esperado ocorrer ao acaso pelo menos duas vezes, dadas as características de ambas sequências dentro do universo de sequências presentes no banco. Assim, quanto menor for o *E-value*, ou mais próximo de zero, mais "significativo" será o *match* entre as sequências com seu respectivo alinhamento.

Já o *Bit-score*, é o valor normalizado atrelado a composição (número de *gaps* e *mismatches*) e a extensão total do alinhamento (FASSLER; COOPER, 2011). O *E-value* decresce exponencialmente, conforme aumenta o *score* do *match* entre o alinhamento (Figura 3D). Diferentemente do *E-value*, o *Bit-score* não depende do tamanho da base de dados. Isso permite que seja usado para comparar as pontuações de alinhamento de diferentes pesquisas/bancos de dados.

1.6. NOVAS TECNOLOGIAS DE ARMAZENAMENTO

Apesar das vantagens, a geração de grandes quantidades de dados traz desafios relacionados ao armazenamento. Nunca foram gerados tantos dados como hoje, e existem projeções de que a velocidade aumentará ainda mais. Estimativas indicam que 175 *zettabytes* serão necessários para armazenar os dados gerados até 2025 (Figura 4). Essa quantidade de dados não pode ser armazenada com os recursos disponíveis atualmente (REINSEL; GANTZ; RYDNING, 2018). Mesmo que as grandes fabricantes de sistemas de armazenamento atual, que são baseadas em silício, fossem capazes de aumentar a produção de *hardware*, previsões indicam que nossas fontes de silício acabarão em 2040 (ZHIRNOV et al., 2016).

Figura 4: Projeção de dados gerados até 2025



Barras azuis representam os dados monitorados até 2018. Após esse ano, as barras representam projeções. A linha amarela representa a tendência dos dados de 2010 a 2025. Fonte: IDC Global DataSphere, 2018 <http://bit.ly/IDCglobal>. Acesso em: 28 abr. 2021.

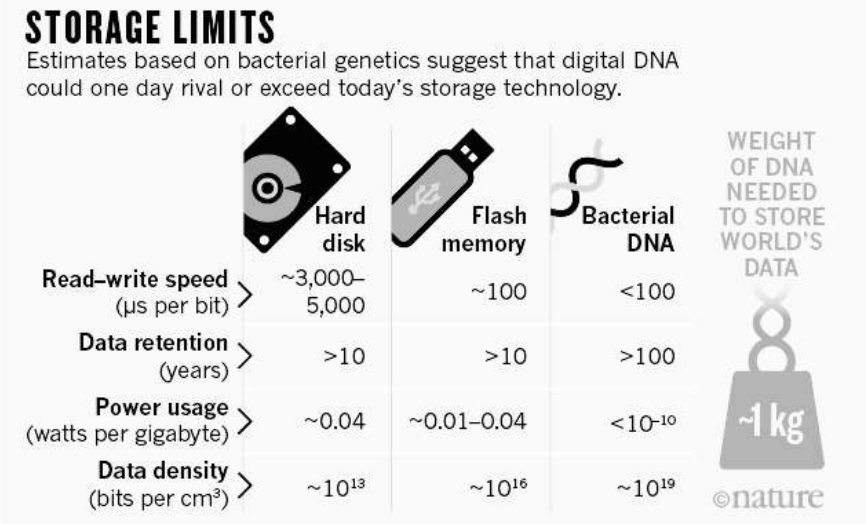
1.6.1. Codificação de dados e *DNA storage*

As estimativas de aumento exponencial na produção dos dados têm fomentado a busca por tecnologias inovadoras de armazenamento. Uma das propostas de destaque é a tecnologia de armazenamento em sequências de DNA (ANAVY et al., 2019; CEZE; NIVALA; STRAUSS, 2019; TAKAHASHI et al., 2019). A capacidade de armazenamento em DNA é conhecida desde a década de 90 (DAVIS, 1996), mas o seu uso para armazenar dados não genômicos é uma ideia recente. Grandes empresas do ramo de tecnologia têm investido em projetos de desenvolvimento comercial dessa nova tecnologia (http://bit.ly/storageMS). Recentemente a Microsoft®, em parceria com a *University of Washington*, lançou o primeiro protótipo automatizado na gravação e recuperação de informações utilizando o DNA (TAKAHASHI et al., 2019).

A molécula de DNA é um meio de armazenamento com alta durabilidade e densidade de informação. Além disso, a possibilidade do armazenamento *in vitro* e *in vivo* oferece flexibilidade, já que é estável com variações de temperatura e umidade. Experimentos têm demonstrado que apenas um grama de DNA é capaz de armazenar 215 *petabytes* de dados (ERLICH; ZIELINSKI, 2017). Estima-se que 1 kg de DNA seria suficiente para armazenar todos os dados produzidos no mundo (EXTANCE, 2016). Atualmente os principais gargalos de armazenamento existentes

como durabilidade, capacidade de leitura e escrita, economia de energia e redução de espaço físico, tornam o DNA uma fonte potencial para o armazenamento comercial (Figura 5) (CEZE; NIVALA; STRAUSS, 2019).

Figura 5: Comparação da capacidade de armazenamento tradicional com a nova tecnologia utilizando o DNA



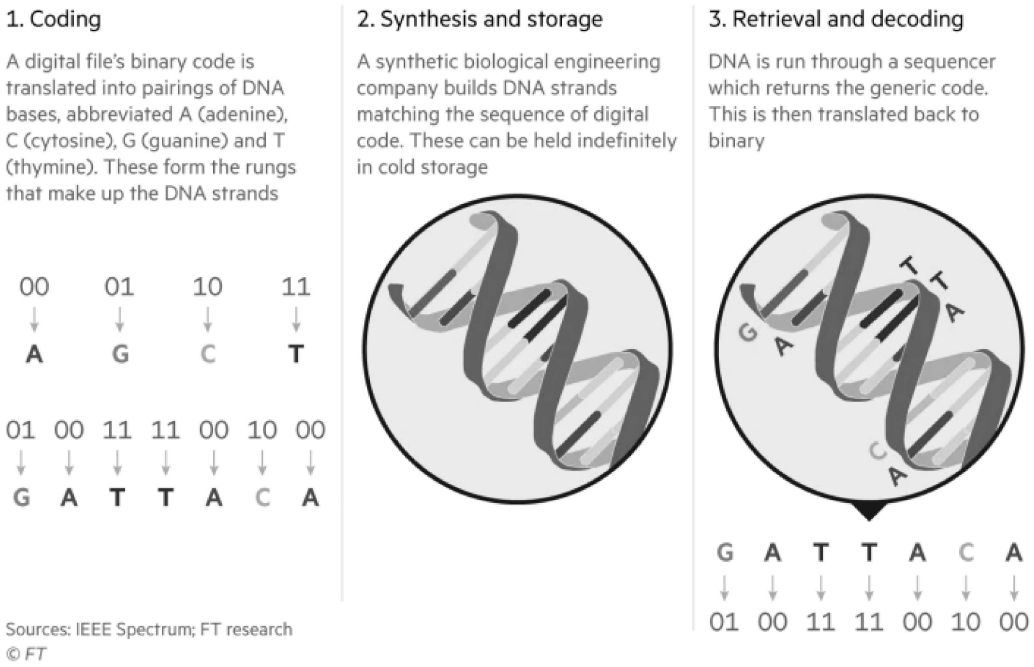
Fonte: Extance, 2016.

Aproximadamente 90% dos dados em *backup* oriundos de computadores e outros sistemas digitais não são mais acessados (GREENGARD, 2019). Mesmo assim, manter uma cópia dos dados é uma forma de garantir que a informação não será perdida. A manutenção de *backups* de longa duração gera alto custo e a estabilidade da molécula do DNA por milhares de anos é uma das maiores vantagens do seu uso nestes casos. Apesar de ser um polímero complexo, o DNA pode ser replicado facilmente, tanto *in vitro* como *in vivo*. Sendo assim, uma possível solução para a realização diária de *backups* em bancos de dados.

Existem diversas propostas relacionadas à conversão de informação em sequências de DNA (ERLICH; ZIELINSKI, 2017; LEE et al., 2019; ORGANICK et al., 2018; SHIPMAN et al., 2016). De modo geral, a codificação em DNA é realizada a partir da linguagem computacional binária (0's e 1's). As informações binárias recebem uma codificação aos pares para compor 4 bases de DNA - Adenina (A), Timina (T), Citosina (C) e Guanina (G). Assim, temos: **00** para **A**, **01** para **G**, **10** para

C e **11** para **T.** Para recuperar a informação codificada no DNA, o processo acontece no processo reverso da codificação (Figura 6).

Figura 6: Codificação de dados em DNA para armazenamento



Codificação binária de 0's e 1's para A (adenina), T (timina), C (citosina) e G (guanina) **(1)**. Síntese da fita de DNA com base na codificação binária **(2)**. Recuperação da informação pela decodificação do DNA em linguagem binária **(3)**. Fonte: IEEE Spectrum FT research, 2018.

Apesar das vantagens existentes na possibilidade de usar o DNA como fonte de armazenamento de dados, existem algumas barreiras que precisam ser superadas para tornar viável o uso comercial. Sem dúvida, o principal obstáculo é o alto custo na síntese e leitura de DNA. Atualmente é necessário utilizar sequenciadores de DNA para ler as informações presentes nas sequências. Pesquisas vêm sendo realizadas a fim de se transpor tais barreiras, como exemplo recentemente foi publicado um novo método capaz de reduzir em 75% o custo com síntese de DNA (ANAVY et al., 2019). Ainda assim, o custo é elevado quando comparado com tecnologias atuais. No entanto, com as proporções da produção de dados, novas tecnologias de armazenamento como o DNA podem ser a solução definitiva para grande produção de dados.

1.7. INTEGRAÇÃO DE DADOS COM FERRAMENTAS ÔMICAS

A conversão de dados de saúde em DNA permite a utilização das ferramentas de bioinformática para integração de grandes bases de dados. Como apresentado na seção 1.4.1., os algoritmos de bioinformática lidam com bilhões de sequências de DNA, devido à complexidade do DNA e a necessidade de ter alta cobertura de sequenciamento (SIMS et al., 2014). Neste trabalho, propomos a reproposição de ferramentas genômicas na análise de diferentes tipos de dados que não tem origem genética. Para que isso seja possível é preciso que as informações administrativas/epidemiológicas estejam codificadas em sequências de DNA. Por isso, desenvolvemos um método de codificação em DNA *in silico*.

Nossa proposta é a possibilidade de utilizar ferramentas genômicas na integração de bases de dados administrativos, desde que estes estejam recodificados em sequências de DNA. De forma simplificada, nosso método converte informações de um determinado indivíduo em pseudo-sequências de DNA. Em seguida, realizamos o alinhamento de sequências similares com o algoritmo BLAST (ALTSCHUL et al., 1990). Esta estratégia inovadora pode fornecer uma solução rápida, barata e prática de lidar com o enorme volume de informação dos diferentes bancos de dados do Ministério da Saúde, bem como qualquer base de dados com informações administrativas.

3. OBJETIVOS

3.1. GERAL

Desenvolver um novo método de integração de bases de dados administrativos utilizando ferramentas genômicas por meio de alinhamentos de sequências similares.

3.2. ESPECÍFICOS

- Desenvolver um método de codificação *in silico* de dados administrativos em DNA;

- Aplicar algoritmos de alinhamento de sequências para fazer cruzamentos *(linkage)* de bases de dados;

- Treinar um modelo de *machine learning* para classificação dos dados obtidos com o algoritmo BLAST;

- Desenvolver uma ferramenta *web* para rápida verificação/curadoria manual do *record linkage*;

- Fazer *benchmark* com outras ferramentas *open source* de RL;

- Realizar integração de 300 milhões de registros usando dados simulados.

4. MATERIAIS E MÉTODOS

4.1. IMPLEMENTAÇÃO E DISPONIBILIDADE DO CÓDIGO

Tucuxi-BLAST foi desenvolvido na linguagem de programação Python 3 [v.3.7.3]. O *record linkage* (RL) entre as bases de dados usa o algoritmo BLAST (Altschul et al., 1990). A implementação do *software* foi feita em Linux. Por ter sido desenvolvido em linguagem multiplataforma, pode ser facilmente adaptado para rodar em qualquer sistema operacional como Microsoft Windows e MacOS. O Tucuxi-BLAST estará disponível e regularmente atualizado no repositório do CSBL (*Computational Systems Biology Laboratory*) no GitHub (https://github.com/csbl-usp/tucuxi_blast).

4.1.1. O porquê do nome Tucuxi

Foi uma homenagem à fauna e à flora amazônica. Por ser natural do Amazonas, sempre estive nos grandes rios de água doce e observava a ligação entre os botos e os pescadores. A presença de botos é um bom indicador da fartura de peixes no local, pois escolhem a região de acordo com a disponibilidade de alimento. O Tucuxi (*Sotalia fluviatilis*) é uma espécie de golfinho de água doce que vive na Bacia Amazônica (FLORES; DA SILVA; FETTUCCIA, 2018). Esta também conhecida como “boto-Tucuxi”. Apesar da semelhança e ser chamado popularmente de boto, sua linhagem filogenética é mais próxima ao grupo dos golfinhos de água salgada e não do verdadeiro boto amazônico, o “boto-Rosa” (*Inia geoffrensis*), famoso no folclore e lendas populares da região (DA SILVA; MARTIN, 2018). Desse modo o Tucuxi é considerado um pseudo boto. Por essa curiosidade, adotamos o Tucuxi como símbolo do programa que cria pseudo-sequências de DNA.

4.2. CODIFICAÇÃO DE DADOS EM DNA *IN SILICO* E MANUTENÇÃO DE PRIVACIDADE

Quatro campos de identificação foram utilizados para realizar o RL entre as bases de dados simuladas e reais do SINAN. Os campos selecionados são amplamente utilizados para realizar RL de bancos de dados administrativos. Antes de executar o Tucuxi-BLAST, convertemos todos os caracteres de *strings* (por exemplo, nomes e sobrenomes) para maiúsculas e removemos caracteres especiais e diacríticos. Disponibilizamos o módulo Tucuxi-clean-data, no repositório principal, que realiza esse procedimento de pré-processamento rapidamente.

A seguir, os campos de identificação foram codificados em DNA na seguinte ordem: (1) nome e sobrenome do indivíduo; (2) data de nascimento; (3) sexo; e (4) nome da mãe e sobrenome. Os bancos de dados de consulta e referência devem ser processados com o mesmo procedimento.

Para remover registros duplicados nas bases de dados, também desenvolvemos a ferramenta Tucuxi-BW (disponível em: https://github.com/csbl-usp/tucuxi_blast) usando Python e a função de *clusters* do programa VSEARCH [v.2.15.2] (ROGNES et al., 2016).

Para garantir a privacidade dos dados, uma chave aleatória de 6 dígitos alfanuméricos é gerada em cada execução do Tucuxi-BLAST. A chave é usada para variar o padrão *in silico* de codificação das sequências de DNA. Isso garante que o conhecimento de uma execução anterior não possa ser usada para quebrar a privacidade da informação de outros bancos em execuções futuras do Tucuxi-BLAST. As sequências geradas serão diferentes a menos que seja imputada manualmente a mesma chave para a codificação.

4.3. ALINHAMENTO DE SEQUÊNCIAS USANDO TUCUXI-BLAST

Após codificar os dados em DNA, utilizamos o algoritmo *open source* BLASTn [v.2.10.0] (ALTSCHUL et al., 1990) para alinhar as sequências entre os bancos de dados. Usamos os seguintes parâmetros no Tucuxi-BLAST: megablast e dc-megablast como task; “-dust no”; “-max_hsps 1”; “-strand plus”; “e-value 1e-10”. Todos os demais parâmetros foram utilizados *default* do manual do BLAST. O alinhamento de sequências longas contendo incompatibilidades pode gerar valores de pontuação (*Bit-score* e *E-value*) maiores do que o alinhamento de sequências menores, mesmo que sejam idênticas. Isso pode penalizar indivíduos com nomes curtos durante o processo de *record linkage*. Para evitar que o tamanho dos nomes tenham impacto direto na análise, os valores de *Bit-score* e *E-value* são representados como a porcentagem do melhor acerto possível para cada sequência (ou seja, a sequência contra si mesma - *itself*). Esta abordagem BLAST-all, em que a sequência é comparada com ela mesma e as pontuações normalizadas, permite a análise de sequências com comprimentos diferentes.

4.4. BANCOS DE DADOS DO SINAN

Os bancos de dados do SINAN e do Sistema de Informações sobre Mortalidade (SIM) do Estado do Amazonas foram fornecidos pela Fundação de Vigilância em Saúde do Amazonas. A utilização dos dados foi aprovada pelo Comitê de Ética em Pesquisa da Fundação de Medicina Tropical Dr. Heitor Vieira Dourado, Amazonas, Brasil (Protocolo nº 3.462.265). As bases de dados do SINAN (2012 a 2017) para Tuberculose (TB), HIV/AIDS e Meningite (MEN) incluem informações sobre os portadores dessas doenças. Nas bases de dados do SINAN, os óbitos podem ser registrados como causados pela doença (óbitos relatados por TB, HIV ou MEN) ou por outras causas (óbitos por morte no trânsito, homicídio, suicídio, entre outros). O banco de dados do SIM (2012 a 2018) centraliza as informações sobre todas as mortes no Estado do Amazonas.

Criamos manualmente um conjunto de dados *gold-standard* que contém todos os indivíduos cujo óbito foi registrado nas bases de dados do SINAN e foi ou não registrado na base de dados do SIM. O *gold-standard* contém 2.382 indivíduos registrados como óbito (MEN = 203; HIV = 936; e TB = 1.243). Destes, 2.183 estavam devidamente cadastrados no SIM (91,6%). Os demais (199), não possuíam pares correspondentes no SIM.

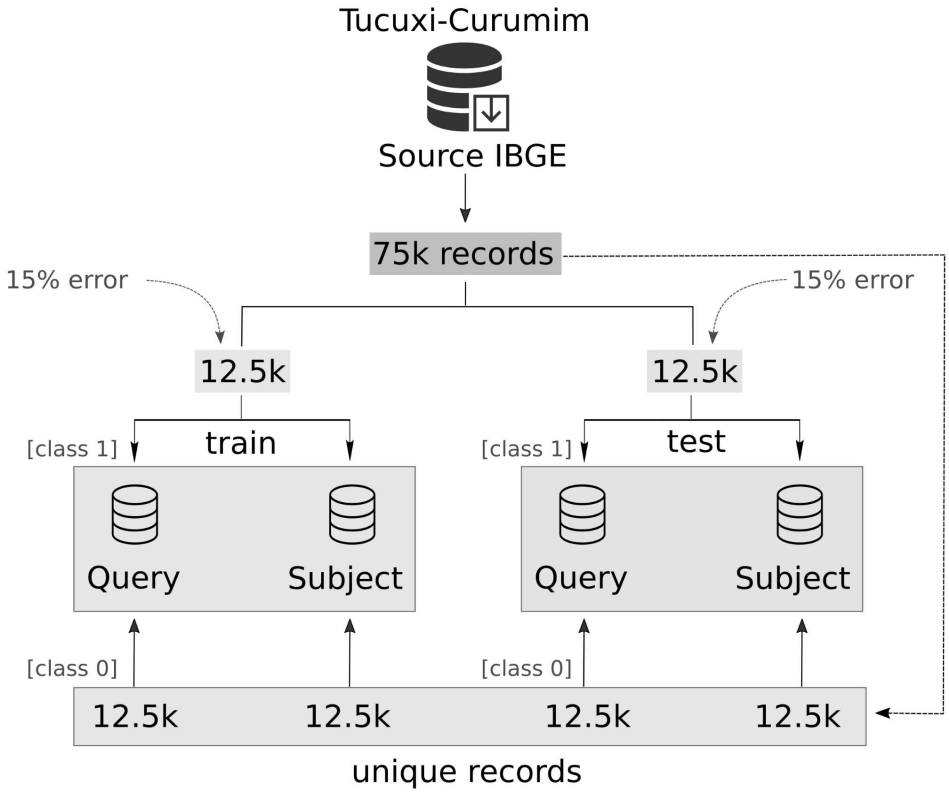
Usando Django Framework [v. 3.0.1], também desenvolvemos a plataforma Tucuxi-Tail (tucuxitail.herokuapp.com), que permite aos usuários verificar manualmente os registros correspondentes encontrados pelo do Tucuxi-BLAST.

4.5. MODELOS DE CLASSIFICAÇÃO E APRENDIZADO DE MÁQUINA

Para a classificação dos RL obtidos pelo BLAST, nós utilizamos as métricas obtidas entre a *query* [*dataset* A] e o *subject* [*dataset* B], foram elas: (1) pontuação de *Bit-score* normalizada; (2) *mismatch* (número de *mismatches*); (3) *gapopen* (número de *gap openings)*; (4) qcovhsp (*query coverage per High Scoring Pairs*); (5) sstart (*start of alignment in subject*); and (6) qstart (*start of alignment in query*). A pontuação de *Bit-score* normalizada foi calculada dividindo o valor de pontuação de *Bit-score* entre a *query* e o *subject* pelo melhor valor de pontuação de *Bit-score* possível para essa mesma *query* (ou seja, o valor de pontuação de *Bit-score* entre a *query* e ela mesma - *itself*).

Modelos de classificação usando aprendizado de máquina, foram desenvolvidos para otimizar a validação do RL, pois o algoritmo BLAST sempre retorna o melhor alinhamento entre as sequências. Esses modelos ajudam na redução da taxa de falso-positivos. Para isso, nós criamos um conjunto de dados simulados com 75k registros usando o programa Tucuxi-Curumim (ver item 4.6). Os dados foram divididos em treino e teste, 25k para cada um dos *subsets*. Depois disso, foram distribuídos aleatoriamente em duas classes: sem correspondência [classe 0] e com correspondência [classe 1]. Por fim, foi imputada a taxa de erro de 15% em cada conjunto de dados, seguindo a mesma distribuição descrita no item 4.3 (Figura 7).

Figura 7: *Dataset* de treino e teste e imputação de erros



Foram gerados 75 mil registros simulados, em seguida divididos em treino e teste (25k), destes 12,5k [classe 0] e 12,5k [classe 1]. Por fim, foi imputado 15% de erros nos registros da classe 1.

Foram criados dois modelos, um baseado em *Random Forest* (RF), e o outro em *Logistic Regression* (LR). O conjunto de treino foi utilizado para treinar os modelos, enquanto o de teste foi utilizado para validar o modelo em dados não

vistos. Utilizamos os algoritmos RF (n_estimators=75; criterion=entropy) e LR (parâmetros *default*) do pacote scikit-learn (PEDREGOSA et al., 2012).

4.6. BANCOS DE DADOS SIMULADOS

Para construir a base de dados simulados contendo 300 milhões de registros brasileiros (300M), desenvolvemos o programa Tucuxi-Curumim (disponível em https://github.com/csbl-usp/tucuxi_curumim). A partir de uma lista de nomes e sobrenomes brasileiros obtida junto ao Instituto Brasileiro de Geografia e Estatística (fonte: censo demográfico oficial de 2010), o programa gera registros aleatórios contendo os seguintes campos: nomes e sobrenomes do indivíduo nomes e sobrenomes da mãe; data de nascimento; e sexo. Para testar a capacidade de processamento do Tucuxi-BLAST, fizemos amostragens aleatórias de conjuntos com 1, 10, 100, 1k, 10k e 100k registros do banco de dados 300M e usamos como *query*. Nestas amostragens, inserimos 10% de erros nos registros a partir dos conjuntos com 100 ou mais registros. Um *script* em Python desenvolvido em nosso laboratório foi usado para introduzir os erros, que por sua vez, são frequentemente encontrados em bancos de dados administrativos, incluindo: substituição e/ou exclusão de um único caractere em nomes/sobrenomes ou de um dígito na data de nascimento, e sobrenomes ausentes (devido a casamento, por exemplo). As taxas de erros foram distribuídas da seguinte forma: 49% com exclusão do nome do meio; 45% com substituição aleatória de um, dois ou três caracteres; 5% de supressão de um ou dois caractere; e 1% exclusão completa da variável (nome da mãe, por exemplo). Essas quantidades de erros são semelhantes às taxas encontradas nas bases de dados epidemiológicas administrativas reais analisadas e descritas no item 5.2 deste trabalho. Por fim, incrementamos a quantidade igual de registros nos *subdatasets* 1, 10, 100, 1k, 10k e 100k que não estão na base de dados de 300M. A inclusão teve a finalidade de inserir ruídos de registros sem par na referência (Figura 10A). Foram utilizados os mesmos conjuntos de dados para o *benchmark*.

4.7. ESTRUTURA COMPUTACIONAL E *BENCHMARK*

Todas as análises foram realizadas em uma estação de trabalho Linux, Intel Core i7-8700, com 32 GB. O *benchmark* foi realizado com as cinco ferramentas de

RL descritas no item 1.4. Foram utilizados os parâmetros *default* conforme a documentação de cada programa. O CIDACS-RL, obrigatoriamente precisa que o *dataset* tenha a coluna de código de município do IBGE, além das 4 variáveis usadas no *gold-standard.* Inserimos essa coluna no *dataset* e imputamos valores de "NA" para toda a coluna. Quando o método de *blocking* foi aplicado, utilizamos o nome e sobrenome do indivíduo para a construção dos *blockings*.

5. RESULTADOS E DISCUSSÃO

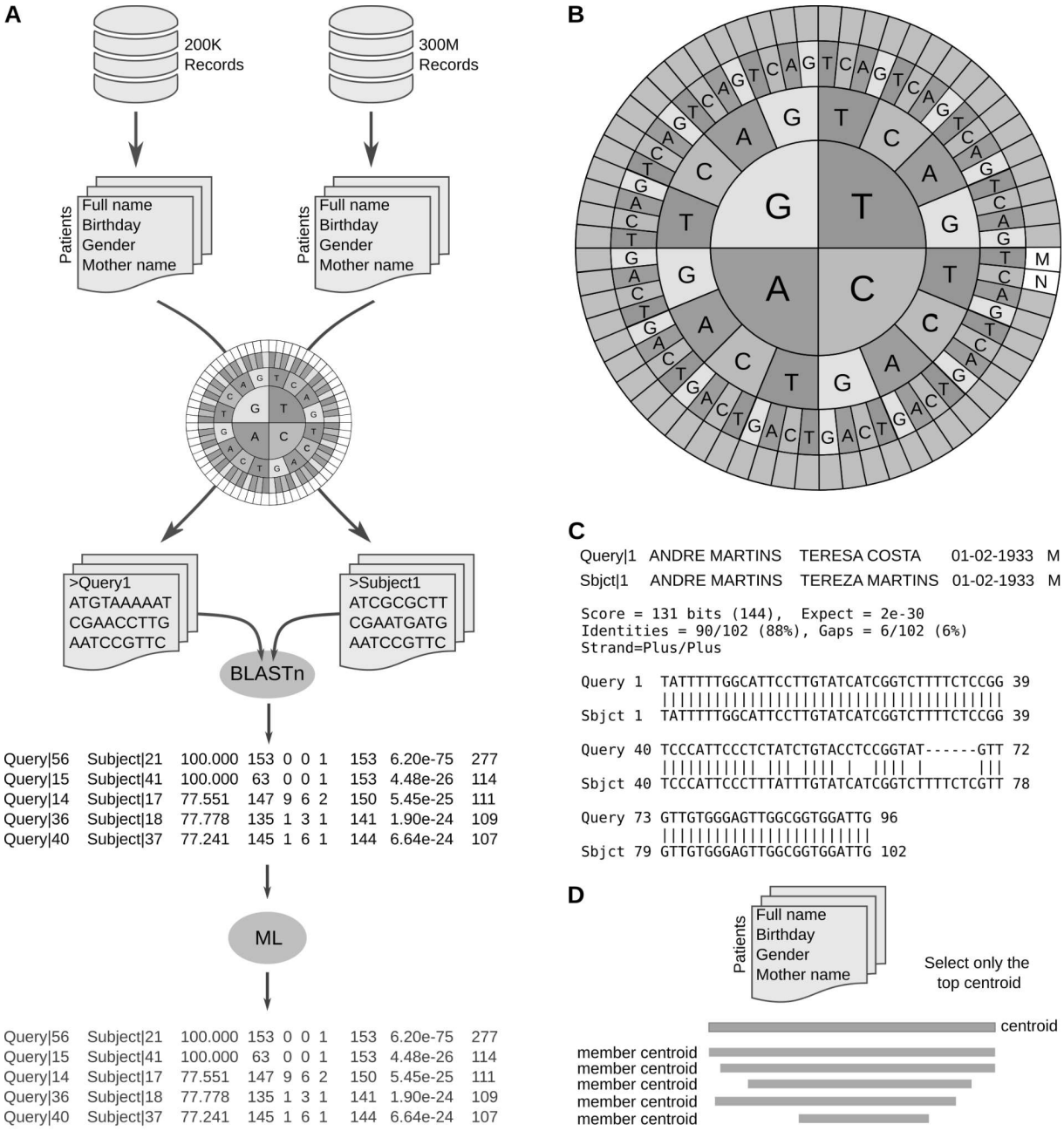
A integração de dados administrativos é uma fonte potencial de informação para pesquisas com objetivo de trazer soluções para problemas públicos, especialmente aqueles relacionados à saúde da população. Esses estudos fornecem subsídios para o desenvolvimento de estratégias de gestão pública baseadas em evidências, que promovem a saúde e o bem-estar da população e reduzem custos a partir do uso racional de recursos (ALI et al., 2019). Muitos países como Inglaterra, Canadá, Austrália, Nova Zelândia e País de Gales investem em iniciativas de sucesso na integração de bases de dados administrativos. Tais iniciativas contam com a construção de grandes centros para integração de dados e desenvolvimento de novas estratégias (CHITTY et al., 2020; EITELHUBER et al., 2018; ELIAS, 2018; TRUDEAU, 2017). Da mesma forma, iniciativas brasileiras têm sido aplicadas com sucesso em grandes bancos de dados governamentais. Este é o caso do Centro de Integração de Dados e Conhecimento para a Saúde - CIDACS. Formado por meio de uma parceria entre o Ministério da Saúde e a Fundação Oswaldo Cruz - Bahia. O CIDACS atualmente conta com uma base de dados que abrange cerca de 55% da população brasileira, com 114 milhões de indivíduos obtidos pela integração de dados administrativos e de saúde (PITA et al., 2018).

Aqui, propomos a utilização do programa Tucuxi-BLAST para RL de bancos de dados de até 300M de registros em um tempo razoável e garantindo a privacidade dos dados sem extenso pré-processamento de dados e dispensando a necessidade de Computação de Alto-Desempenho (HPC, do inglês *High Performance Computing*).

O Tucuxi-BLAST usa um sistema de codificação de DNA *in silico* que atua como uma camada de criptografia dos dados e permite o uso de ferramentas

genômicas como o BLAST (ALTSCHUL et al., 1990), que processa de maneira eficiente grandes conjuntos de dados, incluindo comparações de sequências dissimilares. A visão geral do fluxograma das etapas do Tucuxi-BLAST está resumida na Figura 8.

Figura 8: *Workflow* Tucuxi-BLAST e estruturação de dados



O método utiliza quatro variáveis em comum entre dois conjuntos de dados. Em seguida, é realizada a codificação em DNA *in silico*. O algoritmo BLASTn é utilizado para realizar o alinhamento entre as sequências e, por fim, aprendizagem de máquina é aplicada para classificar o RL como verdadeiro ou

falso (A). Roda de códon utilizada na codificação de DNA (B). Exemplo do resultado do BLAST para RL (C) e módulo para desduplicação de dados (Tucuxi-BW) (D).

5.1. EFICIÊNCIA DO TUCUXI-BLAST COM UM BANCO DE DADOS DE 300 MILHÕES DE REGISTROS

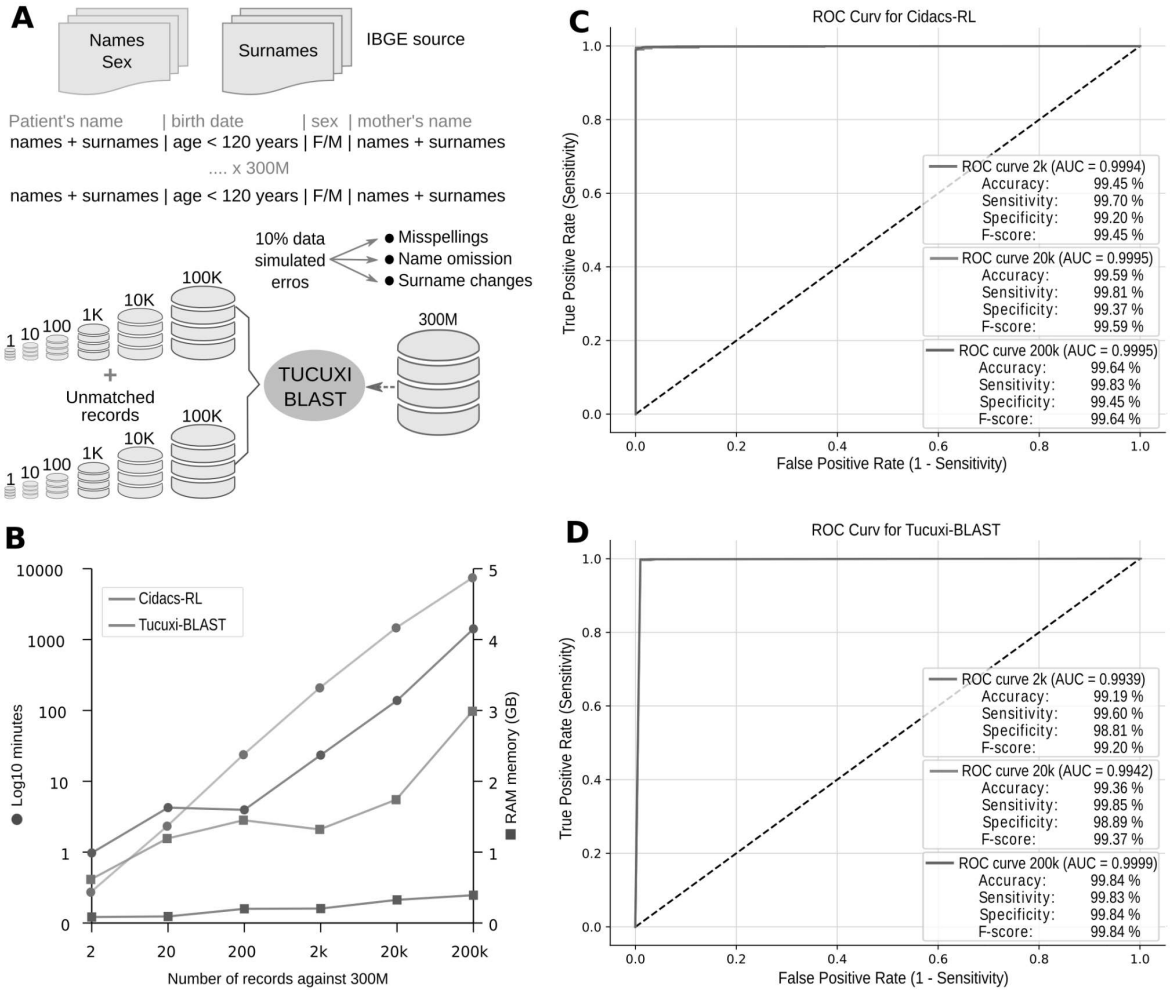
Para demonstrar a robustez, velocidade e precisão do nosso método, testamos o Tucuxi-BLAST usando *datasets* que variam de 2 a 200k registros em relação a uma base de dados de 300M de referência (Figura 9A). As consultas e bancos de dados de referência foram construídos conforme descrito na seção de métodos. Vale ressaltar que metade dos registros de 2 a 200k são ruídos que não existem na base de 300M. Isso serve para demonstrar a capacidade que o Tucuxi-BLAST tem de lidar com ruídos que aumentam exponencialmente o tempo de busca com métodos tradicionais de RL (HARRON et al., 2017). O tempo de processamento e o uso da memória RAM foram monitorados e a média aritmética de 10 repetições foi calculada.

Nós tentamos aplicar os mesmos conjuntos de dados simulados para todos os *softwares* usados no *benchmark* do *gold-standard* (ver item 5.3). No entanto, devido às limitações de memória da máquina utilizada para os testes (32GB), apenas o CIDACS-RL foi capaz de realizar RL em um banco de dados com 300M de registros. Tanto o Tucuxi-BLAST, quanto o CIDACS-RL utilizam em seu método de busca construção de index da base de dados de referência. Para os dois programas foram criados os index previamente e contabilizado o uso de memória e tempo de processamento. O Tucuxi-BLAST levou ~55 minutos e usou 3.8GB de memória para converter todo o *dataset* em sequências de DNA *in silico* e criar os index, enquanto o CIDACS-RL levou ~80 minutos e utilizou 1.7GB para criar os index.

Após a criação dos index, realizamos o RL para os *datasets* 2 a 200k *versus* 300M. Tanto o Tucuxi-BLAST, quanto o CIDACS-RL foram capazes de rodar com baixo consumo de memória, atingindo o pico de 0.4GB e 3GB, respectivamente (Figura 9B). Quando observamos o tempo de processamento, verificamos que nos *datasets* pequenos (2-20), os dois programas tiveram desempenho semelhantes, com desempenho um pouco melhor do CIDACS-RL. A partir do *dataset* de 200 registros, observamos que as curvas de tempo se distanciam conforme aumenta a

quantidade de registros (*query*). O maior tempo de processamento foi observado para o *dataset* de 200k, onde o CIDACS-RL levou 5 dias e 7 horas para finalizar o RL, enquanto o Tucuxi-BLAST levou apenas 23 horas para finalizar o mesmo *dataset*, uma diferença de 4 dias de processamento (Figura 9B).

Figura 9: Dados simulados e desempenho de RL



O Tucuxi-Curumim foi utilizado para gerar todos os dados simulados com os dados obtidos do IBGE (A). O tempo de execução e uso de memória RAM para cada simulação de RL foi avaliado (B). Curva ROC dos dados simulados para o CIDACS-RL e Tucuxi-BLAST (a partir de +2k) (C e D, respectivamente). Todas as simulações foram realizadas em uma estação de trabalho Linux, Intel Core i7-8700, com 32 GB.

Verificamos ainda, as métricas obtidas pelos dois programas. O valor de corte e os parâmetros foram usados conforme descritos pelos autores do CIDACS-RL (BARBOSA et al., 2020). Vale ressaltar que os dados foram projetados para ter erros em 1/10 dos registros e adicional de 50% de ruídos (ou seja, sem correspondência

nos 300M). Ambos os programas obtiveram métricas semelhantes para todos os *datasets* (Figura 9 C e D). Um destaque para o maior conjunto de dados, dentre os 200k, tanto o Tucuxi-BLAST, quanto o CIDACS-RL foram capazes de encontrar 99.832/100k com seus respectivos pares. Entretanto, o Tucuxi-BLAST retornou 159 falsos positivos (FP) e 168 falsos negativos (FN), enquanto o CIDACS-RL apresentou maior número de falsos positivos (555 FP e 168 FN).

O consumo de memória também foi eficiente em ambos os métodos considerando um banco de dados de 300M de registros. Os recursos de memória é uma das principais limitações que muitos programas têm ao processar bancos de dados de médio e grande porte, como pode ser observado em nosso *benchmark* com o *gold-standard* (ver item 5.3).

O Tucuxi-BLAST apresentou excelentes métricas nos dados simulados comparado com o *software* CIDACS-RL, ferramenta está consolidada na integração de grandes conjuntos de dados reais (Projeto CIDACS/Fundação Oswaldo Cruz - Bahia). Nosso método exige baixo consumo de memória RAM no RL de todos os *datasets* testados e ainda é 5x mais rápido do que o CIDACS-RL.

5.2. RL DE BANCOS DE DADOS REAIS DO ESTADO DO AMAZONAS

Além dos dados simulados trabalhamos com dados reais fornecidos pela Fundação de Vigilância em Saúde do Amazonas, Manaus, Brasil. As informações são de origem médicas e socioeconômicas de pacientes com diagnóstico confirmado de tuberculose (TB), HIV/AIDS ou meningite (MEN) entre os anos de 2012 a 2017. O banco de dados de notificação de óbito (SIM) (2012 a 2018) foi usado aqui como referência para RL. Todos os registros foram verificados manualmente para produzir um banco de dados *gold-standard* no qual o *status* do RL entre cada registro e a referência é conhecido. Apenas os casos fatais para cada base de dados foram incluídos no banco de dados *gold-standard*.

Caracteres que apresentam similaridade fonética são comumente trocados durante a inserção manual das informações. Inspirados no código genético degenerado (CRICK et al., 1961), que por sua vez, é composto por três bases de DNA, por exemplo “ATG”. Projetamos a codificação para que na terceira posição do códon haja uma base variável entre caracteres foneticamente semelhantes em

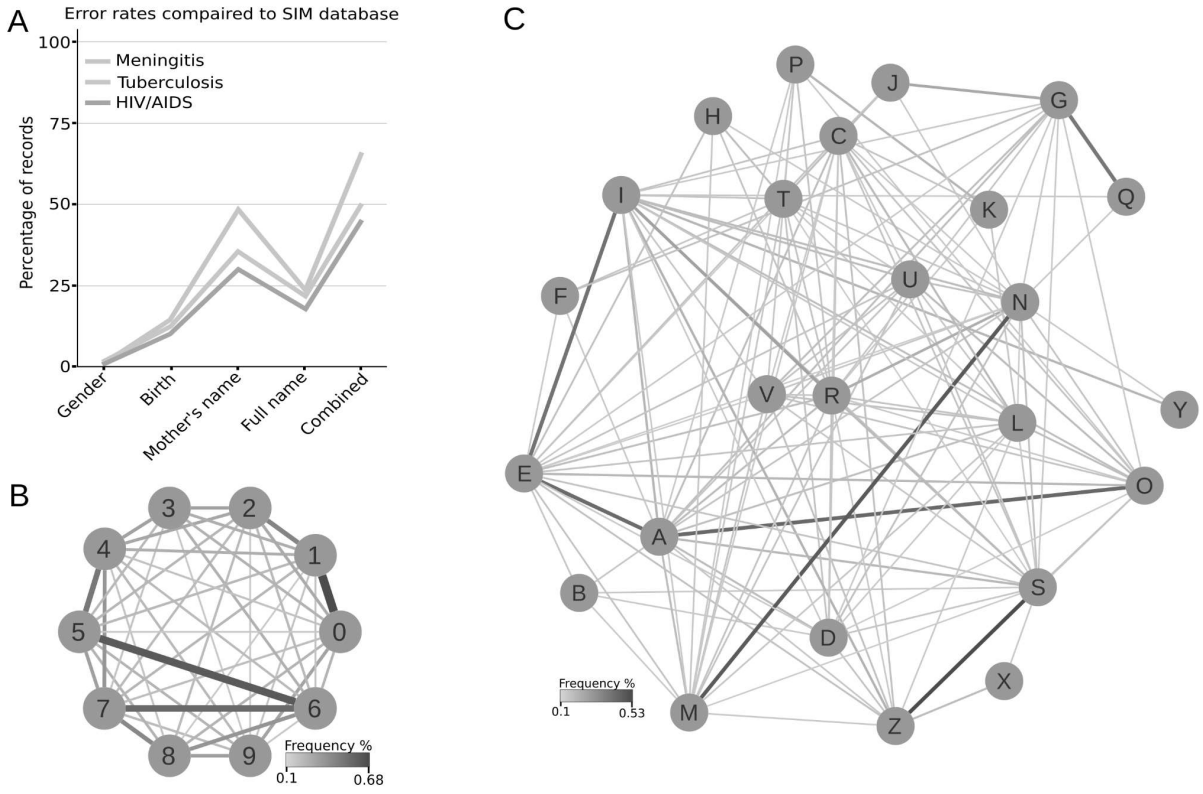
português. Desse modo, letras como S e Z, frequentemente encontradas no mesmo fonema (Tereza e Teresa, por exemplo), são distantes em apenas uma base, ATA e ATC respectivamente (Figura 8B). Essa organização melhorou os *scores* do algoritmo BLAST em casos de substituições de uma única letra, como por exemplo “Souza” e “Sousa”, que são ocorrências/erros ortográficos comuns em dados administrativos. Essa abordagem também dispensa o pré-processamento com algoritmos fonéticos como SOUNDEX, utilizado por alguns *softwares* de RL para padronização de dados (CAMARGO; COELI, 2015; ENAMORADO; FIFIELD; IMAI, 2019). No entanto, nem sempre essa padronização condiz com a realidade diatópica do idioma português, criando codificação que diverge da realidade dos dados (JORDÃO; ROSA, 2012).

Os resultados do BLAST mostraram que pelo menos um erro foi encontrado em aproximadamente ~51% dos registros com *linkage* (Figura 10A). As incompatibilidades de caracteres (*mismatch*), foi o tipo de erro mais frequente encontrado (34%), enquanto os indels (inserção/deleção) foram encontrados em ~25% dos registros. Ao lado dos erros tipográficos, as omissões/inserções do nome do meio foram frequentemente encontradas principalmente nos nomes das mães, o que torna esse campo a informação mais discordante. Uma variação maior nos sobrenomes das mães era esperada, já que é um problema relatado na literatura (DUSETZINA et al., 2014). Mudanças no nome da família são comuns em mulheres devido à tradição de casamento em que as esposas passam a usar o sobrenome do marido. Considerando todas as variáveis, o banco de dados de MEN mostrou a taxa mais alta em comparação com HIV e TB. Um dos motivos que explica as maiores taxas de erros em MEN é a maior presença de recém-nascidos no banco de dados, cujo diagnóstico é informado antes do registro oficial de nascimento.

A taxa de substituição entre caracteres alfanuméricos foi calculada usando registros que mostram apenas *mismatch* nos resultados do BLAST, ou seja, registros com o mesmo comprimento. As principais alterações encontradas nos números da data de nascimento foi a troca de “1” por “0” (0,68%), seguido de “5” por “6” (0,60%), e “6” por “7” (0,53%) (Figura 10B). As principais trocas de caracteres alfabéticos ocorreram entre letras que compartilham fonética semelhante no português brasileiro, e/ou confusão ortográfica como N-M; S-Z; G-Q, I-E (Figura 10C).

A inconsistência de dados entre bancos de dados é uma das principais barreiras que precisam ser superadas durante o processo de RL (JUNIOR et al., 2018). Assim, a maioria das ferramentas de RL necessita de pré-processamento de algoritmos fonéticos, como SOUNDEX e Metaphone (ENAMORADO; FIFIELD; IMAI, 2019). Esses algoritmos foram originalmente desenvolvidos para o idioma inglês e adaptados para o português (JORDÃO; ROSA, 2012; MARCELINO, 2015). Embora o tratamento fonético melhore o RL, esses algoritmos ainda são sensíveis a tipos incorretos que não foram previstos pelo pré-processamento fonético, ocorrendo em perdas das métricas de *linkage* das bases de dados.

Figura 10: Distribuição de erros encontrados nos dados do SINAN



Taxas de erros encontradas no SINAN em comparação com o banco de dados SIM (A). As redes representam a taxa de substituição dentro dos números (B) e letras (C). Os caracteres representam os nós, enquanto as arestas são as frequências de substituições entre eles.

Apesar da alta frequência de erros encontrada nos dados reais, o Tucuxi-BLAST foi capaz de realizar RL exibindo acurácia, sensibilidade e especificidade maior que 98% para todas as bases de dados. O algoritmo BLAST é otimizado para lidar com a variação de sequências que inclui substituição de

nucleotídeos, exclusão e inserção de segmentos. Essa variabilidade é comumente encontrada em sequências de genes ortólogos pertencentes a diferentes espécies. O uso do BLAST para realizar RL foi proposto pela primeira vez por (HONG et al., 2008). Os autores compararam diferentes sistemas de codificação com um, dois e quatro nucleotídeos, sendo o último apresentando os melhores resultados. No entanto, aumentar o comprimento da codificação representa um aumento nos requisitos de disco rígido e memória e também impacta o desempenho direto do BLAST, o que pode representar uma demanda maior de processamento para o uso do BLAST para RL entre bancos de dados grandes. Os autores não implementaram uma ferramenta para que pudéssemos comparar com nossa nova abordagem. Além disso, implementamos melhoria pela utilização de códons de DNA que torna o BLAST mais rápido, sem perder a sensibilidade.

O algoritmo BLAST é muito eficiente em consultas contra bilhões de sequências, é o algoritmo de busca mais utilizado para sequências biológicas (PERKEL, 2021). A comparação de bancos de dados com centenas de bilhões de sequências pode ser verificada com melhor precisão com o BLASTn. No entanto, outras ferramentas foram desenvolvidas, como USEARCH (EDGAR, 2010) e VSEARCH (ROGNES et al., 2016) que podem ser mil vezes mais rápidas que o BLASTn. No entanto, o ganho em velocidade é compensado por uma perda na sensibilidade e também um uso elevado de memória RAM por essas ferramentas. Embora apenas o BLASTn tenha sido usado como método de pesquisa no Tucuxi-BLAST, a abordagem de codificação de DNA torna acessível um grande conjunto de ferramentas genômicas que podem ser aplicadas em estudos de banco de dados não genômicos.

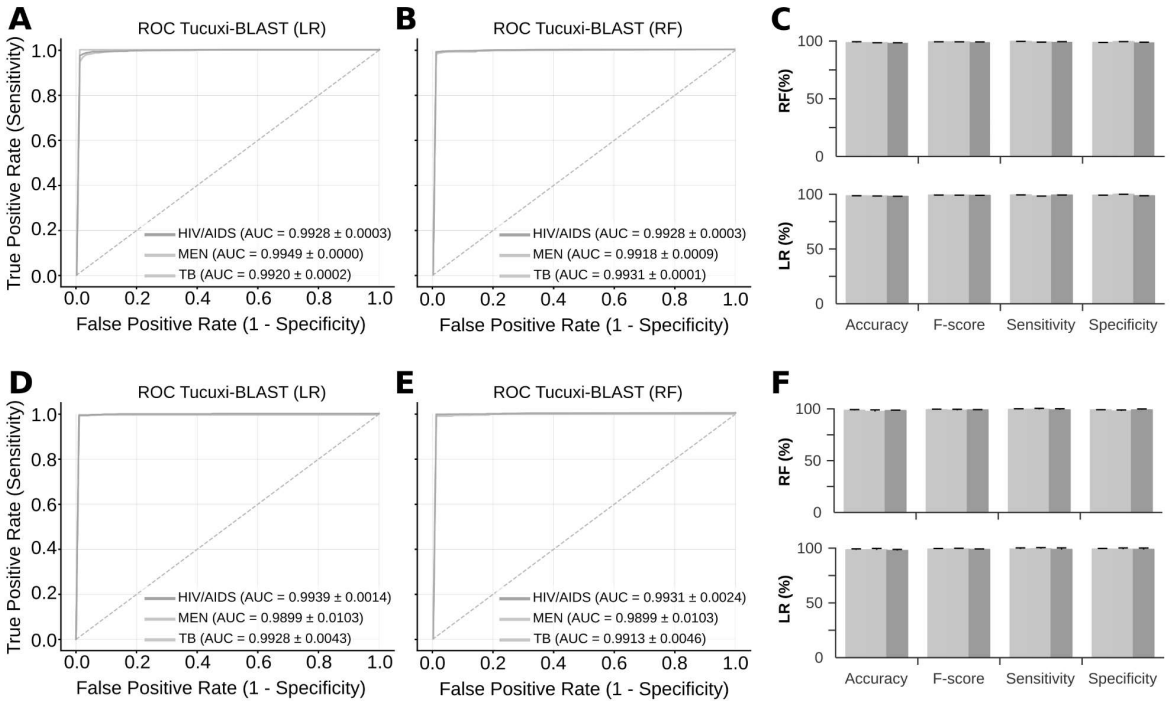
5.3. MÉTRICAS DE AVALIAÇÃO E *BENCHMARK*

Vários estudos de RL usam diferentes algoritmos derivados dos métodos de Jaro-Winkler (WINKLER, 1990) e Damerau-Levenshtein (LEVENSHTEIN, 1966). Cada um tem prós e contras e os resultados são variáveis, dependendo do tipo de dado analisado. Neste trabalho, comparamos *softwares* e métodos de RL que são *open source,* com suporte ativo e não requerem infraestrutura computacional de alto desempenho (HPC), o que frequentemente não está disponível para pequenos

grupos de pesquisa e gestores de Departamentos de Vigilância Epidemiológica. Todas as análises foram realizadas usando dados reais com os mesmos bancos de dados como entrada (ou seja, *gold-standard* contra mortes notificadas - SIM, como referência) na mesma máquina.

Para o modo de produção do nosso método, definimos os dados de treinamento do classificador com dados simulados, como descrito nos métodos. Essa abordagem permite que possamos aplicar em diferentes conjuntos de dados, sem que haja *overfitting* (sobreajuste) no modelo de classificação, uma vez que, o mesmo paciente pode estar em diferentes bases de dados do SINAN. Ainda, essa abordagem permitirá que o Tucuxi-BLAST tenha diferentes conjuntos de dados de treinamentos além do descrito aqui, de modo a incluir novos campos, tais como: endereço ou código do município de origem do paciente, por exemplo. Os resultados do classificador usando dados simulados (Figura 11A, B e C), demonstram que as métricas são muito próximas do classificador com dados reais (Figura 11D, E e F), este por sua vez, obtido por *cross-validation*, que permite obter a média das métricas com diferentes conjuntos de treino e teste para todo o conjunto de dados.

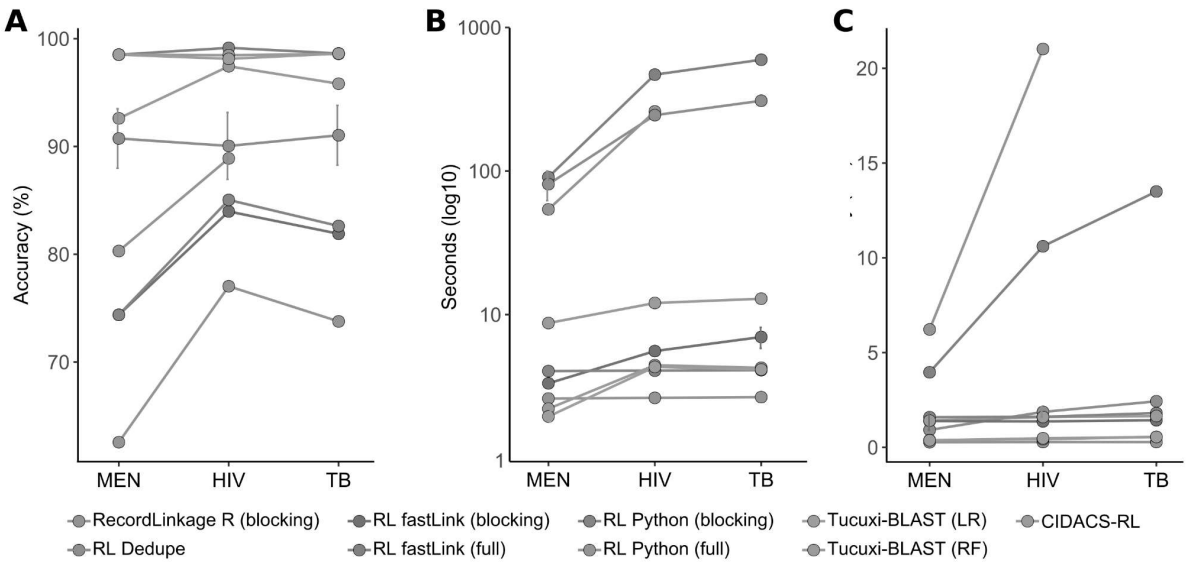
Figura 11. Métricas de desempenho com dados de treinamento simulados e com dados reais



Curvas ROC do Tucuxi-BLAST para o RL de dados simulados (A e B), em comparação com as métricas obtidas *por cross-validation* dos dados reais do SINAN de doenças de meningite (MEN), AIDS/HIV (HIV) e tuberculose (TB) (D e E) usando *Logistic regression* - LR e *Random Forest* - RF com suas respectivas métricas (C e F).

Nos dados de meningite obtivemos F-score = 99,23% e AUC = 0,9949; tuberculose com F-score = 99,25% e AUC = 0,9920; e, finalmente, HIV com F-score = 99,14% e AUC = 0,9928 (Figura 11 A e B). Os classificadores RF e LR apresentaram acurácia média de 98,53% e 98,42%, respectivamente. Caso o banco de dados a ser linkado requeira melhores métricas de especificidade, recomendamos o classificador LR. Para bancos de dados gerais que consideram o equilíbrio de todas as métricas (principalmente balanceamento de sensibilidade/especificidade), o classificador RF pode ser aplicado (Figura 11C). O RL foi realizado de acordo com a documentação de cada método e os *cut-offs default* utilizados. O Tucuxi-BLAST apresentou desempenho de processamento semelhante aos métodos que usam *blocking*, além disso, a precisão é muito superior. Inclusive superando os métodos com as abordagens RL *full* (*linkage* envolvendo todas as combinações possíveis) (Figura 12 A). O programa fastLink (R), junto com o Tucuxi-BLAST obtiveram as melhores métricas do *benchmark*. Na contramão, quando observamos o tempo de processamento, o Tucuxi-BLAST é 100x mais rápido e consome menos memória que o fastLink (Figura 12 B e C).

Figura 12. *Benchmark* com ferramentas de RL em dados reais do SINAN



Acurácia de cada método do *benchmark* (A). Tempo de processamento (em $\log_{10}$ segundos) (B). Consumo de memória RAM em GB (C). *As execuções do pacote RecordLinkage R com o método *full linkage* não foram possíveis para o banco de dados TB, quando usando a máquina de testes mencionada nos métodos, devido à extrapolação de memória disponível.

De modo geral as menores métricas obtidas foram observadas pelos programas que utilizam *blocking*. Isso já era esperado, uma vez que, o método de *blocking* agiliza o processamento por redução da quantidade de comparações entre as bases de dados, consequentemente compromete as métricas obtidas (SAYERS et al., 2016). Uma abordagem mista tem sido implementada na utilização de *blocking* e *full linkage*. Esta abordagem tem se mostrado eficiente em muitos trabalhos, obtendo boas métricas e baixo consumo computacional (BARBOSA et al., 2020; PITA et al., 2018). O CIDACS-RL utiliza essa abordagem mista de *blocking* e *full linkage*, mesmo assim o Tucuxi-BLAST apresentou melhores métricas e foi mais rápido no RL do *gold-standard*.

A principal limitação que consideramos no uso do Tucuxi-BLAST para bancos de dados contendo mais de 300M de registros é o armazenamento temporário em disco rígido. A codificação em DNA *in silico* e o processamento das etapas do BLAST requerem um armazenamento temporário de pelo menos 2 vezes a soma dos bancos de dados de consulta e referência, ou seja, se ambas entradas de banco de dados somarem o total de 5 GB, o Tucuxi-BLAST exigirá pelo menos 10 GB de espaço em disco rígido para prosseguir com o RL. Apesar de ser uma limitação para armazenamento temporário, isso não ocorre com uso de memória RAM (Figura 9B). Além disso, a alocação de mais armazenamento é uma barreira fácil de ser resolvida, enquanto a alocação de memória RAM é limitada à infraestrutura computacional disponível e acaba por demandar mais recursos financeiros para aquisição.

Em um cenário ideal, existe um consenso na área RL da utilização de todas as combinações possíveis de registros (*full linkage*) (ALI et al., 2019; DUSETZINA et al., 2014). No entanto, isso exige alta demanda computacional, muitas vezes disponível apenas em grandes centros de pesquisas. Isso impossibilita trabalhos de pequenos grupos de pesquisa sem recursos para obtenção desses equipamentos e mão de obra especializada em processamento HPC.

O Tucuxi-BLAST apresenta excelente desempenho para RL de grandes bases de dados sem a necessidade de grandes recursos computacionais. Isso em parte deve-se ao uso de ferramentas “ômicas” otimizadas e amplamente usadas em biologia molecular e bioinformática como o BLAST. O método aqui implementado para a codificação de sequências de DNA, juntamente com a matriz de códons adaptada, bem como, com a manutenção da privacidade dos dados (não oferecida pela grande maioria do programas de RL atuais), colocam o Tucuxi-BLAST como uma ferramenta promissora na exploração de pequenas e grandes bases de dados. Além disso, o nosso programa não necessita de recursos computacionais proibitivos e não exige etapas dispendiosas de pré-processamento com algoritmos fonéticos e remoção de *missing data*.

6. CONCLUSÃO

Nossos resultados mostraram um desempenho de alta resolução de *record linkage* usando o sistema de codificação de DNA *in silico* e o algoritmo BLAST. A ferramenta desenvolvida aqui foi capaz de superar erros de digitação e erros tipográficos em bancos de dados administrativos. O Tucuxi-BLAST não necessita da instalação de nenhuma dependência, assim dispensa qualquer conhecimento prévio de gerenciamento de *softwares* e suas respectivas permissões de sistemas Linux. Além disso, nossa abordagem revela vantagens em termos de tempo e acurácia, mesmo usando apenas 4 variáveis, em comparação com outras ferramentas RL de código aberto. A integração de grandes *datasets* pode ser realizada utilizando uma máquina *desktop* com configuração padrão do mercado. Em adição, o sistema de codificação de DNA permite o anonimato completo das informações pessoais, garantindo o sigilo em todo o processo de RL. As informações dos dados codificadas em DNA, possibilitam ainda, a exploração com outras ferramentas genômicas disponíveis não descritas aqui, como por exemplo buscas de sequências *motifs* e polimorfismos (padrões de variação encontrados numa população).

7. REFERÊNCIAS

1000 GENOMES PROJECT CONSORTIUM et al. A global reference for human genetic variation. **Nature**, v. 526, n. 7571, p. 68–74, 1 Oct. 2015.

ALI, M. S. et al. Administrative data linkage in brazil: potentials for health technology assessment. **Frontiers in pharmacology**, v. 10, p. 984, 23 Sep. 2019.

ALTSCHUL, S. F. et al. Basic local alignment search tool. **Journal of Molecular Biology**, v. 215, n. 3, p. 403–410, 5 Oct. 1990.

ANAVY, L. et al. Data storage in DNA with fewer synthesis cycles using composite DNA letters. **Nature Biotechnology**, v. 37, n. 10, p. 1229–1236, 9 Sep. 2019.

BARBOSA, G. C. G. et al. CIDACS-RL: a novel indexing search and scoring-based record linkage system for huge datasets with high accuracy and scalability. **BMC Medical Informatics and Decision Making**, v. 20, n. 1, p. 289, 9 Nov. 2020.

BRASIL. **Portaria de Consolidação nº 4 de setembro de 2017. Consolidação das normas sobre os sistemas e os subsistemas do Sistema Único de Saúde**, 2017.

CAMARGO, K. R. DE; COELI, C. M. Going open source: some lessons learned from the development of OpenRecLink. **Cadernos de Saúde Pública**, v. 31, n. 2, p. 257–263, 2015.

CEZE, L.; NIVALA, J.; STRAUSS, K. Molecular digital data storage using DNA. **Nature Reviews. Genetics**, v. 20, n. 8, p. 456–466, 2019.

CHITTY, K. M. et al. Australian Suicide Prevention using Health-Linked Data (ASHLi): Protocol for a population-based case series study. **BMJ Open**, v. 10, n. 5, p. e038181, 11 May 2020.

COX, M.; ELLSWORTH, D. Application-Controlled Demand Paging for Out-of-Core Visualization. **NASA Ames Research Center**, 1997.

CRICK, F. H. et al. General nature of the genetic code for proteins. **Nature**, v. 192, p. 1227–1232, 30 Dec. 1961.

DATASUS. **Departamento de Informática do Sistema Único de Saúde**. Disponível em: <https://datasus.saude.gov.br/sobre-o-datasus/>. Acesso em: 27 apr. 2021.

DAVIS, J. Microvenus. **Art Journal**, v. 55, n. 1, p. 70–74, Mar. 1996.

DA SILVA, V. M. F.; MARTIN, A. R. Amazon River Dolphin. In: **Encyclopedia of marine mammals**. Elsevier, 2018. p. 21–24.

DE BRUIN, J. Python Record Linkage Toolkit: A toolkit for record linkage and

duplicate detection in Python. **Zenodo**, 2019.

DIMENSIONAL INSIGHT. **Understanding Hospitals' Data Governance Maturity and Challenges**. Disponível em: <https://www.dimins.com/wp-content/uploads/2019/10/s-understanding-hospitals-data-governance-maturity-and-challenges-01012020.pdf>. Acesso em: 18 nov. 2019.

DUSETZINA, S. B. et al. **Linking Data for Health Services Research: A Framework and Instructional Guide**. Rockville (MD): Agency for Healthcare Research and Quality (US), 2014.

EDGAR, R. C. Search and clustering orders of magnitude faster than BLAST. **Bioinformatics**, v. 26, n. 19, p. 2460–2461, 1 Oct. 2010.

EITELHUBER, T. W. et al. Western Australia unveils its new data linkage system. **International Journal for Population Data Science**, v. 3, n. 3, 12 Nov. 2018.

ELIAS, P. The UK administrative data research network: its genesis, progress, and future. **The Annals of the American Academy of Political and Social Science**, v. 675, n. 1, p. 184–201, Jan. 2018.

ENAMORADO, T. E. D.; FIFIELD, B.; IMAI, K. Using a Probabilistic Model to Assist Merging of Large-Scale Administrative Records. **The American political science review**, v. 113, n. 2, p. 353–371, May 2019.

ERLICH, Y.; ZIELINSKI, D. DNA Fountain enables a robust and efficient storage architecture. **Science**, v. 355, n. 6328, p. 950–954, 3 Mar. 2017.

EXTANCE, A. How DNA could store all the world's data. **Nature**, v. 537, n. 7618, p. 22–24, 1 Sep. 2016.

FANG, X.-H. et al. Factors influencing completion of treatment among pulmonary tuberculosis patients. **Patient preference and adherence**, v. 13, p. 491–496, 10 Apr. 2019.

FASSLER, J.; COOPER, P. **BLAST Glossary. 2011 Jul 14. In: BLAST® Help [Internet].** Bethesda (MD): National Center for Biotechnology Information (US), 2011.

FLORES, P. A. C.; DA SILVA, V. M. F.; FETTUCCIA, D. DE C. Tucuxi and guiana dolphins. In: **Encyclopedia of marine mammals**. Elsevier, 2018. p. 1024–1027.

GREENGARD, S. The future of data storage. **Communications of the ACM**, v. 62, n. 4, p. 12–12, 20 Mar. 2019.

GREGG, F.; EDER, D. **Dedupe: A python library for accurate and scalable fuzzy matching, record deduplication and entity-resolution.** Disponível em: <https://github.com/dedupeio/dedupe>. Acesso em: 16 jun. 2020.

HARRON, K. et al. Utilising identifier error variation in linkage of large administrative data sources. **BMC Medical Research Methodology**, v. 17, n. 1, p. 23, 7 Feb. 2017.

HONG, Y. et al. **Record Linkage as DNA Sequence Alignment Problem**. In: 6TH INTERNATIONAL CONFERENCE ON VERY LARGE DATABASE. CVLDB, 2008. Disponível em: <http://nike.psu.edu/publications/qdb08.pdf>. Acesso em: 27 jan. 2020.

JORDÃO, C. C.; ROSA, J. L. G. Metaphone-pt_BR: The Phonetic Importance on Search and Correction of Textual Information. In: GELBUKH, A. (Ed.). **Computational linguistics and intelligent text processing**. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. v. 7182p. 297–305.

JUNIOR, A. A. G. et al. Building the National Database of Health Centred on the Individual: Administrative and Epidemiological Record Linkage - Brazil, 2000-2015. **International Journal for Population Data Science**, v. 3, n. 1, 14 Nov. 2018.

KUO, A. et al. A hadoop/mapreduce based platform for supporting health big data analytics. **Studies in Health Technology and Informatics**, v. 257, p. 229–235, 2019.

LANDER, E. S. et al. Initial sequencing and analysis of the human genome. **Nature**, v. 409, n. 6822, p. 860–921, 15 Feb. 2001.

LEE, H. H. et al. Terminator-free template-independent enzymatic DNA synthesis for digital information storage. **Nature Communications**, v. 10, n. 1, p. 2383, 3 Jun. 2019.

LEINONEN, R. et al. The sequence read archive. **Nucleic Acids Research**, v. 39, n. Database issue, p. D19-21, Jan. 2011.

LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. **Soviet physics doklady**, p. 707, 1966.

MARCELINO, D. **soundexBR: Phonetic-Coding For Portuguese**. Disponível em: <https://rdrr.io/cran/SoundexBR/man/soundexBR.html>. Acesso em: 23 jun. 2020.

MICHELSON, M.; KNOBLOCK, C. Learning Blocking Schemes for Record Linkage. **American Association for Artificial Intelligence**, 2006.

NEWCOMBE, H. B. Record linking: the design of efficient systems for linking records into individual and family histories. **American Journal of Human Genetics**, v. 19, n. 3 Pt 1, p. 335–359, May 1967.

ORGANICK, L. et al. Random access in large-scale DNA data storage. **Nature**

Biotechnology, v. 36, n. 3, p. 242–248, 19 Feb. 2018.

OUSSOUS, A. et al. Big Data technologies: A survey. **Journal of King Saud University - Computer and Information Sciences**, Jun. 2017.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. **The Journal of Machine Learning Research**, 2012.

PERKEL, J. M. Ten computer codes that transformed science. **Nature**, v. 589, n. 7842, p. 344–348, Jan. 2021.

PITA, R. et al. On the accuracy and scalability of probabilistic data linkage over the brazilian 114 million cohort. **IEEE journal of biomedical and health informatics**, v. 22, n. 2, p. 346–353, 2018.

REINSEL, D.; GANTZ, J.; RYDNING, J. The Digitization of the World From Edge to Core. **International Data Corporation**, 1 Nov. 2018.

REIS-SANTOS, B. et al. Tuberculosis in Brazil and cash transfer programs: A longitudinal database study of the effect of cash transfer on cure rates. **Plos One**, v. 14, n. 2, p. e0212617, 22 Feb. 2019.

ROGNES, T. et al. VSEARCH: a versatile open source tool for metagenomics. **PeerJ**, v. 4, p. e2584, 18 Oct. 2016.

SÁNCHEZ-GONZÁLEZ, G. et al. Prediction of dengue outbreaks in Mexico based on entomological, meteorological and demographic data. **Plos One**, v. 13, n. 8, p. e0196047, 6 Aug. 2018.

SARIYAR, M.; BORG, A. The recordlinkage package: detecting errors in data. **The R journal**, v. 2, n. 2, p. 61, 2010.

SAYERS, A. et al. Probabilistic record linkage. **International Journal of Epidemiology**, v. 45, n. 3, p. 954–964, 2016.

SHARMA, R.; AGARWAL, P.; MAHAPATRA, R. P. Evolution in big data analytics on internet of things: applications and future plan. In: TANWAR, S.; TYAGI, S.; KUMAR, N. (Eds.). **Multimedia big data computing for iot applications: concepts, paradigms and solutions**. Intelligent systems reference library. Singapore: Springer Singapore, 2020. v. 163p. 453–477.

SHIPMAN, S. L. et al. Molecular recordings by directed CRISPR spacer acquisition. **Science**, v. 353, n. 6298, p. aaf1175, 29 Jul. 2016.

SIMS, D. et al. Sequencing depth and coverage: key considerations in genomic analyses. **Nature Reviews. Genetics**, v. 15, n. 2, p. 121–132, Feb. 2014.

SINAN. **Sistema de Informação de Agravos de Notificação**. Disponível em:

<https://portalsinan.saude.gov.br/o-sinan>. Acesso em: 27 apr. 2021.

SMITH, T. F.; WATERMAN, M. S. Identification of common molecular subsequences. **Journal of Molecular Biology**, v. 147, n. 1, p. 195–197, 25 Mar. 1981.

SRA. **Sequence Read Archive**. Disponível em: <https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi>. Acesso em: 27 apr. 2021.

TAKAHASHI, C. N. et al. Demonstration of End-to-End Automation of DNA Data Storage. **Scientific Reports**, v. 9, n. 1, p. 4998, 21 Mar. 2019.

TRUDEAU, R. Social data linkage environment. **International Journal for Population Data Science**, v. 1, n. 1, 13 Apr. 2017.

WINKLER, W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. 1990.

ZHANG, Y. et al. Predicting seasonal influenza epidemics using cross-hemisphere influenza surveillance data and local internet query data. **Scientific Reports**, v. 9, n. 1, p. 3262, 1 Mar. 2019.