UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS AGRÁRIAS

DEPARTAMENTO DE ENGENHARIA DE PESCA

PROGRAMA DE PÓS-GRADUAÇÃO EM BIOTECNOLOGIA DE RECURSOS NATURAIS

MATHIAS COELHO BATISTA

DESENVOLVIMENTO DE UMA FERRAMENTA PARA ANÁLISE DE DADOS DE SEQUENCIAMENTO NGS DE BIBLIOTECAS DE SCFVS SELECIONADOS POR PHAGE DISPLAY

FORTALEZA

2023

MATHIAS COELHO BATISTA

DESENVOLVIMENTO DE UMA FERRAMENTA PARA ANÁLISE DE DADOS DE SEQUENCIAMENTO NGS DE BIBLIOTECAS DE SCFVS SELECIONADOS POR PHAGE DISPLAY

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Biotecnologia de Recursos Naturais do Centro de Ciências Agrárias da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Biotecnologia de Recursos Naturais. Área de Concentração: Biologia Estrutural e Computacional.

Orientador: Prof. Dr. Marcos Roberto Lourenzoni.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

B337d Batista, Mathias Coelho.
Desenvolvimento de uma ferramenta para análise de dados de sequenciamento NGS de bibliotecas de scFvs selecionados por phage display / Mathias Coelho Batista. – 2023.
80 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências Agrárias, Programa de Pós-Graduação em Biotecnologia de Recursos Naturais, Fortaleza, 2023.
Orientação: Prof. Dr. Marcos Roberto Lourenzoni.

1. Ferramenta Computacional. 2. Phage Display. 3. NGS. 4. Anticorpo. I. Título.

CDD 660.6

MATHIAS COELHO BATISTA

DESENVOLVIMENTO DE UMA FERRAMENTA PARA ANÁLISE DE DADOS DE SEQUENCIAMENTO NGS DE BIBLIOTECAS DE SCFVS SELECIONADOS POR PHAGE DISPLAY

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Biotecnologia de Recursos Naturais do Centro de Ciências Agrárias da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Biotecnologia de Recursos Naturais. Área de Concentração: Biologia Estrutural e Computacional.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Marcos Roberto Lourenzoni (Orientador)
Fundação Oswaldo Cruz (Fiocruz)

Dr. Antônio Edson Rocha Oliveira
Universidade de Fortaleza (UNIFOR)

Prof. Dr. Nicholas Costa Barroso Lima
Universidade Federal do Ceará (UFC)

À minha mãe.

AGRADECIMENTOS

Agradeço primeiramente à minha mãe, Glória, pelo apoio incondicional durante toda a minha vida. Seu amor e orientação foram fundamentais para que eu enfrentasse cada obstáculo de cabeça erguida.

Aos meus irmãos, Matheus e Marcos, por todas as nossas vivências.

Às minhas queridas tias, Elisângela, Lidiane e Tiá, sempre dispostas a ajudar em momentos difíceis e por quem tenho grande carinho.

Ao meu orientador, Prof. Dr. Marcos Roberto Lourenzoni, pela excelente orientação, que se estendeu para além das atividades acadêmicas, por ter me propiciado a oportunidade de desenvolver pesquisa junto a pessoas geniais, por ter me permitido vislumbrar o mundo do empreendedorismo e da inovação, e pela paciência e compreensão.

Aos participantes da banca examinadora da defesa de Dissertação, Dr. Antonio Edson Rocha Oliveira e Prof. Dr. Nicholas Costa Barroso Lima, pelo tempo disponibilizado e pelas valiosas contribuições.

Aos participantes da banca examinadora do exame de Qualificação, Prof. Dr. Marcelo de Macedo Brígido e Dr. Raphael Trevizani Roque de Oliveira, por terem se disposto e pelas importantes considerações.

À Profa. Dra. Andrea Maranhão, por todo o conhecimento repassado de forma tão inspiradora.

Aos colegas de laboratório, Ana Júlia, Ana Virgínia, Alison, Maísa e Natália pela troca de experiências. Agradeço especialmente ao amigo Cássio Pinheiro pela amizade e suporte sempre presentes.

À Amanda Silveira, Nailson Lima, e Rhonaldo Parente pelas importantes contribuições para o desenvolvimento deste trabalho.

Aos companheiros Jennifer, Júlia, Lucilânia, Josiane, Jamile e Anderson, pela dedicação diária a um propósito maior.

Aos amigos Wilker, Raquel, Thais, Marcilene e Arley, pelo suporte nos momentos mais fundamentais.

À minha namorada, Liandra Éllen, por estar sempre presente sendo amiga e companheira, por me apoiar, compreender e incentivar.

Aos colegas da DIALOG, em especial ao Dirceu, Jack, Jorge, Rafael, Rogério, Tâmara, Vinícius, Yhago e Ynayara, do incomparável time da GETEC.

Ao Programa de Pós-Graduação em Biotecnologia de Recursos Naturais e todo o seu corpo docente, pelo conhecimento de valor inestimável repassado. Agradeço ao Prof. Dr. André Luís Coelho e ao Iuri Saraiva, coordenador e secretário deste programa de pós-graduação, por todo suporte administrativo.

À Universidade Federal do Ceará por ter possibilitado minha formação no ensino superior até aqui.

À Fundação Oswaldo Cruz, pelos recursos e infraestrutura disponibilizados para o desenvolvimento deste trabalho.

À Fundação Cearense de Apoio Científico e Tecnológico (FUNCAP) pela bolsa de estudos durante o Mestrado.

À Fundação para o Desenvolvimento Científico e Tecnológico em Saúde (Fiotec) pelos recursos financeiros que possibilitaram o início deste projeto.

À todos que contribuíram direta ou indiretamente para o desenvolvimento deste trabalho.

"Descobri como é bom chegar quando se tem paciência. E para se chegar, onde quer que seja, aprendi que não é preciso dominar a força, mas a razão. É preciso, antes de mais nada, querer."
(Amyr Klink, 2005, p. 11.)

RESUMO

Os anticorpos monoclonais (mAbs) são proteínas globulares capazes de reconhecer, se ligar e elicitar resposta imunológica contra alvos moleculares específicos e são largamente aplicados no tratamento de diversas doenças. Dentre as técnicas que podem ser aplicadas no desenvolvimento de mAbs, o phage display se destaca por sua capacidade de selecionar fragmentos de mAbs capazes de se ligar a um alvo específico. Ao ser associada a tecnologias de sequenciamento de nova geração (NGS), o phage display se torna uma ferramenta poderosa por possibilitar que seja acompanhado com grande precisão quais e como cada fragmento foi selecionado. O ATTILA é um pipeline computacional capaz de identificar quais candidatos foram mais enriquecidos ao longo da seleção, através da análise de dados provenientes do sequenciamento NGS de bibliotecas de phage display. Embora eficiente no desempenho dessa função, o ATTILA possui processos de instalação, configuração e uso pouco amigáveis aos usuários, além de não ser capaz de aproveitar bem o poder computacional disponível. Por conta disso, este projeto tem como objetivo desenvolver uma ferramenta computacional a partir do ATTILA com melhor desempenho, usabilidade e novas funções. O desenvolvimento resultou em uma nova ferramenta (ATTILA 2.0) que inclui uma interface gráficas para inserção dos parâmetros de entrada e visualização dos resultados, processamento otimizado para melhor aproveitamento dos recursos computacionais, e processo de análise reestruturado para possibilitar a análise simultânea de múltiplos rounds e execução de parte do processamento através de computação em nuvem. Além disso, foi criado um instalador único para facilitar e dar celeridade ao processo de instalação e foi feita a validação da ferramenta. O ATTILA 2.0 é mais amigável à utilização, tanto pela existência de uma interface gráfica, como pela facilitação da instalação pelo instalador. Através de implementação de multiprocessamento, o ATTILA 2.0 se mostrou capaz de utilizar melhor os recursos computacionais disponíveis. A reestruturação do código permitiu a analisar bibliotecas obtidas do sequenciamento NGS de múltiplos ciclos de seleção por phage display, assim como a divisão do processamento entre a máquina do usuário e uma API hospedada em servidor da Fiocruz, além de possibilitar alimentação de um banco de dados que permitirá integrações futuras com outras ferramentas. A validação foi feita através da análise de dados NGS provenientes do sequenciamento de bibliotecas de seleção por phage display que teve como alvo uma das alças da proteína CD20.

Palavras-chave: ferramenta computacional; phage display; ngs; anticorpo.

ABSTRACT

Monoclonal antibodies (mAbs) are globular proteins capable of recognizing, binding and eliciting an immune response against specific molecular targets, being widely applied in the treatment of various diseases. Among the techniques that can be applied in the development of mAbs, phage display stands out for its ability to select mAb fragments capable of binding to a specific target. When associated with next-generation sequencing (NGS) technologies, phage display becomes a powerful tool because it allows precise tracking of which and how each fragment was selected. ATTILA is a computational pipeline capable of identifying which candidates were most enriched during selection, through the analysis of data from NGS sequencing of phage display libraries. Although efficient in performing this function, ATTILA has a unfriendly installation, configuration, and usage processes, and is unable to fully utilize available computational power. Therefore, this project aims to develop a computational tool based on ATTILA with better performance, usability, and new functions. The development resulted in a new tool (ATTILA 2.0) that includes a graphical interface for input parameter insertion and result visualization, optimized processing for better use of computational resources, and a restructured analysis process to enable simultaneous analysis of multiple rounds and execution of part of the processing through cloud computing. In addition, a single installer was created to facilitate and speed up the installation process, and the tool was validated. ATTILA 2.0 is more user-friendly, both due to the existence of a graphical interface and the facilitation of installation by the installer. Through the implementation of multiprocessing, ATTILA 2.0 was shown to better utilize available computational resources. The code restructuring allowed for the analysis of libraries obtained from NGS sequencing of multiple rounds of phage display selection, as well as the division of processing between the user's machine and an API hosted on a Fiocruz server, and the feeding of a database that will allow future integrations with other tools. Validation was performed through the analysis of NGS data from the sequencing of phage display selection libraries targeting one of the loops of the CD20 protein.

Keywords: computational tool; phage display; ngs; antibody.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Anvisa	Agência Nacional de Vigilância Sanitária
API	Interface de Programação de Aplicação
ATTILA	*AuTomated Tool for Immunoglobulin Analysis*
BD	Banco de Dados Relacional
CAR	Receptor de Antígeno Quimérico
CDR	Região Determinante de Complementariedade
ddNTP	Didesoxirribonucleotídeo Trifosfato
dNTP	Desoxirribonucleotídeo Trifosfato
epPCR	*error-prone Polymerase Chain Reaction* / reação em cadeia da polimerase propensa ao erro
Fab	Fragmento de Ligação ao Antígeno
Fc	Fragmento Cristalizável
FDA	*Food and Drug Administration*
Fv	Fragmento Variável
GEPeSS	Grupo de Pesquisa em Engenharia de Proteínas e Soluções para a Saúde
HACA	*Human Anti-Chimeric Antibody*
HAMA	*Human Anti-Mouse Antibody*
mAb	anticorpo monoclonal
MER	Modelo Entidade-Relacionamento
NGS	Sequenciamento de Nova Geração
RAM	Memória de Acesso Aleatório
scFv	Fragmento Variável de Cadeia Única
VH	domínio variável de cadeia pesada
VHH	nanocorpo
VL	domínio variável de cadeia leve

SUMÁRIO

1 INTRODUÇÃO

Os anticorpos são proteínas globulares com a distinta capacidade de reconhecer, se ligar e elicitar resposta imunológica contra alvos moleculares específicos que vão desde moléculas de DNA a peptídeos. Uma de suas primeiras aplicações documentadas, visando imunização passiva, ocorreu ainda em 1890, quando Behring e Kitasato mostraram que um animal poderia ser protegido de um inóculo fatal de difteria através do soro de outro animal que já havia sido exposto e sobrevivido ao patógeno (LLEWELYN *et al.*, 1992). Um ano mais tarde, o mesmo princípio seria usado para tratar um menino acometido da mesma doença. Os soros obtidos se mostraram específicos, ou seja, o soro usado para tratar tétano não se mostrou eficiente no tratamento de difteria e vice-versa.

Ao se mostrar eficaz, a soroterapia passou a ser aplicada no tratamento de animais e seres humanos, sendo usada até hoje, como é o caso dos soros antiofídicos (ALANGODE *et al.*, 2020; LIN *et al.*, 2022). Contudo, em muitos pacientes, a administração desse tipo de tratamento levou a reações anafiláticas conhecidas como doença do soro, e que muitas vezes levavam à morte (SRIAPHA *et al.*, 2022). O desenvolvimento de técnicas de purificação de anticorpos humanos a partir do soro de pacientes convalescentes em 1944 sanou esse problema, com exceção das doenças em que o soro não poderia ser obtido de forma segura. Desde então, diversas técnicas foram desenvolvidas visando a obtenção de anticorpos humanos e humanizados e seus fragmentos.

O desenvolvimento de novas técnicas levou a um crescimento considerável no número de anticorpos humanos ou humanizados aprovados para terapia por organizações como *Food and Drug Administration* (FDA) e Agência Nacional de Vigilância Sanitária (Anvisa), dos 20 anticorpos com maior volume de venda no mundo em 2019, 18 são humanos ou humanizados, ultrapassando US$ 91 bilhões em vendas (MULLARD *et al.*, 2021). A técnica de phage display foi utilizada na criação de um dos primeiros Abs aprovados pela FDA, e hoje, associada a tecnologia de sequenciamento de nova geração, (NGS) tem se mostrado eficiente na proposição de novos anticorpos e fragmentos para tratamento e diagnóstico de doenças (KROHN *et al.*, 2022).

2 REFERENCIAL TEÓRICO

2.1 Aplicação terapêutica de anticorpos

Nas últimas décadas observou-se um aumento expressivo no desenvolvimento de soluções farmacológicas, baseadas em anticorpos, visando o tratamento de doenças como câncer, esclerose múltipla, asma e artrite reumatoide (DAI *et al.*, 2021).

Anticorpos podem ser policlonais ou monoclonais. Os anticorpos policlonais constituem um conjunto de anticorpos que têm como característica reconhecer regiões diferentes do mesmo alvo, desencadeando também respostas diversas. Já os anticorpos monoclonais (mAbs) reconhecem o alvo sempre na mesma região e têm os mesmos mecanismos de elicitação, tendo resultados mais previsíveis quando aplicados tanto em ferramentas de diagnóstico quanto em tratamentos (SINGH *et al.*, 2018).

Um dos primeiros grandes saltos no desenvolvimento de anticorpos se deu com a criação da técnica de hibridomas. A técnica de hibridomas voltada para a produção de anticorpos foi desenvolvida por Milstein e Khöler, dando-lhes o prêmio Nobel em 1984. A técnica se baseia na fusão de linfócitos produtores de um anticorpo monoclonal e células de mieloma, a partir da qual surge uma linhagem capaz de produzir um anticorpo monoclonal por tempo indefinido.

No início, os anticorpos de origem não-humana foram majoritariamente utilizados na indústria farmacêutica, contudo, observou-se o desencadeamento de reações adversas nos pacientes pelo fato da proteína ser exógena. Esse conjunto de reações deletérias oriundas do próprio sistema imunológico do indivíduo foi chamado de *Human Anti-Mouse Antibody* (HAMA), responsável também pela diminuição da eficácia do tratamento ao afetar as propriedades farmacocinéticas do anticorpo em questão (HWANG *et al.*, 2005). Visando a diminuição dos efeitos deletérios, foram desenvolvidos os anticorpos quiméricos. Nestes, as porções constantes murinas são substituídas por humanas, mantendo-se as regiões variáveis murinas a fim de conservar a especificidade do anticorpo. Contudo, embora em menor grau, ainda se observaram reações imunológicas indesejadas denominadas *Human Anti-Chimeric Antibody* (HACA) (HWANG *et al.*, 2005). A partir desse cenário foram desenvolvidas diversas técnicas de humanização de anticorpos, tendo como base o objetivo de manter a menor porção murina possível na estrutura (SAFDARI *et al.*, 2013). Tanto para a diminuição da imunogenicidade indesejada, como para a otimização de fatores de interesse, como aumento da afinidade e termoestabilidade, o entendimento da estrutura e propriedade dos anticorpos é fundamental.

2.2 Anticorpos: estrutura e função

Os anticorpos (Abs) são proteínas capazes de reconhecer e se ligar a alvos moleculares específicos. Abs geralmente são formados por repetições de um dímero constituído por uma cadeia leve e uma cadeia pesada, associadas por pontes dissulfeto (CHIU *et al.*, 2019). São conhecidos 5 tipos de cadeia pesada com funções distintas no sistema imunológico: IgA, IgD, IgE, IgG e IgM. Os Abs que contém cadeias do tipo IgA e IgM têm a capacidade de formar dímeros e pentâmeros por possuírem uma cadeia de junção (J). Existem dois tipos de cadeia leve, kappa (κ) e lambda (λ), que possuem propriedade similares (CHIU *et al.*, 2019). Os Abs com cadeias do tipo IgG, mais estudados e aplicados, possuem uma cadeia pesada composta de 3 domínios constantes (CH1, CH2 e CH3) e 1 domínio variável (VH). A cadeia leve de Abs IgG é formada de 1 domínio constante (CL) e um domínio variável (VL) (Figura 2).

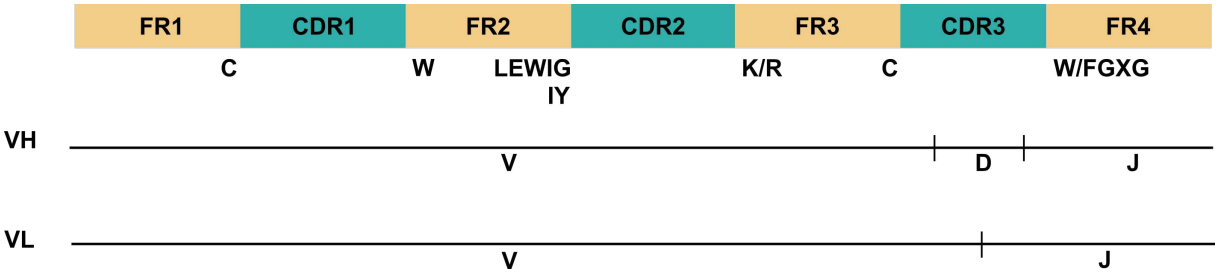
Ao serem clivados por proteases como a papaína, a estrutura dos Abs se divide em uma Fragmento Cristalizável (Fc) e um Fragmento de Ligação ao Antígeno (Fab). Enquanto a porção Fc tem a função de elicitar resposta imunológica e possui relativamente pouca variação de sequência de aminoácidos e estrutura tridimensional dentre os diferentes Abs, a porção Fab tem grande variabilidade, possuindo a função de reconhecer os antígenos (CHIU *et al.*, 2019).

A porção Fab engloba os domínios domínio variável de cadeia pesada (VH) e domínio variável de cadeia leve (VL), que tem aproximadamente 125 e 110 resíduos, respectivamente (Figura 1). As sequências codificadoras dos domínios VH e VL são formadas nas células-B através da combinação de segmentos gênicos conhecidos como V, D e J. O processo de recombinação dos segmentos gênicos V(D)J é fundamental na geração da variabilidade que garante a diversidade dos anticorpos (ALBERTS *et al.*, 2002). Por ter uma região de recombinação a mais, devido ao segmento D, os domínios VH tendem a ter maior variabilidade. Outro importante gerador de variabilidade são as mutações somáticas que ocorrem durante o processo de maturação das células-B nas regiões codificadoras dos domínios variáveis e que são capazes de aumentar a afinidade e especificidade do anticorpo a ser gerado pelo antígeno.

São regiões específicas dos domínios variáveis, chamadas de Regiões Determinantes de Complementariedade (CDRs), que concentram a maior variabilidade de sequência e estrutura, além de serem as principais responsáveis pelo reconhecimento do alvo (WU; KABAT, 1970). As CDRs se apresentam em forma de loops formando o paratopo complementar ao antígeno, e são circundadas por regiões pouco variáveis, denominadas de *frameworks*. Por conta da pouca variabilidade das regiões framework, é possível identificá-las, assim como às CDRs, através de

um processo de numeração baseado em padrões conhecidos na sequência de aminoácidos, do alinhamento múltiplo com sequências de referencia e através da aplicação de Modelo Ocuto de Markov (ABHINANDAN *et al.*, 2008; DUNBAR; DEANE, 2015). Existem diferentes esquemas de numeração, sendo os mais aplicados o de Kabat, Chothia e IMGT.

Figura 1 – Representação da organização interna dos domínios variáveis, destacando-se as regiões framework (amarelo) e as CDRs (verde). Em seguida, tem-se a identificação dos aminoácidos conservados nas sequências de domínios variáveis de IgG humana, representados pelo código de uma letra, em que X representa um aminoácido qualquer. Ressalta-se que o padrão WGXG é específico de VH enquanto FGXG é específico de VL. A baixo estão representados os segmentos gênicos (V, D e J) que se combinam para formar as sequências codificadoras dos domínios VH e VL, respectivamente.



Fonte: Adaptado de Silva (2016)

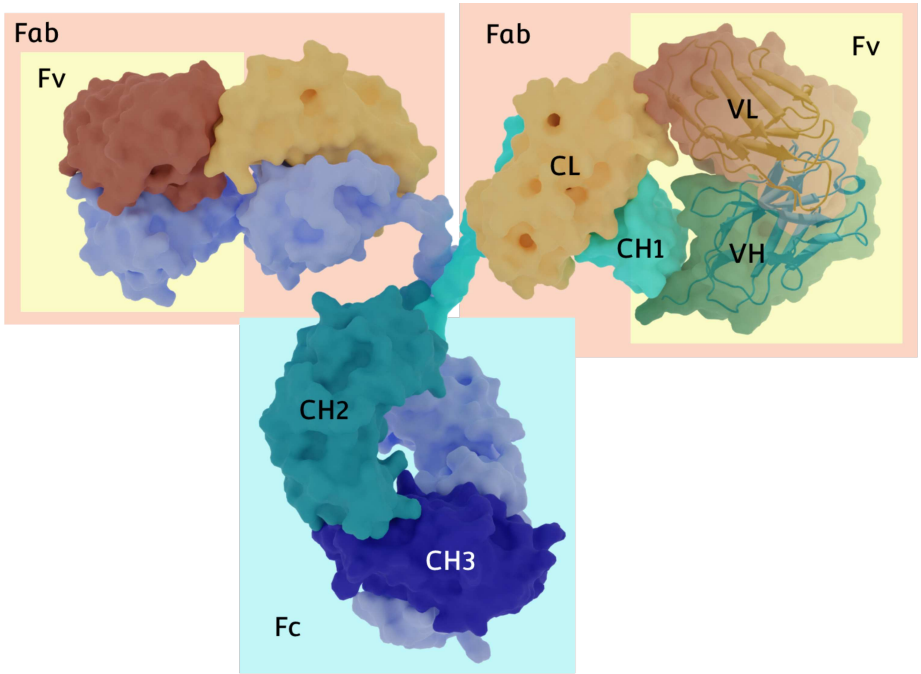
O esquema de numeração de Kabat foi desenvolvido observando-se a variação nas sequências de aminoácidos de domínios variáveis alinhadas (KABAT *et al.*, 1977). Já o esquema de Chothia foi estabelecido tendo como diretriz aspectos conformacionais dos loops formados pelas CDRs (CHOTHIA; LESK, 1987). Assim, foram observados resíduos que flanqueiam as CDRs e que determinam sua conformação, uma vez que, exceto pela CDR-H3, há pouca variação de tamanho (classe) e conformação das CDRs. As conformações observadas para as diferentes classes de CDRs são denominadas por Chothia e Lesk (1987) de estruturas canônicas. Já o esquema IMGT tem como finalidade ser capaz de numerar de forma equivalente não apenas domínios variáveis de anticorpos, mas também de receptores de células-T e outras proteínas da superfamília das imunoglobulinas (LEFRANC, 1997).

A numeração tem grande importância no entendimento da estrutura dos domínios variáveis, por possibilitar a identificação de regiões importantes para a manutenção da estrutura dos loops, como os resíduos da zona de Vernier (FOOTE; WINTER, 1992; MAKABE *et al.*, 2008). Consequentemente, costuma ter grande impacto na seleção de posições candidatas a mutações sítio-dirigidas e no processo de humanização de anticorpos.

Uma vez que cada Fragmento Variável (Fv), VH e VL, mantêm a estrutura e a função mesmo quando separados do resto do Ab, foram desenvolvidos Fragmentos Variáveis de Cadeia

Única (scFvs) formados pela associação de VH e VL através de um peptídeo de ligação (*linker*). Os scFvs retém a capacidade de reconhecimento de um Ab integral, sendo, contudo, mais simples de manipular e produzir in vitro, além de ter maior capacidade de difusão em sistemas biológicos como o corpo humano (BATRA *et al.*, 2002). Por conta disso, o desenvolvimento de scFvs e derivados, como conjugados fármaco-scFv, têm se tornado um foco da indústria farmacêutica e alimentar visando a criação de novos biofármacos, testes de diagnóstico e, mais recentemente, terapias celulares (MUÑOZ-LÓPEZ *et al.*, 2022; LIMA, 2022; OLIVEIRA, 2020).

Figura 2 – Representação da estrutura de um anticorpo do tipo IgG (PDB: 1IGT). No retângulo azul está destacado o fragmento cristalizável (Fc) formado de dois pares dos domínios constantes CH2 e CH3, sendo um par de cada cadeia pesada. Nos retângulos rosa estão delimitadas as porções Fab, constituídas pelos domínios VL e CL das cadeias leves e pelos domínios VH e CH1 das cadeias pesadas. As porções Fv, ressaltadas pelos retângulos amarelos, são formadas pelos domínios variáveis das cadeias pesadas e leves. Ainda no retângulo amarelo da direita, é possível observar que o paratopo complementar ao antígeno é formado pelos loops das cadeias variáveis.



Fonte: O autor

2.3 Técnicas de produção de scFvs para desenvolvimento de anticorpos monoclonais

Phage display é uma das técnicas mais utilizadas e eficientes na descoberta de novos anticorpos não-humanos e humanos (Figura 3), e se baseia na construção de uma grande biblioteca de fagos que expressam proteínas capazes de reconhecer alvos, podendo ser scFvs, Fabs, dentre outros, conectados a uma de suas proteínas estruturais. No caso do fago mais

comumente utilizado, o M13, a proteína III serve de âncora ao scFv de modo que o scFv permanece acessível ao solvente e capaz de reconhecer o alvo (SMITH *et al.*, 1985). Diversos *rounds* de seleção são feitos com esses fagos contra uma molécula alvo imobilizada em uma superfície. Aqueles com maior afinidade resistem às etapas de lavagem sendo selecionados, amplificados através de incubação com bactérias, e submetidos ao próximo *round* de seleção.

O desenvolvimento das técnicas de Sequenciamento de Nova Geração (NGS) possibilitam o sequenciamento com alta cobertura das bibliotecas antes, durante e após a seleção por *phage display* (DIAS-NETO *et al.*, 2009), permitindo avaliar com precisão quais sequências únicas foram enriquecidas, que definimos como sequência, além de permitir acessar quais os aminoácidos mais frequentes para cada posição das proteínas selecionadas.

2.4 Tecnologias de sequenciamento

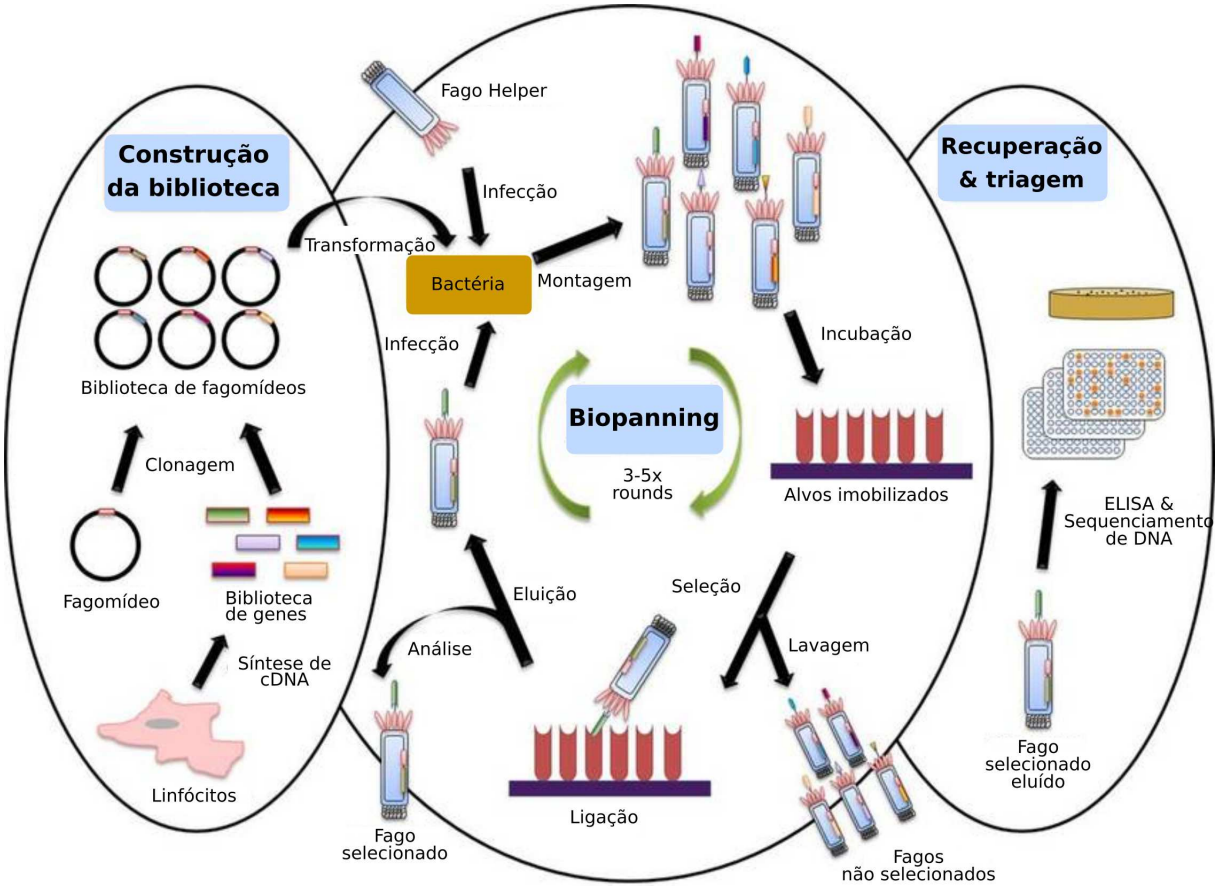
2.4.1 Sanger

O desenvolvimento da primeira geração de tecnologias de sequenciamento foi marcado pela técnica desenvolvida por SANGER *et al.* (1977). Neste método, um mesmo fragmento de DNA de fita simples é amplificado em quatro reações realizadas em recipientes separados. Cada recipiente, deve conter uma mistura dos quatro tipos de Desoxirribonucleotídeo Trifosfato (dNTP) necessários à amplificação e um tipo de Didesoxirribonucleotídeo Trifosfato (ddNTP) (em menor quantidade), que, por não ter a hidroxila 3', impede a continuação da amplificação. Deste modo, são gerados fragmentos de diferentes tamanhos, sempre terminados no ddNTP em questão (HEATHER *et al.*, 2016). Ao fazer correr os produtos das quatro reações em paralelo em um gel de poliacrilamida, é possível deduzir a sequência de nucleotídeos. Essa tecnologia foi aperfeiçoada através da utilização de detecção baseada em fluorometria, permitindo a execução das quatro reações em um mesmo recipiente. Além disso, a criação de uma técnica de eletroforese baseada em capilaridade permitiu a automação do sequenciamento de Sanger, capaz de obter sequências menores que uma kilobase.

2.4.2 Segunda geração de sequenciadores

O desenvolvimento da segunda geração de sequenciadores foi marcado pelo surgimento da capacidade de executar diversos sequenciamentos em paralelo, diminuindo drasticamente o tempo necessário para se obter a sequência de interesse.

Figura 3 – Esquematização de seleção de fragmentos de mAbs baseada na técnica de *Phage Display*. O processo se inicia com a geração de uma biblioteca de sequências codificadoras de fragmentos de mAbs, como Fabs ou scFvs, que podem ser obtidos por diversos meios, como a partir de DNA de linfócitos de doadores. Cada sequência codificante de um fragmento de mAb é inserida em um fagomídeo, que por sua vez é inserido em bactérias através de transformação genética. Pela ação de um fago *Helper*, os fagomídeos são ativados e ocorre a geração de fagos no interior das bactérias transformadas, idealmente com cada um associado a um fragmento de mAbs. Após incubação dos fagos gerados com o alvo imobilizado, aqueles que possuem fragmentos de mAbs com afinidade ao alvo são selecionados através de um processo de lavagem e, então, eluídos e replicados através da infecção de bactérias. O ciclo de seleção se repete geralmente por 3 a 5 vezes, e, ao fim, os fragmentos de mAbs selecionados são analisados por técnicas como sequenciamento ou ELISA.



Fonte: Adaptado de (LEOW *et al.*, 2017)

Os primeiros sequenciadores de segunda geração disponíveis para comercialização foram desenvolvidos pela empresa 454 Life Sciences, liderada por Jonathan Rothburg, mais tarde incorporada pela empresa Roche (HEATHER *et al.*, 2016). O seu funcionamento se baseia na detecção do pirofosfato liberado durante a amplificação de sequências de DNA (Figura 4). Nessa técnica, apenas um dos quatro nucleotídeos possíveis é adicionado, seguindo-se da verificação da liberação do pirofosfato, que, caso seja detectado, indica que o nucleotídeo em questão foi adicionado à sequência. A detecção se dá através da conversão do pirofosfato em ATP pela enzima ATP sulfurilase; o ATP, por sua vez, ativa a enzima Luciferase, que emite luz detectável.

Esse processo é repetido com diferentes nucleotídeos a cada rodada até que não se tenha mais liberação do pirofosfato, indicando o fim da amplificação (RONAGHI *et al.*, 2001).

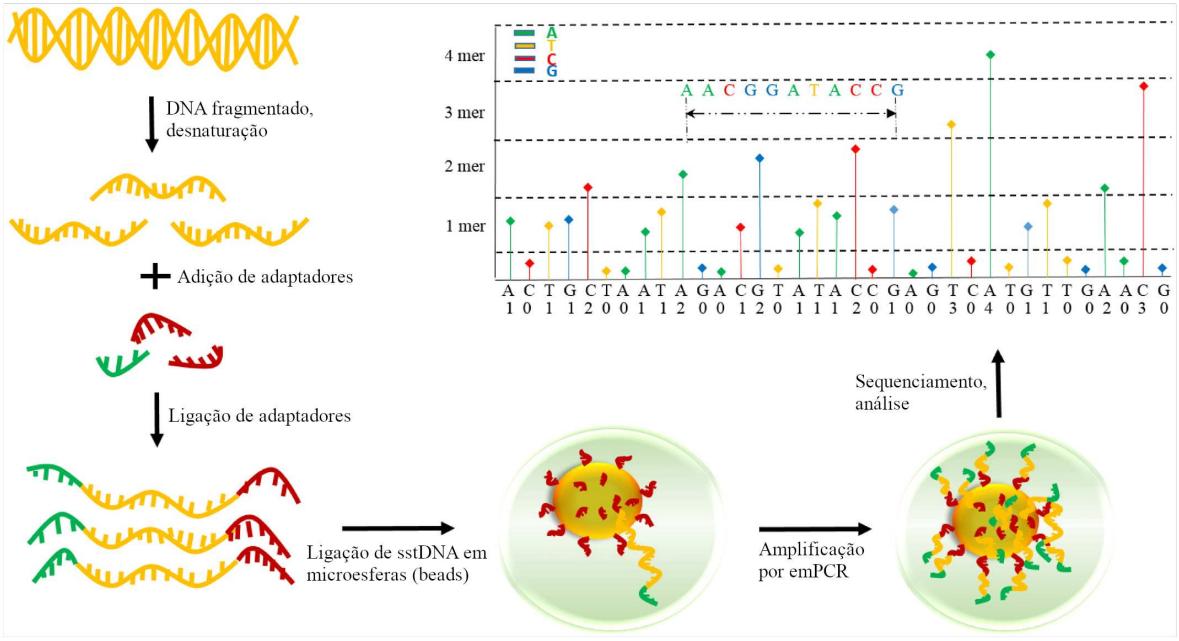
Outro avanço empregado nesse tipo de sequenciador foi a utilização de microesferas (beads) às quais os fragmentos a serem sequenciados são ligados através de adaptadores localizados em suas extremidades (WONG *et al.*, 2019). Os fragmentos associados às suas respectivas esferas são submetidos a uma amplificação por PCR que popula cada esfera com seu respectivo sequência. As esferas são transferidas para placas com micropoços nas quais são submetidas ao processo de amplificação já descrito.

Recentemente a técnica de sequenciamento de segunda geração empregada pelos aparelhos Illumina, desenvolvida pela empresa Solexa, passou a dominar grande parte dos laboratórios em detrimento de métodos mais antigos como o da Roche 454. Nos equipamentos Illumina, os fragmentos são sintetizados com um par de adaptadores diferentes, um em cada extremidade (METZKER *et al.*, 2009) (Figura 5). À princípio, os fragmentos são colocados sobre uma placa, onde uma de suas extremidades se liga através da sequência adaptadora à uma sequência complementar já imobilizada. Em seguida, a outra extremidade, que contém o outro adaptor, se liga à placa, de modo que a fita de DNA se estabelece num formato de ponte sobre a placa. As fitas imobilizadas passam, então, por vários ciclos de amplificação de modo que se forma numa mesma região um agrupamento das mesmas sequências. Após essa etapa, as fitas complementares são eliminadas e as remanescente passam por mais um ciclo de amplificação, contudo, dessa vez sendo adicionados nucleotídeos que possuem um fluoróforo imobilizado em sua extremidade 3', de modo que, para que o nucleotídeo seguinte seja adicionado, deve haver a remoção do fluoróforo por ação enzimática. Contudo, antes de ser removido, o tipo de nucleotídeo incorporado pode ser detectado através de excitação do fluoróforo por lasers e sensores apropriados. Por fim, o mesmo processo é feito com a fita complementar. A possibilidade de sequenciar os fragmentos de DNA de ambas as extremidades se mostrou um grande avanço em relação às tecnologias anteriores, podendo levar a resultados mais fidedignos através da sobreposição e combinação dos resultados gerados pelo sequenciamento iniciado das direções opostas (MARDIS, 2013).

2.5 ATTILA

A partir do momento em que passou-se a produzir grande quantidade de fragmentos sequenciados, iniciou-se o desenvolvimento e aplicação de tecnologias computacionais que

Figura 4 – Esquematização das etapas do sequenciamento feito pela plataforma Roche 454. O DNA a ser sequenciado é inicialmente clivado gerando fragmentos menores. Os fragmentos gerados passam por um processo de desnaturação de modo que tornam-se de fita única (ssDNA). É feita então a adição de adaptadores diferentes à cada extremidade dos fragmentos formando DNA molde de fita única (sstDNA) e permitindo a ligação de cada fragmento a uma microesfera (bead). A amplificação por PCR ocorre no interior de gotículas de uma emulsão de água em óleo. A cada nucleotídeo adicionado ocorre a liberação de pirofosfato que, após tratado, reage com a luciferase produzindo sinal luminoso que é lido por um sensor do sequenciador. Os sinais são processados e permitem a obtenção da sequência de cada fragmento.

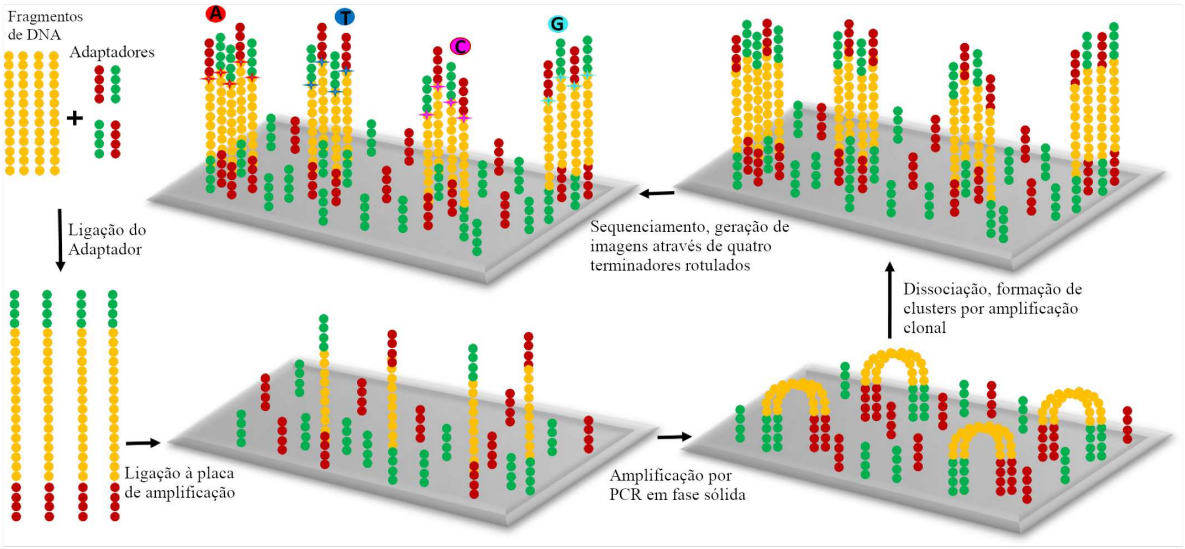


Fonte: Adaptado de WONG *et al.* (2019)

permitissem a sobreposição e combinação de fragmentos contíguos de uma mesma sequência em uma só (STADEN, 1979). Desde então uma série de ferramentas computacionais com as mais variadas aplicações foram criadas (MEHMOOD, 2014), e diversos protocolos de análise de dados biológicos foram estabelecidos baseados no conjunto dessas ferramentas, onde os dados de saída de uma ferramenta se tornam os dados de entrada de outra até que se obtenha o resultado final. Esse encadeamento de ferramentas, geralmente orquestrado por um conjunto de scripts, é chamado de *pipeline*.

O *AuTomated Tool for Immunoglobulin Analysis* (ATTILA) é um *pipeline* criado para a análise de dados NGS provenientes do sequenciamento de bibliotecas de *phage display* (MARANHÃO *et al.*, 2020). De modo geral, o ATTILA é capaz de quantificar a abundância, ou número de repetições, de cada sequência dentro de uma mesma biblioteca. Fazendo a comparação da abundância entre bibliotecas de *rounds* diferentes, pode identificar as sequências que apresentaram aumento de abundância mais considerável. Essa comparação é chamada de

Figura 5 – Sequenciamento pela plataforma Illumina. Adaptadores distintos são ligados a cada extremidade de fragmentos de DNA a serem sequenciados formando sstDNA. Os fragmentos são ligados através dos adaptadores em uma de suas extremidades a uma placa de amplificação se distribuindo de forma aleatória. É induzida a ligação da outra extremidade na placa, de modo que o fragmento assume uma conformação de ponte na qual passa por um processo de amplificação. Através da amplificação, formam-se agrupamentos de sequências idênticas. Uma nova amplificação é feita utilizando nucleotídeos que liberam fluoróforos ao serem incorporados, podendo ser detectados pelos sensores do sequenciador.



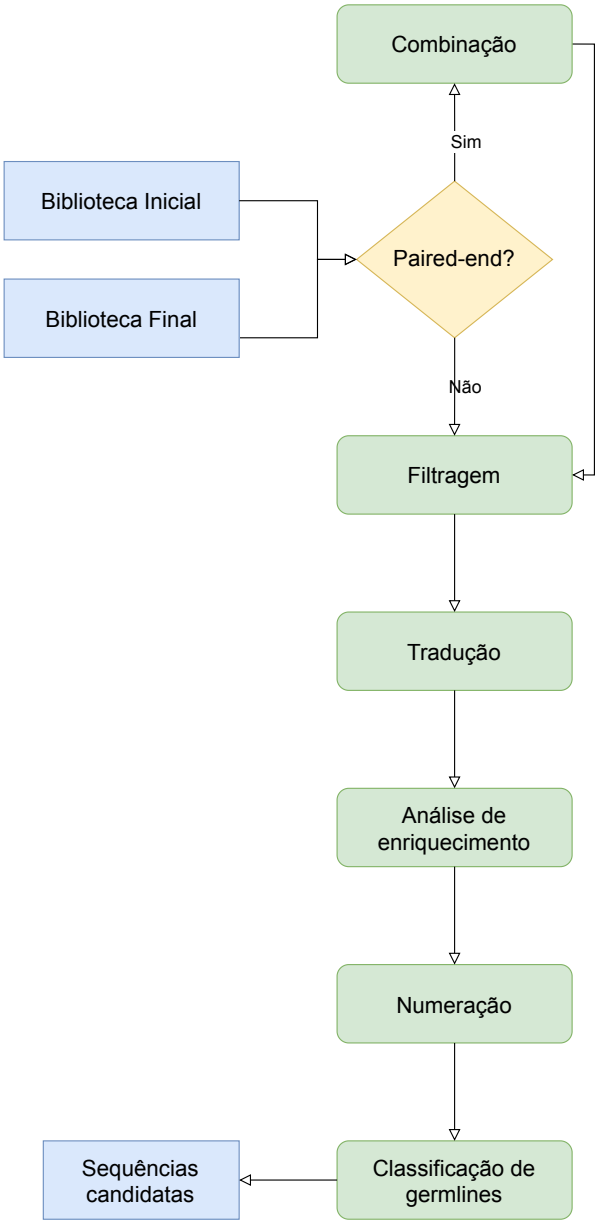
Fonte: Adaptado de WONG *et al.* (2019)

análise de enriquecimento. O ATTILA ainda fornece outras informações úteis como dados de qualidade das bibliotecas e as sequências germinais (*germlines*) das sequências candidatas mais enriquecidas.

O processo de análise se inicia com o recebimento dos arquivos fastq obtidos do sequenciamento NGS de bibliotecas de *phage display* (Figura 6). O formato de arquivo fastq tem como característica fornecer não apenas as sequências (*reads*) e seus identificadores, mas também uma avaliação da qualidade de cada nucleotídeo sequenciado. Quando o sequenciamento feito é do tipo single-end o sequenciamento é feito a partir de apenas uma das extremidades dos fragmentos e as *reads* obtidas são agrupadas em um único arquivo. Já em sequenciamentos do tipo paired-end, dois arquivos com *reads* são obtidos, cada um referente a uma das extremidades sequenciadas (*foward* e *reverse*). O sequenciamento paired-end favorece resultados mais precisos, com uma taxa de erro menor, pois, sendo as *reads* obtidas dos mesmos fragmentos, podem ser sobrepostas e comparadas levando à eliminação de erros.

Ressalta-se que, com as tecnologias de sequenciamento usadas atualmente, pode-se obter *reads* de tamanho próximo a 300 pb. A junção de *reads foward* e *reverse*, que possuem uma

Figura 6 – Representação em fluxograma das etapas executadas pelo *pipeline* ATTILA. O processo inicia-se com o recebimento dos dados NGS (bibliotecas VH e VL finais e iniciais). Caso o sequenciamento tenha sido do tipo paired-end, é feita a combinação das *reads* das duas extremidade com a ferramenta fastq-join, em seguida é feita a filtragem e o controle de qualidade das sequências montadas com o softwares Prinseq-lite (SCHMIEDER, 2011) e FastQC, em que aquelas com *Phred Score* menor que o valor definido pelo usuário são eliminadas. As etapas subsequentes são desempenhadas por scripts, com exceção da numeração de sequências, feita através da ferramenta Abnum, e da classificação de germlines, feita com o software IgBlast.



Fonte: O autor

região de sobreposição, obtidos de sequenciamento paired-end, permitem a obtenção de uma sequência consenso de tamanho maior, tornando possível a obtenção da sequência completa até mesmo do domínio VH que pode ter mais de 375 pb. Entretanto, não é possível combinar *reads* que não tenham sobreposição; com tecnologias capazes de gerar *reads* com até 300 pb, torna-se

inviável a obtenção da sequências de scFvs completos, que tendem a ter mais de 600 pb. Deste modo, costuma-se sequenciar as regiões codificantes para os domínios VH e VL separadamente. Esta estratégia tem como desvantagem a impossibilidade de identificar quais pares VH e VL estavam formados na biblioteca sequenciada.

Independentemente do tipo de sequenciamento, o ATTILA executa uma análise de qualidade das bibliotecas com o software FastQC (ANDREWS *et al.*, 2010). Caso o sequenciamento seja paired-end, o ATTILA executa uma etapa de combinação das *reads* das duas extremidades utilizando o software Fastq-join (ARONESTY, 2013). Posteriormente, as bibliotecas são submetidas a um processo de filtragem, baseado nos dados fornecidos através dos arquivos fastq e em um valor numérico de corte fornecido pelo usuário (Phred Score). O *Phred Score* é uma representação da probabilidade de uma determinada base sequenciada estar errada. O valor do *Phred Score* e sua relação com a probabilidade de erro no sequenciamento é determinado pela Equação 2.1 em que Q representa o valor do *Phred Score* e P a probabilidade de erro. Por padrão, o valor de *Phred Score* usado é 20, equivalente a 1 erro a cada 100 pares de bases.

$$Q = -10log_{10}P. \tag{2.1}$$

Após a filtragem, uma nova análise de qualidade é feita com o FastQC. Em seguida, as sequências de nucleotídeos são direcionadas a um processo de tradução com o programa nativo do ATTILA chamado translateab9 e que executa simultaneamente uma nova etapa de filtragem. O programa leva em consideração a distância esperada entre as cisteínas presentes no fim dos *frameworks* 1 e 3 (Figura 1), bem como a presença de aminoácidos canônicos das sequências de VH e VL humanas, eliminando as sequências que não correspondem ao padrão esperado.

As sequências traduzidas são por sua vez direcionadas a uma análise de enriquecimento. A análise de enriquecimento considera a abundância relativa (fr_i) de cada sequência de aminoácido dentro da biblioteca dos *rounds* inicial e final e é calculada segundo a Equação 2.2, em que fr_i é a frequência relativa da sequência *i*, F_i é o número de repetições da sequência *i* e *N* é o número total de sequências na biblioteca antes da tradução. O quociente da divisão da frequência relativa do *round* final pela do inicial é denominado *fold-change* e representa o número de vezes em que determinada sequência aumentou ou diminuiu em abundância no *round* final em relação ao *round* inicial (Equação 2.3).

$$fr_i = \frac{F_i}{n}. \tag{2.2}$$

$$fr_i = \frac{fr_{if}}{fr_{i0}}. \tag{2.3}$$

As sequências que tiveram maior *fold-change* são selecionados e passam por um processo de numeração, originalmente através da ferramenta web Abnum (ABHINANDAN *et al.*, 2008), e alinhamento contra um banco de sequências germinais humanas com o IgBlast (YE *et al.*, 2013). Por fim, a partir das informações obtidas ao longo do processo de análise, é gerado um relatório para cada cadeia que inclui um gráfico que expressa o número de sequências descartadas ao longo do processo e o decaimento de sequências válidas em cada etapa de análise.

Embora o ATTILA seja inovador, uma vez que não existia ferramenta capaz de unir todas essas funcionalidades e realizar análise tão completa de dados NGS de bibliotecas de *phage display*, o ATTILA possui limitações descritas a baixo.

A popularização e barateamento das tecnologias de sequenciamento NGS permitiu a pesquisadores experimentalistas que trabalham com *phage display* a obtenção de dados NGS de seus experimentos através de facilities de sequenciamento sem, necessariamente, terem amplo conhecimento de como fazer a análise desses dados. Ainda que o ATTILA estabeleça uma metodologia de análise e ofereça os componentes para que seja feita, o processo de instalação de todos os componentes do *pipeline* é laborioso, demorado e exige conhecimento considerável de sistemas operacionais Linux por parte do usuário, uma vez que todo o procedimento, que conta com várias etapas, é majoritariamente feito no terminal. Além disso, atualmente alguns de seus componentes estão desatualizados, inviabilizando o funcionamento de parte do *pipeline*. A execução da análise também se dá inteiramente pelo terminal e apenas o relatório final apresenta uma interface gráfica. O conjunto de análises realizadas pelo ATTILA tende a sobrecarregar a memória Memória de Acesso Aleatório (RAM) do computador em que é executado, uma vez que os arquivos trabalhados são relativamente grandes, e, por falta de otimização, são em alguns momentos carregados inteiramente podendo levar a travamentos do computador. Por fim, o ATTILA originalmente é capaz de analisar apenas dois *rounds* por vez, o final e o inicial, de modo que se o experimento incluir o sequenciamento dos *rounds* intermediários, o processo de análise tem de ser repetido pelo usuário para cada par, e deste modo os resultados ficam segmentados em relatórios diferentes, além de tornar o processo de análise mais lento.

2.6 Engenharia de proteínas *in silico*

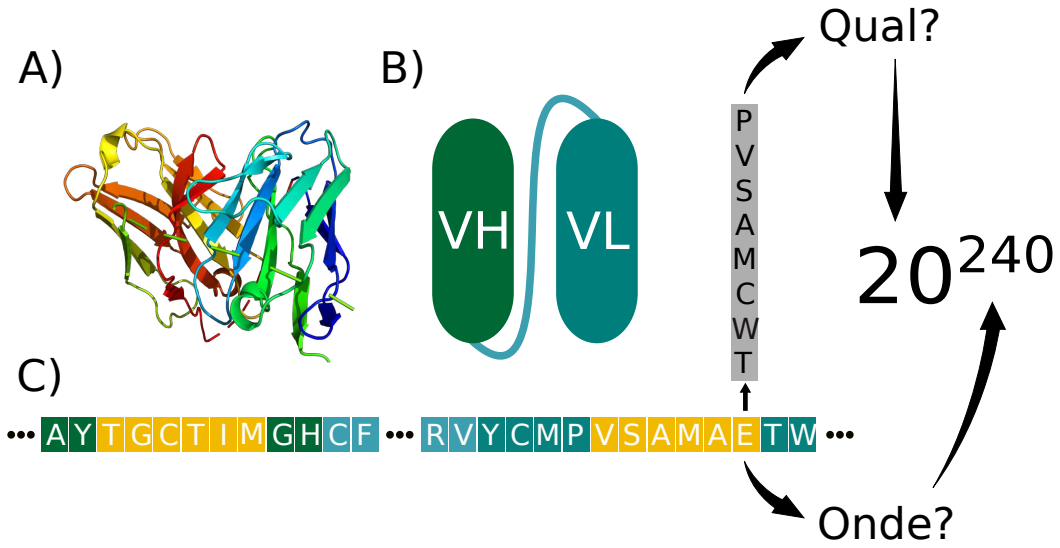
O Grupo de Pesquisa em Engenharia de Proteínas e Soluções para a Saúde (GEPeSS) tem atuado no desenvolvimento de estudos e ferramentas com foco em anticorpos e soluções derivadas, como células-T com Receptor de Antígeno Quimérico (CAR). O trabalho de Rodrigues (2014) permitiu uma visão inicial da forma como as estruturas dos anticorpos, em especial os *loops* que compõem as CDRs, se comportam em sua forma tridimensional, baseada em dados experimentais obtidos de bancos de dados públicos. Outros trabalhos no GEPeSS como o de (REBOUÇAS, 2018; OLIVEIRA, 2020; LIMA, 2022) buscaram analisar o comportamento da estrutura de fragmentos de anticorpos de interesse comercial quando simulados *in silico* e viabilizaram a definição de parâmetros de avaliação da interação de fragmentos variáveis com alvos moleculares de interesse.

Os trabalhos desenvolvidos pelo grupo representam os primeiros passos em direção a proposição de novos anticorpos e derivados com características de interesse como: alta afinidade pelo antígeno, alta termoestabilidade e baixa antigenicidade. Contudo, hoje, é impossível percorrer e testar todo o espaço de busca de anticorpos possíveis *in silico*, uma vez que, considerando apenas os domínios variáveis, tem-se uma média de 240 resíduos de aminoácidos, sendo possíveis pelo menos 20 tipos de resíduos por posição, ou seja, 20^{240} sequências possíveis (Figura 7).

Uma forma de direcionar a busca, tornando mais eficiente a proposição de novas sequências é através da análise das sequências e estruturas de scFvs e Fabs já obtidos experimentalmente. A técnica de *phage display* associada ao sequenciamento NGS é capaz de gerar uma quantidade massiva de dados de sequências de anticorpos ou derivados que são experimentalmente viáveis, podendo servir de base para proposição de mutações em anticorpos de interesse.

Este projeto tem como finalidade recriar o *pipeline* ATTILA na forma de uma nova versão do ATTILA, denominada ATTILA 2.0, estável, de uso e instalação mais simples, e com processamento otimizado. A otimização do processamento inclui a refatoração do código, aplicando paralelização das tarefas e divisão da análise dos dados. Na nova estratégia projetada, os dados parcialmente processados na máquina do usuário são transferidos para nuvem em infraestrutura da Fiocruz para término das análises. Ao fim do processamento, os resultados gerados na nuvem são retornados para o usuário, incluindo sequências enriquecidas, numeradas e com classificação de germlines. A execução de parte do processamento em

Figura 7 – Representação do espaço de busca de novos scFvs em termos de sequência. A) Representação da estrutura tridimensional de um scFv. B) Esquematização simplificada de um scFv formado por um VH e um VL unidos por um *linker*. C) Sequência ilustrativa de scFv, com CDRs destacadas em amarelo, onde geralmente são focadas as mutações que visam melhoria da afinidade e especificidade de um scFv. Destaca-se que para cada posição a ser mutada tem-se 20 possíveis aminoácidos candidatos. Considerando-se um scFv com 240 aminoácidos, tem-se 20^{240} possíveis sequências. O conhecimento de quais resíduos são mais frequentes para cada posição (representados em cinza) a partir de dados de repertório de sequências possibilita o direcionamento das mutações propostas, e um repertório de estruturas possibilita a modelagem massiva de estruturas tridimensionais das CDRs, que possuem estrutura canônica definida a partir do tamanho.



Fonte: O autor

infraestrutura da Fiocruz possibilitará que o usuário se beneficie de informações complementares a serem implantadas futuramente no ATTILA 2.0. Essa estratégia visa ainda possibilitar o compartilhamento de informações das sequências obtidas pelo usuário através da deposição em Banco de Dados Relacional (BD) mantido pelo GEPeSS. A existência do BD possibilita o recebimento de informações e a integração com com BDs públicos de sequências e estruturas de anticorpos e seus fragmentos objetivando, futuramente, a proposição mais assertiva de mutações que levem a melhorias nas propriedades de interesse de scFvs e Fabs para aplicação do desenvolvimento de mAbs.

3 OBJETIVOS

3.1 Objetivo geral

Desenvolver uma ferramenta computacional a partir do ATTILA com melhor desempenho, usabilidade e novas funções.

3.2 Objetivos específicos

1. Desenvolver uma interface gráfica para inserção dos parâmetros de entrada e visualização dos resultados;
2. Otimizar o processamento visando diminuir o tempo de análise e a sobrecarga da memória RAM;
3. Possibilitar a análise de múltiplos *rounds* simultaneamente;
4. Diminuir a carga computacional para o usuário através de computação em nuvem;
5. Facilitar o processo de instalação através de um instalador único;
6. Validar o funcionamento através de análise de dados NGS experimentais obtidos pelo sequenciamento de bibliotecas de phage display tendo como alvo uma das alças do CD20.

4 METODOLOGIA

4.1 Desenho e implementação de uma interface gráfica para recebimento de parâmetros e visualização dos resultados

A primeira função da interface gráfica é o recebimento dos parâmetros de forma amigável ao usuário, portanto o seu desenho foi baseado nos parâmetros já esperados pelo ATTILA, sendo eles: identificador do projeto, diretório para salvamento dos arquivos gerados, se o sequenciamento é paired-end ou não, qualidade mínima desejada, tamanho mínimo desejado, e número de sequências a serem selecionadas. Para o ATTILA 2.0, também foram acrescentados o número de *rounds* a serem analisados e o tipo de comparação entre *rounds* desejada.

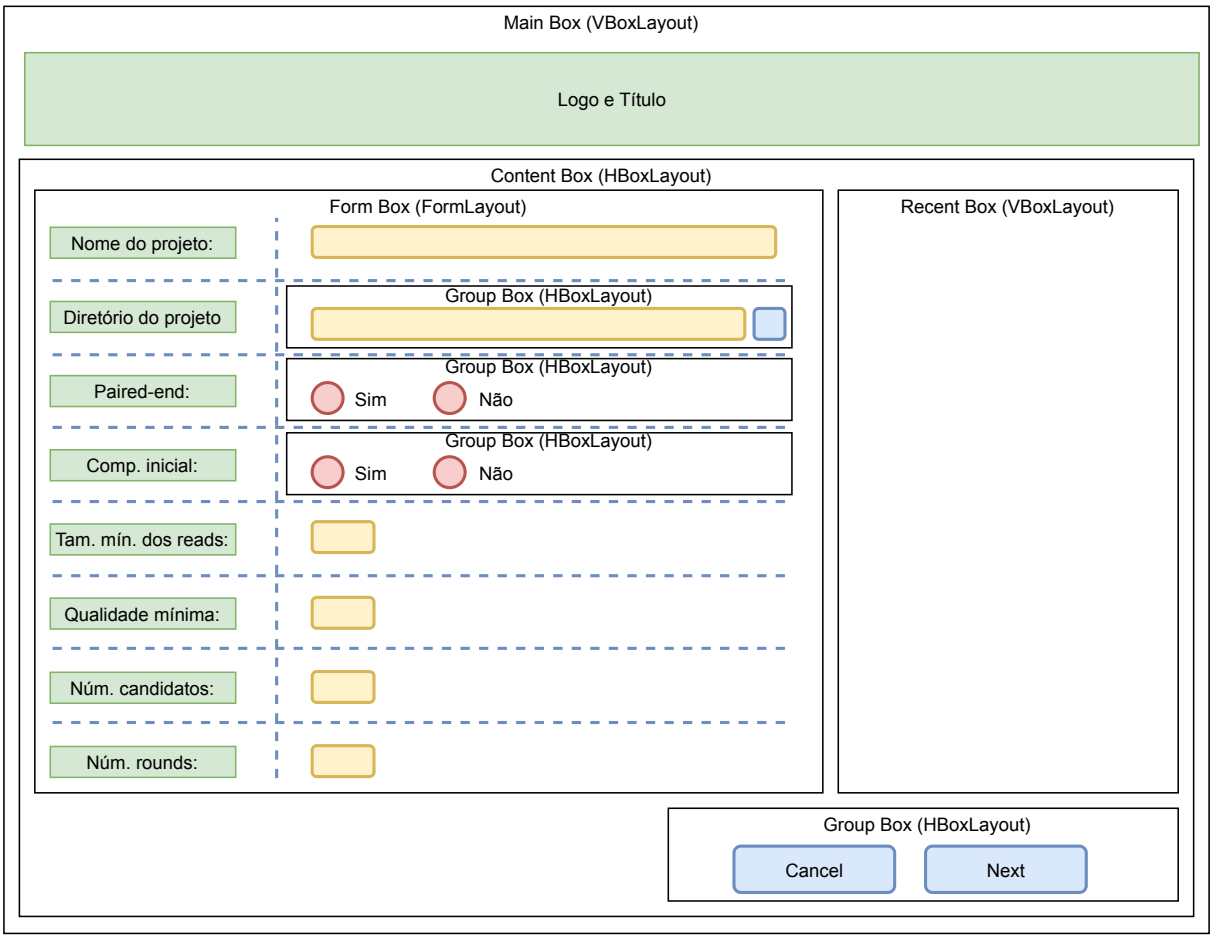
O projeto inicial da interface foi feito através do software QtDesigner. A implementação da interface, incluindo suas três telas (Figuras 8, 9 e 10), foi realizada utilizando-se a linguagem Python3 acompanhada da biblioteca PyQt5. Também foi utilizado o sistema gerenciador de banco de dados (SGBD) SQLite para armazenamento dos parâmetros inseridos pelo usuário a fim de gerar um histórico ao qual o usuário pudesse recorrer frente à necessidade de repetir uma análise, não precisando, assim, preencher todos os campos novamente.

4.2 Otimização do processamento para o ATTILA 2.0

O *pipeline* ATTILA é composto de vários scripts escritos na linguagem Perl e diversos softwares de terceiros, sendo todos orquestrados por um script principal, o *autoiganalisis.pl*, também escrito em Perl. Visando alcançar maior flexibilidade para implementação de melhorias no ATTILA 2.0, o código do script principal foi reescrito utilizando-se a linguagem de programação Python3, sendo segmentado em dois módulos. O primeiro módulo, *autoiganalysis3.py*, foi criado para atuar na comunicação direta com ferramentas de terceiros e comandos do terminal. O segundo módulo, o *execute6.py*, foi desenvolvido como orquestrador do fluxo de análise. Por fim, também foi necessário adaptar o script *parserid.pl*, responsável por eliminar espaços dos identificadores das sequências, uma vez que identificou-se instabilidade nos resultados da execução deste script quando analisando sequências cujos identificadores continham caracteres especiais.

Visando obter melhor aproveitamento do poder computacional disponível na máquina do usuário e, assim, diminuir o tempo de processamento, o módulo *execute6.py* foi arquitetado

Figura 8 – Projeto da Tela 1, responsável pelo recebimento dos parâmetros de análise. Em verde, etiquetas de texto; em amarelo, campos para digitação de texto; em azul, botões de ação; em vermelho, botões de opção. As linhas pretas definem o escopo de agrupamentos e disposições dos elementos.



Fonte: O autor

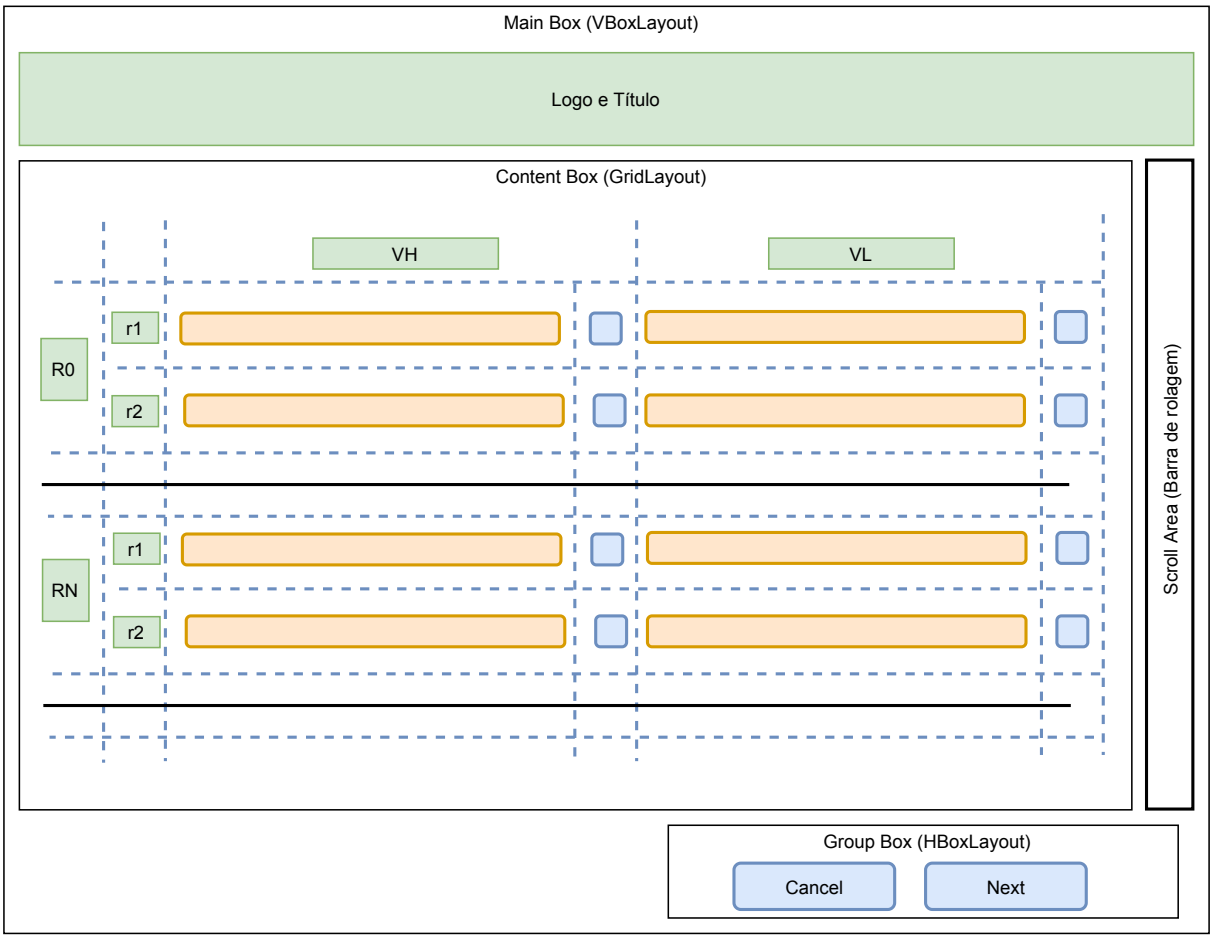
de modo a possibilitar a ação da biblioteca *multiprocessing* do Python, responsável por permitir a execução simultânea de processos em múltiplos núcleos de uma mesma CPU (11).

4.3 Implementação de rotina de análise simultânea de múltiplos *rounds* no ATTILA 2.0

O ATTILA, ao processar as bibliotecas NGS recebidas do usuário, executa as etapas da análise em pares de *rounds* (final e inicial) de cada domínio. O código responsável pela execução das etapas de montagem, filtragem e tradução foi remodelado para tornar possível a análise de cada *round* de forma independente. A etapa de análise de enriquecimento, responsável por comparar as abundâncias relativas de cada clone em um par de rounds, foi reestruturada para comparar múltiplos *rounds*.

Um problema já existente no *pipeline* original, e aumentado com a execução simultânea, principalmente, da etapa de tradução, foi o uso intenso de memória RAM, sobrecarregando-a

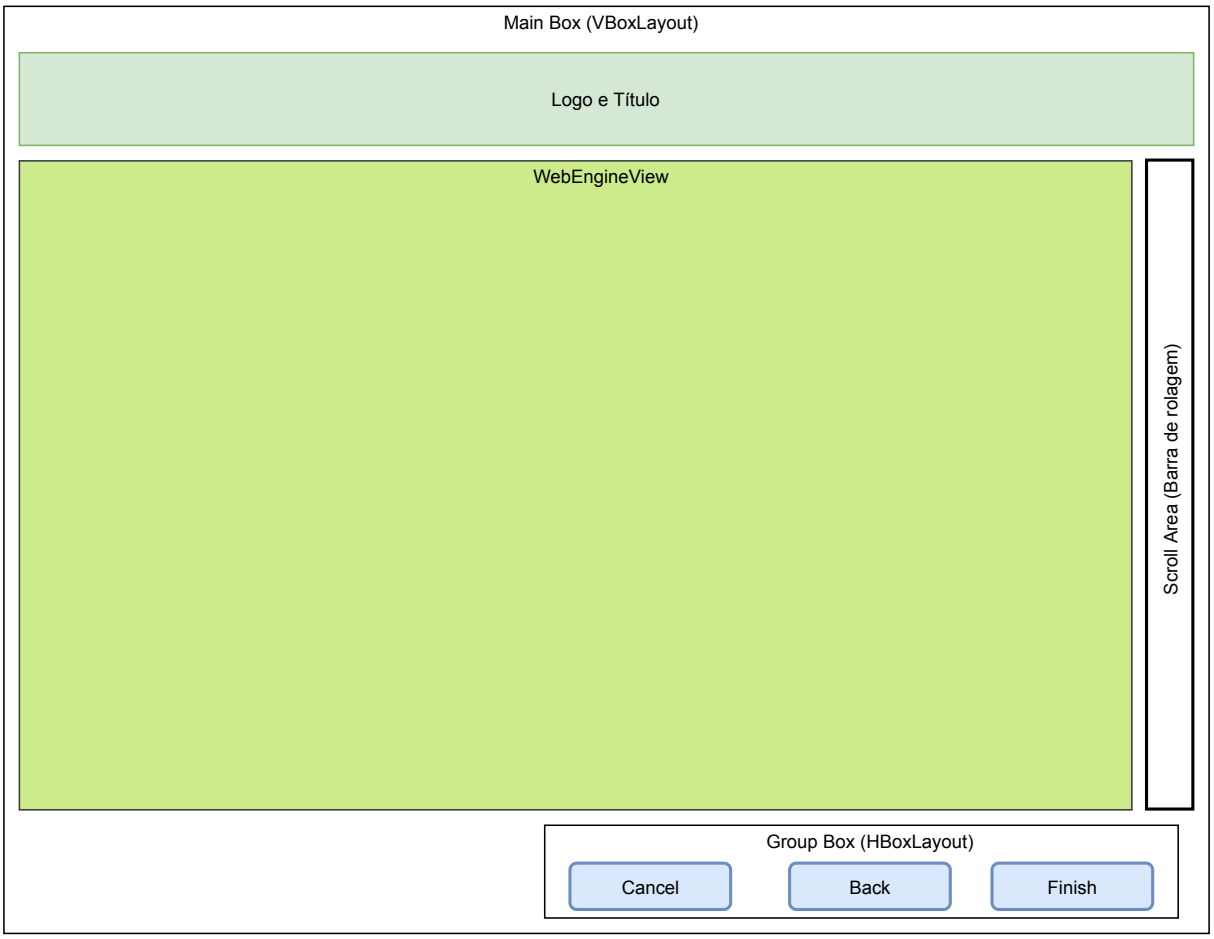
Figura 9 – Projeto da Tela 2, direcionada ao recebimento das bibliotecas NGS em arquivos do formato fastq. Em verde, etiquetas de texto; em laranja, campos para digitação de texto; em azul, botões de ação. As linhas pontilhadas indicam o arranjo dos elementos dentro de um sistema de grid estabelecido através de layout do PyQt5.



Fonte: O autor

e levando a travamentos do computador. Para solucionar esse problema, o arquivo fasta obtido após a etapa de filtragem passou a ser segmentado em múltiplos arquivos, cada um com 1000 sequências (Figura 12). A etapa de tradução, baseada na execução da ferramenta *translateab9*, passou a ser executada com cada um dos arquivos gerados após a segmentação. Assim, apenas parte das sequências a serem traduzidas são carregadas na memória em cada ciclo de tradução. Ao fim do processo de tradução, os arquivos resultantes, contendo as sequências traduzidas, são unidos gerando o arquivo final que seguirá para a etapa de contagem dos clones e cálculo da frequência relativa.

Figura 10 – Projeto da Tela 3, espaço dedicado à visualização dos resultados das análises A área em verde (*WebEngineView*) permite a visualização de arquivos HTML, funcionando de forma semelhante a um navegador web. Em azul, os botões de ação que permitem fechamento do programa e retorno para telas anteriores.



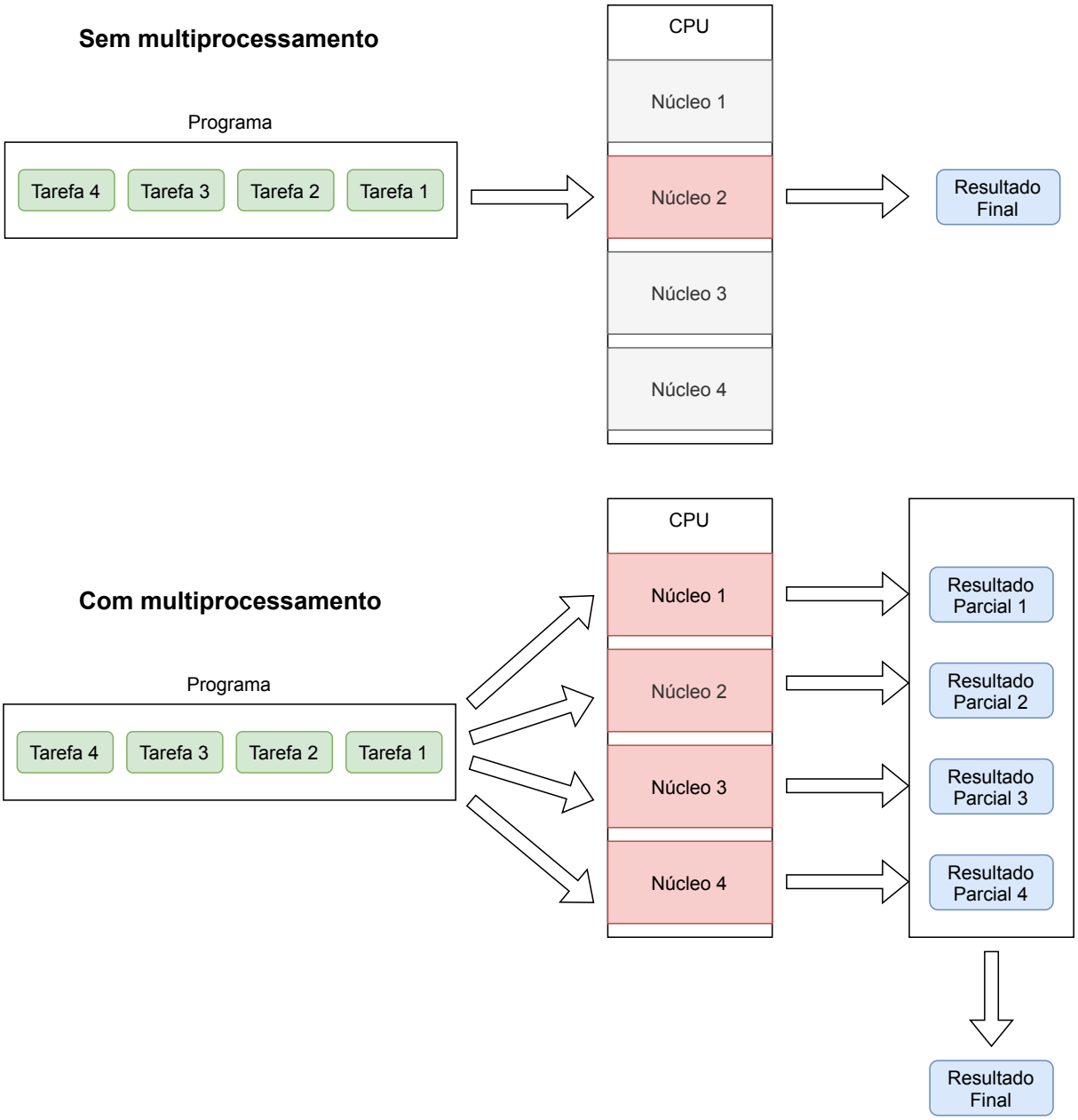
Fonte: O autor

4.4 Estruturação do ATTILA 2.0, criação e instalação de uma API para computação em nuvem

As etapas de análise foram divididas em três fases (Figura 13). A primeira, a ser executada na máquina do usuário, compreende as etapas de montagem, filtragem, tradução e análise de frequência. A segunda foi realocada para execução em uma Interface de Programação de Aplicação (API) hospedada em servidor da Fiocruz, e compreende as etapas de análise de enriquecimento, seleção de sequências, numeração e busca de sequências germinais. A terceira e última fase é a responsável pela finalização da análise de enriquecimento, geração dos relatórios e apresentação dos resultados.

O ATTILA utiliza o servidor web Abnum na etapa de numeração. Embora ainda esteja disponível para uso, sendo um serviço hospedado em servidores de terceiros, está sujeito

Figura 11 – Projeto da Tela 3, espaço dedicado à visualização dos resultados das análises A área em verde (*WebEngineView*) permite a visualização de arquivos HTML, funcionando de forma semelhante a um navegador web.

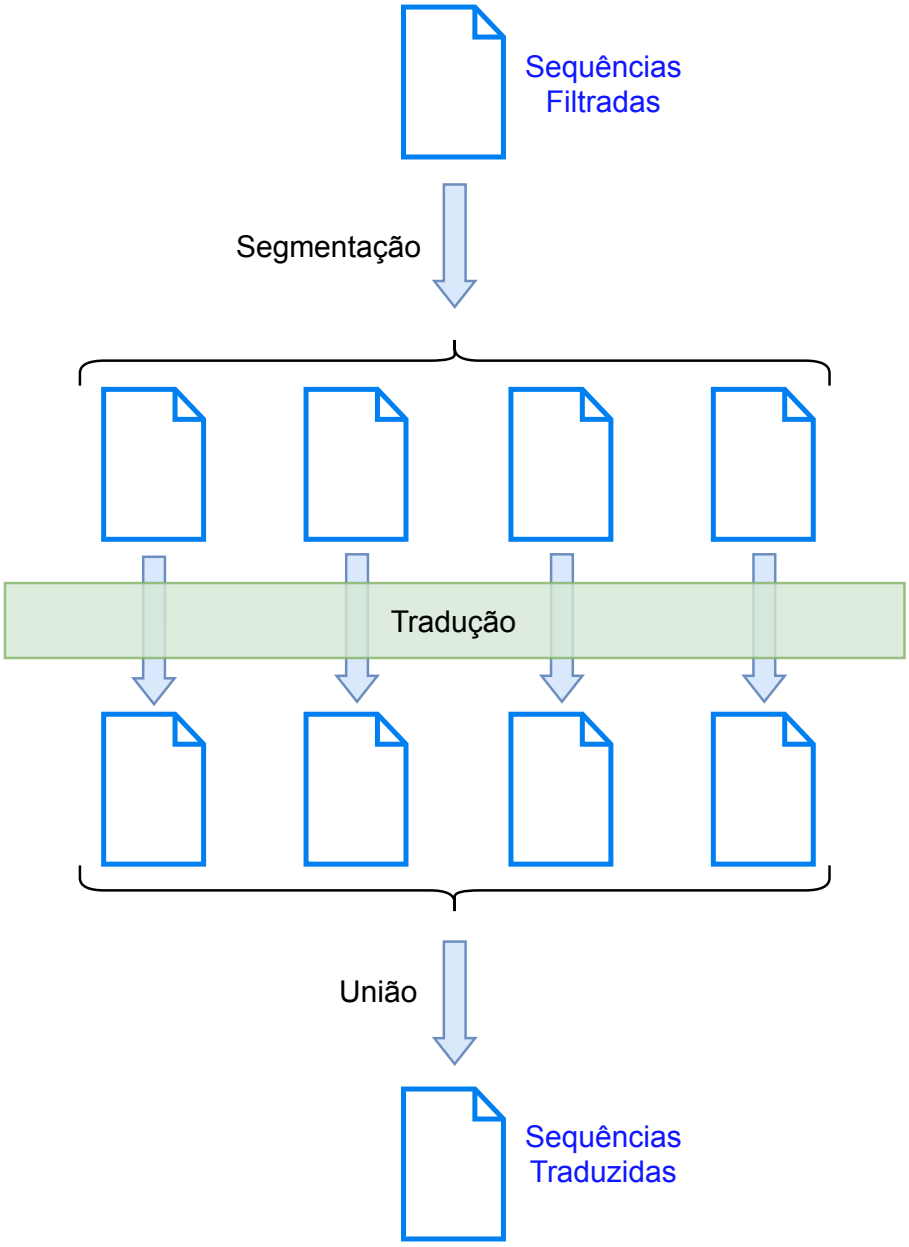


Fonte: O autor

a indisponibilidade imprevista, temporária ou definitiva. Além disso, o número de sequências traduzidas pode ser limitado para evitar sobrecarga e como política de segurança contra ataques. Deste modo, para o ATTILA 2.0 foi feita a substituição da ferramenta de tradução para o ANARCI (DUNBAR; DEANE, 2015). O script convertofasta.pl foi adaptado para tratar a saída do ANARCI em vez da saída do Abnum.

A API foi desenvolvida utilizando-se como base a estrutura fornecida pelo framework Django complementado pela biblioteca Django Rest Framework (Figura 14). MySQL foi

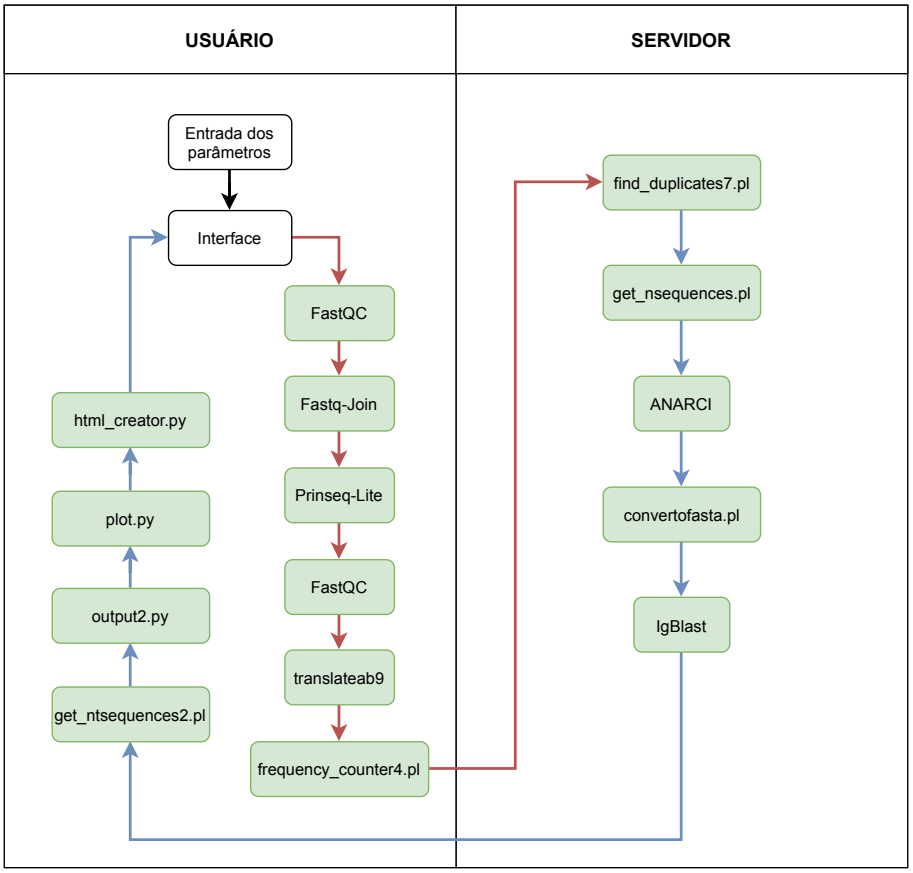
Figura 12 – Representação do processo de segmentação das sequências filtradas antes da etapa de tradução. O arquivo com as sequências filtradas é segmentado em múltiplos arquivos com até 1000 sequências cada. Cada novo arquivo passa pelo processo de tradução de forma independente. Assim, apenas 1000 sequências são carregadas na memória RAM por vez. Os arquivos resultantes do processo de tradução são, então, unidos.



Fonte: O autor

utilizado como SGBD, estando instalado em um servidor de dados externo também providenciado pela Fiocruz. O servidor de arquivos estáticos utilizado foi o NGINX. O Gunicorn foi utilizado como servidor HTTP. Docker e Docker-compose foram utilizados para criação de um sistema de containers com cada um dos requerimentos já citados. A API foi hospedada em servidor com sistema operacional Ubuntu Server 20.04. A API se tornou responsável por receber os dados resultantes da primeira fase da análise e encaminhá-los para a segunda fase. Terminada a análise,

Figura 13 – Fases de análise do ATTILA 2.0. A partir do recebimento dos parâmetros de análise e das bibliotecas NGS pela interface, inicia-se a análise ainda na máquina do usuário (setas vermelhas). Após a contagem de sequências por clone de cada biblioteca através do script *frequency_counter4.pl*, os arquivos fasta resultantes são enviados para a API desenvolvida e hospedada em servidor da Fiocruz onde a análise prossegue. Ao fim da busca por sequências germinais feita através da ferramenta IgBlast, os arquivos resultantes das etapas de análise do servidor são enviados para o computador do usuário, onde a análise geral é finalizada e o relatório de resultados é gerado.

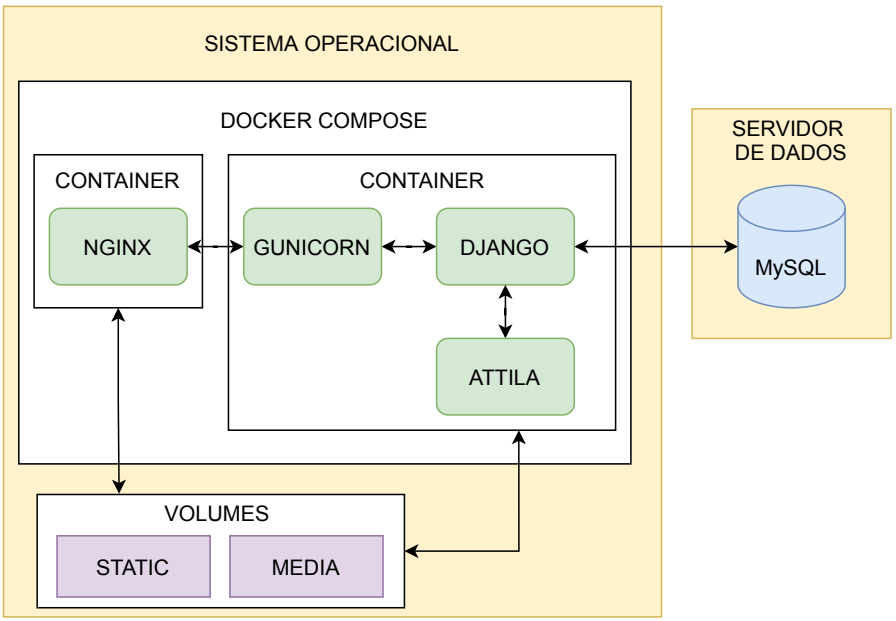


Fonte: O autor

os arquivos resultantes são enviados novamente para a máquina do usuário onde a terceira fase da análise deve iniciar.

A terceira fase de execução do ATTILA 2.0 compreende as etapas de obtenção das sequências de nucleotídeos a partir das sequências selecionadas, contagem do número de sequências ao longo das etapas de análise, confecção dos gráficos e geração do relatório final. O script gerador do relatório, antes baseado em um arquivo html base, foi reescrito com Python3 utilizando-se a biblioteca Dominate. Os scripts geradores dos gráficos, antes feitos com a linguagem R, foram recriados com Python e com as bibliotecas Pandas, Seaborn e Matplotlib, visando maior dinamicidade e permitindo o recebimento de dados de múltiplos *rounds*.

Figura 14 – Estrutura usada na implementação da API. Os dois containers são instâncias independentes, mas interagem entre si através de protocolos HTTP e através dos volumes de dados, espaços em disco que podem ser acessados por ambos os containers. A interação com o servidor de dados fica a cargo do framework Django.

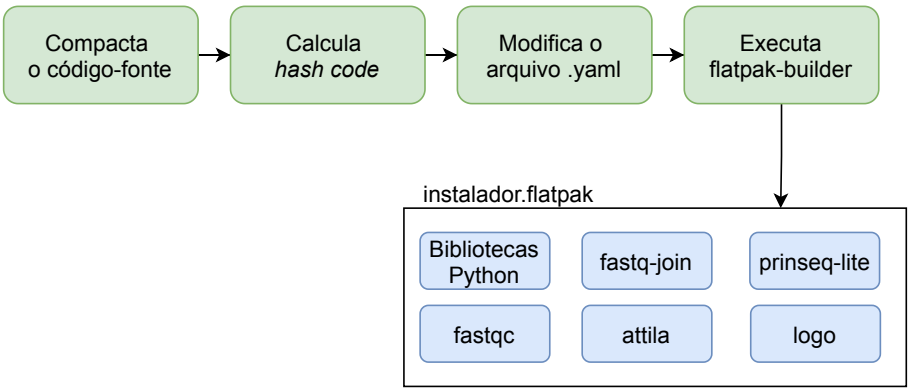


Fonte: O autor

4.5 Desenvolvimento de um instalador

Um instalador foi desenvolvido com a plataforma Flatpak, uma tecnologia que permite o encapsulamento de scripts e programas na forma de módulos em um microambiente próprio isolado do resto do sistema operacional. É altamente versátil, sendo de fácil instalação e compatível com todos os sistemas operacionais Linux. Para criação do instalador, todos os componentes necessários ao funcionamento dos programas e scripts do segmento do usuário do ATTILA 2.0 foram definidos em módulos de um arquivo de configuração, do tipo yaml, exigido pelo construtor de pacotes, *flatpak-builder*. Em cada módulo foi definido a forma como o programa, biblioteca ou script estava guardado, se deveria ser descompactado e quais deveriam ser suas etapas de instalação. De modo a realizar a construção dos pacotes automaticamente após cada nova versão, foi criado um script em bash chamado compila.sh capaz de compactar a pasta com os códigos do ATTILA 2.0, calcular um código hash único para a nova versão, editar o arquivo yaml sinalizando uma nova versão, e executar a ferramenta de criação de instalador (Figura 15).

Figura 15 – Processo de geração de uma nova versão do instalador através do novo processo desenvolvido.



Fonte: O autor

4.6 Validação do ATTILA 2.0

O ATTILA 2.0 foi testado através da análise de dados NGS advindos do sequenciamento paired-end de bibliotecas de quatro *rounds* de seleção por *phage display*. A biblioteca inicial foi obtida através da técnica de *error-prone Polymerase Chain Reaction* / reação em cadeia da polimerase propensa ao erro (epPCR), tendo passado por filtragem através do método de *biopanning*, onde uma das alças da proteína CD20 humana foi utilizada como alvo. Os dados foram cedidos pelo Dr. Gilvan Pessoa Furtado, pesquisador da Fundação Oswaldo Cruz - Ceará (Fiocruz-CE).

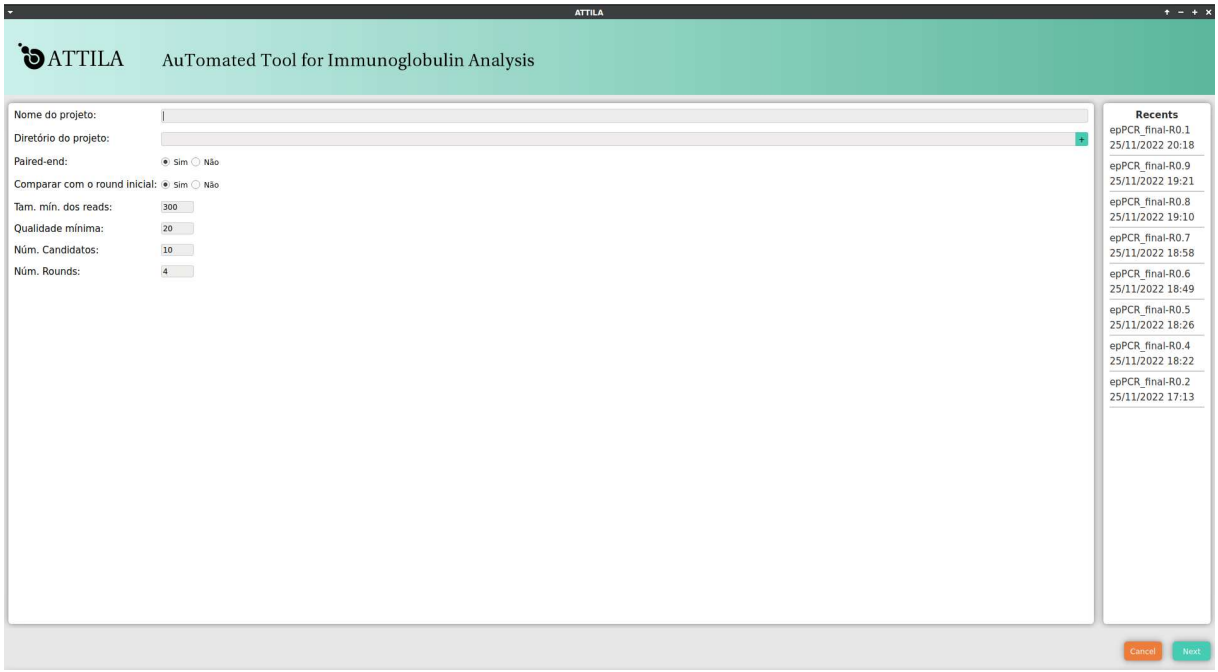
Para validação do ATTILA 2.0, foram feitas duas execuções, ambas com: paired-end definido como “sim”; tamanho mínimo das *reads* como 100; qualidade mínima como 20; número de sequências selecionadas como 10; e número de *rounds* definido como 4. Na primeira execução, a opção “comparar com *round* inicial” foi definida como “sim”, e, na segunda, definida como “não”. Foi executada com o ATTILA original, utilizando-se os mesmos parâmetros, através de repetidas execuções do *pipeline*. Os testes foram feitos em um computador com processador Intel Core i7, com 8 núcleos e 8 GB de memória RAM, além de um SSD de 256 GB.

5 RESULTADOS

5.1 Desenvolvimento da interface gráfica

A primeira tela da interface gráfica (Figura 16) foi desenvolvida com a função de receber os parâmetros da análise. As informações obtidas na primeira tela são: o nome do projeto, o diretório do projeto, o tipo de sequenciamento, o tipo de comparação entre rounds, o tamanho mínimo das *reads*, a qualidade mínima das *reads* na escala Phred, o número mínimo de candidatos que serão selecionados, e número de *rounds* sequenciados que serão analisados. Na lateral direita pode ser acessado o histórico de análises feitas recentemente, preenchendo os campos com os valores já utilizados. A descrição mais detalhada de cada campo pode ser encontrada no Apêndice A. Ressalta-se que a opção de escolha do tipo de sequenciamento se manteve para possibilitar a análise de dados antigos.

Figura 16 – Tela 1 Interface gráfica do ATTILA 2.0. Só é permitido o avanço quando todos os campos obrigatórios estão preenchidos e caso já não haja diretório com caminho proposto nos campos 1 e 2.

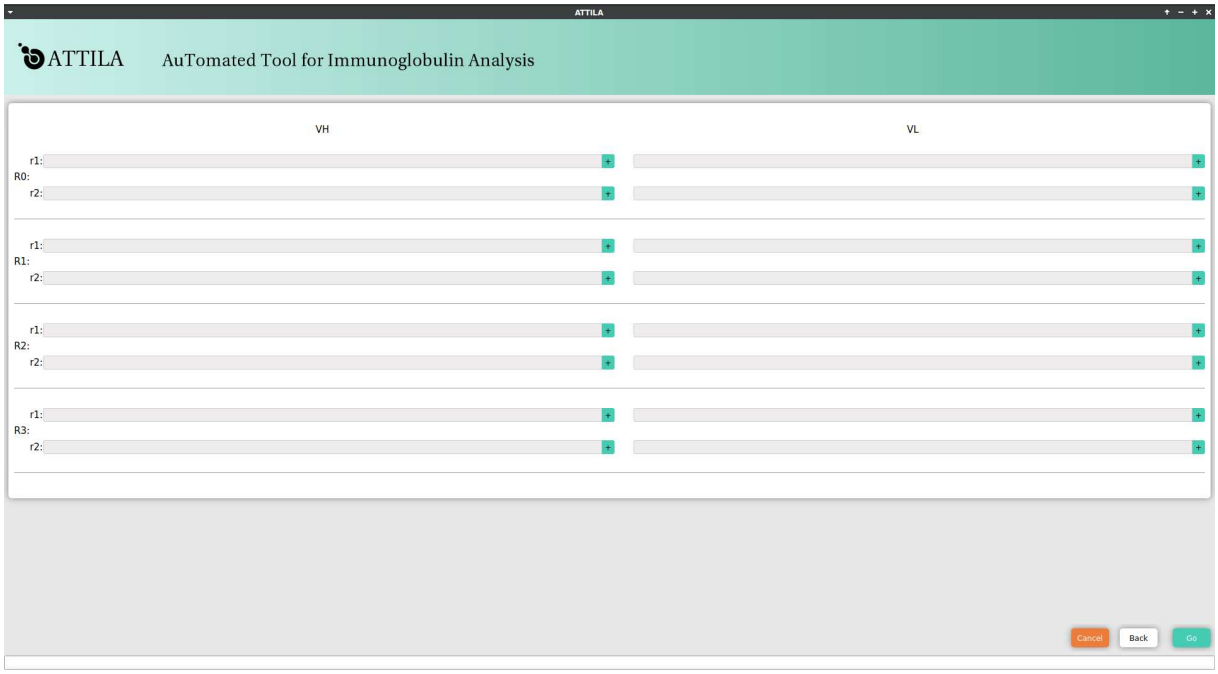


Fonte: O autor

Na segunda tela, o usuário pode fazer a inserção dos caminhos para os arquivos fastq oriundos do sequenciamento. No exemplo da Figura 16, devido à seleção da opção “sim” no campo “paired-end”, na Figura 17 tem-se dois campos para cada cadeia e *round* referentes aos arquivos de sequenciamento *foward* e *reverse*. Ainda na tela anterior (Figura 16), foi

definido o número de *rounds* como sendo 4, de modo que na Tela 2 (Figura 17), há campos para recebimento de arquivos de 4 *rounds*. Os campos são dinamicamente gerados com base nos parâmetros definidos na tela anterior.

Figura 17 – Tela de recebimento dos caminhos para os arquivos fastq oriundos do sequenciamento na máquina do usuário. É possível a inserção de duas formas: clicando no botão verde com o símbolo "+" e buscando pelo arquivo fastq nas pastas do sistema, ou copiando o arquivo fastq e colando no campo de texto.



Fonte: O autor

Por fim, na terceira tela, são apresentados os resultados na forma de um relatório (Figura 18). O relatório é gerado na forma de um arquivo html, de modo que também pode ser aberto pelo usuário quando quiser, através de um navegador web. Originalmente, o ATTILA gerava o relatório final tomando como base um arquivo html estático, que limitava a apresentação de resultados a apenas dois *rounds* de seleção. Para tornar a geração do relatório dinâmica, foi criado um módulo em Python3 utilizando-se a biblioteca Dominate capaz de abstrair os componentes de uma página web, permitindo a geração do relatório, independentemente, de um *template* base e direcionado pelos parâmetros da análise, como o número de *rounds*.

No relatório final os resultados estão divididos entre duas seções principais. A primeira seção, denominada de "*Reads Information*" reúne o quantitativo das sequências ao longo das etapas de análises (Figura 18). Abaixo de *reads Information* há uma barra de informações sobre percentuais de perda de *reads*. Informações da mesma natureza podem ser visualizadas através do gráfico à esquerda da tela, que exibe o percentual de *reads* descartadas por terem

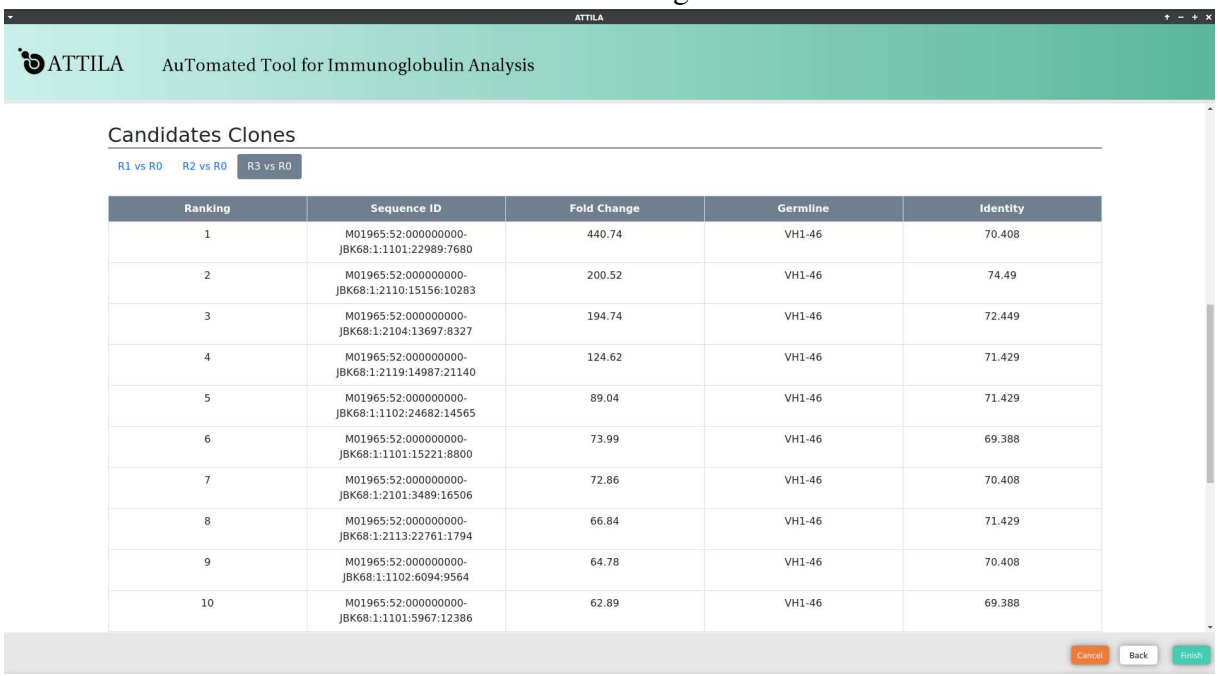
tamanho menor que o mínimo definido pelo usuário. No gráfico à direita da Tela 3, ainda na Figura 18, tem-se o número de sequências remanescentes em cada etapa e em cada *round* Tanto a tabela quanto os gráficos aqui citados são gerados de forma dinâmica, a partir dos dados e parâmetros inseridos pelo usuário.

Figura 18 – Exemplo de apresentação de relatório final na Tela 3. É possível ver as guias para alternar entre resultados dos domínios variáveis de cadeia pesada (*VH Library*) e leve (*VL Library*). A barra cinza no início da seção "Reads Information" indica a perda de sequências ao longo das etapas de análise que incluem, também, os processos de filtragem. O gráfico à esquerda mostra a proporção de *reads* eliminadas por tamanho inadequado. O gráfico à direita mostra o número de sequências remanescentes em cada etapa da análise para cada *round*

Fonte: O autor

Na segunda seção, "*Candidates Clones*", ainda na tela de resultados (Figura 19), são apresentados resultados mais específicos, com foco nas sequências selecionadas para cada *round* Através dos botões superiores (neste exemplo, R1 vs R0, R2 vs R0 e R3 vs R0), o usuário pode alternar entre os resultados de cada comparação. Os resultados desta seção são apresentados em duas tabelas. Na primeira é possível identificar o *fold-change* e a sequência germinal com maior identidade de cada candidato. As sequências são ranqueados do maior *fold-change* para o menor. Na segunda tabela, "*Regions of Variable Domain of Candidates Clones*", as sequências de cada candidato são apresentadas na mesma ordem da tabela anterior e com *frameworks* e CDRs identificados (Figura 20).

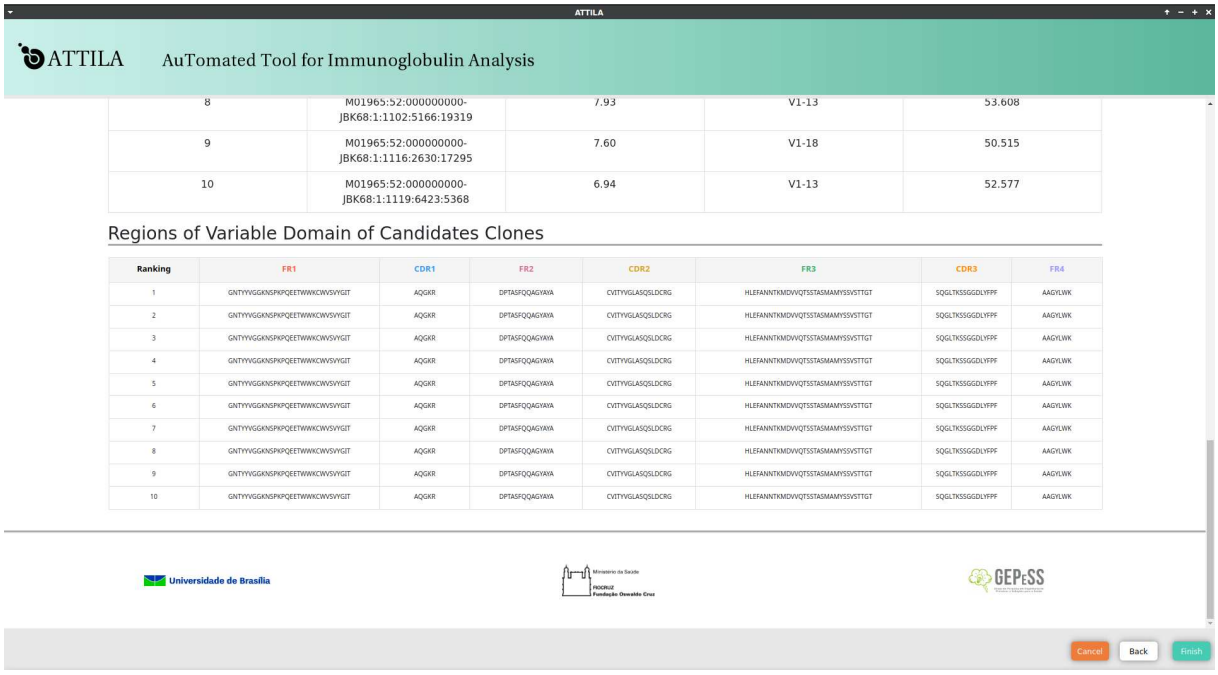
Figura 19 – Segunda seção da tela de resultados. Logo abaixo de “Candidates Clones” estão os botões seletores que permitem alternar entre os resultados de cada *round* O número de botões e a nomenclatura muda de acordo com o número de *rounds* analisados e o tipo de comparação. Abaixo dos seletores pode ser vista uma tabela, referente à comparação R3xR0 (selecionada) que identifica as sequências selecionadas pelo ID da sequência, fornece o fold change, a germline com maior identidade e o valor da identidade com a germline encontrada.



Ranking	Sequence ID	Fold Change	Germline	Identity
1	M01965:52:000000000-JBK68:1:1101:22989:7680	440.74	VH1-46	70.408
2	M01965:52:000000000-JBK68:1:2110:15156:10283	200.52	VH1-46	74.49
3	M01965:52:000000000-JBK68:1:2104:13697:8327	194.74	VH1-46	72.449
4	M01965:52:000000000-JBK68:1:2119:14987:21140	124.62	VH1-46	71.429
5	M01965:52:000000000-JBK68:1:1102:24682:14565	89.04	VH1-46	71.429
6	M01965:52:000000000-JBK68:1:1101:15221:8800	73.99	VH1-46	69.388
7	M01965:52:000000000-JBK68:1:2101:3489:16506	72.86	VH1-46	70.408
8	M01965:52:000000000-JBK68:1:2113:22761:1794	66.84	VH1-46	71.429
9	M01965:52:000000000-JBK68:1:1102:6094:9564	64.78	VH1-46	70.408
10	M01965:52:000000000-JBK68:1:1101:5967:12386	62.89	VH1-46	69.388

Fonte: O autor

Figura 20 – Segunda seção da tela de resultados. As sequências dos clones são mantidas na mesma ordem dos seus identificadores na tabela anterior. Os *frameworks* e as CDRs são segmentadas e destacadas para cada candidato. Neste exemplo, as sequências exibidas são meramente ilustrativas.



Ranking	FR1	CDR1	FR2	CDR2	FR3	CDR3	FR4
1	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
2	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
3	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
4	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
5	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
6	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
7	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
8	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
9	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK
10	GNTYYVGGKNSPKPQEETWWKCWVSVYGIT	AQGKR	DPTASFQQAGYAYA	CVITYVGLASQSLDCRG	HLEFANNTKMDVVQTSSTASMAMYSSVSTTGT	SQGLTKSSGGDLYFPF	AAGYLWK

Fonte: O autor

5.2 Otimização do processamento

Embora a aplicação de multiprocessamento tenha dado mais celeridade à análise, também levou a um gargalo na utilização da memória RAM disponível. O uso excessivo da memória RAM causou travamentos na etapa de tradução, uma vez que o script *translateab9* carrega todas as sequências na memória antes de traduzi-las e estava fazendo isso para todos os *rounds* ao mesmo tempo. A segmentação dos arquivos de sequências de nucleotídeos em arquivos menores, com até 1000 sequências, permitiu a diminuição da carga de uso da memória RAM, eliminando completamente problemas com travamentos. Para garantir que esse problema não voltaria a acontecer, em decorrência da análise de um número maior de rounds, foi criado um funil no multiprocessamento de modo que, na etapa de tradução, apenas um processo seja executado por vez.

Por outro lado, durante a execução do ATTILA original com apenas dois rounds, observaram-se dois picos no uso de memória RAM também decorrente da execução do script *translateab9* para VH e VL. Embora seja possível a execução manual de múltiplas análises simultâneas com o ATTILA original, os picos no uso da memória RAM podem causar travamentos na máquina do usuário.

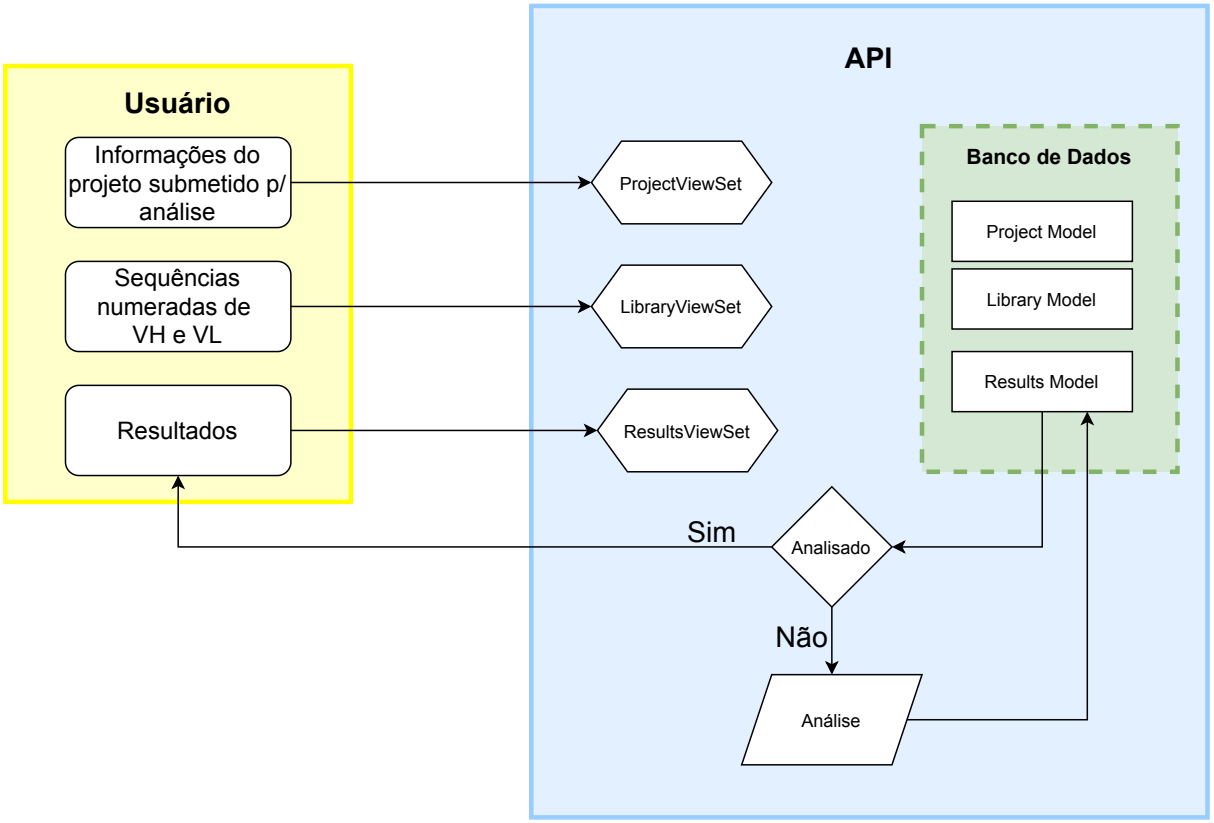
5.3 Estruturação do ATTILA 2.0, criação e instalação de uma API

A API desenvolvida se mostrou capaz de receber os resultados gerados na primeira etapa da análise. A comunicação do cliente com a API se deu por protocolos HTTP através de três endpoints construídos para recebimento e envio de dados. Cada endpoint foi conectado à sua respectiva tabela em um banco de dados SQL através dos models, uma construção padrão do *fold-change* Django (Figura 21).

O endpoint Project é o responsável por receber as informações do projeto, como data de submissão, e os parâmetros definidos pelo usuário no início da análise. Library é o endpoint responsável pelo recebimento dos dados referentes às bibliotecas, que são: identificador do projeto, cadeia, round, código hash calculado a partir do arquivo de sequências de aminoácidos recebido do usuário, caminho do arquivo de sequências e espécie analisada. O último endpoint é o Results, através dele o usuário pode obter os resultados das análises feitas no servidor como arquivo de sequências enriquecidas, arquivo de sequências enriquecidas numeradas no formato resultante do processamento pela ferramenta ANARCI, arquivo de sequências enrique-

cidas numeradas no formato fasta, arquivo de classificação de sequências germinais e log de processamento.

Figura 21 – Esquematização da comunicação entre as partes do ATTILA 2.0, da máquina do usuário com a API. As caixas brancas na seção do representam os conjuntos de dados que são enviados e/ou recebidos. Os hexágonos na seção da API representam os endpoints da API, ou seja, canais por onde é feita a comunicação entre a API e serviços externos visando o recebimento e envio de dados. Ainda na seção do servidor, dentro da área verde Estão representados os models, construções do Django que dão origem e gerenciam as tabelas no BD. O losango representa uma tomada de decisão que é acionada quando dados de resultados são requisitados da API, caso os resultados já estejam disponíveis, são enviados para o usuário, se não, é iniciado o processo de análise para gerá-los.



Fonte: O autor

Uma vez que a primeira fase do processamento é finalizada na máquina do usuário e os dados resultantes são submetidos para a API através dos endpoints Project e Library, visando a continuação da análise, o segmento da parte do usuário faz uma requisição para o endpoint de resultados (Results), o que dispara o início do processamento. Em seguida uma nova requisição é feita para o mesmo endpoint, dessa vez para o recebimento dos dados resultantes da análise.

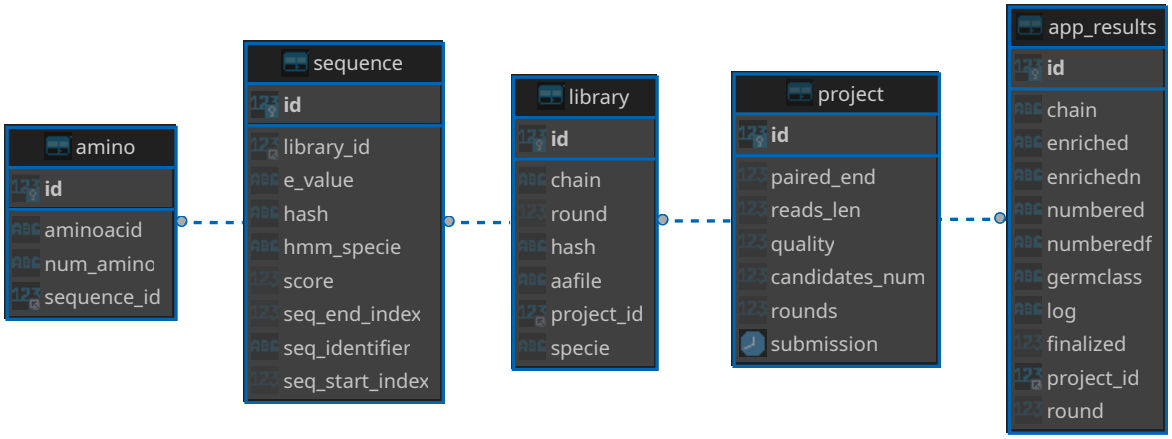
A existência de uma API, para onde as sequências traduzidas e tratadas são encaminhadas, permite o enriquecimento de um BD de sequências (Figura 22). A API também viabiliza a integração dos dados obtidos por *phage display* com dados de estrutura e de sequências de

mAbs obtidos por outros meios, como, por exemplo, de BD públicos. Possibilita ainda o acoplamento de uma série de outras ferramentas e *pipelines* que podem usufruir dos dados enriquecidos para geração de conhecimento útil ao desenho de mAbs.

O BD associado à API (Figura 22) foi projetado de modo a armazenar os dados do projeto, na tabela project. Nesta tabela, são armazenados os metadados do projetos: tipo de sequenciamento (paired-end), tamanho mínimo das *reads* (reads_len), qualidade mínima das *reads* (quality), número de candidatos a serem selecionados (candidates_num), número de *rounds* analisados (rounds) e data da submissão (submission).

Cada projeto cadastrado na tabela project pode estar associado a uma ou mais bibliotecas, representadas pela tabela library, que referencia o projeto de origem através do campo project_id. Para cada biblioteca é identificada: a cadeia à qual é referente (chain), o *round* ao qual é referente (round), o caminho do arquivo com as sequências de aminoácidos da biblioteca (aafile), um código hash que garante que o arquivo da biblioteca em questão não está duplicado (hash), e a espécie animal de origem (specie), que passará a ser preenchida em versões posteriores da ferramenta.

Figura 22 – Esquematização do MER do BD associado à API. Na tabela app_results estão os campos responsáveis por armazenar os caminhos dos arquivos gerados durante o processamento feito no servidor, sendo totalmente preenchida para cada cadeia e *round* analisado de cada projeto. A tabela project é responsável por armazenar os parâmetros utilizados na análise, advindos do computador do usuário. A tabela library guarda metadados associados às bibliotecas de sequências de cada projeto. A tabela sequence guarda os metadados de cada sequência presente nas bibliotecas submetidas para análise, incluindo resultados gerados pelo programa ANARCI. A tabela amino armazena cada aminoácido que compõem as sequências e o número atribuído durante o processo de numeração.



Fonte: O autor

A tabela sequence é responsável por armazenar os metadados das sequências identificadas nos arquivos de cada biblioteca. Como esta tabela tem o objetivo de armazenar apenas

as sequências de aminoácidos únicas de cada biblioteca, para cada sequência é calculado e armazenado um código hash (hash), evitando que sequências repetidas sejam guardadas. Aqui também são guardados os metadados gerados no processo de numeração das sequências através do software ANARCI, como: a espécie mais próxima identificada (hmm_specie), o score do alinhamento (score), a probabilidade do alinhamento ter sido realizado ao acaso (e_value), o aminoácido de início e de fim da numeração na sequência original (seq_start_index e seq_end_index, respectivamente). Por fim, a tabela sequence também armazena o identificador da sequência definido no arquivo de origem (sequence_id), e o identificador da biblioteca da qual faz parte (library_id)

A tabela app_results guarda majoritariamente os caminhos dos arquivos de resultados gerados pelo processamento realizado na API (Apêndice B). Dentre os campos que guardam caminhos de arquivos, tem-se: O campo enriched, para o arquivo enriched.fasta; enrichedn, arquivo list<num>.fasta, em que <num> é o número do *round* do arquivo em questão; numbered, arquivo list<num>numbered.txt; numberedf, para o arquivo list<num>numbered.fasta; germclass, arquivo list<num>numberedgermlineclassification.txt; e log, que guarda o caminho do arquivo log2.txt. Em relação aos demais campos, tem-se: chain, para o identificador da cadeia ao qual os arquivos são referentes, round, para identificação do *round* que originou os resultados; finalized, que informa se o processamento já foi finalizado; e project_id, o identificador do projeto ao qual os resultados pertencem.

5.4 Implementação da análise simultânea de múltiplos *rounds* no ATTILA 2.0

A arquitetura do ATTILA 2.0 tem diferenças consideráveis em relação ao ATTILA original. Além da utilização do Python3 como principal linguagem de programação no lugar da linguagem Perl, foi implementado um sistema de multiprocessamento e foi feita a divisão da execução entre o computador do usuário e uma API hospedada em infraestrutura da Fiocruz. Deste modo, fez-se necessário verificar se o tempo de análise foi afetado.

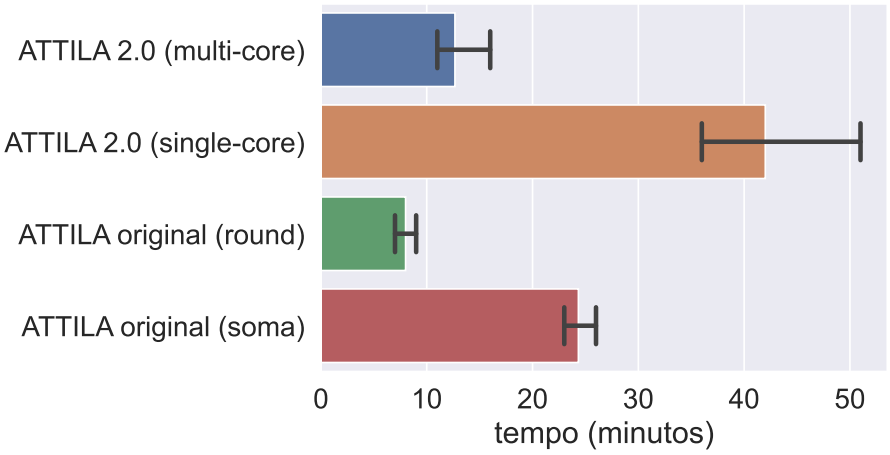
Para analisar o tempo de execução, os mesmos dados foram processados em triplicata com e sem o multiprocessamento ativado (*multi-core* e *single-core*, respectivamente). A mesma análise foi feita com o ATTILA original através de sucessivas execuções. Como resultado, com multi-core o tempo médio para obtenção do resultado foi de aproximadamente 16 minutos (11, 16 e 11 minutos), já para o single-core o tempo médio de execução foi de 42 minutos (36, 39 e 51 minutos). Para o ATTILA original o tempo de análise foi calculado pela soma das médias do

tempo de análise para cada par de *round* comparados (RnxR0, n=1...3), resultando em um valor arredondado de 24 minutos (Figura 23). O tempo médio por execução do ATTILA original ficou em 8 minutos. Pode-se notar que o multiprocessamento levou à redução significativa do tempo de análise dos dados em relação ao single-core ao possibilitar o processamento simultâneo de múltiplos *rounds*.

O tempo médio de análise no modo multi-core do ATTILA 2.0 foi menor que o da análise manual de múltiplos *rounds* com o ATTILA original, no qual a ferramenta foi executada para cada par de *rounds* comparados de forma sucessiva. O tempo médio de análise por *round* do ATTILA original, de 8 minutos, pode representar de forma teórica o que seria a execução do ATTILA original em modo multi-core. Vale lembrar, contudo, que um dos fatores limitantes da sobreposição de processos do ATTILA original é a sobrecarga de memória, que pode levar a travamentos. De fato, durante as execuções, mesmo sendo executado com apenas dois *rounds* por vez, em mais de uma vez o processo de tradução foi interrompido pelo sistema operacional devido ao pico no uso de memória. Como consequência disso, apenas parte das sequências foram traduzidas e os valores de *fold-change* foram alterados.

O maior tempo de análise no modo multi-core do ATTILA 2.0 em relação ao tempo médio por *round* do ATTILA original pode ser explicado pela reestruturação da etapa de tradução e pela alocação de parte do *pipeline* em nuvem. A remodelagem da etapa de tradução representa a inserção de novas etapas no processamento, o que demanda esforço computacional e tempo. A comunicação com a API é necessariamente dependente da conexão com a internet, de modo que está sujeita a variação na velocidade da transmissão de pacotes, o que também explica a maior variação entre os tempos mínimo e máximo de processamento no ATTILA 2.0 em relação ao ATTILA original.

Figura 23 – Comparação do tempo de processamento do ATTILA 2.0 com e sem o multiprocessamento ativado, multi-core e single-core, respectivamente. Em relação ao ATTILA original, é comparado também o tempo médio de execução por par de *rounds* (round) e a média da soma do tempo de execuções sucessivas para múltiplos *rounds* (soma). As barras pretas indicam os tempos de execução mínimos e máximos de cada triplicata.



Fonte: O autor

5.5 Desenvolvimento de um instalador

O Flatpak tem como um de seus principais diferenciais permitir a instalação de um programa em praticamente todos os sistemas operacionais Linux que estejam devidamente configurados. Uma outra grande vantagem é a facilitação do processo de instalação, que depende de poucas linhas de comando, e que em certos sistemas pode ser feita através da própria interface, sem nenhuma interação com o terminal. Por fim, a celeridade do processo de instalação é um diferencial importante no ATTILA 2.0, uma vez que a compilação de pacotes, a partir do código fonte, tende a ser um processo complexo, demorado e sujeito a falhas. O processo de instalação, que inclui a instalação do Flatpak, do Flathub, das dependências operacionais e do ATTILA 2.0, levou aproximadamente 15 minutos.

O instalador desenvolvido para o ATIILA 2.0 se mostrou estável em todos os testes, instalando com precisão todos os pacotes necessários ao processamento a ser executado na máquina do usuário. O instalador, que ficou limitado a um tamanho de 155 MB, foi capaz de englobar o código da interface, programas como FastQC, prinseq-lite e fastq-join, todas as bibliotecas do Python necessárias ao funcionamento, e os scripts do ATTILA 2.0 responsáveis por coordenar as análises.

5.6 Validação da ferramenta

CD20 é uma proteína de membrana encontrada principalmente em linfócitos B, por conta disso, é um alvo importante no tratamento de displasias imunológicas, como linfoma de não-Hodgkin e leucemia (SALLES *et al.*, 2017; HALLEK *et al.*, 2018). O rituximabe é um anticorpo monoclonal quimérico que se liga ao CD20 e é amplamente utilizado no tratamento de doenças imunológicas, incluindo linfomas, leucemias e doenças autoimunes. Por conta de sua relativa estabilidade e ampla gama de aplicações, o rituximabe tem estado no rol dos anticorpos monoclonais mais vendidos no mundo nos últimos anos (GRILO; MANTALARIS, 2019).

Embora tenha eficácia significativa, o que explica parte de seu sucesso mercadológico, o rituximabe ainda gera reações adversas em uma parcela dos pacientes, estando as reações anafiláticas dentre as mais comuns (LEVIN *et al.*, 2017). Em decorrência disso, e também visando maior eficácia em tratamentos, tem-se buscado alternativas ao rituximabe, baseadas principalmente em anticorpos humanizados e humanos. Nesse sentido, a técnica de *phage display* se torna uma importante aliada, ao possibilitar a proposição de um número considerável de mAbs e scFvs.

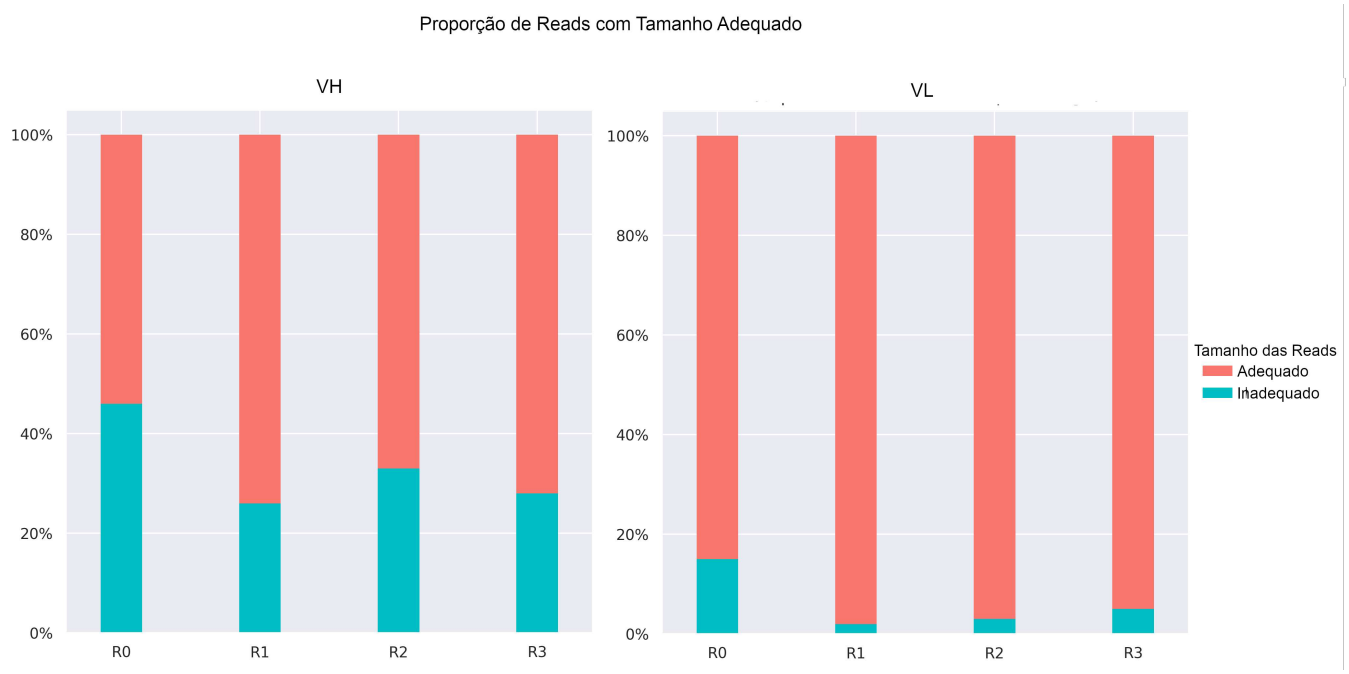
A validação da ferramenta desenvolvida foi feita através do processamento de bibliotecas proveniente do sequenciamento de quatro *rounds* de seleção por *phage display* de scFvs candidatos contra um peptídeo cíclico desenhado a partir de uma das alças da proteína CD20. A biblioteca inicial foi gerada através da técnica de *error-prone* PCR, em que a sequência de DNA codificadora do rituximabe foi replicada através da técnica de PCR com indução de mutações randômicas com o objetivo de gerar variação. Após a seleção, pela técnica de *biopanning*, as bibliotecas dos quatro *rounds* passaram por sequenciamento paired-end na plataforma Illumina, a partir da qual foram gerados os arquivos fastq utilizados no processo de validação do ATTILA 2.0. As bibliotecas NGS foram cedidas pelo Dr. Gilvan Pessoa Furtado, pesquisador da Fiocruz-CE.

A análise foi feita alterando-se como parâmetro o *round* de comparação. Em um primeiro momento, todos os *rounds* foram comparados com o *round* 0, em seguida a análise foi executada novamente comparando-se cada *round* com o seu antecessor (R3 x R2, R2 x R1, R1 x R0). Os arquivos listados no Apêndice B foram gerados para cada *round* (R_n; n = 1, 2, 3 ... n) com exceção do *round* 0, para o qual não são gerados resultados de enriquecimento, servindo ele como *round* de referência.

Através dos gráficos nas Figuras 24 e 25 pode-se observar o número de sequências remanescentes ao longo das etapas de processamento. Embora haja uma queda no número

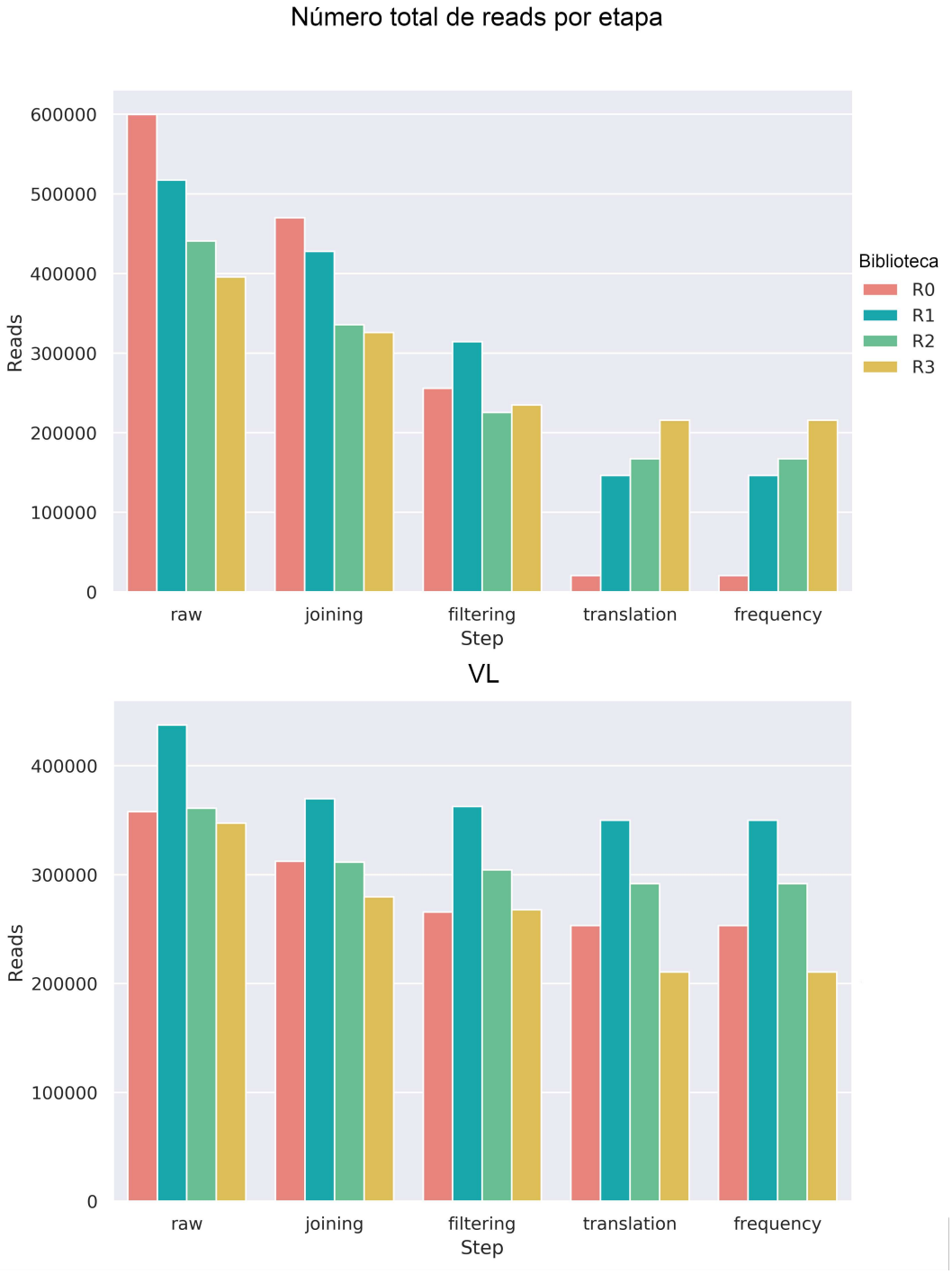
de sequências já na etapa de combinação de sequências (*joining*), a maior redução acontece nas etapas de filtragem, quando todas as sequências abaixo do tamanho mínimo (100 pb), com qualidade menor do que a mínima estabelecida (Phred Score = 20) ou com ausência dos aminoácidos conservados foram eliminadas. O R0 de VH foi o que mais perdeu sequências na etapa de filtragem. Nos resultados gerados pelo FastQC observa-se uma grande variação nos valores de qualidade na região inicial das sequências, entre o nucleotídeo na posição 1 e 34 (Figura 26). Observa-se contudo que a média dos valores de qualidade por nucleotídeo se mantém acima de 20 e só tende a decrescer nas bases finais das sequências, um resultado esperado em sequenciadores Illumina. Quando consideradas as sequências inteiras, tem-se uma grande quantidade de sequências com baixa qualidade (Phred Score = 2), mas a maioria tendo qualidade acima de 36. O R0 de VL, por outro lado, apresentou maior qualidade, como pode ser visto nos exemplos das Figuras 26, 27 e 28, e, por isso, apresentou menor perda de *reads* ao longo da análise (Figuras 24 e 25). Os resultados gerados pela ferramenta FastQ corroboram os que foram gerados pelo ATTILA 2.0 em termos de eliminação de sequências com baixa qualidade. Idealmente a biblioteca de VH apresentaria qualidade semelhante a de VL, contudo, o ATTILA 2.0 se mostrou capaz de tratar mesmo as bibliotecas com qualidade a baixo do ideal.

Figura 24 – Proporção de *reads* com tamanho adequado. Representação do percentual do total das *reads* de cada *round* eliminadas por terem tamanho menor do que o definido no início da análise (100 bases). Em verde e em vermelho, representação da proporção de *reads* com tamanho inadequado e adequado, respectivamente.



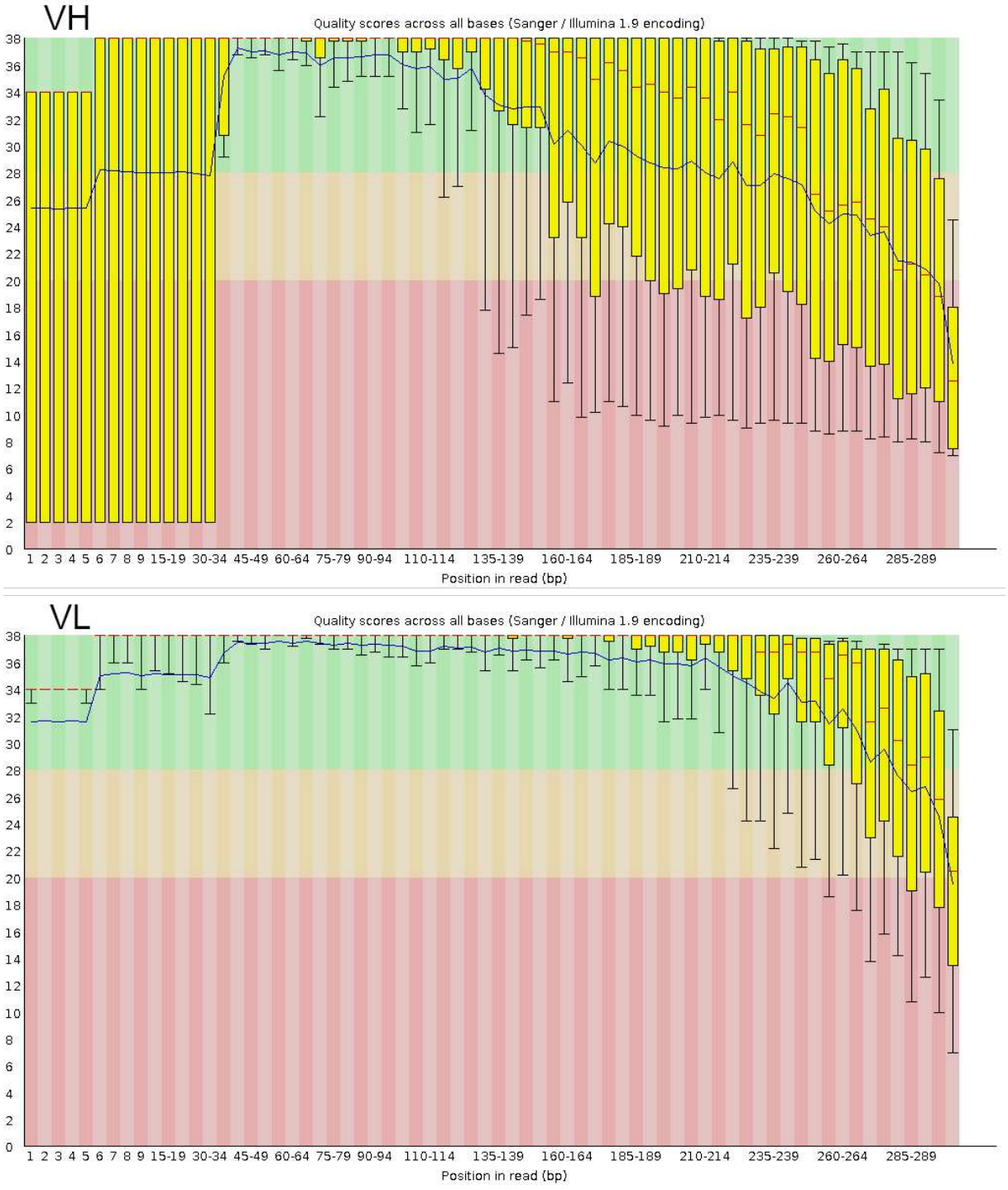
Fonte: O autor

Figura 25 – Número de *reads* em cada etapa a análise. Em "raw" estão representadas as quantidades de sequências em cada biblioteca antes do início do processamento. Em joining, filtering, translation e frequency estão representadas as contagens das sequências após as etapas de montagem, filtragem, tradução e cálculo de frequência relativa, respectivamente. Cada *round* é representado por uma cor.



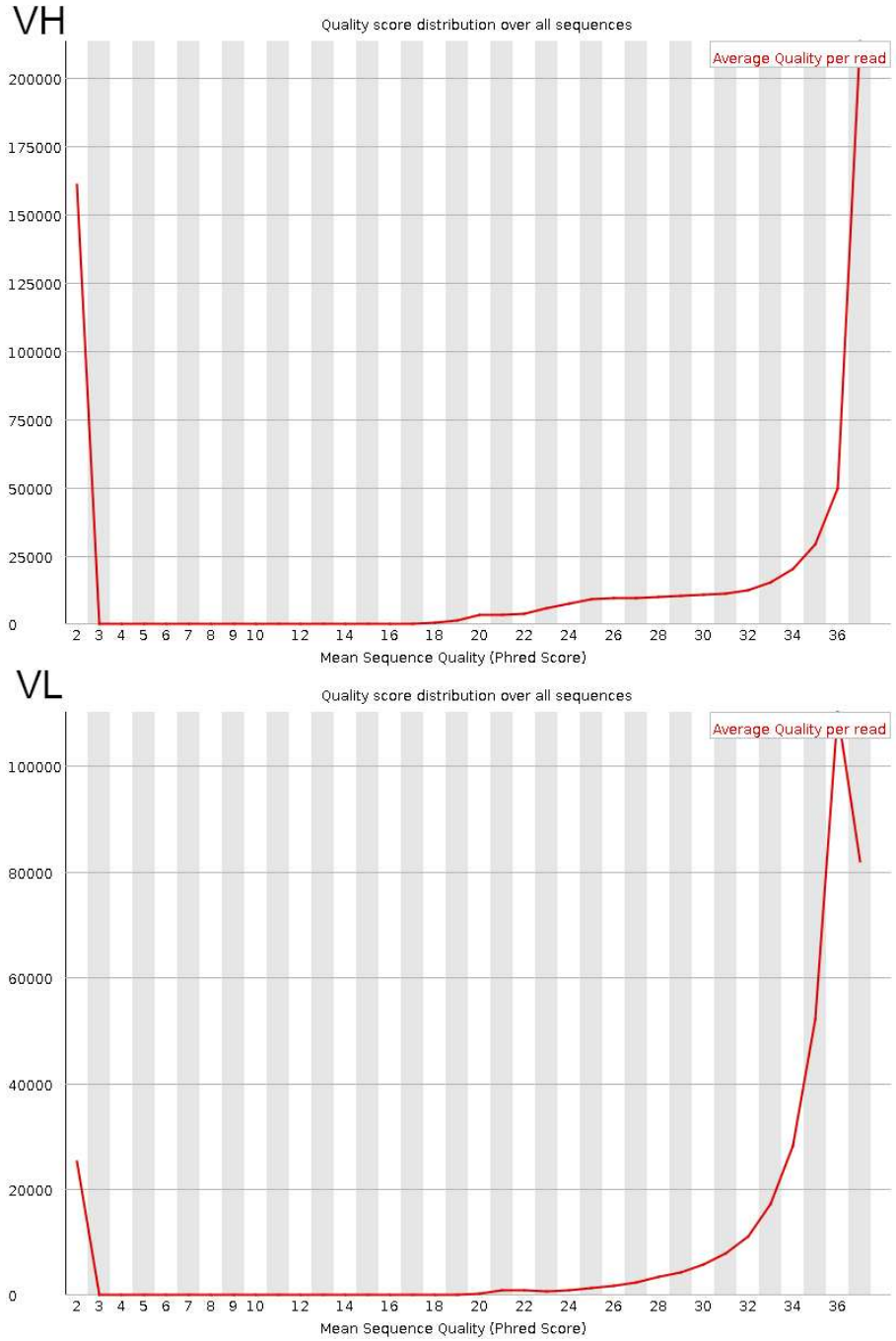
Fonte: O autor

Figura 26 – Gráfico gerado pelo software FastQC para as bibliotecas *foward* do R0 de VH e VL. As barras amarelas indicam a variação no nível de qualidade das sequências por posição (entre os percentis 25 e 75). As barras pretas também representam a variação de qualidade por posição, mas indo dos percentis 10 a 90. A linha vermelha dentro de cada barra representa a mediana dos valores de qualidade e a linha azul que atravessa o gráfico é uma representação da média. As regiões verde, laranja e vermelho indicam os valores de qualidade considerados pela documentação do FastQC como altos, médios e baixos, respectivamente.



Fonte: O autor

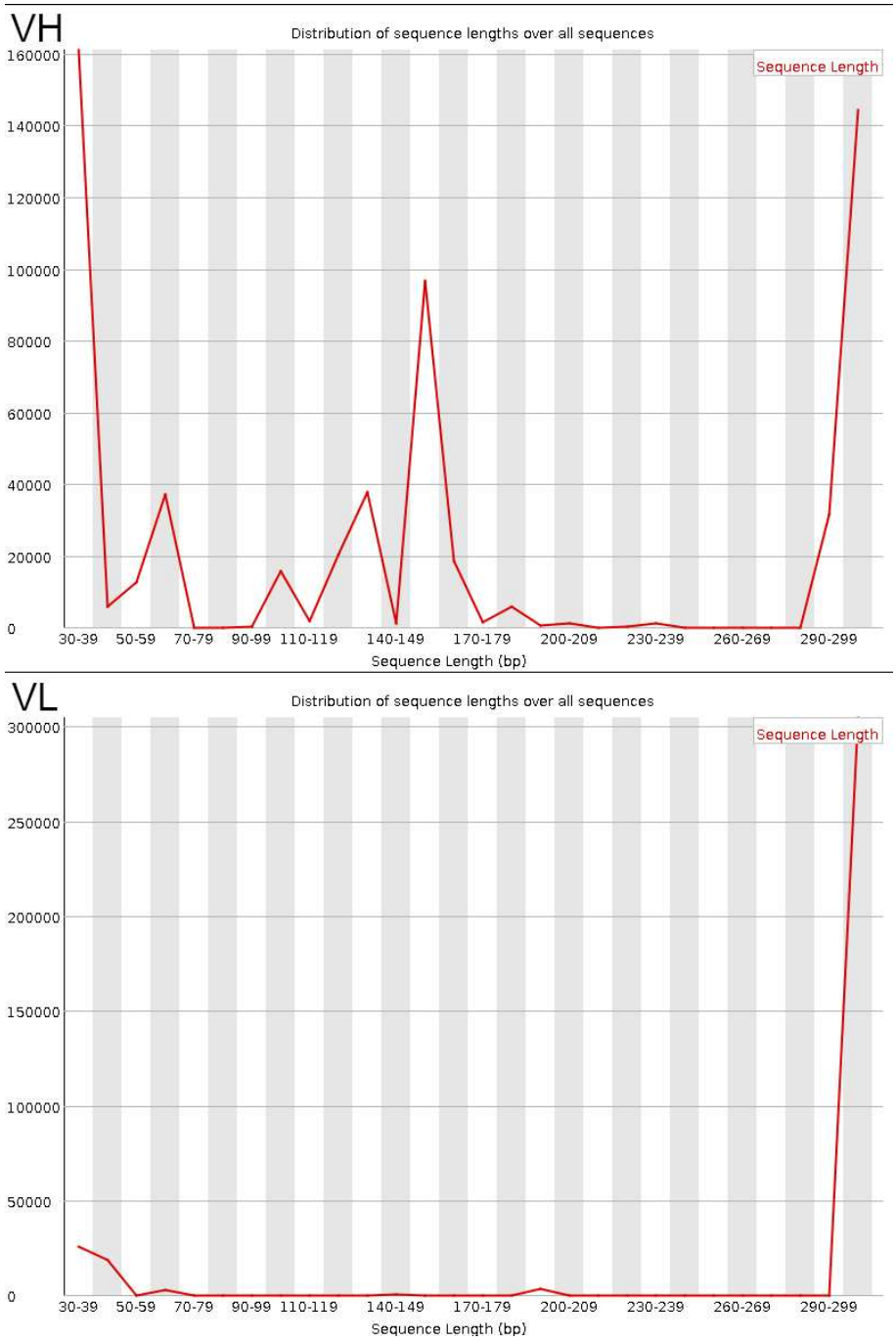
Figura 27 – Gráfico gerado pelo software FastQC para as bibliotecas *foward* do R0 de VH e VL. Representação do número de *reads* por qualidade média (Phred Score).



Fonte: O autor

Em relação aos modos de comparação, diferenças significativas foram encontradas nos resultados das sequências enriquecidas (R3 x R0 e R3 x R2). Das 10 sequências mais enriquecidas em cada modo de comparação, apenas 1 estava presente em ambos os resultados da análise de VH e 2 na análise de VL. Essa diferença era esperada, pois, como mostrado pelo trabalho de MARANHÃO *et al.* (2020), o enriquecimento das sequências nem sempre é ascendente ao longo dos *rounds* de seleção. A comparação R3 x R2, favoreceu sequências que

Figura 28 – Gráfico gerado pelo software FastQC para as bibliotecas *foward* do R0 de VH e VL. Representação da distribuição do número de sequências por tamanho.



Fonte: O autor

foram enriquecidas no último ciclo de seleção em detrimento das que vinham sendo enriquecidas nos *rounds* anteriores mas que foram menos evidentes no último *round* (R3). Já a comparação R3 x R0 favoreceu as sequências que foram mais enriquecidas no *round* final em relação ao inicial, independentemente das variações que ocorreram nos *rounds* intermediários.

Nas tabelas 1 e 2, pode-se observar que todos as sequências selecionadas tiveram maior identidade com a mesma germline (VH1-46), independentemente do modo de comparação.

A partir desse resultado pode-se supor que as sequências selecionadas não diferem consideravelmente entre si, mesmo assim há uma grande variação no *fold-change*, o que indica que as mutações foram decisivas na capacidade dos candidatos selecionados de se ligarem ao alvo. O mesmo pode ser observado para VL nas tabelas 3 e 4 contudo, com valores de identidade menores e com uma exceção nas germlines que em sua maioria foram V1-13, mas que em um dos casos foi V1-4. As informações de sequências germinais mais similares e dados de numeração são fundamentais em processos de otimização de anticorpos. As estratégias de humanização comumente usadas, por exemplo, se baseiam na busca por sequências humanas (em especial sequências germinais) com alta similaridade em relação a murina, que serve como arcabouço para recebimento das CDRs exógenas (PAVLINKOVA *et al.*, 2001).

Tabela 1 – Dados dos 10 candidatos de VH mais enriquecidos na comparação R3 x R0.

Ranking	**Fold Change**	**Germline**	**Identity**
1	1947.05	VH1-46	70.408
2	885.82	VH1-46	74.49
3	860.32	VH1-46	72.449
4	550.52	VH1-46	71.429
5	393.33	VH1-46	71.429
6	326.87	VH1-46	69.388
7	321.86	VH1-46	70.408
8	295.27	VH1-46	71.429
9	286.19	VH1-46	70.408
10	277.84	VH1-46	69.388

Fonte: O autor.

Tabela 2 – Dados dos 10 candidatos de VH mais enriquecidos na comparação R3 x R2.

Ranking	**Fold Change**	**Germline**	**Identity**
1	287.85	VH1-46	69.388
2	146.80	VH1-46	70.408
3	30.22	VH1-46	69.388
4	21.11	VH1-46	69.388
5	18.23	VH1-46	69.388
6	16.31	VH1-46	68.367
7	15.35	VH1-46	70.408
8	15.35	VH1-46	70.408
9	13.91	VH1-46	69.388
10	13.67	VH1-46	69.388

Fonte: O autor.

As sequências selecionadas pela execução do ATTILA original foram os mesmos dos selecionados pelo ATTILA 2.0. Os valores de *fold-change* também se mantiveram iguais, bem

Tabela 3 – Dados dos 10 candidatos de VL mais enriquecidos na comparação R3 x R0.

Ranking	Fold Change	Germline	Identity
1	18.20	V1-13	53.608
2	14.89	V1-13	51.546
3	11.42	V1-13	52.577
4	9.93	V1-13	52.577
5	8.93	V1-13	52.577
6	8.93	V1-13	52.577
7	8.44	V1-4	50.515
8	7.94	V1-13	53.608
9	7.61	V1-18	50.515
10	6.95	V1-13	53.608

Fonte: O autor.

Tabela 4 – Dados dos 10 candidatos de VL mais enriquecidos na comparação R3 x R2.

Ranking	Fold Change	Germline	Identity
1	90.97	V1-13	52.577
2	20.85	V1-13	53.608
3	10.23	V1-13	52.577
4	7.96	V1-13	51.546
5	6.82	V1-13	51.546
6	6.44	V1-4	50.515
7	5.69	V1-13	52.577
8	5.69	V1-13	52.577
9	5.69	V1-13	52.577
10	5.69	V1-13	51.546

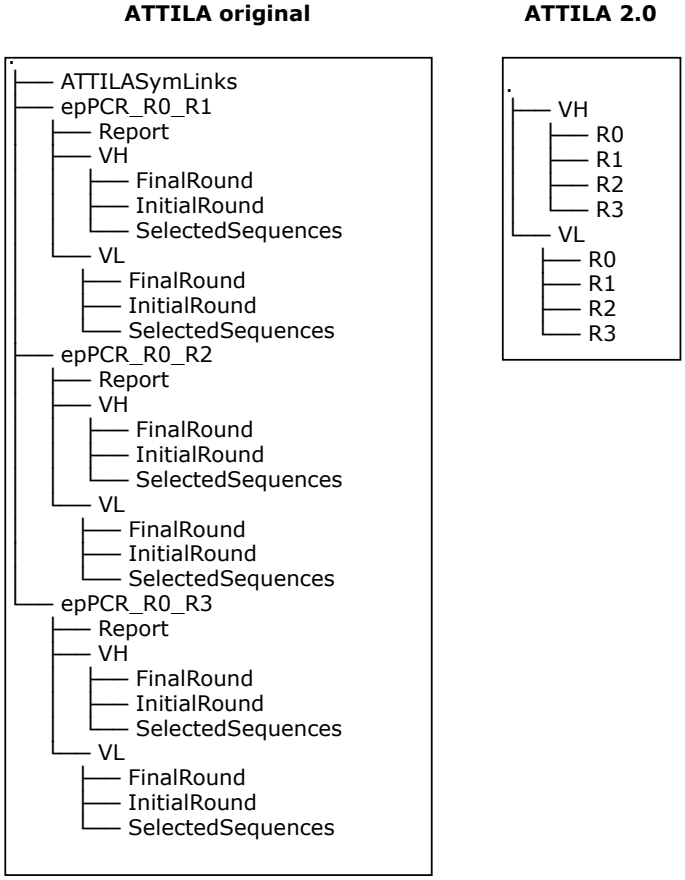
Fonte: O autor.

como as germlines selecionadas e os valores de identidade. Vale ressaltar que, para possibilitar a execução do ATTILA original, foram necessários ajustes em partes da ferramenta. Inicialmente, não foi possível alterar o tamanho mínimo para filtragem das reads devido a um erro no código do script responsável por receber os parâmetros da análise no ATTILA original. Além disso, o processo de tradução também não funcionou à princípio em virtude da desatualização do endereço de comunicação do ATTILA original com o servidor Abnum. Por fim, foi necessário corrigir o script *parserid.pl* do ATTILA original como foi feito para o ATTILA 2.0, uma vez que as sequências analisadas para validação da ferramenta contém o caracter especial "+"em seus identificadores, o que causava erro nos resultados e impedia o correto prosseguimento da análise.

Uma diferença nos resultados obtidos por ambas as versões está nos gráficos gerados. O ATTILA 2.0, por ser capaz de executar a análise de múltiplos rounds, gerou gráficos que comparam todos os *rounds* analisados. Já o ATTILA original, gerou um gráfico para cada comparação entre *rounds*. Da mesma forma, enquanto o ATTILA 2.0 possibilitou a comparação das sequências selecionadas em cada *round* em um mesmo relatório, as sequências selecionadas

pelo ATTILA original ficaram segmentadas em diretórios separados (Figura 29). Por fim, como resultado da repetida execução do ATTILA original para comparação dos múltiplos rounds, foram gerados dados redundantes ocupando 34.4 GB em disco, considerando os dois modos de comparação, enquanto os resultados gerados pelo ATTILA 2.0 ocuparam 26 GB.

Figura 29 – Esquematização dos diretórios gerados pelo ATTILA original e pelo ATTILA 2.0 para apenas um modo de comparação (R_n vs R0, para n = 1,2 e 3).



Fonte: O autor

Tabela 5 – Comparação das principais características do ATTILA 2.0 e do ATTILA original.

Características	**ATTILA 2.0**	**ATTILA original**
Interface gráfica	possui	não possui
Histórico de análises feitas	possui	não possui
Análise de múltiplos *rounds*	possui	não possui
Tempo médio de execução por *round*	14 minutos[1]	8 minutos
Estrutura de diretórios	simples	complexa
Ferramenta de numeração	ANARCI[2]	Abnum[3]
Modo de execução	local e remoto	local[4]
Principal linguagem de programação	Python	Perl
Instalador único	possui	não possui

Fonte: O autor.
Nota: [1] multi-core.
Nota: [2] Software instalável capaz de executar numeração de domínios variáveis de mAbs (DUNBAR; DEANE, 2015).
Nota: [3] Servidor web para numeração de domínios variáveis de mAbs (ABHINANDAN *et al.*, 2008).
Nota: [4] depende de serviços remotos de terceiros (Abnum).

6 DISCUSSÃO

Uma das dificuldades comuns na utilização do ATTILA é a falta de uma interface gráfica que torne a experiência do usuário mais amigável, facilitando a entrada dos parâmetros, das bibliotecas NGS e visualização dos resultados. Para utilizar a versão original do ATTILA, o usuário precisa possuir familiaridade com sistemas operacionais Linux, uma vez que todo o processo de execução se dá pelo terminal. Essa é uma característica que tem se tornado comum entre softwares de bioinformática desenvolvidos na academia, e tem como consequência o distanciamento dos pesquisadores de áreas experimentais, que tendem a ter apenas conhecimento básico do uso de computadores (SMITH, 2013).

O ATTILA 2.0 foi projetado tendo como premissa ser amigável ao usuário, de modo a facilitar o acesso às funcionalidades da ferramenta mesmo à pesquisadores de áreas experimentais. Assim, foi implementada uma interface gráfica com três telas para recebimento dos parâmetros e visualização dos resultados de forma simples. O segmento de histórico de análises feitas, denominado "*recents*" possibilita que o usuário rapidamente preencha os campos de parâmetros e os caminhos para as bibliotecas, levando a uma economia considerável de tempo. Além disso, é permitido ao usuário corrigir erros de digitação, e alternar entre a primeira e a segunda tela antes do início da análise, o que não era possível no ATTILA originalmente.

O ATTILA 2.0 vai de encontro ao princípio das ferramentas completamente baseadas em servidores web como a plataforma H++ (GORDON *et al.*, 2005). O ATTILA 2.0 tem parte considerável do processamento executada na máquina do usuário, o que dá liberdade para que o pesquisador busque computadores mais potentes que permitirão análise mais rápida, além de dar acesso integral a todos os arquivos gerados no processamento e permitir o acompanhamento próximo de cada etapa executada.

O ATTILA original tem como ponto forte o modo de geração do relatório final, em um arquivo HTML de fácil entendimento que pode ser aberto e visualizado em navegadores web. Por esse motivo, foi optado por construir a tela final da interface como uma *WebView*, com função semelhante a de um navegador, para possibilitar que o relatório continuasse sendo gerado em um arquivo HTML. A vantagem disso é a possibilidade de abertura e visualização do relatório diretamente em navegadores web, independentemente do ATTILA, após a análise. Um diferencial do relatório gerado pelo ATTILA 2.0 em relação ao original é que os gráficos e tabelas também passaram a ser gerados de forma dinâmica, se adaptando ao número de *rounds*. Deste modo, o usuário pode ter, reunido em um único lugar, os resultados principais referentes a

todos os *rounds* analisados. A migração da estruturação do HTML para Python3 com a biblioteca *Dominate*, garantiu uma grande melhoria na legibilidade do código, facilitando manutenções futuras. É válido ressaltar que a falta de manutenção em softwares desenvolvidos está entre as principais causas da baixa adesão ao uso de ferramentas desenvolvidas para análises de bioinformática (MANGUL *et al.*, 2019).

Uma vez submetidos os dados, é iniciado o processo de análise, tanto no ATTILA original, quanto no ATTILA 2.0. A reestruturação do código do ATTILA aconteceu em quatro fases. Na primeira, os scripts orquestradores principais, antes escritos na linguagem de programação Perl foram refeitos em Python3. As seções do código responsáveis por interagir diretamente com o sistema Linux e com softwares de terceiros foram segmentadas em um módulo responsável apenas por essa comunicação. Cada processo de comunicação externa foi convertido em uma função Python3, que passou a ser invocada pelo código principal de forma simplificada e padronizada. A refatoração do código na linguagem Python3 permite uma maior legibilidade e consequentemente facilita o processo de manutenção da ferramenta por outros desenvolvedores. Além disso, o Python tem grande importância na computação científica em grande parte devido a variedade de bibliotecas disponíveis e a sua capacidade de integração com ferramentas externas (LYU, 2022), fatores importantes tendo em vista o objetivo de oferecer novas funcionalidades para os usuários do ATTILA 2.0 no futuro.

Na segunda fase, as alterações visaram flexibilizar o código para o recebimento de dados de múltiplos *rounds*. Originalmente, o ATTILA é capaz de receber dados de dois rounds, final e inicial. Por conta disso, todo o código do ATTILA foi construído de modo que as etapas prévias à fase de comparação eram executadas com cada par de rounds, restringindo o número de *rounds* analisados. Assim, as etapas prévias à comparação entre rounds, executadas pelo ATTILA 2.0, foram reestruturadas para serem executadas com apenas um *round* por vez, tornando-se processos unitários, eliminando a restrição de análise aos pares e possibilitando a flexibilização do número de *rounds* analisados. As etapas de comparação entre os *rounds* também precisaram ser adaptadas, uma vez que anteriormente era feita apenas uma comparação, entre o *round* final e o inicial, e que passou a ser necessário fazer N-1 comparações, sendo N o número de *rounds*.

Já a terceira fase das alterações teve como foco principal favorecer o melhor aproveitamento dos recursos computacionais da máquina em que o ATTILA 2.0 for executado. Para isso, o código orquestrador da análise na máquina do usuário (execute6) foi adaptado para executar o processamento de todas as bibliotecas ao mesmo tempo, distribuindo as tarefas de análise para

os núcleos disponíveis no processador. Em decorrência disso, foi observada uma sobrecarga no uso de memória RAM que levou a travamentos sucessivos. A sobrecarga passou a acontecer durante a fase de tradução, devido ao carregamento de todas as sequências a serem traduzidas ao mesmo tempo. Assim, foi criado um sistema de segmentação das sequências a serem traduzidas, de modo que o processo de tradução passou a ser feito em bateladas. Foi criado ainda um funil que impede a paralelização do processo de tradução, eliminando completamente a sobrecarga da memória RAM e, consequentemente, problemas com travamentos.

A quarta fase visou, majoritariamente, a criação de um canal de integração dos resultados gerados pelo ATTILA 2.0 com outras ferramentas e serviços que venham a ser desenvolvidos pelo GEPeSS ou por grupos parceiros. Para isso, parte do processo de análise foi segmentado em uma API, hospedada em servidor da Fiocruz. De acordo com que análises são feitas, um BD é alimentado com as sequências de aminoácidos de VH e VL enviadas para a API. Tomou-se o cuidado de calcular o código hash dos arquivos de sequências e das próprias sequências recebidas a fim de minimizar redundâncias no banco de dados.

A paralelização do processamento trouxe redução significativa do tempo de análise feita pelo ATTILA 2.0 ao possibilitar um melhor aproveitamento dos recursos computacionais disponíveis em relação ao modo single-core. Já quando comparado ao ATTILA original, embora o tempo médio de análise por *round* tenha sido menor no ATTILA original, a sobreposição de processos pode levar a travamentos por uso excessivo de memória. Deste modo, faz-se necessária a execução consecutiva dos processos de análise para mais de 2 rounds, e neste caso o ATTILA original apresentou maior tempo de análise do que o ATTILA 2.0 em modo *multi-core*. Vale ressaltar que, por mais de uma vez, o processo de tradução no ATTILA original foi interrompido pelo sistema operacional, gerando apenas parte das sequências traduzidas e, consequentemente, alterando os valores de *fold-change*. Em momento algum o usuário é avisado deste problema, podendo tomar resultados errôneos como verdadeiros. Ao fazer melhor gerenciamento de memória, o ATTILA 2.0 evita este problema.

O maior tempo médio de análise por *round* no ATTILA 2.0 em relação do ATTILA original também pode ser explicado pela divisão da execução da ferramenta entre o computador do usuário e a API, o que torna o tempo de análise dependente da qualidade da conexão com a internet no computador do usuário. A necessidade de modificação do processo de tradução adicionou novas etapas à análise tendo também impacto no tempo de processamento.

Uma vez que, através do sequenciamento NGS, é possível obter a sequência de

centenas de milhares de VHs e VLs, acredita-se que o BD em questão tem potencial para revelar tendências e restrições naturais das sequências e estruturas de VHs e VLs viáveis e, assim, direcionar o processo de evolução e aprimoramento através de mutações sítio-dirigidas de sequências já selecionadas, usando evolução *in silico* ou desenho racional (CANNON *et al.*, 2019; WARSZAWSKI *et al.*, 2019; PITTALA; BAILEY-KELLOGG, 2020).

Devido ao grande número de programas que constituem o ATTILA, o processo de instalação e configuração do *pipeline* é laborioso e demorado. Vários programas encontravam-se desatualizados ou não podiam mais ser encontrados nas páginas de download originais. Outros programas, como o Prinseq-lite, necessitam de alterações no código fonte para funcionarem em alguns sistemas Linux. Por isso foi desenvolvido um instalador com a tecnologia Flatpak que tem como principal diferencial a capacidade de funcionar em praticamente todos os tipos de sistemas Linux, e necessitar de poucos passos para executar a instalação. O processo de instalação que, quando executado manualmente, podia levar horas, passou a ser executado em aproximadamente 15 minutos. Mangul *et al.* (2019) constatou que 49% de 99 ferramentas de bioinformática analisadas não apresentavam modo de instalação simples e eficiente, e que a dificuldade no processo de instalação tem impacto considerável na popularidade dos softwares desenvolvidos gerando menos citações.

A fim de demonstrar o funcionamento do ATTILA 2.0, foi feita a análise de dados provenientes do sequenciamento NGS de bibliotecas de quatro *rounds* de seleção por *phage display*. A seleção teve como alvo molecular uma das alças da proteína de membrana CD20, usada com peptídeo cíclico. A partir dos resultados gerados pela ferramenta foi possível obter o número de sequências obtidas pelo sequenciamento de em cada biblioteca, assim como o número de sequências remanescentes após cada etapa do processamento. Através da execução do software FastQC, foram gerados relatórios de qualidade das bibliotecas antes e após o processo de filtragem, que se mostrou eficiente na remoção das reads com tamanho inferior ao desejado e com baixa qualidade.

A análise foi executada duas vezes, com os dois modos de comparação entre *rounds* possíveis. O ATTILA 2.0 foi capaz de identificar as sequências mais enriquecidas em cada *round*, assim como quantificar o nível do enriquecimento (*fold-change*). Em relação à comparação do *round* das bibliotecas de VH, observou-se que os valores de *fold-change* foram consideravelmente maiores na comparação R3xR0 do que na R3xR2, resultado esperado já que a divergência de composição entre bibliotecas dos *rounds* R3 e R2 tende a ser menor, tendo ambas passado por

mais de um ciclo de seleção. Já para as bibliotecas de VL observou-se o contrário, provavelmente por ter havido pouca variação das sequências com maior frequência relativa. Naturalmente, o cálculo de enriquecimento pode ter favorecido sequências com baixa frequência relativa no R2, mas que tenham se tornado mais representativos no R3. Além disso, assim como os valores de *fold-change*, as sequências selecionadas também foram diferentes em ambos os tipos de comparação. Apenas uma sequências de VH e duas de VL estavam em ambos os conjuntos de sequências selecionadas.

Por fim, a comparação dos resultados do ATTILA 2.0 com o original mostrou que os resultados obtidos são idênticos, em especial as sequências de VH e VL selecionadas, seus respectivos valores de *fold-change* e as *germlines* identificadas. As sequências selecionadas ao fim do processo são candidatas à análise posterior tanto estrutural quando experimental, sendo necessária a combinação de diferentes VHs e VLs para a formação de scFvs que possam ser aplicados para o fim desejado. Essa combinação é necessária por conta da incapacidade dos sequenciadores mais utilizados nos dias de hoje de obter as sequências completas dos scFvs selecionados em cada *round* de *phage display*.

7 CONCLUSÃO

O desenvolvimento da ferramenta ATTILA 2.0, baseada no *pipeline* ATTILA, possibilitou prover diferenciais como a facilidade de uso e instalação, bem como a estabilidade e melhor desempenho computacional. A interface gráfica desenvolvida torna viável a utilização do ATTILA 2.0 por usuários menos experientes com sistemas operacionais Linux. O processo de instalação foi consideravelmente simplificado ao unificar em um único instalador todas as dependências necessárias à análise dos dados NGS. Através da reestruturação do código, foi possível flexibilizar o número de *rounds* analisados e comparados em uma mesma submissão, tornando desnecessária a execução consecutiva da ferramenta. Além disso, as otimizações feitas permitiram o melhor aproveitamento dos recursos computacionais disponíveis na máquina do usuário, diminuindo o tempo de análise.

Embora eficiente na identificação das sequências que melhor se destacaram ao longo dos *rounds* de seleção, o método de análise apresenta limitações. O fato de receber as bibliotecas de VH e VL de forma independente inviabiliza a identificação dos pares que formavam os scFvs originais, tornando necessário o teste experimental de diferentes combinações dos VHs e VLs mais enriquecidos. Outras limitações herdadas do *pipeline* original são a provável incapacidade de analisar sequências de espécies que apresentem imunoglobulinas consideravelmente diferentes das humanas, como as de galinha, tubarão e camelídeo. Isso ocorre porque no processo de tradução o programa translateab9 se baseia na identificação de aminoácidos canônicos que podem não estar presentes nas cadeias variáveis de imunoglobulinas de outras espécies.

O processo de contagem dos clones se baseia na identificação exata da subsequência formada pelas regiões CDR1, FR2, CDR2, FR3, CDR3, de forma que a busca se torna extremamente sensível a pequenas variações que podem ter sido causadas por erros no sequenciamento. Desta forma, pode acontecer de sequências que deveriam ser consideradas como mesmo clone, serem tratadas como clones diferentes, o que acabaria por alterar os valores de fold-change e, potencialmente, a lista de clones mais enriquecidos. Em todo caso, como ressaltado por Silva (2016), a alternativa, que seria a comparação das sequências por alinhamento, poderia tornar o processo de análise extremamente lento e não garantiria um resultado satisfatório.

Por ser associado a uma API, o ATTILA 2.0 permite o compartilhamento voluntário de sequências de VH e VL com a Fiocruz através do GEPeSS por parte do usuário. O compartilhamento de sequências possibilitará a alimentação massiva do banco de dados de sequências que deve ocorrer a partir da utilização da ferramenta possibilitando uma maior compreensão das

tendências naturalmente existentes nas sequências de scFvs viáveis, como, por exemplo, quais aminoácidos são mais frequentes em cada posição. Esse entendimento associado a técnicas de análise de dados e inteligência artificial diminuirá o espaço de busca por novos scFvs e dará celeridade ao desenvolvimento de novos tratamentos e ferramentas de diagnóstico baseadas em mAbs.

8 TRABALHOS FUTUROS

Com o rápido desenvolvimento da tecnologias de sequenciamento NGS, já é possível a obtenção de reads com mais de 600 bases, o suficiente para cobrir toda a sequência da maioria dos scFvs (NANNINI, 2020; MAGI *et al.*, 2016). Deste modo, o ATTILA 2.0 deverá futuramente aceitar esse novo tipo de dado, que solucionará o problema da associação entres os candidatos VH e VL enriquecidos.

Em relação à aceitação de sequências de outras espécies, estudos estão sendo feitos no grupo, visando a busca por padrões em nanocorpos (VHH) de camelídeos, o que possibilitará o aumento do número de espécies aceitas para análise pelo ATTILA 2.0. Consequentemente, será favorecido o recebimento de sequências de outras espécies que alimentarão do BD de repertório possibilitando o desenvolvimento de novos VHHs, que possuem diversas vantagens para aplicações terapêuticas (LI *et al.*, 2022; GREVE *et al.*, 2020).

Diante do surgimento de tecnologias inovadoras no campo da bioinformática, como o AlphaFold (JUMPER *et al.*, 2021; VARADI *et al.*, 2021), torna-se possível também a implementação da função de modelagem da estrutura de candidatos selecionados integrada ao ATTILA 2.0. Baseado na estrutura obtida, será possível a análise das interações intramoleculares e até das interações intermoleculares, caso haja também a estrutura do alvo.

A partir do conhecimento das estruturas dos scFvs e das tendências observadas no banco de dados de sequências, associado a técnicas de Aprendizado de Máquina, espera-se que seja possível a proposição automática de otimizações na sequência de aminoácidos que visem o aumento da afinidade, a diminuição da imunogenicidade, aumento da termoestabilidade e diminuição da tendência à formação de corpos de inclusão. Desta forma, tornando mais rápido o processo de desenvolvimento de biofármacos e ferramentas de diagnósticos baseados em mAbs.

A análise integral do repertório de sequências contidas em cada biblioteca pode revelar informações valiosas. Novas análises podem ser implementadas no sentido de identificar a distribuição de germlines nas bibliotecas inteiras. Além disso, a construção de árvores filogenéticas a partir das sequências únicas de cada candidato, podem possibilitar um entendimento mais profundo das mutações que são favorecidas ao longo da seleção (MANKOWSKA *et al.*, 2016).

Por fim, deverá ser construído um termo de consentimento que será apresentado ao usuário, para que ele possa aceitar ou não o compartilhamento voluntário das sequências de VH e VL analisadas.

REFERÊNCIAS

ABHINANDAN, K. *et al.* Analysis and improvements to kabat and structurally correct numbering of antibody variable domains. **Molecular Immunology**, [*s.l.*], v. 45, n. 14, p. 3832–3839, ago. 2008.

ALANGODE, A.; RAJAN, K.; NAIR, B. G. Snake antivenom: challenges and alternate approaches. **Biochemical Pharmacology**, [*s.l.*], v. 181, p. 1–8, nov. 2020.

ALBERTS, B.; JOHNSON, A.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WALTER, P. **Molecular Biology of the Cell**. [*S. l.*]: Garland Science, 2002. v. 1.

ANDREWS, S.; KRUEGER, F.; Segonds-Pichon, A.; BIGGINS, L.; KRUEGER, C.; WINGETT, S. **FastQC: a quality control tool for high throughput sequence data. A Quality Control Tool for High Throughput Sequence Data**. Babraham: [*Babraham Institute*], 2010. Disponível em: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. Acesso em: 01 feb. 2023.

ARONESTY, E. Comparison of sequencing utility programs. **The Open Bioinformatics Journal**, [*s.l.*], v. 7, n. 1, p. 1–8, jan. 2013.

BATRA, S. K. *et al.* Pharmacokinetics and biodistribution of genetically engineered antibodies. **Current Opinion In Biotechnology**, [*s.l.*], v. 13, n. 6, p. 603–608, dez. 2002.

CANNON, D. A. *et al.* Experimentally guided computational antibody affinity maturation with de novo docking, modelling and rational design. **Plos Computational Biology**, [*s.l.*], v. 15, n. 5, p. 1–22, 2019.

CHIU, M. L. *et al.* Antibody structure and function: the basis for engineering therapeutics. **Antibodies**, [*s.l.*], v. 8, n. 4, p. 1–80, dez. 2019.

CHOTHIA, C.; LESK, A. M. Canonical structures for the hypervariable regions of immunoglobulins. **Journal of Molecular Biology**, [*s.l.*], Elsevier BV, v. 196, n. 4, p. 901–917, ago. 1987. Disponível em: https://doi.org/10.1016/0022-2836(87)90412-8.

DAI, J.-M. *et al.* Modified therapeutic antibodies: improving efficacy. **Engineering**, [*s.l.*], v. 7, n. 11, p. 1529–1540, nov. 2021.

DIAS-NETO, E. *et al.* Next-generation phage display: integrating and comparing available molecular tools to enable cost-effective high-throughput analysis. **Plos One**, [*s.l.*], v. 4, n. 12, p. 1–11, dez. 2009.

DUNBAR, J.; DEANE, C. M. Anarci: antigen receptor numbering and receptor classification. **Bioinformatics**, [*s.l.*], v. 32, n. 2, p. 298–300, set. 2015.

FOOTE, J.; WINTER, G. Antibody framework residues affecting the conformation of the hypervariable loops. **Journal of Molecular Biology**, [*s.l.*], Elsevier BV, v. 224, n. 2, p. 487–499, mar. 1992. Disponível em: https://doi.org/10.1016/0022-2836(92)91010-m.

GORDON, J. C.; MYERS, J. B.; FOLTA, T.; SHOJA, V.; HEATH, L. S.; ONUFRIEV, A. H: a server for estimating pKas and adding missing hydrogens to macromolecules. **Nucleic Acids Research**, [*s.l.*], Oxford University Press (OUP), v. 33, n. Web Server, p. W368–W371, jul. 2005. Disponível em: https://doi.org/10.1093/nar/gki464.

GREVE, H. d. *et al.* Simplified monomeric vhh-fc antibodies provide new opportunities for passive immunization. **Current Opinion In Biotechnology**, [*s.l.*], v. 61, p. 96–101, fev. 2020.

GRILO, A. L.; MANTALARIS, A. The increasingly human and profitable monoclonal antibody market. **Trends In Biotechnology**, [*s.l.*], v. 37, n. 1, p. 9–16, jan. 2019.

HALLEK, M.; SHANAFELT, T. D.; EICHHORST, B. Chronic lymphocytic leukaemia. **The Lancet**, [*s.l.*], v. 391, n. 10129, p. 1524–1537, abr. 2018.

HEATHER, J. M. *et al.* The sequence of sequencers: the history of sequencing dna. **Genomics**, [*s.l.*], v. 107, n. 1, p. 1–8, jan. 2016.

HWANG, W. Y. K. *et al.* Immunogenicity of engineered antibodies. **Methods**, [*s.l.*], v. 36, n. 1, p. 3–10, 2005.

JUMPER, J. *et al.* Highly accurate protein structure prediction with alphafold. **Nature**, [*s.l.*], v. 596, n. 7873, p. 583–589, jul. 2021.

KABAT, E. A.; WU, T. T.; BILOFSKY, H. Unusual distributions of amino acids in complementarity-determining (hypervariable) segments of heavy and light chains of immunoglobulins and their possible roles in specificity of antibody-combining sites. **Journal of Biological Chemistry**, [*s.l.*], Elsevier BV, v. 252, n. 19, p. 6609–6616, out. 1977. Disponível em: https://doi.org/10.1016/s0021-9258(17)39891-5.

KROHN, S. *et al.* Identification of new antibodies targeting malignant plasma cells for immunotherapy by next-generation sequencing-assisted phage display. **Frontiers In Immunology**, [*s.l.*], v. 13, n. 1, p. 1–16, jun. 2022.

LEFRANC, M.-P. Unique database numberings system for immunogenetic analysis. **Immunology Today**, [*s.l.*], Elsevier BV, v. 18, n. 11, p. 509, nov. 1997. Disponível em: https://doi.org/10.1016/s0167-5699(97)01163-8.

LEOW, C.; FISCHER, K.; LEOW, C.; CHENG, Q.; CHUAH, C.; MCCARTHY, J. Single domain antibodies as new biomarker detectors. **Diagnostics**, [*s.l.*], MDPI AG, v. 7, n. 4, p. 52, out. 2017. Disponível em: https://doi.org/10.3390/diagnostics7040052.

LEVIN, A. S. *et al.* Reactions to rituximab in an outpatient infusion center: a 5-year review. **The Journal Of Allergy And Clinical Immunology: In Practice**, [*s.l.*], v. 5, n. 1, p. 107–113, jan. 2017.

LI, Q.; ZHANG, F.; LU, Y.; HU, H.; WANG, J.; GUO, C.; DENG, Q.; LIAO, C.; WU, Q.; HU, T. Highly potent multivalent vhh antibodies against chikungunya isolated from an alpaca naïve phage display library. **Journal Of Nanobiotechnology**, [*s.l.*], v. 20, n. 1, p. 1–15, maio 2022.

LIMA, A. J. F. **Estudo da influência do epítopo c-myc na formação da interface entre o scFv e o CD19 para aplicação em CAR.** 2022. 99 f. Dissertação (Mestrado) – Curso de Mestrado em Biotecnologia de Recursos Naturais, Universidade Federal do Ceará, Fortaleza, 2022.

LIN, C.-C.; SHIH, C.-P.; WANG, C.-C.; OUYANG, C.-H.; LIU, C.-C.; YU, J.-S.; LO, C.-H. The clinical usefulness of taiwan bivalent freeze-dried hemorrhagic antivenom in protobothrops mucrosquamatus- and viridovipera stejnegeri-envenomed patients. **Toxins**, [*s.l.*], v. 14, n. 11, p. 1–13, nov. 2022.

LLEWELYN, M. B.; HAWKINS, R. E.; RUSSELL, S. J. Discovery of antibodies. **Bmj**, [*s.l.*], v. 305, n. 6864, p. 1269–1272, nov. 1992.

LYU, J. Applications of python programming language in bioinformatics field. **Journal of Proteomics Bioinformatics**, [*s.l.*], v. 15, n. 1000588, p. 1, 2022.

MAGI, A. *et al.* Characterization of minion nanopore data for resequencing analyses. **Briefings In Bioinformatics**, [*s.l.*], p. 940–953, ago. 2016.

MAKABE, K.; NAKANISHI, T.; TSUMOTO, K.; TANAKA, Y.; KONDO, H.; UMETSU, M.; SONE, Y.; ASANO, R.; KUMAGAI, I. Thermodynamic consequences of mutations in vernier zone residues of a humanized anti-human epidermal growth factor receptor murine antibody. **Journal of Biological Chemistry**, [*s.l.*], Elsevier BV, v. 283, n. 2, p. 1156–1166, jan. 2008. Disponível em: https://doi.org/10.1074/jbc.m706190200.

MANGUL, S.; MARTIN, L. S.; ESKIN, E.; BLEKHMAN, R. Improving the usability and archival stability of bioinformatics software. **Genome Biology**, [*s.l.*], Springer Science and Business Media LLC, v. 20, n. 1, fev. 2019. Disponível em: https://doi.org/10.1186/s13059-019-1649-8.

MANKOWSKA, S. A. *et al.* A shorter route to antibody binders via quantitative in vitro bead-display screening and consensus analysis. **Scientific Reports**, [*s.l.*], v. 6, n. 1, p. 1–11, nov. 2016.

MARANHÃO, A. Q. *et al.* Discovering selected antibodies from deep-sequenced phage-display antibody library using attila. **Bioinformatics And Biology Insights**, [*s.l.*], v. 14, p. 1–8, jan. 2020.

MARDIS, E. R. Next-generation sequencing platforms. **Annual Review Of Analytical Chemistry**, [*s.l.*], v. 6, n. 1, p. 287–303, jun. 2013.

MEHMOOD, M. A. Use of bioinformatics tools in different spheres of life sciences. **Journal Of Data Mining In Genomics Proteomics**, [*s.l.*], v. 05, n. 02, p. 1–13, jan. 2014.

METZKER, M. L. *et al.* Sequencing technologies — the next generation. **Nature Reviews Genetics**, [*s.l.*], v. 11, n. 1, p. 31–46, dez. 2009.

MULLARD, A. *et al.* Fda approves 100th monoclonal antibody product. **Nature Reviews Drug Discovery**, [*s.l.*], v. 20, n. 7, p. 491–495, maio 2021.

MUÑOZ-LÓPEZ, P. *et al.* Single-chain fragment variable: recent progress in cancer diagnosis and therapy. **Cancers**, [*s.l.*], v. 14, n. 17, p. 1–26, ago. 2022.

NANNINI, F. e. a. Combining phage display with smrtbell next-generation sequencing for the rapid discovery of functional scfv fragments. **Mabs**, [*s.l.*], v. 13, n. 1, p. 1–12, dez. 2020.

OLIVEIRA, N. F. F. **Simulação de dinâmica molecular de um modelo de CAR em interação com CD10, marcador de células cancerosas.** 2020. 128 f. Dissertação (Mestrado) – Curso de Mestrado em Biotecnologia de Recursos Naturais, Universidade Federal do Ceará, Fortaleza, 2020.

PAVLINKOVA, G. *et al.* Effects of humanization and gene shuffling on immunogenicity and antigen binding of anti-tag-72 single-chain fvs. **International Journal Of Cancer**, [*s.l.*], v. 94, n. 5, p. 717–726, 2001.

PITTALA, S.; BAILEY-KELLOGG, C. Learning context-aware structural representations to predict antigen and antibody binding interfaces. **Bioinformatics**, [*s.l.*], v. 36, n. 13, p. 3996–4003, abr. 2020.

REBOUÇAS, A. d. S. **Estudo da interação do scFv do anticorpo rituximab com a alça do receptor CD20: avaliação da energia livre de ligação pelo método abf para proposição de *biobetters*.** 2018. 84 f. Dissertação (Mestrado) – Curso de Mestrado em Programa de Pós-Graduação de Biologia Computacional e Sistemas, Instituto Oswaldo Cruz, Rio de Janeiro, 2018.

RODRIGUES, F. N. **Análise das estruturas de fragmentos de anticorpos VH e VL com potencial para aplicação *in silico*** 2018. 44 f. Monografia (Bacharelado) – Bacharelado em Biotecnologia, Universidade Federal do Ceará, Fortaleza, 2014.

RONAGHI, M. *et al.* Pyrosequencing sheds light on dna sequencing. **Genome Research**, [*s.l.*], v. 11, n. 1, p. 3–11, jan. 2001.

SAFDARI, Y. *et al.* Antibody humanization methods – a review and update. **Biotechnology And Genetic Engineering Reviews**, [*s.l.*], v. 29, n. 2, p. 175–186, out. 2013.

SALLES, G. *et al.* Rituximab in b-cell hematologic malignancies: a review of 20 years of clinical experience. **Advances In Therapy**, [*s.l.*], v. 34, n. 10, p. 2232–2273, out. 2017.

SANGER, F. *et al.* Dna sequencing with chain-terminating inhibitors. **Proceedings Of The National Academy Of Sciences**, [*s.l.*], v. 74, n. 12, p. 5463–5467, dez. 1977.

SILVA, H. M. **Metodo in silico para analise de sequencias de imunoglobulinas produzidas por tecnologia de phage display.** 2016. 89 f. Dissertação (Mestrado) – Curso de Mestrado em Programa de Pós-Graduação de Biologia Computacional e Sistemas, Instituto Oswaldo Cruz, Brasília, 2016.

SINGH, S. *et al.* Monoclonal antibodies: a review. **Current Clinical Pharmacology**, [*s.l.*], v. 13, n. 2, p. 85–99, out. 2018.

SMITH, D. R. The battle for user-friendly bioinformatics. **Frontiers in Genetics**, [*s.l.*], Frontiers Media SA, v. 4, 2013. Disponível em: https://doi.org/10.3389/fgene.2013.00187.

SMITH, G. P. *et al.* Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface. **Science**, [*s.l.*], v. 228, n. 4705, p. 1315–1317, jun. 1985.

SRIAPHA, C. *et al.* Early adverse reactions to snake antivenom: poison center data analysis. **Toxins**, [*s.l.*], v. 14, n. 10, p. 1–19, out. 2022.

STADEN, R. A strategy of dna sequencing employing computer programs. **Nucleic Acids Research**, [*s.l.*], v. 6, n. 7, p. 2601–2610, mar. 1979.

VARADI, M. *et al.* Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. **Nucleic Acids Research**, [*s.l.*], v. 50, n. 1, p. 439–444, nov. 2021.

WARSZAWSKI, S. *et al.* Optimizing antibody affinity and stability by the automated design of the variable light-heavy chain interfaces. **Plos Computational Biology**, [*s.l.*], v. 15, n. 8, p. 1–24, ago. 2019.

WONG, K.-C. *et al.* Dna sequencing technologies. **Acm Computing Surveys**, [*s.l.*], v. 52, n. 5, p. 1–30, set. 2019.

WU, T. T.; KABAT, E. A. An analysis of the sequences of the variable regions of bence jones proteins and myeloma light chains and their implications for antibody complementarity. **Journal of Experimental Medicine**, [*s.l.*], Rockefeller University Press, v. 132, n. 2, p. 211–250, ago. 1970. Disponível em: https://doi.org/10.1084/jem.132.2.211.

YE, J. *et al.* Igblast: an immunoglobulin variable domain sequence analysis tool. **Nucleic Acids Research**, [*s.l.*], v. 41, n. 1, p. 34–40, maio 2013.

APÊNDICE A – DESCRIÇÃO DOS CAMPOS DA TELA 1

Campo	Descrição
Nome do projeto	Nomeia o diretório onde são salvos os arquivos resultantes do processamento.
Diretório do projeto	Local onde a pasta nomeada é criada.
Tipo de sequenciamento	Define se é paired-end ou não. Afeta a construção da Tela 2, uma vez que, sendo paired-end o número de arquivos gerados pelo sequenciamento é o dobro em relação ao single-end.
Comparar com round inicial	Determina como os rounds serão comparados entre si; se marcado “sim”, as comparações acontecem em relação ao round inicial, se marcado “não”, cada round é comparado com seu antecessor.
Tamanho mínimo das reads	Define o limiar de corte das reads oriundas do sequenciamento por tamanho, tendo 300 nucleotídeos como valor padrão.
Qualidade mínima	Valor na escala Phred utilizado na filtragem das reads; tem 20 como valor padrão o que representa a probabilidade máxima de 1 erro a cada 100 nucleotídeos.
Mínimo de candidatos	Número de candidatos que serão selecionados ao fim do processo.
Número de rounds	Número de ciclos de seleção sequenciados; afeta a construção da Tela 2.
Recents	Histórico das últimas análises feitas. Ao clicar em um registro todos os campos são preenchidos com os valores utilizados na análise do registro em questão.
Botões	Cancelar análise e avançar para a próxima tela.

Fonte: O autor.

APÊNDICE B – ARQUIVOS GERADOS PELO ATTILA 2.0

Ferramenta	Arquivo	Descrição
FastQC	<biblioteca>_fastqc.html	Relatório de qualidade das bibliotecas
	<biblioteca>_fastqc.zip	Arquivos estáticos do relatório
FastqJoin	<round>join	Arquivo fastq com sequências montadas
	<round>un1	Reads antes da montagem
	<round>un2	
parserid.pl	<round>newid.fq	<round>join com espaços removidos dos identificadores
Prinseq-Lite	<round>.before.fasta	<round>newid.fq antes da filtragem
	<round>.after.fasta	Sequências filtradas no formato fasta
	<round>.after.fastq	Sequências filtradas no formato fastq
	<round>.after.qual	Valores de qualidade para cada nucleotídeo das sequências filtradas
FastQC	<round>_fastqc.html	Relatório de qualidade das das sequências após filtragem
	<round>_fastqc.zip	Arquivos estáticos do relatório
translateab	<round>.aa.fasta	Sequências filtradas pela presença dos aminoácidos marcadores e traduzidas
	<round>.nt.fasta	Sequências filtradas pela presença dos aminoácidos marcadores
execute.py	log.txt	Relatório de execução do cliente
frequency_counter4.pl	<round>.aa.freq.txt	Agrupamento e contagem de sequências iguais
find_fuplicates7.pl	enriched.fasta	Sequências enriquecidas
get_nsequences.pl	list<num>.fasta	Lista das N sequências de aminoácidos mais enriquecidas, como definido pelo usuário na interface
	list<num>numberednt.fasta	Lista das N sequências de nucleotídeos mais enriquecidas
ANARCI	list<num>numbered.txt	Lista das N sequências de aminoácidos mais enriquecidas e numeradas
convertofasta.pl	list<num>numbered.fasta	Lista das N sequências de aminoácidos mais enriquecidas e numeradas em formato fasta
IgBlast	list<num>numbered germlineclassification.txt	Classificação de germlines das N sequências mais enriquecidas
execute_server.py	log2.txt	Relatório de execução do servidor

Fonte: O autor.