Universidade de Brasília
Instituto de Ciências Biológicas
Departamento de Biologia Celular
Programa de Pós-Graduação em Biologia Molecular

Método *in silico* para análise de sequências de imunoglobulinas produzidas por tecnologia de *phage display*

Heidi Muniz Silva

Brasília, março de 2016

Universidade de Brasília
Instituto de Ciências Biológicas
Departamento de Biologia Celular
Programa de Pós-Graduação em Biologia Molecular

Método *in silico* para análise de sequências de imunoglobulinas produzidas por tecnologia de *phage display*

Dissertação apresentada ao Departamento de Biologia Celular do Instituto de Ciências Biológicas da Universidade de Brasília, como requisito parcial para obtenção do grau de mestre em Biologia Molecular.

Heidi Muniz Silva

Orientador:
Prof. Dr. Marcelo de Macedo Brígido

Coorientador:
Prof. Dr. Nalvo Franco de Almeida Jr.

Brasília, março de 2016

Dissertação de mestrado sob o título "Método *in silico* para análise de sequências de imunoglobulinas produzidas por tecnologia de *phage display*", defendida por Heidi Muniz Silva no dia 03 de março de 2016 em Brasília, pela banca examinadora constituída pelos doutores Maria Emília Machado Telles Walter, Werner Treptow e Andrea Queiroz Maranhão.

Marcelo de Macedo Brígido (CEL-IB/UnB)
orientador

Nalvo Franco de Almeida Jr. (FACOM/UFMS)
coorientador

Werner Treptow (CEL-IB/UnB)
examinador interno

Maria Emília Machado Telles Walter (CIC/UnB)
examinadora externa

Andrea Queiroz Maranhão (CEL-IB/UnB)
suplente efetivo

Dedico esta tese a minha irmã Julia Harumi,
a quem tanto amo.

Agradecimentos

Quero agradecer a minha mãe, por não medir esforços para me ajudar a fazer o mestrado em Brasília, por ter me ensinado a tomar decisões sensatas, por ter me preparado para viver sozinha quando chegasse a hora, pelo seu apoio constante, pelo seu amor e por ter me dado a Júlia. A minha educação sempre foi sua prioridade, eu cresci vendo você lutar por mim, e assim percebi que esforço e estudo formavam um caminho promissor para mim. Obrigada, madrecita, por tudo, principalmente por ter me dado tudo o que eu precisava para chegar onde estou. Eu amo você !

Gostaria de agradecer ao meu melhor amigo, Cláudio. Sem este rapaz eu não conseguiria sobreviver a todos os problemas que enfrentei em Brasília, logo que cheguei na cidade. Obrigada por ter tido paciência comigo, por ter me apoiado quando eu sentia falta de casa, por ter pensado em soluções e ter sofrido junto comigo quando tive problemas de moradia, quando resolvi fazer a disciplina maluca que deveria durar 3 semanas e durou 3 meses, e quando estive totalmente perdida na análise do projeto, num momento de grande pressão com prazos. Obrigada meu amigo querido por ter me ajudado tanto a ter forças para lidar com tudo o que deu errado no primeiro ano do mestrado.

Obrigada tio Franski e dona Cida, por terem me acolhido por 3 meses em sua casa, por terem me apoiado e me ajudado em tantas coisas, principalmente a encontrar uma boa moradia e por me darem uma laço de família, do qual eu senti muita falta no primeiro ano. Obrigada pelo carinho, pela paciência, pela bondade e por sempre torcerem por mim, para que o mestrado desse certo.

Agradeço ao Faheem, o primeiro amigo que fiz na cidade. Obrigada guri por ter me ajudado bastante com a questão da minha adaptação em Brasília, pelas dicas de ônibus, sobre a UnB, de moradia, pela companhia agradável na república, por me proteger e cuidar de mim, pela hora do chá com leite, pelas comidas caseiras deliciosas e por ter me apresentado a Suellen.

Outra pessoa que não poderia deixar de mencionar é minha amiga Suellen. Eu fico feliz só de lembrar o primeiro dia em que conheci essa moça. Quando eu estava bem mal, bem desanimada, a alegria da semana era o horário do chá. De noite, lá pelas 21h, a Suellen passava na república e vinha tomar chá com o Faheem, e assim conheci minha amiga.

Obrigada por ser uma pessoa tão iluminada, tão cheia de vida, por ter me ajudado todas as vezes que precisei, por ter me dado teto quando fui expulsa do apartamento da velha maluca, sem nem me conhecer direito. Obrigada por ter me ensinado a aproveitar os momentos bons quando eles apareciam, por ter me ensinado a ser menos chorona e entender que nem tudo é o fim do mundo, por ser tão boa comigo, por ter sido a irmã mais velha que sempre quis ter.

Gostaria de agradecer a Chris, por ter me dado muitas dicas de programação, pela indicação de boas fontes para estudar Perl, e pela paciência de olhar meus algoritmos quando eu ainda não tinha confiança nos meus programas.

Agradeço aos amigos do laboratório de Bioinformática, Julien, Guilherme, Waldeyr, Danimel, João e Andressa, pelo companheirismo, pelos dias divertidos, pelo apoio, e por todos os conhecimentos de linux, programação em C, em Perl e Java, e de análises de bioinformática que compartilharam comigo. Obrigada meus amigos !

Obrigada Julien por ter me dado dicas imprescindíveis em Perl, por ter me ajudado em todas as coisas com as quais me desesperei, por ter me ensinado a ter confiança na minha capacidade de resolver os problemas da análise, por ter tanta consideração comigo, pela sua amizade, por me incentivar a não desistir da carreria acadêmica mesmo quando eu já tinha dado tudo por perdido, e é claro por todas as risadas. Julien você é íntegro, tem um coração muito generoso e sempre será querido para mim. Admiro você e fico feliz por ter tido a oportunidade de te conhecer e de trabalhar no mesmo grupo de pesquisa que você.

Obrigada Waldeyr, por alegrar meus dias, por sempre chegar sorridente no laboratório, pelo incentivo, por ter me apresentado ao desenvolvimento Web e Java, e pelo companheirismo. Waldeyr é uma pessoa valiosa em qualquer grupo que ele participe, pela sua capacidade como profissional e pela maneira simples de conseguir integrar um grupo inteiro, unir as pessoas e deixá-las mais próximas entre si. Obrigada por acreditar em mim e por me ensinar a trabalhar em grupo, Fantástico Waldeyr, sem você os dias não seriam tão divertidos.

Agradeço ao Guilherme, meu amor, pelas dicas de C, que me ajudaram a fazer um ótimo programa de tradução, por me ensinar Java, por ter sido paciente comigo, por tentar me acalmar quando eu estava estressada e cansada com o mestrado, por acreditar no meu potencial e pelo incentivo constante para que eu avançasse nos estudos de bioinformática.

Obrigada Rafa, pelas diversas dúvidas que você sanou, por sempre estar disposto a me

explicar detalhes sobre os dados e sobre o experimento, por ter produzido dados de qualidade excepcional de tal maneira que pude desenvolver o método inteiro de análise de imunoglobulinas a partir de tais dados. Obrigada também por ter tido a coragem de testar o pacote automatizado e por dar sugestões valiosas para a melhoria do pacote. O Rafa é um rapaz muito inteligente, experiente em Imunologia Molecular, ótimo para trabalhar em grupo e ainda sempre disposto a ajudar. A participação do Rafa foi essencial a este trabalho. Muito obrigada Rafa !!!

Agradeço a Tainá, pelo direcionamento inicial no meu projeto, por ter confiado no meu trabalho, pela compreensão com a minha pouca experiência, e pelos conhecimentos sobre análise de dados NGS.

Obrigada professor Nalvo, por acreditar que eu poderia seguir o caminho em Bioinformática, por ter sugerido o mestrado na UnB, com o professor Marcelo, pelo carinho e por sempre me ajudar quando eu precisava. Você é como um pai para mim, sempre me indicando boas rotas para me tornar uma bioinformata e acreditando no meu potencial como cientista. Tudo começou com você, e acho que já sabe o lugar especial que você ocupa no meu coração.

Obrigada professor Marcelo por me conceder a oportunidade de realizar um mestrado em um dos melhores programas de pós-graduação do país, por me ensinar tantas coisas sobre imunologia molecular e bioinformática, pela paciência, pela simpatia, por toda a experiência de pesquisa e desenvolvimento de artigos, e por ter me concedido um projeto tão interessante em que eu pudesse integrar conhecimentos de imunologia molecular e computação, fazendo programas que tentam "imitar o seu olhar", sobre sequências de imunoglobulinas. Estou muito feliz com o trabalho que fizemos, e sempre serei grata ao senhor pela inestimável experiência de vida que pude ter aqui em Brasília. Muito obrigada por tudo !

Sumário

Resumo

Com o advento das plataformas de sequenciamento de alto desempenho (HTS), tornou-se possível obter amplas amostragens das bibliotecas produzidas por *phage display*, cujo enorme volume dificulta a análise da diversidade das bibliotecas bem como a detecção de clones selecionados, a qual classicamente é realizada por ensaios de afinidade do anticorpo pelo antígeno. Considerando tal desafio, foi desenvolvido um método *in silico* automatizado para a análise de sequências de imunoglobulinas produzidas por *phage display*, que permite encontrar clones selecionados, a partir de bibliotecas sequenciadas por plataformas HTS. O método é composto por 6 etapas: montagem de *reads*, filtragem de sequências, tradução, análise de enriquecimento, numeração de resíduos e classificação de *germlines*. Para validar o método, foram analisados três conjuntos de dados, cada um contendo as bibliotecas original e final, sendo dois deles sequenciados pela plataforma Illumina, e o terceiro pela plataforma 454 Roche. A análise completa de cada par de bibliotecas foi executada em menos de 3 horas. Os tempos de execução promissores devem-se principalmente aos programas de tradução e cálculo de frequência dos clones, os quais foram desenvolvidos com estratégias inteligentes para analisar bibliotecas contendo mais de 10^6 *reads*, em menos de 5 minutos. Como saída final, é produzida uma lista de clones candidatos, enriquecidos e reconhecidos como domínio variável de imunoglobulina, ordenados por *fold change* de frequência e com sua respectiva classificação de *germlines*, os quais muito provavelmente foram selecionados pelo experimento de *phage display*. Além da eficiência do método no que diz respeito ao curto tempo necessário para sua execução, a abordagem utiliza um critério biológico para detectar clones candidatos, baseando-se nas marcas canônicas de domínio variável de imunoglobulina.

Abstract

Since high-throughput sequencing (HTS) platforms provide larger sampling of phage display libraries, the amount of data imposes challenges to analyze libraries diversity and to find selected clones, which are traditionally tested by antibody affinity assays. Considering that, we developed an automated *in silico* method to analyze immunoglobulin sequences produced by phage display, which allows the detection of selected clones, from libraries sequenced by HTS platforms. The method consists of 6 steps: reads joining, sequence filtering, translation, enrichment analysis, residues numbering and germline classification. In order to validate the method, 3 sets of data were analysed, each containing initial and final phage display libraries, being 2 sets sequenced by Illumina and one by 454 Roche platform. The complete analysis of each pair of libraries was performed in less than 3 hours. The promising execution time is mainly due to the translation and frequency calculation programs, which were developed with intelligent strategies to process libraries composed of more than 10^6 reads, in less than 5 minutes. As final output, the method creates a list of candidate clones, enriched and recognized as immunoglobulin variable domain, sorted by fold change of frequency and classified by germline, which probably were selected by phage display experiments. Besides the efficiency of the method concerning the fast performance, the present approach uses a biological criterion to find candidate clones, based on canonical signature of immunoglobulin variable domain.

1 Introdução

1.1 Imunoglobulinas

Entre os diversos tipos de células sanguíneas, originadas a partir de células-tronco da medula óssea, destacam-se os linfócitos B ou células B. Estas consistem em efetores indispensáveis do sistema imune adaptativo[1], o qual é mediado por linfócitos (B e T) e por exposição a antígenos. As células B possuem como características singulares seu receptor de superfície chamado Receptor de Célula B (**BCR**), e a capacidade de produzir enormes quantidades de anticorpos. Cada célula B madura produz somente um tipo de anticorpo, e no entanto, o repertório de anticorpos presentes em um único indivíduo é altamente diverso (Sompayrac, 2012). Dessa maneira, repertórios de anticorpos tem sido foco de muitos trabalhos na área de Imunologia Molecular, com diferentes aplicações, tais como desenvolvimento de vacinas, prognóstico e diagnóstico clínico, e produção de anticorpos recombinantes (Naylor & Capra, 1999; Wang & Yu, 2004).

A superfamília das imunoglobulinas compreende uma vasta diversidade de moléculas componentes do sistema imune, entre as quais estão as imunoglobulinas (anticorpos) e estruturas caracterizadas por dobramento similar ao de imunoglobulina (Ig-like fold), tais como receptores de células T (TCR), moléculas de histocompatibilidade (MHC I e II) e receptores de imunoglobulinas (Williams & Barclay, 1988). No entanto, esta seção se limitará a descrever os anticorpos, baseando-se na estrutura de um IgG (imunoglobulina da classe G).

Um anticorpo consiste numa glicoproteína[2], formada por dois pares idênticos de cadeias de aminoácidos. Cada par é formado por dois tipos de cadeias, uma cadeia pesada e uma cadeia leve (Figura 1), produzido por células B, como uma das várias estratégias do sistema imune adaptativo (Marchalonis *et al.*, 1996; Sompayrac, 2012). Ambas as cadeias possuem domínio variável e domínio constante. O domínio variável apresenta variação significativa de tamanho e sequência de aminoácidos, enquanto o domínio constante apresenta-se mais conservado entre diferentes imunoglobulinas. Torna-se re-

[1]Sistema imune adaptativo: imunidade mediada por células e/ou anticorpos, presente somente em vertebrados (Elgert, 1998).

[2]Glicoproteína: proteína que possui uma ou mais moléculas de carboidratos ligados a sua estrutura.

levante notar porém, que a cadeia pesada possui 3 domínios constantes (C_{H1}, C_{H2} e C_{H3}), já a cadeia leve possui apenas um (C_L). Assim, as cadeias leve e pesada estão espacialmente orientadas de tal maneira que apenas os domínio V_H e C_{H1} mantenham interação com os domínios V_L e C_L, respectivamente. Os demais domínios da cadeia pesada, C_{H2} e C_{H3}, interagem com seus homólogos da outra cadeia pesada (Owen *et al.*, 2013).

O domínio constante está ligado a uma molécula de carboidrato. A porcentagem e a localização do carboidrato varia de acordo com a classe do anticorpo (Elgert, 1998), conceito comentado mais adiante. Os oligossacarídeos são moléculas formadas por três a dez unidades de carboidratos, e têm papel crítico na função biológica do anticorpo, uma vez que anticorpos desprovidos de carboidratos se ligam ao complemento[3] com menor eficácia, e também perdem a habilidade de se ligar a alguns receptores de domínio constante, FcR (Coloma *et al.*, 2000).

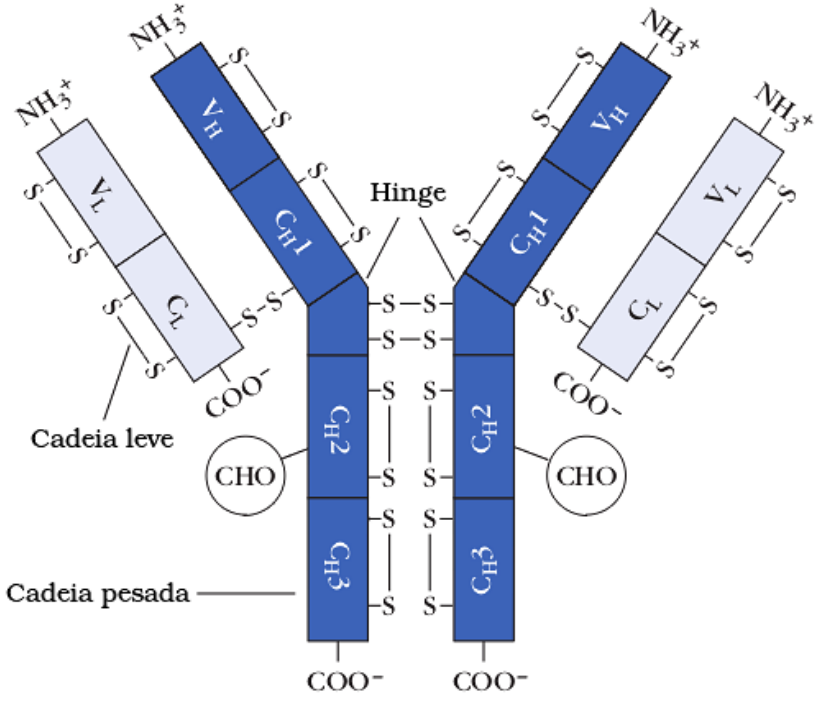

Figura 1: Estrutura de imunoglobulina, destacando os domínios V_H, V_L, C_{H1}, C_{H2}, C_{H3} e C_L. Fonte: (Owen *et al.*, 2013).

Comumente, descreve-se uma imunoglobulina por 2 tipos de fragmentos, **Fab** e

[3]Complemento: termo coletivo que designa uma série de proteínas plasmáticas, cuja ativação contribui para defesa contra agentes estranhos e para muitas características da resposta inflamatória (Elgert, 1998).

Fc, obtidos pela clivagem da enzima papaína (Porter, 1958). Esta cisteíno-protease hidrolisa ligações peptídicas em sítios que contenham resíduos de cisteína e, em um anticorpo, o sítio onde ocorre esta clivagem corresponde à região chamada de dobradiça ou *hinge* (Brezski & Jordan, 2010). A região *hinge* não está incluída em nenhum domínio variável ou constante, e corresponde a um grupo de resíduos que, por meio de pontes dissulfeto entre duas cisteínas da cadeia pesada, conectam as regiões **Fab** à região **Fc**. Além disso, devido à flexibilidade da região *hinge*, as regiões **Fab** podem se mover uma em relação à outra (Elgert, 1998).

A região **Fab** ("fragmento de ligação ao antígeno") possui o sítio de ligação ao antígeno, sendo composta pelos domínios variáveis V_H e V_L, e pelos domínios constantes C_{H1} e C_L. Antígeno corresponde a qualquer molécula que se ligue especificamente a um anticorpo ou a um TCR. Já a região **Fc** ("fragmento cristalizável") determina a classe do anticorpo, e é responsável por desencadear uma resposta imune, por meio da interação com receptores de imunoglobulinas da superfície de células do sistema imune ou com moléculas efetoras (Elgert, 1998; Owen *et al.*, 2013).

O domínio variável é formado por sete regiões, três regiões determinantes de complementaridade (CDRs) e quatro regiões chamadas *framework* (Figura 2). O domínio variável não é uniformemente variável, pois as CDRs apresentam uma variação de tamanho e sequência proteica consideravelmente maior que as regiões *framework*. Estas por sua vez, apresentam resíduos bastante conservados principalmente nas regiões que flanqueiam as CDRs. As CDRs formam o arcabouço do sítio de ligação ao antígeno e portanto, contribuem para a especificidade do anticorpo pela molécula alvo. Vale ressaltar que as regiões *framework* também desempenham papel relevante para especificidade ao antígeno, pois muito provavelmente afetam a conformação ou a flexibilidade dos loops formados pelas CDRs (Eisen, 2014). Neste ponto, torna-se relevante mencionar a organização dos genes que formam uma imunoglobulina bem como alguns detalhes sobre seu enovelamento.

As cadeias leve e pesada possuem estrutura modular, isto é, são constituídas por segmentos gênicos diferentes. Uma célula B precursora, que ainda não teve contato com seu antígeno cognato (antígeno que se liga especificamente aos seus receptores), inicialmente possui múltiplas versões de cada um dos segmentos gênicos, e precisa passar por eventos de recombinação para compor uma combinação única de segmentos para

a cadeia leve e para a pesada (Sompayrac, 2012). A cadeia leve é construída pela reunião de 3 segmentos, o segmento V (variável), o segmento J (junção) e o segmento C (constante). Os dois primeiros segmentos formam o domínio variável, e o último segmento forma o domínio constante.

A cadeia pesada por sua vez, também é composta pelos mesmos segmentos, todavia, notam-se duas diferenças. A primeira é a presença de um segmento adicional no domínio variável da cadeia pesada, o segmento D (de diversidade), que ao ser rearranjado situa-se entre os segmentos V e J. E a segunda corresponde ao seu segmento C, um tanto mais longo por conter duas regiões adicionais (C_{H2} e C_{H3}).

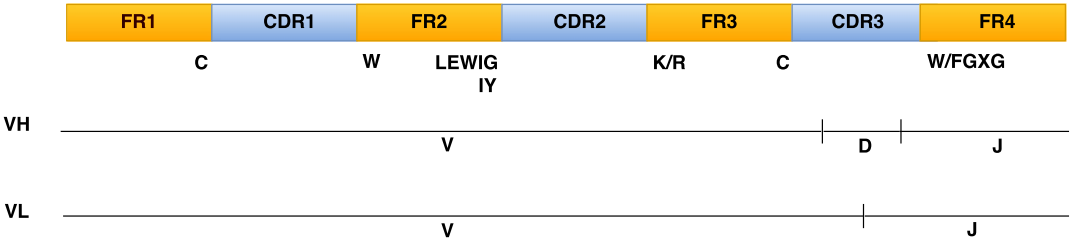

Figura 2: Esquema de regiões do domínio variável. FR: *framework*. CDR: região determinante de complementaridade. Abaixo do domínio variável são denotados resíduos conservados das regiões *framework* que flanqueam as CDRs, de cadeia pesada e cadeia leve. As barras apresentam a combinação de segmentos gênicos para cadeia pesada e leve.

Quanto às CDRs, estas situam-se no segmento V de ambas as cadeias, contudo, a CDR3 ocorre na junção VDJ da cadeia pesada e, na junção VJ da cadeia leve. Ressalta-se ainda que apenas a célula B precursora possui todas as versões dos segmentos V, D, J e C, já a célula B madura dispõe somente dos segmentos recombinados que irão compor o anticorpo que sua linhagem se comprometeu a produzir (Sompayrac, 2012; Owen *et al.*, 2013).

No contexto de Imunonologia, o termo *germlines* refere-se aos segmentos gênicos do lócus de imunoglobulina, presentes em linhagens germinativas, isto é, em células indiferenciadas que são precursoras de células do sistema imune. Cada molécula de anticorpo é codificada por múltiplos segmentos *germline* de domínio variável, os quais são rearranjados diferentemente em cada célula precursora do sistema imune para produzir um repertório primário e diverso. Os genes rearranjados passam então por hipermutação somática e seleção antigênica, resultando em um repertório expandido e aperfeiçoado de células B antígeno-específicas (Owen *et al.*, 2013).

Distinguem-se 5 padrões básicos de sequência do segmento C_H: *mu* (μ), *delta* (δ), *gama* (γ), *epsilon* (ϵ) e *alfa* (α). Cada tipo de sequência padrão do segmento C_H é chamado de isotipo, e o isotipo das cadeias pesadas de uma imunoglobulina é denominado classe (Owen *et al.*, 2013). Por conseguinte, as imunoglobulinas são divididas em 5 classes : IgM (μ), IgD (δ), IgG (γ), IgA (α) e IgE (ϵ).

O rearranjo produtivo dos segmentos gênicos constituintes das imunoglobulinas permite a expressão de cadeias leve e pesada funcionais, as quais irão interagir entre si por pontes dissulfeto, pontes de hidrogênio e interações hidrofóbicas, de tal modo que o heterodímero assuma uma estrutura tridimensional típica de imunoglobulina, o chamado dobramento de imunoglobulina (Branden & Tooze, 1999; Jung *et al.*, 2001).

O enovelamento típico de imunoglobulina consiste em 2 folhas-β antiparalelas, proximamente empacotadas e unidas por pontes dissulfeto, de modo que uma folha esteja voltada para a outra. Este tipo de dobramento ocorre tanto na cadeia pesada quanto na cadeia leve. O domínio constante é formado por uma folha-β de 3 fitas e outra folha-β de 4 fitas. Já o domínio variável tem arranjo similar, formado por uma folha-β de 4 fitas, no entanto, apresenta uma outra folha-β com 5, e não 3 fitas como no domínio constante, pois as duas fitas adicionais estão conectadas pelo *loop* que contem a CDR2.

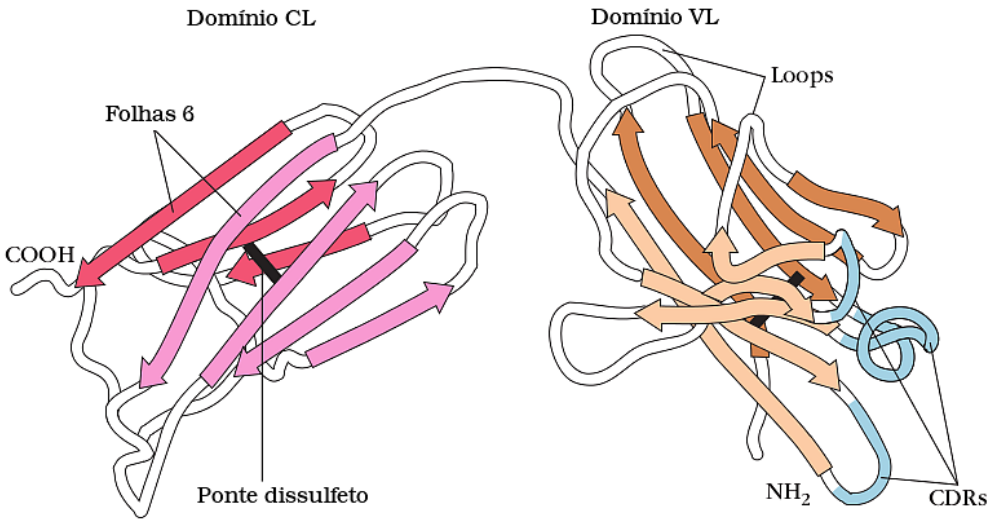

Figura 3: Dobramento de imunoglobulina da cadeia leve. Fonte: (Owen *et al.*, 2013).

Os resíduos que formam o *core* das folhas-β (resíduos *framework*) são altamente conservados entre diferentes imunoglobulinas, sendo responsáveis não somente por es-

tabilizar a estrutura das folhas-β por meio de interações hidrofóbicas, mas também por estabelecer interações estáveis entre os domínios constantes leve e pesado, e entre os domínios variáveis leve e pesado (Branden & Tooze, 1999; Tramontano, Chotia & Lesk, 1990). Por flanquear as CDRs, os resíduos *framework* são usados para definir o posicionamento no genoma das regiões determinantes de complementaridade (Elgert, 1998). Considerando a participação essencial das CDRs e das regiões *framework* na interação com o antígeno e nas vantagens de testes biológicos e experimentos que envolvem a expressão de proteínas menores, pesquisas sobre anticorpos recombinantes frequentemente utilizam sequências que codificam apenas domínios variáveis.

1.2 Produção de anticorpos recombinantes

O desenvolvimento de hibridomas possibilitou a produção de anticorpos monoclonais ainda na década de 70 (Köler & Milstein, 1975). Visto que células B morrem rapidamente ao serem cultivadas *in vitro*, a tecnologia de hibridomas solucionou este obstáculo, tornando possível o cultivo de linfócitos B imortais, capazes de produzir anticorpos monoclonais. Brevemente, tal método pode ser compreendido em duas etapas. A primeira consiste em isolar linfócitos B, provenientes do baço de um doador imunizado contra o antígeno de interesse. Já a segunda etapa, resume-se a fusionar um linfócito B com uma célula de mieloma (célula mielóide cancerosa). Dessa maneira, a célula híbrida resultante originará um clone[4] imortal, capaz de produzir anticorpos de mesma especificidade por um dado antígeno, chamados de anticorpos monoclonais, por serem produzidos por um único clone de célula B (Walsh, 2007).

Embora os hibridomas tenham sido um avanço notável nos estudos de imunologia, os anticorpos monoclonais foram aprovados para o uso terapêutico somente na década de 80. Inicialmente, as pesquisas focavam em tratamentos de câncer, porém, atualmente anticorpos monoclonais são utilizados para diferentes propósitos, tais como, indução de imunidade passiva, diagnóstico e terapêutica (câncer, transplante e doenças cardiovasculares) (Walsh, 2007).

De acordo com o banco de estudos em fase de ensaio clínico, ClinicalTrial.gov[5], do NIH (US National Institutes of Health), atualmente existem 3572 estudos sobre

[4]Clone: Linhagem de células originadas a partir de uma única célula.
[5]ClinicalTrial.gov: ⟨https://clinicaltrials.gov/ct2/home⟩

anticorpos monoclonais em fase de ensaio clínico, em 191 países. Deste total, 144 pertencem à América do Sul, onde o Brasil lidera com 81 estudos. Desde 1986 até 2015, o mercado farmacêutico dos EUA e da Europa conta com 47 anticorpos monoclonais terapêuticos (Ecker, Jones & Levine, 2015). Diante disso, mostram-se auspiciosas as pesquisas envolvendo métodos de desenvolvimento de anticorpos monoclonais, já que estes constituem produtos promissores para o mercado farmacêutico.

Os hibridomas permitem obter diferentes anticorpos monoclonais específicos para um mesmo antígeno. E no intuito de produzir anticorpos em larga escala para fins terapêuticos, podem ser utilizadas bibliotecas de anticorpos recombinantes e assim encontrar quais anticorpos apresentam afinidade pelo antígeno alvo. Uma técnica promissora que pode usada para tal finalidade corresponde à tecnologia de *phage display*, descrita na seção seguinte.

1.3 *Phage display*: expressão de peptídeos em fagos filamentosos

Phage display consiste na expressão de proteínas ou peptídeos na superfície de fagos filamentosos[6]. O gene da proteína de interesse é fusionado ao gene de uma proteína do capsídeo do fago, o qual infecta a célula bacteriana e assim, assegura-se a expressão da proteína de estudo durante a etapa de produção de proteínas essenciais à montagem da partícula viral. Desse modo, uma biblioteca de genes de interesse, por exemplo, genes codificantes de fragmentos de anticorpos recombinantes, é gerada utilizando como veículo de expressão o genoma de fagos filamentosos (Maranhão & Brígido, 2000; Willats, 2002; Walsh, 2007).

Usualmente, a biblioteca de fagos passa por 3 a 5 ciclos de seleção de maneira que os membros componentes da biblioteca sejam genes de proteínas específicas para um dado alvo, ou mais precisamente, fragmentos de anticorpos específicos para um antígeno de interesse. A seleção por afinidade (*biopanning*) resume-se a expor a biblioteca de fagos à moléculas alvo imobilizadas, de maneira que apenas os fagos expressando a proteína de especificidade desejada sejam retidos. Por eluição, recuperam-se os fagos selecionados,

[6]Fagos: vírus que infectam bactérias. Fagos filamentosos são um tipo de fago que não possui cauda, e cuja simetria é helicoidal (Madigan *et al.*, 2009).

o que permite obter os genes codificantes das proteínas que se ligam especificamente a um alvo de interesse (Willats, 2002; Walsh, 2007).

Uma metodologia bastante utilizada para expressar proteínas na superfície de fagos compreende um sistema de fago híbrido, baseado em fagomídeo. Fagomídeo pode ser compreendido como um plasmídeo que contém uma origem de replicação e o gene da proteína do capsídeo fusionada a proteína de interesse. O fagomídeo coinfecta células bacterianas com um fago *helper*, que possui todos os outros genes do fago, exceto a origem de replicação. Visto que somente o fagomídeo possui origem de replicação, sua sequência é replicada e suas cópias são incorporadas nas novas partículas virais. O genoma do fago helper não é replicado, afinal é desprovido de origem de replicação. Todavia, é possível montar novas partículas virais, pois os genes do fago *helper* são expressos. Tem-se como resultado a produção de partículas virais funcionais contendo o fagomídeo, o qual por sua vez, possui o gene codificante da proteína de interesse (Maranhão & Brígido, 2000; Willats, 2002). Esta abordagem tem sido utilizada pelo grupo de pesquisa em Imunologia Molecular da UnB (Universidade de Brasília). O esquema geral das etapas de *phage display* pode ser visto na Figura 4.

O monitoramento da seleção é realizado por titulação[7] da biblioteca. O título dos fagos da biblioteca inicial, anterior ao experimento, é então comparado com o título dos fagos da biblioteca final, após a seleção (Barbas *et al.*, 2001; Maranhao & Brigido, 2000). Espera-se que os valores de título diminuam ao longo dos ciclos de *phage display*, afinal a seleção por afinidade reduz gradativamente a diversidade de clones da biblioteca. Clone é um conjunto de fagos que foram originados a partir de um dado fago, e portanto, possuem em seu genoma o mesmo gene codificante de fragmento de anticorpo e, expressam em sua superfície o mesmo fragmento de anticorpo.

Ressalta-se que a titulação compreende a contagem de partículas virais da biblioteca como um todo, e não de clones individuais. Isso permite observar mudanças no número de fagos da biblioteca, e não do número de fagos de cada clone. Ao final do experimento tem-se um biblioteca de fagos que foram selecionados, os quais são amplificados por PCR e caracterizados por sequenciamento (Kay, Winter & McCafferty, 1996). A partir das sequências dos clones selecionados, são realizados testes biológicos *in vitro* a fim de

[7]Titulação: técnica laboratorial que permite quantificar a concentração de um reagente conhecido. Em *phage display*, a titulação produz uma estimativa da quantidade de partículas de fagos de uma biblioteca.

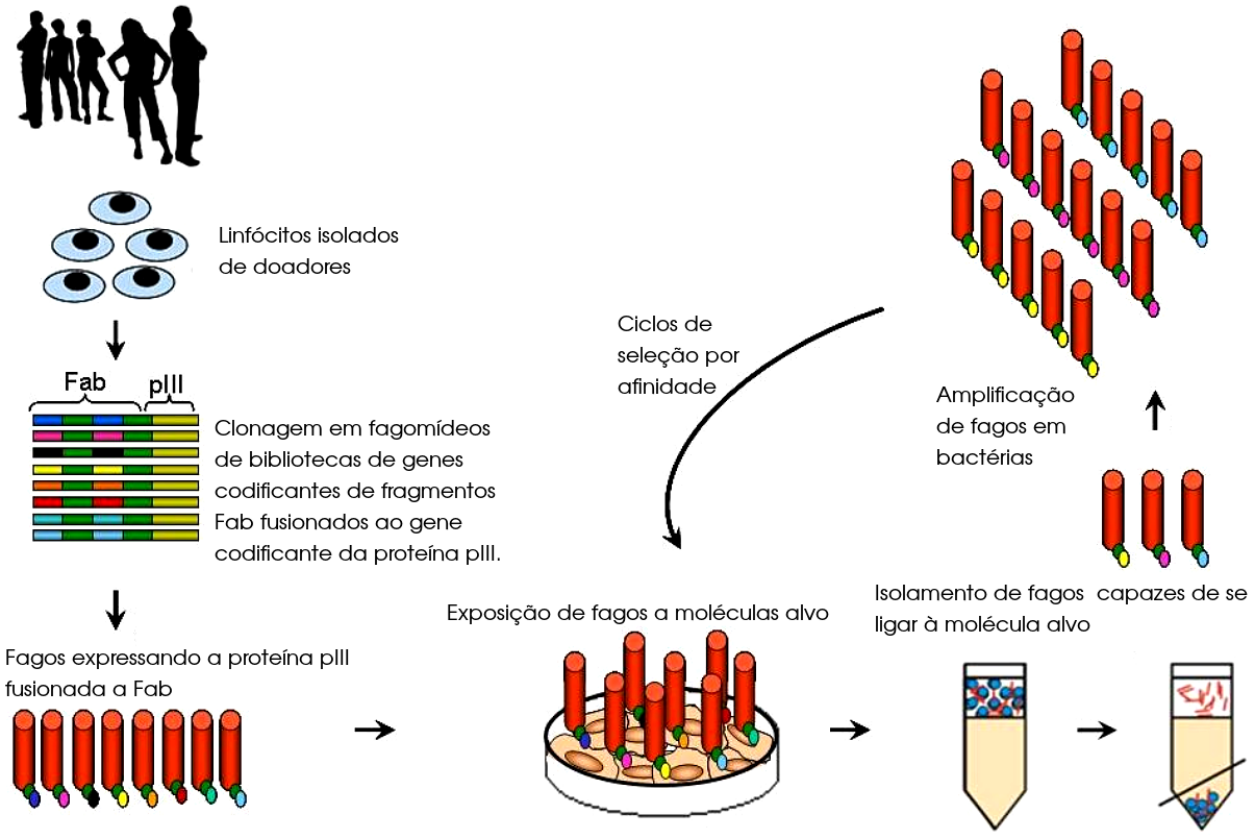

Figura 4: Esquema geral da tecnologia de phage display. Modificado de (Dantas-Barbosa, Brigido & Maranhao, 2012).

avaliar a afinidade dos anticorpos pela molécula alvo.

O sequenciamento Sanger normalmente é utilizado para avaliar a diversidade das bibliotecas de *phage display* e rastrear sequências de interesse a cada etapa do experimento. No entanto, o método de Sanger permite amostrar apenas uma pequena fração das bibliotecas, o que limita a análise de diversidade e a detecção de genes candidatos, isto é, genes de clones que foram selecionados no experimento de *phage display* (Christiansen *et al.*, 2015a; Dias-Neto *et al.*, 2009).

Nesse contexto, plataformas de sequenciamento de alto desempenho surgem como alternativas mais eficazes para amostrar bibliotecas de maneira ampla, produzindo grandes quantidades de sequências para cada biblioteca sequenciada. A combinação de *phage display* com tecnologias de sequenciamento de alto desempenho proporciona não somente uma amostragem mais profunda como também a possibilidade de substituir algumas etapas da metodologia *wetlab* tais como a titulação de fagos, conferindo assim

vantagens em relação a abordagem clássica, caracterizada pelo consumo considerável de recursos e de tempo (Ravn *et al.*, 2010).

1.4 Sequenciamento de alto desempenho

A tecnologia de sequenciamento, desenvolvida por Sanger em 1977 (Sanger, Nicklen & Coulson, 1977), revolucionou os métodos utilizados em Biologia Molecular, pois tornou possível obter a sequência de nucleotídeos de uma molécula de DNA. O sequenciamento Sanger consiste num tipo de sequenciamento por síntese, o qual fundamenta-se na adição de dideoxinucleotídeos terminadores, isto é, nucleotídeos cujo carbono 3' não possui hidroxila e que portanto, terminam a polimerização ao impedir a adição do próximo nucleotídeo. Quando a técnica surgiu, os fragmentos eram ordenados por tamanho por meio de eletroforese em gel, que foi substituída pela eletroforese capilar.

Além disso, os dideoxinucleotídeos anteriormente eram identificados por marcadores radioativos, e atualmente são marcados com fluoróforos (uma cor para cada base nitrogenada) (Kircher & Kelso, 2010). A eletroforese capilar assim como a em gel, separa moléculas por tamanho e carga. As moléculas de DNA deslocam-se em capilares finíssimos em direção a um pólo positivo, e de acordo com a carga (proporcional ao tamanho), algumas moléculas chegam mais rapidamente que outras por serem mais curtas. Antes de chegar ao pólo positivo, um detector identifica qual é o dideoxinucleotídeo que termina cada sequência. Assim, ordenadas por tamanho e carga, as moléculas geram um gráfico com picos de fluorescência para cada um dos fluróforos. Tal gráfico permite obter a sequência da molécula de DNA (Biosystems, 2009).

As plataformas de sequenciamento de alto desempenho resolveram algumas limitações do sequenciamento Sanger, tais como contaminação da amostra, erros inseridos nas sequências durante a clonagem, baixa cobertura e alto custo. No sequenciamento de nova geração (NGS), a amplificação da biblioteca de DNA ocorre em uma superfície sólida, e sistemas ópticos substituem a eletroforese capilar de Sanger (Kircher & Kelso, 2010; Myllykangas, Buenrostro & Ji, 2012; Hert, Fredlake & Barron, 2008).

De modo geral, as diferentes plataformas de sequenciamento de nova geração compartilham três etapas: preparação da biblioteca de DNA, imobilização e sequenciamento. A preparação da biblioteca resume-se a fragmentar o DNA, e ligar adapta-

dores[8] às extremidades dos fragmentos. Já na etapa de imobilização, os fragmentos são ancorados em uma superfície sólida por meio dos adaptadores, e assim é definido o sítio onde ocorrerá a reação de sequenciamento. Quanto ao sequenciamento, cada plataforma utiliza um tipo diferente de reação, porém, todas são dotadas de sistemas ópticos que monitoram os eventos moleculares (Myllykangas, Buenrostro & Ji, 2012). Dentre as principais tecnologias de sequenciamento de alto desempenho, destacam-se a 454 Roche e a Illumina, cada qual com particularidades que determinam diferenças pontuais nas etapas de análise de dados.

A plataforma 454 Roche utiliza o método de pirosequenciamento (Figura 5). As moléculas de DNA fragmentadas e dotadas de adaptadores são ligadas à superfície de microesferas ou *beads*, as quais servem de sítio de amplificação. À medida que a DNA polimerase adiciona um nucleotídeo complementar, o pirofosfato liberado e um substrato adenosina 5'-fosfosulfato formam ATP, numa reação catalisada pela enzima sulforilase. O ATP formado participa por sua vez da conversão de luciferina em oxi-luciferina, pela enzima luciferase (Scientific, 2015). Esta conversão libera luz, a qual é detectada por uma câmera de CCD (Dispositivo de Carga Acoplada), indicando que um nucleotídeo foi adicionado. Os nucleotídeos são adicionados separadamente e sequencialmente, o que permite descobrir qual nucleotídeo é incorporado a cada ciclo, sendo os picos de sinal luminoso, proporcionais à quantidade de nucleotídeos incorporados (Sciences, 2012). Geralmente, neste tipo de sequenciamento usam-se adaptadores para apenas uma das fitas do DNA, e portanto, são produzidos *reads*[9] de apenas uma das fitas, chamados de *reads single-end*.

Quanto à plataforma Illumina, esta trabalha com método de sequenciamento por síntese, porém se distingue de Sanger por explorar a terminação reversível cíclica para cessar temporariamente a síntese de DNA. Tanto a amplificação quanto a reação de sequenciamento ocorrem em uma plataforma de vidro, chamada de *flow cells*. As *flow cells* são recobertas com adaptadores complementares aos que estão ligados aos fragmentos de DNA, o que permite imobilizar as fitas de DNA (Kircher & Kelso, 2010). Para produzir *reads single-end*, um dos tipos de adaptadores é removido, e por con-

[8]Adaptadores: Oligonucleotídeos ligados às extremidades da molécula de DNA, usados para imobilizar a molécula em uma superfície sólida. Exemplo: adaptadores conjugados com biotina se ligam às microesferas recobertas por estreptavidina, devido a afinidade da biotina pela estreptavidina, e assim é possível imobilizar as moléculas de DNA nas microesferas (Rizzi *et al.*, 2012).

[9]*Reads*: Sequências curtas de DNA produzidas pelo sequenciador.

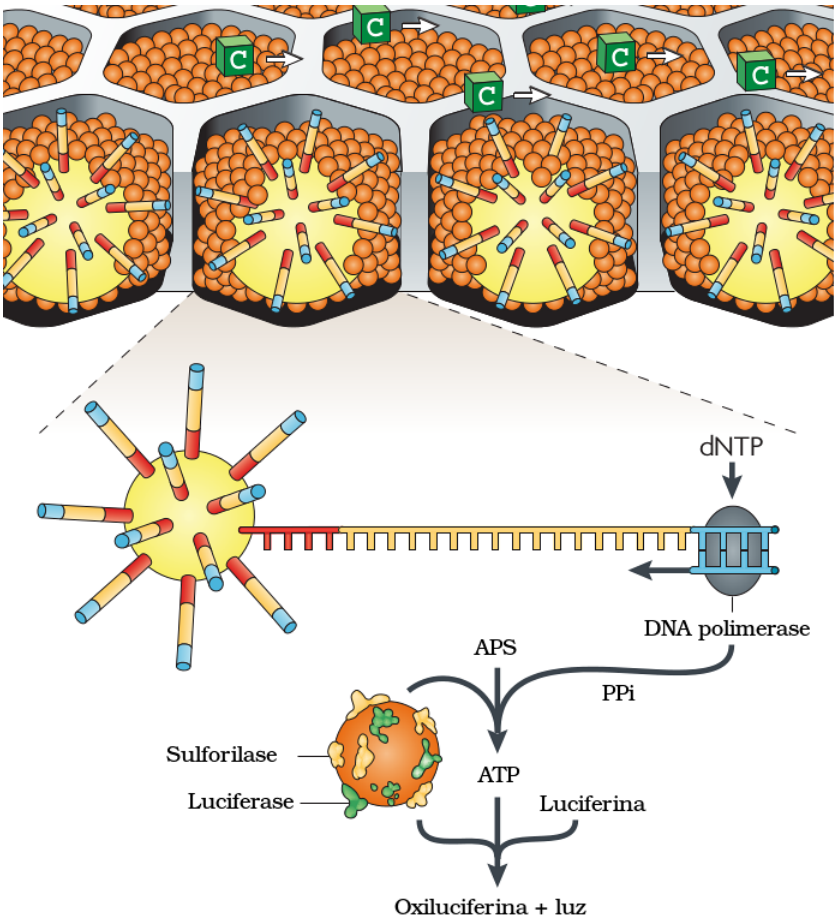

Figura 5: Esquema de pirosequenciamento. Milhões de microesferas contendo fitas simples de DNA são colocadas num suporte de vidro onde ocorre a reação de pirosequenciamento. APS: 5'-adenosina fosfosulfato. PPi: pirofosfato. Em detalhe, uma fita em processo de polimerização pela DNA polimerase, e a consequente liberação de luz uma vez que seja incorporado um novo nucleotídeo. Fonte: (Metzker, 2010).

sequência restam nas *flow cells* moléculas de DNA correspondentes a apenas uma das fitas (um único sentido). Se forem realizados ciclos diferentes contendo cada um dos dois tipos de adaptadores, então ambas as fitas do DNA serão sequenciadas, produzindo *reads* chamados de *paired-end* (Mardis, 2013).

Na reação de sequenciamento são usados nucleotídeos terminadores fluorescentes reversíveis, que terminam a síntese ao serem incorporados na sequência, pois possuem o carbono 3' contendo um grupo funcional ligado à hidroxila, chamado de terminador, que impede a inserção do próximo nucleotídeo. Para cada tipo de nucleotídeo é usado um fluoróforo de cor diferente, de modo que uma câmera CCD registra imagens das

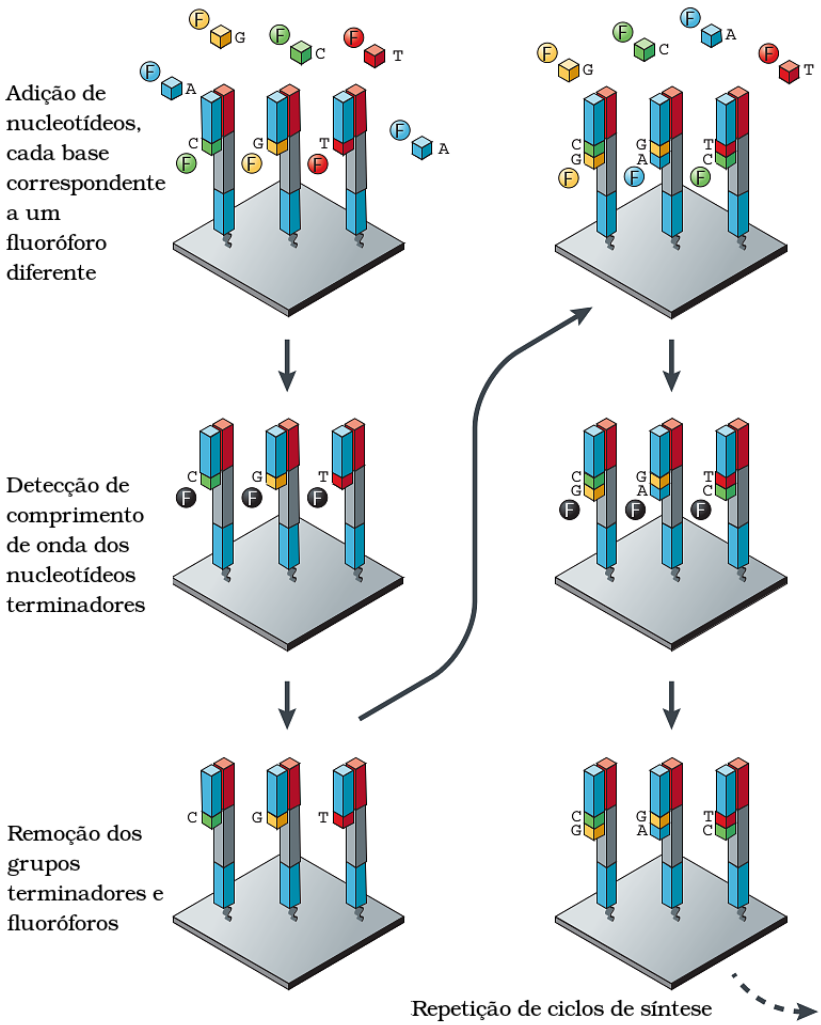

Figura 6: Esquema de sequenciamento por síntese, da plataforma Illumina. Fonte: (Metzker, 2010).

flow cells e identifica pelo comprimento de onda qual nucleotídeo foi incorporado na sequência. Para inserir o próximo nucleotídeo, o terminador e os fluoróforos são removidos do nucleotídeo terminador, e assim novos nucleotídeos podem ser incorporados para dar continuidade à síntese (Metzker, 2010).

1.5 Sequenciamento de alto desempenho aplicado a *phage display*

O sequenciamento de bibliotecas de *phage display* produz bibliotecas de *reads*, os quais correspondem a sequências codificadoras de fragmentos de anticorpos. Na bibli-

oteca NGS, um clone corresponde a um grupo de sequências que foram recuperadas de um conjunto de fagos, os quais foram originados a partir de um mesmo fago. Considerando tais conceitos, diferentes grupos tem descrito propostas *in silico* de análise de bibliotecas de *phage display* sequenciadas por plataformas de alto desempenho.

O primeiro trabalho que associou sequenciamento de alto desempenho com a tecnologia de *phage display* utilizou a plataforma 454 Roche (Dias-Neto *et al.*, 2009). Este estudo demonstrou que a amostragem pelo sequenciamento NGS é muito mais ampla que pelo método Sanger, pois foram produzidas bibliotecas com cerca de 10^5 *reads* pela plataforma 454 Roche, em contrapartida a bibliotecas com tamanho de 10^3 produzidas pelo sequenciamento Sanger. Quanto ao tempo necessário para gerar tais bibliotecas, estimativas indicam que a amostragem com sequenciamento Sanger aumenta à medida que aumenta o tamanho da biblioteca, enquanto a abordagem da plataforma 454 Roche apresenta tempo constante para produzir bibliotecas com 10^3 até 10^6 *reads*. Para gerar bibliotecas de 10^6 *reads* por meio de sequenciamento Sanger foi estimado o tempo de 4106 dias, já a plataforma 454 Roche leva 74,8 horas (Dias-Neto *et al.*, 2009).

A partir deste trabalho vários outros estudos passaram a aplicar sequenciamento de alto desempenho para caracterizar as bibliotecas geradas por *phage display* (Glanville *et al.*, 2009; Ravn *et al.*, 2010; Matochko *et al.*, 2012; Christiansen *et al.*, 2015b; Wu *et al.*, 2012), com diferentes finalidades, dentre as quais destacam-se a análise de diversidade das bibliotecas e a identificação de clones selecionados por *phage display*. No contexto de Imunologia Molecular, a diversidade das bibliotecas de *phage display* são em geral analisadas em termos de CDRs ou de CDR3 (Ravn *et al.*, 2010; Glanville *et al.*, 2009; Maranhão *et al.*, 2013), afinal estas regiões apresentam maior variação de resíduos, em especial CDR3, a qual é considerada por muitos autores como a região que de fato determina a afinidade do anticorpo pelo antígeno, muito embora as demais CDRs e as regiões *framework* participem de maneira essencial para determinar a conformação do sítio de ligação bem como a afinidade pelo antígeno e entre as cadeias V_H e V_L (Tramontano, Chotia & Lesk, 1990; Masuda *et al.*, 2006).

Dentre estes estudos, dois deles classificam as bibliotecas por meio do alinhamento das CDRs ou da CDR3 das sequências das bibliotecas contra um banco de sequências de *germlines* (Glanville *et al.*, 2009; Ravn *et al.*, 2010), e assim analisam a diversidade pelo uso de *germlines* nas bibliotecas de *phage display*. Já um trabalho sobre repertório

de imunoglobulinas de *Gallus gallus* (galinha) realizou a análise de diversidade baseado no desvio de composição de aminoácidos das sequências das bibliotecas, em relação às sequências da *germline* de cadeia pesada, usando uma macro desenvolvida no Excel (Wu *et al.*, 2012). Outro grupo de pesquisa propôs ainda a análise de diversidade de bibliotecas de *phage display* por meio de *scripts* MathLab, que calculam a abundância das sequências de nucleotídeos e dos aminoácidos por posição, nas sequências mais frequentes (Matochko *et al.*, 2012).

No que diz respeito à detecção de clones selecionados por *phage display*, um trabalho publicado em conjunto por duas farmacêuticas da Suíça comparou as sequências de CDR3 de V_H, e usou o termo "clones candidatos", para denominar as sequências mais frequentes, que por aumentarem em proporção da biblioteca inicial para final, eram consideradas selecionadas pelo experimento de *phage display*. O conjunto de clones considerados candidatos apresentou afinidade pelo antígeno, e ainda continha clones que não haviam sido detectados no ensaio clássico de ELISA[10], geralmente usado para avaliar a afinidade dos clones selecionados por *phage display*.

A afinidade ao antígeno foi correlacionada ao enriquecimento[11] de muitos clones analisados, e dessa forma, o estudo descreve a detecção de clones candidatos baseado no critério de frequência de clones, isto é, na proporção de sequências que pertencem a clones individuais. Caso a proporção de sequências de um dado clone aumente da biblioteca inicial, antes do experimento de *phage display*, para a biblioteca final, após a seleção de *phage display*, a sequência que representa o clone é vista como candidata, pois considera-se que o aumento de sua proporção é resultante da seleção de *phage display* (Ravn *et al.*, 2010).

Uma das farmacêuticas do trabalho mencionado publicou um trabalho mais recente, em que foi desenvolvido um *workflow* de análise de bibliotecas NGS, produzidas por *phage display* (Ravn *et al.*, 2013). O grupo propõe as seguintes etapas: controle de qualidade, cálculo de frequência de clones baseado na sequência de nucleotídeos e de aminoácidos, identificação da sequência do anticorpo e sua respectiva *germline*, visualização dos resultados, remoção de erros de sequenciamento, identificação de *clusters*

[10]ELISA (Enzyme-Linked Immunosorbent Assay): ensaio que permite a detecção da interação entre antígeno e anticorpo por meio da mudança de cor da solução que contém as moléculas de teste.

[11]Enriquecimento: na abordagem *wetlab*, enriquecimento consiste no aumento do número de partículas virais ao longo dos ciclos de seleção por afinidade. Na abordagem *in silico* enriquecimento corresponde ao aumento na proporção de sequências que compõem um clone.

de CDR3 e recuperação de clones.

Exceto os dois últimos passos, todos os demais são executados pelo programa N^2GSAb, desenvolvido pelo grupo usando o servidor Microsoft SQL (Ravn *et al.*, 2013). As sequências são consideradas candidatas de acordo com a frequência, como descrito acima, e neste trabalho, o grupo apresenta a recuperação dos clones, a qual é realizada por meio da montagem das sequências candidatas de V_H e V_L, e amplificação do fragmento montado, seguido de sequenciamento Sanger para caracterizar o scFv (fragmento variável de cadeia simples).

Apesar de existirem diferentes trabalhos que descrevem a combinação entre *phage display* e plataformas de sequenciamento de alto desempenho, bem como ferramentas de bancos de dados ou versões *stand-alone* para análise de sequências de imunoglobulinas (Abhinandan & Martin, 2008; Raghavan, 2009; Ye *et al.*, 2013; Lefranc *et al.*, 2009), não foi descrito ainda um método *in silico* automatizado para identificar clones selecionados por *phage display*, a partir de bibliotecas NGS.

E mesmo os estudos dedicados a identificar clones candidatos não apresentam um *workflow* automatizado e utilizam apenas o critério de frequência de clones para detectar candidatos, sem considerar características biológicas mais detalhadas das sequências (Ravn *et al.*, 2010; Ravn *et al.*, 2013). Além disso, a análise destes trabalhos limita-se à CDRs ou à CDR3 de V_H, e dessa maneira as demais regiões de V_H bem como o domínio V_L são desconsiderados.

Embora o critério de frequência de clones garanta a escolha das sequências mais frequentes, não assegura que as sequências possuam marcas canônicas de anticorpo por toda a extensão do domínio variável, requisito este verificado apenas na etapa de bancada, acompanhada com sequenciamento Sanger nas abordagens anteriores. Além disso, a identificação de clones e/ou análise de diversidade não deveria limitar-se à CDR3 das sequências de V_H, mas sim usar um estratégia de análise mais ampla, que incluísse todas as regiões *framework* e CDRs de V_H e de V_L.

Desse modo, torna-se evidente a relevância do desenvolvimento de um método *in silico* automatizado, capaz de analisar bibliotecas de *phage display* sequenciadas por plataformas de alto desempenho, a fim de encontrar clones selecionados, e que utilize critérios de detecção baseados não somente na frequência de clones, mas também na assinatura de imunoglobulinas, tanto de V_H quanto de V_L.

1.6 Objetivo Geral

O presente trabalho tem por objetivo propor um método *in silico* para análise de sequências de imunoglobulinas, produzidas por tecnologia de *phage display*.

1.7 Objetivos específicos

- Propor e implementar um método *in silico* automatizado de detecção de sequências de imunoglobulinas, selecionadas por *phage display*;
- analisar a diversidade de bibliotecas de *phage display*, formadas por sequências codificantes de domínio variável de imunoglobulinas;

2 Metodologia

2.1 Critérios do método

O método desenvolvido considera os seguintes conceitos :

- Clone é um grupo de sequências codificadoras de fragmentos de anticorpos, recuperadas de um conjunto de fagos, os quais foram originados a partir de um único fago.
- Um clone é representado pela sequência membro mais longa.
- Enriquecimento consiste no aumento do número de sequências que compõem um clone, ao longo dos ciclos de seleção de *phage display*.
- Clone candidato é aquele cuja sequência representativa foi considerada candidata, isto é, atende aos critérios do método.

Sequências que muito provavelmente pertencem a clones selecionados por *phage display* são chamadas de sequências candidatas, como denominado por outros trabalhos da literatura (Ravn *et al.*, 2010; Ravn *et al.*, 2013). Na presente abordagem, são propostos dois critérios para detectar sequências candidatas a partir de bibliotecas NGS de *phage display*:

1. A sequência candidata deve conter as regiões canônicas do domínio variável, quatro regiões *framework* e três CDRs (Figura 2). Devido à presença de resíduos *framework* bastante conservados que flanqueiam as CDRs e ao fato de que as CDRs assumem um número limitado de conformações (Abhinandan & Martin, 2008; Al-Lazikani, Lesk & Chothia, 1997), é possível traçar padrões do domínio variável, para V_H e para V_L. O reconhecimento do domínio variável é realizado em duas etapas. Na etapa de tradução, são traduzidas somente as sequências que possuírem os dois resíduos de cisteína que flanqueiam as regiões CDR1 até CDR3 e os resíduos que flanqueiam CDR3, que correspondem a uma cisteína e à *substring* WGXG de V_H e FGXG de V_L, em que X é um resíduo de aminoácido qualquer. Dessa maneira, uma sequência é traduzida somente se tiver as marcas canônicas de domínio variável. Na segunda etapa, numeração de resíduos, uma sequência

atende ao critério de marcas canônicas caso seja numerada. A numeração consiste em atribuir um número a cada resíduo de aminoácido, que corresponde a uma posição estruturalmente equivalente em diferentes moléculas, e que é realizada a partir de um alinhamento da sequência contra um perfil de domínio variável. O perfil de domínio variável utiliza alinhamento múltiplo e o Modelo Hidden Markov (HMM) (Abhinandan & Martin, 2008).

2. A sequência candidata deve pertencer a um clone que foi enriquecido, ou seja, um clone cuja proporção de sequências aumentou em ciclos sucessivos de seleção de *phage display*. Na abordagem *in vitro*, o monitoramento da seleção é realizado pela titulação das bibliotecas, e portanto, a estimativa de partículas virais refere-se às bibliotecas inteiras (Barbas *et al.*, 2001). A análise de clones individuais é inviável na metodologia *wetlab*, já que seria necessário o monitoramento manual de cada um dos vários clones presentes nas bibliotecas de *phage display*, cuja diversidade inicial é de cerca de 10^7 a 10^8 clones (Kay, Winter & McCafferty, 1996). Além disso, o uso de plataformas de sequenciamento de alto desempenho possibilita amostragens mais profundas que o sequenciamento Sanger (Dias-Neto *et al.*, 2009) e, por conseguinte, gera bibliotecas muito maiores, cujo volume adiciona mais um obstáculo para o monitoramento de clones individuais. Após um ciclo de seleção por afinidade, os clones selecionados passam por amplificação em bactéria, e assim a quantidade de partículas virais correspondente a um clone selecionado aumenta da biblioteca inicial para a biblioteca final. Visto que nas bibliotecas NGS os clones são representados por sequências, a análise de enriquecimento de clones individuais será baseada na proporção de sequências que constituem um clone. Desse modo, um clone será considerado enriquecido se a proporção de sequências que o compõem aumentar da biblioteca inicial para a biblioteca final, e assim, a sequência representativa do clone atende ao segundo requisito para ser considerada candidata. Este raciocínio é suportado pelos trabalhos de duas farmacêuticas da Suíça, que estimaram o enriquecimento de clones por meio da proporção de sequências, e que reuniram evidências de que o enriquecimento de clones pode ser correlacionado à afinidade do anticorpo pelo antígeno (Ravn *et al.*, 2010).

2.2 Bibliotecas de *phage display*

A fim de validar o método, foram analisados três conjuntos de dados. Um deles foi sequenciado pela plataforma 454 Roche, e os demais plataforma Illumina MiSeq. Para diferenciar os conjuntos Illumina, um deles será denominado conjunto Illumina S1, e o outro, conjunto Illumina S2.

Cada conjunto possui quatro bibliotecas, duas de V_H e duas de V_L. Para cada tipo de cadeia, há uma biblioteca original, sequenciada antes dos ciclos de seleção e uma biblioteca final, sequenciada após os ciclos de seleção. Diferenças pontuais determinaram a execução de algumas etapas de processamento específicas para cada conjunto. No entanto, em um panorama geral, todos os conjuntos compartilham a maioria das etapas de análise.

2.3 Método *in silico* para detecção de sequências de imunoglobulinas selecionadas por *phage display*

A Figura 7 apresenta as etapas do método *in silico* para análise das bibliotecas de *phage display*. A entrada são bibliotecas NGS de *phage display*, e como saída, tem-se uma lista de clones candidsatos para produção de anticorpos recombinantes, escolhidos de acordo com os critérios mencionados anteriormente. Um esquema mais detalhado contendo os arquivos de entrada e saída, bem como os programas utilizados na análise podem ser vistos na Figura 8. Nas seções seguintes, são descritas as etapas de análise.

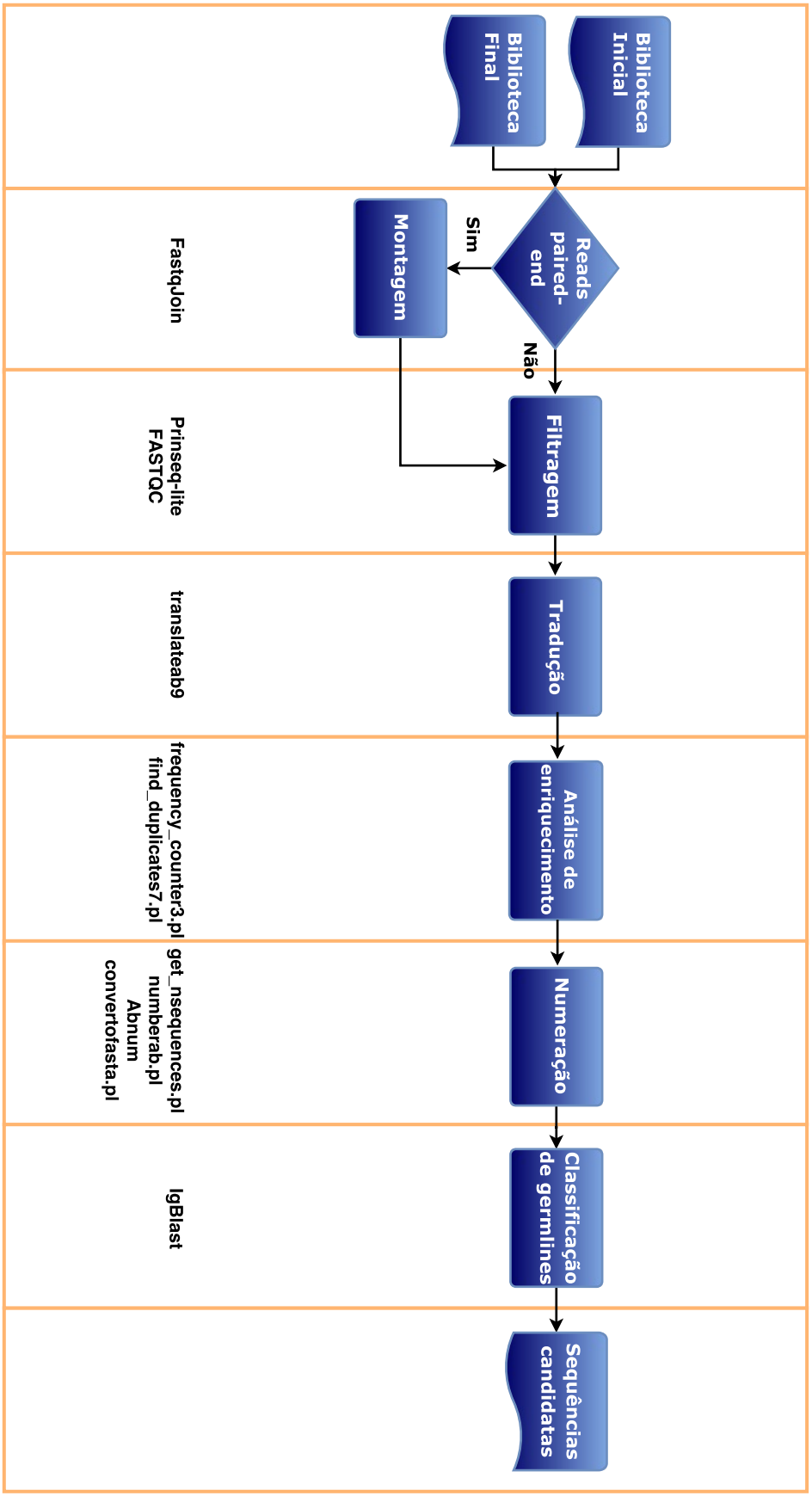

Figura 7: Etapas do método *in silico* para a análise de sequências de imunoglobulinas, produzidas por *phage display*, a partir de bibliotecas NGS.

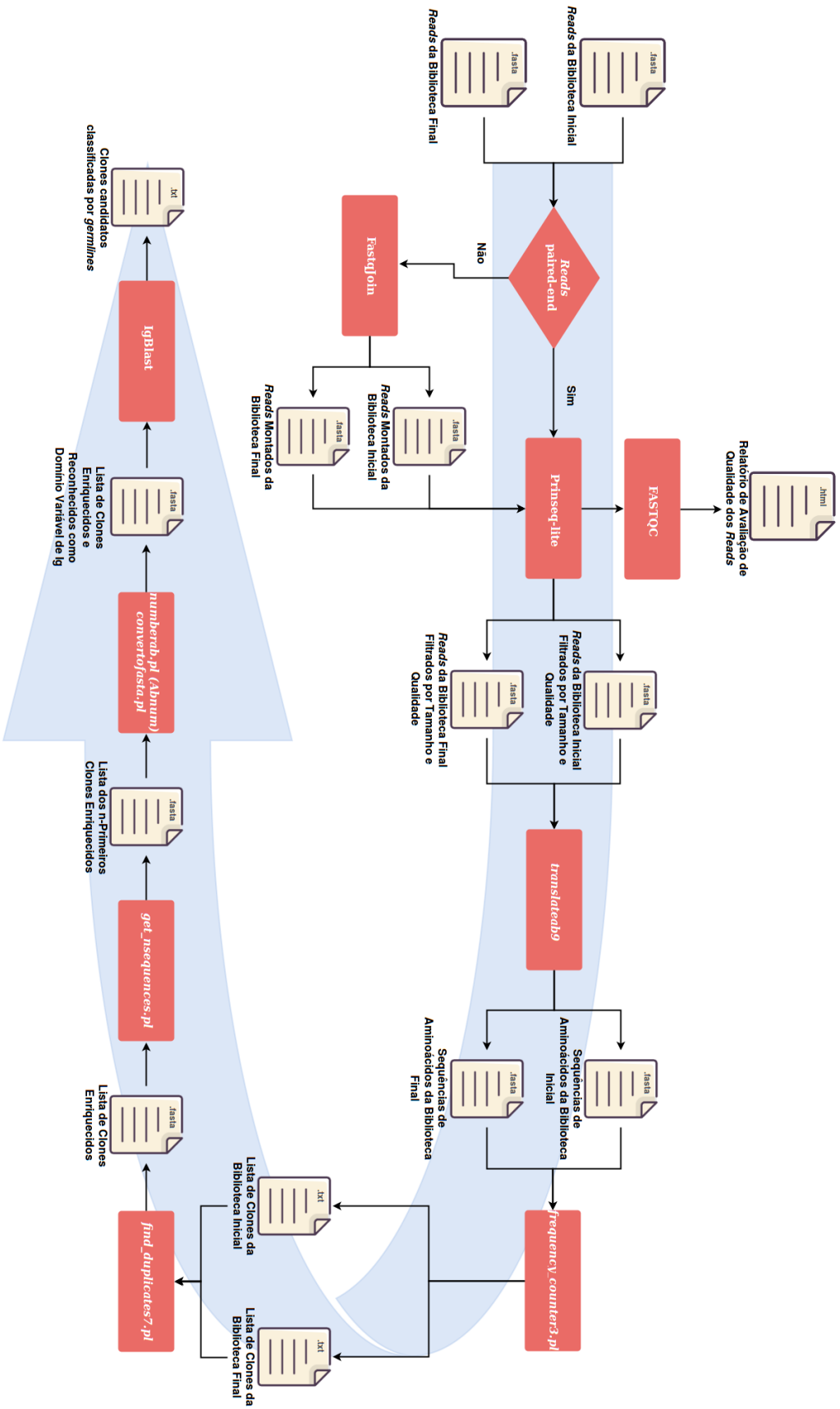

Figura 8: Esquema geral do método, mostrando os programas e seus respectivos arquivos de entrada e saída.

2.4 Filtragem e controle de qualidade

O primeiro passo do pipeline (Figura 7) consiste em avaliar a qualidade das sequências e executar filtragem, caso necessário. A qualidade das sequências é avaliada pelo *software* FastQC (Andrews, 2012), o qual executa controle de qualidade de dados NGS, permitindo identificar problemas gerados pelo sequenciador ou durante a preparação da biblioteca. O seguinte comando foi utilizado para executar o FastQC:

```
fastqc input -q -o destiny
```

A opção `-q` silencia as mensagens impressas na saída padrão, a opção `-o` permite indicar um diretório diferente do diretório da entrada, para salvar os arquivos produzidos pelo controle de qualidade, e `input` é substituído pelo caminho do arquivo `fastq`, que corresponde à entrada para o FastQC. Esta ferramenta gera como saída um arquivo html, contendo um relatório de avaliação da qualidades dos *reads* das bibliotecas analisadas (Figura 8).

Quanto à filtragem, utiliza-se o *software* PRINSEQ (Schmieder & Edwards, 2011) para remover sequências de baixa qualidade e que não possuam tamanho adequado. A qualidade é representada pela pontuação de qualidade PHRED (Ewing *et al.*, 1998), medida comumente usada para avaliar a acurácia de uma plataforma de sequenciamento, que expressa a probabilidade de erro de cada nucleotídeo sequenciado (Equação 1), onde Q corresponde à pontuação de qualidade e P à probabilidade de erro (Illumina, 2011). Neste método, é exigida uma qualidade mínima de 20, que se substituída na Equação 1, equivale a 1 erro a cada 100 pb sequenciados ou 99% de acurácia. Quanto ao tamanho da sequência, esta deve possuir no mínimo o tamanho do gene do domínio variável, de cerca de 250 a 300 pb.

$$Q = -10log_{10}P \tag{1}$$

O *software* PRINSEQ é usado em dois processos da análise. No primeiro processo, o PRINSEQ converte o formato `fastq` para `fasta`. Um arquivo em formato `fastq` possui informações sobre a qualidade na escala PHRED e sobre a sequência. Já o arquivo em formato `fasta` é mais compacto, contendo apenas o identificador e a sequência de nucleotídeos ou de aminoácidos. Visto que a informação necessária para a análise

das bibliotecas resume-se às sequências e seus respectivos identificadores, o formato `fasta` apresenta-se como o mais apropriado para a execuções das etapas seguintes. A conversão direta do formato `fastq` para o formato `fasta` é realizada para que seja calculado o número de *reads* das bibliotecas de entrada usando expressão regular, a fim de gerar gráficos de qualidade descritos mais adiante. Para tal conversão de formato, o seguinte comando foi utilizado:

```
prinseq-lite -fastq input -out_format 1 -out_good output
```

A opção `-fastq` indica o formato da entrada, a opção `out_format` permite escolher o formato dos arquivos de saída (opção 1 gera somente arquivo `fasta`, 5 gera arquivos `fastq`, `fasta` e `qual`), já a opção `-out_good` permite escolher o nome dos arquivos de saída.

Finalmente, o segundo processo em que é executado o PRINSEQ consiste na filtragem por tamanho e qualidade e, desta vez, além da conversão para `fasta`, os *reads* com tamanho abaixo de 300 pb e/ou com qualidade abaixo de 20 são removidos das bibliotecas, com o seguinte comando:

```
prinseq-lite -fastq input min_len 30 min_qual_mean 20 -out_format 5
-out_bad null -out_good output
```

As opções `min_len` e `min_qual_mean` permitem configurar respectivamente, o tamanho e a qualidade mínima dos *reads*. Quanto à opção `out_bad`, esta permite descartar as sequências de má qualidade e tamanho inadequado caso seja configurada com o valor "null".

2.5 Identificação de bibliotecas V_H e V_L

A identificação de bibliotecas V_H e V_L é um passo específico para o conjunto 454 Roche, cujas sequências não foram identificadas pela *facility* de sequenciamento. O experimento que produziu o conjunto usou *primers* identificadores que permitem distinguir V_H e V_L. Inicialmente, foi desenvolvido um *script* Perl, *antibodyid8.pl*, que recebe como entrada o arquivo `fasta`, busca pelas sequências dos *primers* usando expressão regular, e gera 4 arquivos de saída contendo, respectivamente, o conjunto de sequências

identificadas como V_H, o conjunto de sequências V_L, sequências não identificadas, e o total de sequências dos três arquivos anteriores.

Este *script* é eficaz em encontrar as sequências dos *primers* que não estejam corrompidas, isto é, que não possuam inserções, deleções ou substituições. No entanto, é ineficaz para lidar com sequências de *primers* diferentes das originais, pois a expressão regular realiza busca exata. Uma quantidade considerável de sequências não pode ser identificada (25,6% da biblioteca inicial e 53,9% da biblioteca final), devido à presença de erros inseridos pela plataforma de sequenciamento (Tabela 1).

Considerando que o tamanho das bibliotecas do conjunto 454 Roche já havia sido notavelmente reduzido na etapa de filtragem por qualidade e tamanho, e que a quantidade de sequências não identificadas corresponde a pouco mais da metade de uma das bibliotecas, descartar estas sequências poderia comprometer as análises de enriquecimento e diversidade, por redução da amostra. Como solução, optou-se por não descartar as sequências com *primers* corrompidos, e usar alinhamento e não mais busca exata para identificar as sequências dos *primers*.

Tabela 1: Identificação de bibliotecas V_H e V_L pelo *script antibodyid8.pl*

Subconjunto	**Número de *r*eads da biblioteca R_0**	**Número de *r*eads da biblioteca Rs**
V_H	34492	28108
V_L	85040	55899
Não identificado	41106	98061
Total	160638	182068

R_0: biblioteca original. Rs: biblioteca final após a seleção de *phage display*.

Sendo assim, a distinção de bibliotecas V_H e V_L passou a ser executada pelo programa Cutadapt (Martin, 2011), o qual remove adaptadores, *primers*, caudas poliA e outros tipos de sequências indesejadas de sequências de bibliotecas NGS. O programa utiliza alinhamento semiglobal[12] para identificar as sequências a serem removidas. A entrada para o Cutadapt são os arquivos das bibliotecas inicial e final, de V_H ou de V_L, e um arquivo contendo as sequências dos *primers*. Como saída o Cutadapt produz um arquivo contendo as sequências que possuíam os *primers*, sendos estes removidos

[12]Alinhamento semiglobal: alinhamento cuja pontuação penaliza diferenças apenas na região de sobreposição entre as sequências comparadas (Martin, 2011).

das sequências. Neste método, o Cutadapt é usado para distinguir sequências V_H e V_L. Foram removidos *primers* tanto na direção 3' quanto 5', de V_H e V_L.

O *script antibodyid8.pl* desempenha então apenas a validação dos conjuntos de sequências identificados pelo Cutadapt, pois espera-se que o Cutadapt identifique um número maior de sequências que o *script* Perl, considerando que o primeiro utiliza alinhamento para comparar sequências e que, portanto, consegue lidar com os erros inseridos pelo sequenciador. Tal passo de validação foi essencial para descobrir que o Cutadapt identificou parte das sequências como V_H e também como V_L, devido à similaridade entre as sequências dos *primers*.

Para tanto, foi desenvolvido um *script*, *mergedatav4.pl*, que recebe como entrada os arquivos V_H e V_L gerados pelo Cutadapt, busca sequências duplicadas, e gera dois arquivos (um para V_H e outro para V_L) de sequências que constavam somente ou no arquivo de V_H ou no arquivo de V_L. O Cutadapt permitiu identificar quase a totalidade das bibliotecas (Tabela 2), recuperando assim um número de sequências bem maior que o *script antibodyid8.pl*.

Tabela 2: Identificação de bibliotecas V_H e V_L pelo programa Cutadapt

Subconjunto	**Número de *r*eads da biblioteca R_0**	**Número de *r*eads da biblioteca Rs**
V_H	48595	38689
V_L	111595	141407
Não identificado	448	1972
Total identificado	160190	180096

R_0: biblioteca original. Rs: biblioteca final após a seleção de phage display.

Para que as demais etapas do pipeline trabalhassem com as sequências originais, sem a remoção de subsequências, foi implementado o *script get_id.pl*, que recebe como entrada o arquivo `fasta` original contendo as bibliotecas NGS mistas, e um arquivo (V_H ou V_L) gerado pelo *mergedatav4.pl*. O *get_id.pl* imprime em um arquivo de saída, todas as sequências originais cujos identificadores existem no arquivo gerado pelo *mergedatav4.pl*.

2.6 Montagem

Os *reads* dos conjuntos Illumina são *paired-end* e foram produzidos de modo que parte do gene do domínio variável estivesse na sobreposição entre R1 e R2. As denominações R1 e R2 referem-se a ambas as fitas do DNA. Assim R1 é um *read* que pode ser correspondente à fita *forward* ou *reverse* e o R2 é correspondente à fita complementar de R1. Tal peculiaridade exige um passo adicional para a análise do conjunto, chamado de montagem. A montagem de *reads paired-end* que possuem sobreposição consiste em alinhar os *reads* e encontrar a região de sobreposição, e concatená-la a trechos não sobrepostos de R1 e R2. Procura-se pela sequência consenso na região de sobreposição, que garante uma confiabilidade extra ao sequenciamento, visto que tem-se o dobro de nucleotídeos referentes a uma mesma sequência. Assim, as sequências dos conjuntos Illumina são formadas por uma região de R1, a sobreposição entre R1 e R2, e uma região de R2. O programa usado para montagem foi o FastqJoin (Aronesty, 2011; Aronesty, 2013), da ea-utils (licença MIT), o qual escolhe a base de maior qualidade caso as bases de uma dada posição sejam iguais, e calcula a diferença entre as qualidades das bases, caso as bases sejam diferentes. Ressalta-se ainda que a filtragem das bibliotecas de *reads paired-end* é realizada após a montagem.

2.7 Tradução

A etapa seguinte consiste na tradução das sequências, pois no presente método o cálculo de frequência dos clones é realizado a partir da comparação de sequências de aminoácidos. Para tanto, foi desenvolvido o programa *translateab9* em linguagem C, que recebe como entrada o arquivo em formato `fasta`, resultante do passo de filtragem, traduz as sequências e as imprime em arquivos de saída. A escolha da fase aberta de leitura (ORF - *Open Reading Frame*) se baseia não somente na ausência de códons de parada, como também na presença de marcas canônicas do domínio variável. O programa busca por *subtrings*[13] que contenham tamanho dentro de um dado intervalo. Um dos padrões corresponde a *substring* que contém CDR1, FR2, CDR2 e FR3, flanqueada por dois resíduos canônicos de cisteína. O outro padrão é formado pela CDR3, a qual é

[13] *String*: tipo de dado definido em linguagens de programação que corresponde a uma sequência de caracteres. Uma subsequência de uma *string* é chamada de *substring*.

delimitada pelo segundo resíduo de cisteína e a sequência canônica **WGXG**, para V_H ou **FGXG**, para V_L, em que **X** é um resíduo de aminoácido qualquer. O *translateab9* admite tamanhos dos padrões que estejam dentro de intervalos específicos para V_H e V_L (Tabela 4), discutidos na seção 3.4. O requisito de encontrar marcas canônicas nas ORFs tem por objetivo aplicar o primeiro critério do método.

Finalmente, como saída, o programa de tradução cria dois arquivos em formato `fasta`. Um deles possui sequências de aminoácidos e o outro as sequências correspondentes de nucleotídeos. Este segundo arquivo é necessário para recuperar as sequências de nucleotídeos que produzem as sequências consideradas candidatas. Outro detalhe sobre o arquivo de sequências de aminoácidos é que para cada entrada é impressa a *substring* contendo as marcas canônicas de domínio variável e a sequência completa, com o respectivo identificador. Esta *substring* contendo CDRs é usada no cálculo de frequência de clones, passo seguinte à tradução.

2.8 Análise de enriquecimento

A análise de enriquecimento é composta por dois passos. O primeiro corresponde ao cálculo da frequência relativa dos clones e o segundo consiste na identificação de clones cuja frequência aumenta da biblioteca inicial para a biblioteca final. Um clone é formado por um grupo de sequências de aminoácidos que possuem a mesma subsequência, contendo as regiões CDR1 até CDR3. Ressalta-se que esta subsequência foi identificada para cada sequência traduzida pelo programa *translateab9*. Um vez que o programa de cálculo de frequência identifique os clones da biblioteca, a frequência relativa de cada clone é calculada baseando-se na proporção de sequências que os compõem.

O programa *counter2* foi desenvolvido em linguagem C, de tal modo que recebe como entrada um arquivo `fasta` contendo sequências traduzidas, calcula a frequência relativa de clones à medida que lê as sequências, e imprime em um arquivo de saída uma lista de sequências em ordem decrescente de frequência relativa. Como alternativa, foi desenvolvido um programa Perl, *frequency_counter3.pl*, que recebe a mesma entrada e produz uma saída bastante similar à do *counter2*, com a diferença de que imprime o tamanho da biblioteca como informação adicional.

Quanto ao cálculo de frequência relativa, inicialmente o total usado correspondia ao

número de sequências traduzidas, no entanto, o total de sequências filtradas, as quais são entrada para o programa de tradução, mostrou-se mais adequado ao cálculo a fim de minimizar os efeitos da aplicação do primeiro critério sobre os valores de *fold change* dos clones. A frequência relativa de um clone deveria ser independente dos critérios do método, uma vez que o cálculo baseado somente nas sequências que possuem todas as marcas de domínio variável usaria como total um subconjunto da biblioteca real. Tal escolha poderia resultar em diferenças nos valores de *fold change*, pois a frequência de um clone poderia ser superestimada caso o tamanho da biblioteca traduzida fosse muito menor que o tamanho da biblioteca filtrada. Assim, o cálculo da frequência relativa de clones individuais pode ser expresso por

$$fr_i = \frac{F_i}{N}, \tag{2}$$

onde fr_i corresponde à frequência relativa de um clone i, F_i corresponde ao número de sequências que constituem um clone i e N corresponde ao total de sequências filtradas.

Finalmente, o arquivo de saída compreende uma lista de sequências, em que cada entrada possui um identificador, tamanho da biblioteca e frequência relativa da *substring*, seguido da *substring* que abrange as regiões de CDR1 até CDR3, e de todas as sequências que possuem a *substring* e respectivos identificadores. Resumidamente, o arquivo de saída contém uma lista de clones de um biblioteca com suas respectivas frequências relativas.

O programa *frequency_counter3.pl* apresentou tempos de execução menores que o *counter2.c*, cuja estratégias e tempos de resposta são discutidos na seção 3.4. Uma vez calculada a frequência relativa dos clones, é possível executar a segunda etapa da análise de enriquecimento. Para tanto, foi implementado um *script* Perl, *find_duplicates7.pl*, que recebe como entrada as listas de clones ordenados por frequência relativa, da biblioteca inicial, anterior ao experimento e da biblioteca final, após o experimento.

O programa *find_duplicates7.pl* busca por clones cuja frequência relativa tenha aumentado da biblioteca inicial para a biblioteca final, e imprime uma lista decrescente de clones ordenados por aumento de frequência. Para cada clone, é impressa a maior sequência membro, que passa a ser representativa do clone, seu identificador e o aumento da frequência, que corresponde ao quociente entre a frequência relativa do clone

na biblioteca final e a frequência relativa do clone na biblioteca inicial, que neste método chamamos de *fold change*. Sendo assim, aplica-se o segundo critério do método na etapa de análise de enriquecimento, visto que é produzida uma lista de clones que foram enriquecidos ao longo dos ciclos de seleção de *phage display*.

2.9 Reconhecimento dos domínios V_H e V_L

Os domínios variáveis de imunoglobulinas são identificados como V_H ou V_L baseando-se no alinhamento da sequência de estudo contra os perfis de domínio variável, os quais foram criados a partir de um banco de sequências de imunoglobulinas humanas e murinas, usando HMM (Abhinandan & Martin, 2008). O alinhamento da sequência de interesse contra o perfil de domínio variável permite realizar a numeração dos resíduos de aminoácidos. A numeração consiste em atribuir um número a cada resíduo de aminoácido que corresponde a uma posição estruturalmente equivalente em diferentes moléculas (Abhinandan & Martin, 2008). Existem diferentes esquemas de numeração, sendo o mais tradicional o esquema de Kabat (Kabat *et al.*, 1992), que se baseia somente na variação de sequências. A numeração da sequência permite identificar todas as regiões *framework* e CDRs do domínio variável bem como inserções e deleções (Abhinandan & Martin, 2008).

Visto que a numeração de resíduos constitui uma maneira eficaz de verificar se uma dada sequência é reconhecida como domínio variável, uma etapa de reconhecimento de domínio variável foi incluída no presente método, a fim de reforçar o primeiro critério, e assegurar que as sequências selecionadas possuam de fato o perfil das regiões do domínio variável. Optou-se por identificar as sequências de acordo com o esquema de numeração de Kabat, em virtude do foco desta análise residir na variabilidade das sequências, tema central do trabalho de Kabat, e não na estrutura de imunoglobulinas.

As primeiras sequências da lista produzida pelo *find_duplicates7.pl* no passo anterior são as sequências com maiores valores de *fold change* e que são representativas de clones enriquecidos e que, portanto, atendem ao segundo critério do método. Estas sequências são extraídas do arquivo de saída do *find_duplicates7.pl*, pelo *script get_nsequences.pl*, que imprime as sequências num arquivo em formato `fasta`. Escolheu-se como valor padrão, extrair as 10 primeiras sequências pois a partir delas é possível fazer várias

combinações de cadeias pesada e leve. Foi implementado um *script*, *numberab.pl*, que envia as dez primeiras sequências para um servidor do grupo de Bioinformática da universidade UCL (University College London), solicitando a identificação e numeração ao programa Abnum (pertencente ao pacote abYsis) (Abhinandan & Martin, 2008), e redireciona a saída de cada uma das sequências para um único arquivo.

O programa Abnum alinha sequências proteicas contra os perfis dos domínios V_H e V_L, gerando como saída um arquivo contendo uma linha referente ao identificador da sequência, seguida por linhas compostas pelo rótulo do tipo de sequência (H para cadeia pesada e L para cadeia leve), posição do resíduo (representada por um número inteiro) e o aminoácido. O Abnum numera somente sequências cujos domínios variáveis estejam completos, e por consequência, garante que apenas sequências reconhecidas como domínio variável de imunoglobulinas sejam numeradas (Abhinandan & Martin, 2008; Raghavan, 2009). Visto que a saída do Abnum consiste num arquivo de colunas e que seria inviável trabalhar com tal formato, foi desenvolvido um *script*, *convertofasta.pl*, que converte o formato de colunas para formato `fasta`.

2.10 Classificação de *Germlines*

A identificação dos genes de *germline*, que deram origem aos domínios de um anticorpo, tem se tornado relevante para aplicações clínicas (Wang *et al.*, 2008), como o prognóstico de Leucemia Linfocítica Crônica (Naylor & Capra, 1999), e para estudos que buscam relacionar mutações com especifidade ao antígeno. Considerando a possibilidade de fornecer um passo inicial para a análise de mutações, nosso método tem como última etapa a classificação de *germlines* dos clones candidatos, realizada pelo *software* IgBlast (NCBI), versão *stand-alone* (Ye *et al.*, 2013).

A ferramenta IgBlast permite identificar genes V, D e J de *germlines*, bem como delinear as regiões *framework* e as CDRs, por meio de alinhamento local contra bancos de dados de *germlines*. Escolhemos bancos de dados humanos, pois as bibliotecas analisadas são de origem humana. Com relação ao tipo de entrada, optou-se por sequências de aminoácidos para assegurar que o IgBlast não escolhesse ORFs incorretas. Desse modo, o arquivo `fasta` produzido pelo *script convertofasta.pl*, é usado como entrada para o IgBlast. O IgBlast, por sua vez, é configurado para produzir um arquivo `txt`

compacto, contendo valores de identidade da sequência com a respectiva *germline* considerada como melhor *hit* e as posições de ínicio e fim de cada região do domínio variável, exceto a FR4, pois os bancos de germlines possuem somente o segmento V.

2.11 Integração de resultados da análise

No intuito de facilitar a visualização dos resultados, estes são integrados em um arquivo `html`. Para tanto foram desenvolvidos dois programas em Perl, o *rscript_creator.pl* e o *html_creator.pl*. O primeiro recebe como entrada o caminho dos arquivos das bibliotecas inicial e final, anteriores à filtragem, o caminho de um arquivo em formato `csv`, que contém o número de sequências por etapa, o diretório onde serão armazenados os *scripts* R e o diretório onde serão armazenados os gráficos criados pelos *scripts* R. O programa gera então dois *scripts* R, um deles cria um gráfico de proporção de *reads* com tamanho adequado baseado nos arquivos `fasta` anteriores à filtragem, e o outro, um gráfico de número de *reads* por etapa.

A saída do IgBlast, juntamente com o arquivo de sequências numeradas pelo Abnum em formato `fasta`, e os gráficos gerados pelos *scripts* R, referentes às bibliotecas V_H e V_L constituem a entrada para o *html_creator.pl*. Este cria um arquivo `html`, de modo que seja apresentada uma saída mais concisa e que integra dados relevantes sobre os clones candidatos e as bibliotecas de V_H e de V_L, tais como melhor *hit*[14] de *germlines*, valores de identidade, nomes de *germlines* do NCBI, valores de *fold change*, regiões do domínio variável (*framework* e CDRs) e os gráficos de proporção de *reads* de acordo com tamanho adequado, e de número de *reads* por etapa.

2.12 Automatização do método

A fim de tornar o método compatível com outras aplicações em Imunologia Molecular, o método foi automatizado. Para tanto, foram desenvolvidos um programa Perl, denominado *autoiganalysis3.pl*, e um *script shell*, denominado *atillacli.sh*. O *script attilacli.sh* interage com o usuário via linha de comando, para obter ou um arquivo de configuração da automatização (caso exista), ou uma série de informações que permi-

[14] *Hit*: *substring* de uma sequência do banco (que neste caso é o conjunto de *germlines*) que pode ser alinhada a uma *substring* de uma sequência *query* (neste caso pertencente a biblioteca NGS).

tam criar um arquivo de configuração, o qual será usado pelo *autoiganalysis3.pl* para executar todos os programas componentes do método. O pacote de programas desenvolvidos neste método bem como os *scripts* de automatização serão disponibilizados em breve para *download* gratuito de modo que a análise posssa ser executada com tempos mais curtos que abordagens que utilizam servidores de análise *online*. Além disso, como o pacote de programas será instalado na máquina local, o usuário poderá acompanhar todo o processo da análise.

Quantos aos *scripts* da automatização, o *attilacli.sh* foi escrito em linguagem *shell* a fim de manter uma das mais poderosas funcionalidade de *shell*, que corresponde a autocompletar caminhos de diretórios. Dessa maneira, a função de autocompletar é um dos mecanismos para evitar erros na configuração da automatização. O *attilacli.sh* possui ainda testes de verificação de diretórios e arquivos e um menu de configuração que permite corrigir os argumentos dados pelo usuário. O *attilacli.sh* pede a confirmação do usuário para criar links simbólicos dos programas desenvolvidos neste trabalho, cria o diretório do projeto e então executa o *autoiganalysis3.pl* para as bibliotecas V_H e em seguida para as bibliotecas V_L.

O *autoiganalysis3.pl* lê o arquivo de configuração criado pelo *attilacli.sh*, cria subdiretórios para as bibliotecas V_H e V_L, e então executa sequencialmente cada uma das etapas do método. O *attilacli.sh* informa ao usuário quando a análise de V_H ou V_L é finalizada. Cada diretório, seja V_H ou V_L, terá três subdiretórios, chamados InitialRound, FinalRound e SelectedSequences e um arquivo `csv`, com o número de sequências a cada etapa. Os diretórios InitialRound e FinalRound possuem os arquivos produzidos pelo controle de qualidade, montagem, filtragem, tradução e cálculo de frequência dos ciclos inicial e final de *phage display*, respectivamente. Já o diretório SelectedSequences possui um arquivo contendo sequências de clones enriquecidos, um arquivo contendo as sequências dos n primeiros clones enriquecidos, um arquivo contendo sequências numeradas pelo Abnum, isto é, o arquivo de clones candidatos e um arquivo com a classificação dos clones candidatos de acordo com as *germlines*. No diretório pai do projeto, além dos subdiretórios V_H e V_L, são criados arquivos log para registrar erros ou a saída padrão dos programas executados pelo método, e um subdiretório chamado Report, que contém o arquivo `html`, o qual apresenta um relatório da análise com os principais resultados, todas as imagens incluídas no `html` e um arquivo de log do

html_creator.pl.

2.13 Análise de distâncias do domínio variável

Foi realizada uma análise de distâncias entre os resíduos usados pelo programa *translateab9*, com o propósito de escolher distâncias mais acuradas para detecção de domínios variáveis. Para observar as distâncias entre os dois primeiros resíduos de cisteína do domínio variável, foram usadas sequências de *germline*, disponibilizadas na seção "Ig Germline Genes" da ferramenta IgBlast, versão online do NCBI (Ye *et al.*, 2013), tanto de germline de V_H quanto de V_L humanos. Para analisar o tamanho da CDR3, a qual por sua vez está presente na junção do segmento V e J (para V_L) ou V, D e J (para V_H), foi necessário obter sequências já recombinadas. Assim, foram obtidas sequências recombinadas aleatórias do NCBI, de tamanho entre 100 e 300 pb, com as seguintes palavras-chave:

- immunoglobulin heavy chain variable region, partial AND "Homo sapiens"[porgn:__txid9606]
- immunoglobulin kappa chain variable region, partial AND "Homo sapiens"[porgn:__txid9606]
- immunoglobulin light chain variable region, partial AND "Homo sapiens"[porgn:__txid9606]

Foram desenvolvidas duas versões de um programa Perl, *count_distance_germline.pl* e *count_distance_cdr3.pl*, os quais leem o arquivo em formato `fasta`, contendo as sequências, e imprimem num arquivo de saída, em formato `csv`, a distância entre os resíduos e o número de sequências que apresentam tal distância. O primeiro programa calcula a distância entre os dois resíduos de cisteína, já o segundo calcula o tamanho da CDR3. Uma vez calculadas as distâncias, foram construídos gráficos com a ferramenta R (R Core Team, 2015).

Os programas de cálculo de distâncias desconsideram sequências contendo mais de dois resíduos de cisteínas, pois estas confundem o motor de expressão regular Perl, cuja característica principal é estender a expressão regular o quanto for possível. Uma vez encontrado o primeiro resíduo de cisteína, o motor Perl estende o padrão até a n-ésima cisteína. Portanto, as distâncias calculadas a partir de tais sequências não iriam refletir as distâncias reais ente resíduos canônicos do domínio variável. Desse modo, tornou-se mais apropriado não utilizar tais sequências na análise de distância. Ressalta-se que

atualmente o NCBI removeu os bancos de *germlines* humanos da seção "Ig Germline Genes"da ferramenta IgBlast. Em virtude disso, não é possível reproduzir a busca por *germlines humanos* no presente momento.

2.14 Análise BLAST de perfil de imunoglobulinas

Para contribuir com os resultados obtidos pela análise de sequências de imunoglobulinas, dos conjuntos 454 Roche e Illumina S1 e S2, as bibliotecas foram alinhadas contra bancos de *germlines* humanos, os mesmos usados na classificação de *germlines*. O programa BLAST (Basic Local Alignment Search Tool) foi utilizado para realizar os alinhamentos (Altschul *et al.*, 1990). Esta ferramenta compara sequências por meio de alinhamento local [15], e atribui um *score* de similaridade ao alinhamento, como uma medida do quão parecidas são as sequências entre si. Nesta análise, as configurações foram usadas com valores *default*, exceto para os valores de *e-value* [16], a fim de garantir confiabilidade aos alinhamentos obtidos.

Considerando que a análise de enriquecimento é baseada na frequência dos clones das bibliotecas filtradas, estas foram a entrada para o BLAST. O alinhamento tem por objetivo não somente corroborar que os valores de frequência calculados de fato correspondam a frequência de clones de imunoglobulinas, mas também demonstrar a capacidade do método em trabalhar com bibliotecas que possuam pelo menos parte das sequências desprovidas de perfil de imunoglobulinas ou com algum outro tipo de problema, como deleções e *frameshift*. Os valores de *e-value* utilizados foram, de 10^{-20} a 10^{-5}, em intervalos regulares de 10^{-5}. As bibliotecas de entrada possuem sequências de nucleotídeos, enquanto os bancos de dados, sequências de aminoácidos. Assim foi usado o programa blastx, do antigo pacote blastall do BLAST, que alinha sequências traduzidas contra um banco de sequência proteicas (Altschul *et al.*, 1990).

[15] Alinhamento local: é o alinhamento entre *substrings* das sequências comparadas (Setubal, Meidanis & Setubal-Meidanis, 1997)

[16] *E-Value*: estimativa proporcional à probabilidade de um alinhamento possuir um dado *score* ao acaso, em um banco de sequências de tamanho conhecido.

2.15 Análise de diversidade das bibliotecas de *phage display*

A análise da diversidade foi realizada usando dois conceitos, *cluster* e entropia. *Cluster* compreende um conjunto de sequências mais similares entre si que com outras sequências. Para as bibliotecas de *phage display*, um *cluster* representará um clone. O programa CD-HIT (Li, Jaroszewski & Godzik, 2001; Li & Godzik, 2006) foi utilizado para encontrar *clusters* de sequências nas bibliotecas V_H e V_L dos conjuntos Illumina S1 e 454 Roche. O CD-HIT é mais rápido que outros programas de agrupamento devido ao seu algoritmo, que evita alinhamentos com *score* de similaridade[17] acima de um dado limite.

Para prever a similaridade de um alinhamento, o algoritmo procura subsequências de tamanho definido (2 a 5 aminoácidos para proteínas, e 8 a 12 nucleotídeos para DNA) que as sequências comparadas possuam em comum. Uma vez prevista a similaridade, o programa constrói o alinhamento local entre as duas sequências comparadas somente se a similaridade estiver acima do limite estabelecido (Holm & Sander, 1998). As sequências são ordenadas por tamanho, de modo que a sequência mais longa se torna representativa do primeiro *cluster*. As demais sequências são comparadas com as representativas dos *clusters* existentes. Se a similaridade de uma sequência com uma representativa qualquer está acima de um limite, a sequência é incluída no *cluster*. Caso a sequência não possa ser incluída em nenhum dos *clusters* existentes, um novo *cluster* é criado tendo esta sequência como representativa (Li & Godzik, 2006).

Na presente abordagem os critérios para incluir membros em um *cluster* foram similaridade e a identidade de sequências[18]. O CD-HIT foi configurado para identificar *clusters* cujas sequências tivessem similaridade entre 90% a 100%. Para cada biblioteca, o CD-HIT foi executado com diferentes valores de identidade de nucleotídeo, de 80% a 100%, a intervalos regulares de 5%, usando 32 processadores.

Uma vez identificado o número de *clusters* nas bibliotecas, a diversidade pode ser estimada por meio do índice de Shannon ou entropia de Shannon. A equação da entropia de Shannon foi originalmente formulada para medir a incerteza média sobre os símbolos

[17] *Score* de similaridade: Pontuação atribuída a um alinhamento baseada em um sistema de pontos para *match* (par idêntico), *mismatch* (par não idêntico) e *gaps* (lacunas no alinhamento).

[18] Identidade: número de resíduos idênticos dividido pelo tamanho da sequência mais curta (Li, 2015).

que compõem mensagens produzidas por uma dada fonte, no contexto da teoria da informação (Blachman, 1968). No entanto, ecólogos têm usado a entropia de Shannon para estimar a diversidade de comunidades ou populações (Magurran, 2013). Nesse sentido, quanto maior o número de espécies em uma comunidade, maior será a incerteza sobre a espécie a que pertence o próximo indivíduo amostrado (Mayer, Donovan & Pawlowski, 2014).

Para as bibliotecas de *phage display* foi usado raciocínio similar, porém, em termos de clones e não de espécies. Os valores de entropia de Shannon permitem inferir sobre a incerteza do clone ao qual pertence uma sequência da amostra e tem como vantagem o fato de ser sensível a variações na abundância, isto é, possibilita lidar com amostras em que as espécies não são igualmente distribuídas (Jost, 2006). Desse modo, foi calculado o índice de Shannon, em bits pela equação

$$H = -\sum_{i=1}^{M} P_i log_2 Pi, \tag{3}$$

onde H corresponde ao índice de Shannon, M corresponde ao número total de *clusters* da biblioteca e P_i corresponde à proporção de sequências que pertencem a um *cluster* i. Foi desenvolvido um *script* Perl, *entropycalculator.pl*, que recebe como entrada um arquivo de saída do CD-HIT da biblioteca inicial e da final de V_H ou de V_L, contendo os *clusters* identificados com suas respectivas sequências membros, e calcula a entropia de Shannon usando a equação descrita acima. Como saída, o programa imprime a entropia de Shannon da biblioteca inicial e da biblioteca final.

3 Resultados e Discussão

3.1 Resultados produzidos pelo método automatizado

O resultado de maior interesse compreende a lista de clones candidatos de V_H e de V_L. No conjunto Illumina S1 foram encontrados 9 candidatos para V_H, e 10 candidatos para V_L, com valores de *fold change* acima de 100 (Figuras 9 e 10), e que puderam ser reconhecidos como domínio variável de imunoglobulina tanto pelo *translateab9* quanto pelo Abnum.

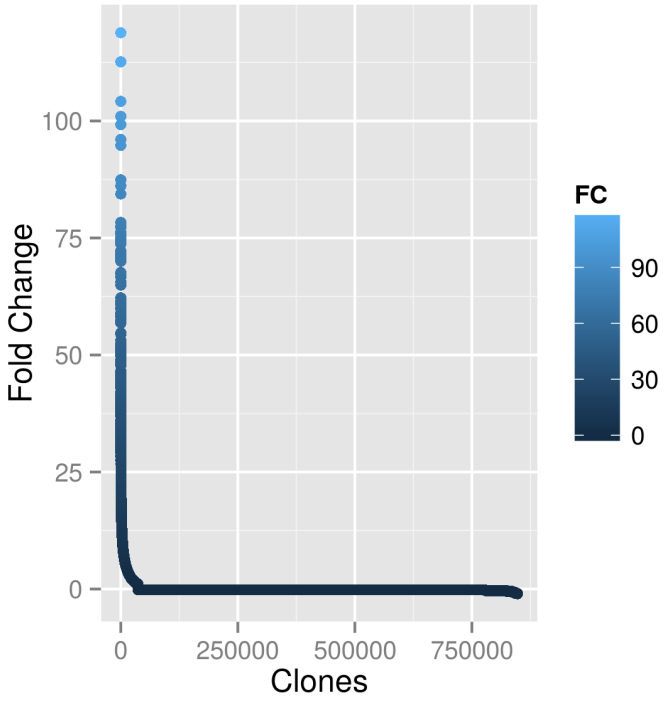

Figura 9: Distribuição de *fold change* do conjunto V_H Illumina. O valor de *fold change* é proporcional à amplificação do clone e, portanto, é maior para clones que sofreram seleção mais acentuada durante o experimento de *phage display*.

Os gráficos desta seção mostram valores de *fold change* de todos os clones das bibliotecas, enriquecidos e não enriquecidos, e para obter estas listas de clones foi desenvolvida uma versão adicional do programa *find_duplicates7.pl*, que diferente da versão original, não imprime somente uma lista de clones enriquecidos, mas sim de todos os clones de uma biblioteca. Foram extraídos então os valores de *fold change* do arquivo de saída

da versão adicional, e com o pacote R foram construídos gráficos para mostrar a distribuição dos clones de acordo com os valores de *fold change*. Ressalta-se que estes gráficos não são gerados pelo método automatizado, mas posteriormente poderia ser incluída no método a criação de um arquivo em formato `csv` contendo os valores de *fold change* de todos os clones de uma dada biblioteca, de modo que o usuário pudesse observar as mudanças na proporção dos clones da biblioteca inicial para final, tanto de V_H quanto de V_L.

Como visto nas Figuras 9 e 10, uma pequena fração das bibliotecas possui *fold change* positivo, o que permite inferir que esta fração de clones foi enriquecida durante a seleção de *phage display* e, portanto, constitui uma evidência de que a seleção do experimento foi bem sucedida para este conjunto.

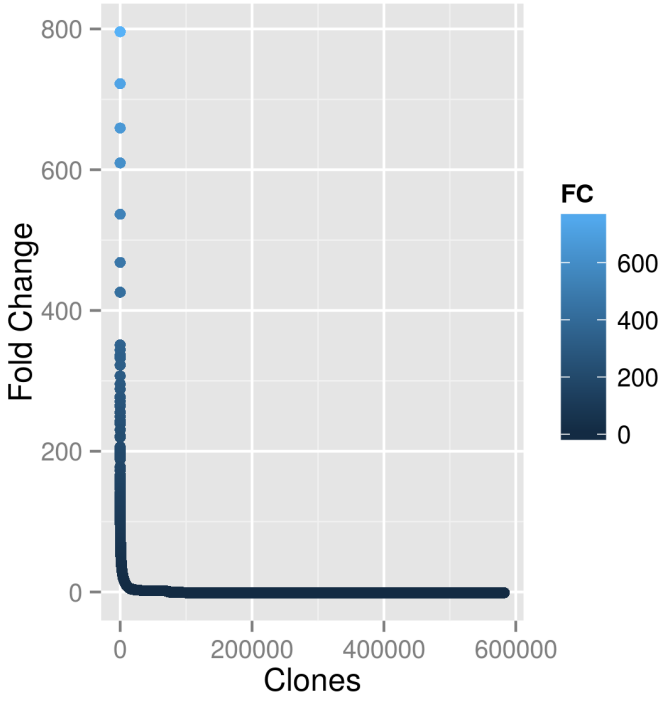

Figura 10: Distribuição de *fold change* do conjunto V_L Illumina.

Com relação à análise do conjunto 454 Roche, foi possível encontrar 10 candidatos para V_H, e nenhum para V_L. Dentre as sequências candidatas de V_H, apenas as duas primeiras apresentam *fold change* acima de 100 (Figura 11). Assim como ocorreu no conjunto Illumina S1, uma pequena fração da biblioteca de V_H foi enriquecida. A lista de candidatos de V_H apresenta particularidades em alguns aspectos dos resultados

gerados pela análise. Dentre as observações importantes está o *fold change* da primeira sequência candidata, que destaca-se por apresentar uma grandeza consideravelmente maior, de 10^4, em comparação aos demais valores, de grandeza de no máximo 10^2.

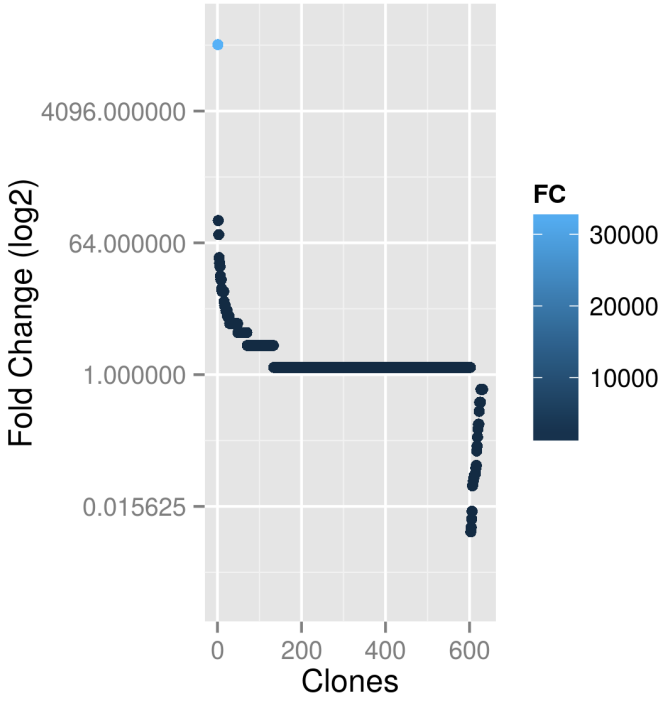

Figura 11: Distribuição de *fold change* do conjunto V_H 454 Roche.

Outro ponto diz respeito à classificação de *germlines*, pois diferentemente do conjunto Illumina S1, todas as sequências candidatas de V_H do conjunto Roche foram classificadas como pertencentes a uma mesma *germline*, VH1-8. Não obstante, as sequências candidatas são bastante similares entre si, o que pode ser observado na tabela de identificação das regiões do domínio variável presente no arquivo `html` (dados não apresentados por exigência de sigilo dos autores).

Além disso, o alinhamento múltiplo de nucleotídeos das sete primeiras[19] sequências, realizado com a ferramenta online Clustal Omega (Sievers *et al.*, 2011; Squizzato *et al.*, 2015), produziu uma matriz de identidade (Anexo A), tal que a segunda e a terceira sequência candidata possuem 98.89% de identidade, já a primeira sequência da lista

[19]Foram alinhadas apenas sete sequências, pois três candidatas não puderam ter as sequências de nucleotídeos recuperadas em virtude da sequência de aminoácidos possuir um ou mais resíduos desconhecidos. O programa *get_ntsequence.pl* recupera apenas sequências de nucleotídeos cujas sequências de aminoácidos possuam todos os resíduos conhecidos.

de candidatos apresenta identidade de 96% com as demais sequências. É provável que os 7 clones ou sequências candidatas, constituam na verdade apenas dois clones, dado a classificação de *germlines* e também os valores de identidade de nucleotídeos do alinhamento múltiplo.

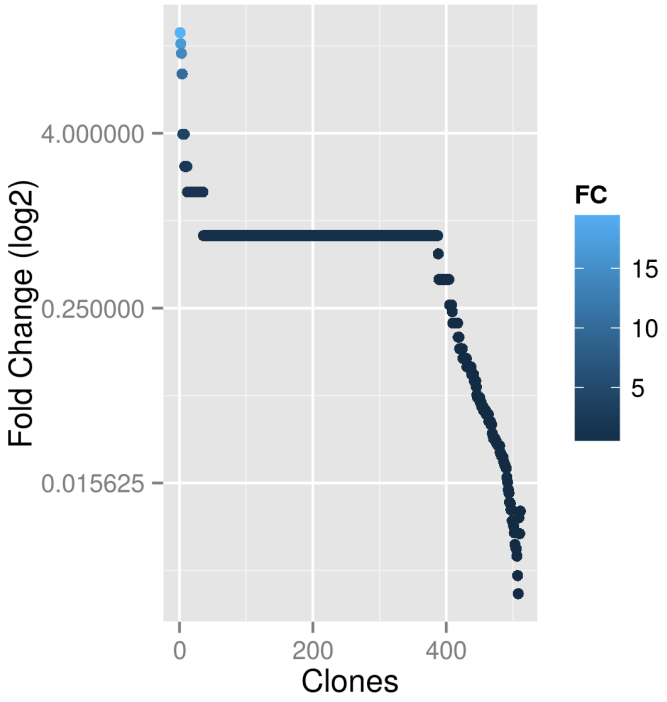

Figura 12: Distribuição de *fold change* do conjunto V_L 454 Roche

Supondo que tais sequências realmente pertençam a dois clones, entre as possíveis causas, pode-se sugerir erros da plataforma de sequenciamento 454 Roche, cujas taxas de inserções, deleções e substituições tem sido registradas na literatura (Prabakaran *et al.*, 2011), e variação natural gerada por hipermutações somáticas. A baixa qualidade média de score PHRED das bibliotecas de V_H e de V_L sugere como causa mais provável erros produzidos pelo sequenciador. Seja qual for o evento que gerou esta variação entre sequências de um mesmo clone, é fato que o presente método apresenta limitações em lidar com variações pontuais em sequências de aminoácidos, afinal as sequências são reunidas em um clone usando busca exata por um subsequência dotada de marcas canônicas de domínio variável.

Apesar desta limitação, o método fornece elementos que permitem investigações mais aprofundadas sobre as sequências candidatas, tais como a tabela de classificação

de *germlines*, a tabela de identificação das regiões do domínio variável, os gráficos referentes à qualidade dos *reads*[20], todos os arquivos `fasta` contendo as sequências de aminoácidos e de nucleotídeos, bem como os valores de *fold change*. Dessa maneira, os resultados produzidos possibilitam contornar limitações relacionadas à variações nas sequências de um mesmo clone.

Quanto às bibliotecas V_L do conjunto 454 Roche, como é discutido mais adiante, a biblioteca final de V_L teve algum problema durante o experimento ou na amplificação por PCR anterior ao sequenciamento, pois a biblioteca está consideravelmente comprometida, de modo que a maioria das sequências foram descartadas ao longo das etapas da análise. A Figura 12 exibe valores de *fold change* anormais, se comparados com os valores de *fold change* das demais bibliotecas, afinal não há nenhum clone cujo enriquecimento tenha sido da ordem de 10^2, e mesmo os primeiros candidatos não foram reconhecidos como domínio variável de imunoglobulina pelo Abnum.

3.2 Proporção de imunoglobulinas nas bibliotecas de *phage display*

Nos alinhamentos realizados pelo BLAST para cada um dos valores de *e-value*, de 10^{-20} a 10^{-5}, as bibliotecas dos conjuntos Illumina S1, 454 Roche e Illumina S2 foram usadas como *query*[21], contra os bancos de *germlines*, de V_H e V_L. A proporção de imunoglobulinas nas bibliotecas foi estimada a partir do número de sequências que não tiveram nenhum *hit* contra o banco de *germlines*. O comando *grep* do terminal permite obter o número de ocorrências de um dado padrão, que neste caso foi a *string* "No hit", que aparece 8 linhas após o identificador da sequência, caso o BLAST não tenha encontrado nenhum *hit* cujo *score* tenha *e-value* acima de um dado valor. O tamanho da biblioteca também foi obtido com o comando *grep*, porém, usando o padrão "^>", isto é, a linha correspondente ao identificador de cada sequência.

A partir do número de sequências sem *hit* e do tamanho da biblioteca, foram calculadas as porcentagens de sequências com e sem *hits* para imunoglobulina. Todos os gráficos foram produzidos com o pacote R (R Core Team, 2015), usando o biblioteca

[20] Exemplos de gráficos gerados pelo método automatizado encontram-se no Anexo B.
[21] *Query*: sequência de interesse que é comparada contra um banco de sequências.

ggplot2.

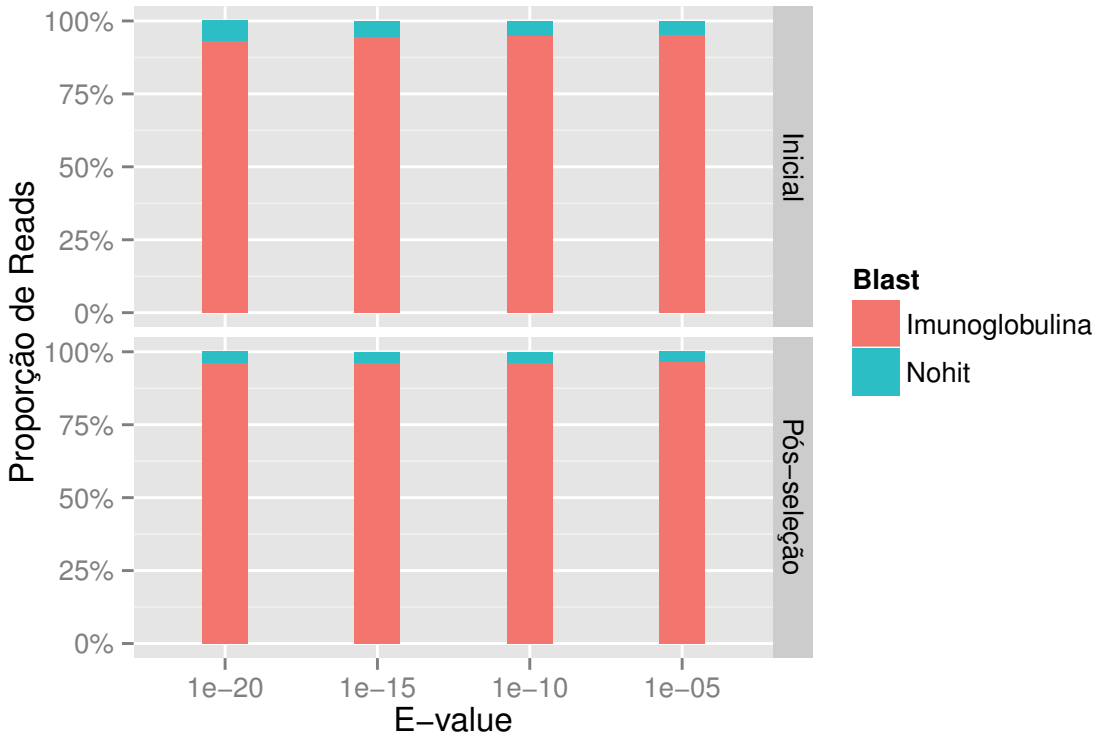

Figura 13: Proporção de imunoglobulinas nas bibliotecas inicial e final de V_H do conjunto Illumina S1, para diferentes valores de *e-value*.

As bibliotecas inicial e final de V_H, do conjunto Illumina S1, apresentaram pouca variação na proporção de imunoglobulinas encontradas pelo BLAST, nos diferentes valores de *e-value*. A biblioteca inicial de V_H apresentou proporção de 93,1% a 95,3% de imunoglobulinas, correspondentes aos *e-values* de 10^{-20} até 10^{-5} (Figura 13).

Já a biblioteca final de V_H apresentou proporção de 96,2% a 96,5% de imunoglobulinas, correspondentes aos *e-values* de 10^{-20} até 10^{-5} (Figura 13). Em todas as execuções, as bibliotecas apresentaram mais de 90% de imunoglobulinas, o que contribui com a suposição de que a maioria das sequências que compõem as bibliotecas são similares a imunoglobulinas.

Os alinhamentos das bibliotecas inicial e final de V_L, do conjunto Illumina S1, demonstraram que a proporção de imunoglobulinas diminui gradativamente para valores de *e-value* mais estringentes (Figura 14). A biblioteca inicial de V_L apresentou de 72,7% a 98% de imunoglobulinas, para a faixa de valores de *e-value* mencionada anteriomente.

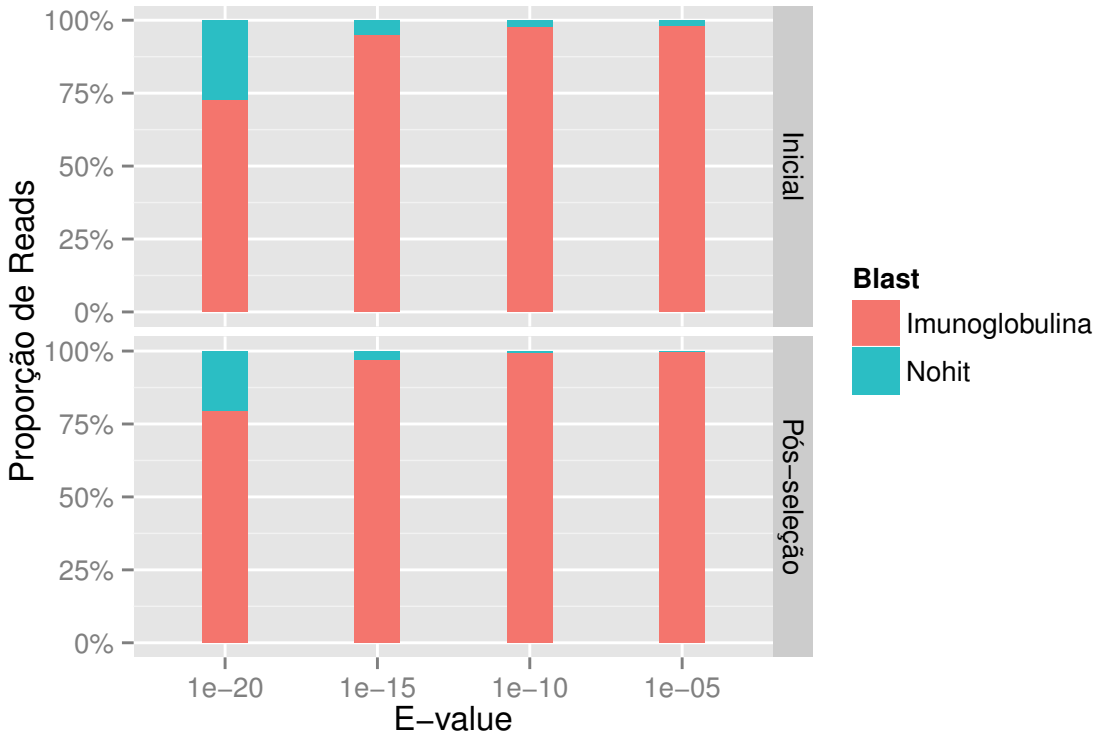

Figura 14: Proporção de imunoglobulinas nas bibliotecas inicial e final de V_L do conjunto Illumina S1, para diferentes valores de *e-value*.

Quanto a biblioteca final de V_L, esta apresentou de 79.8% a 99.7% de imunoglobulinas. Exceto para o *e-value* de 10^{-20}, todas as execuções do BLAST encontraram fração de imunoglobulinas acima de 90% nas bibliotecas V_L.

Com relação às bibliotecas V_H do conjunto Roche, é possível notar proporções semelhantes entre as bibliotecas inicial e final (Figura 15). O BLAST encontrou de 97,1% a 99,7% de imunoglobulinas na biblioteca inicial de V_H, intervalo de valores consideravelmente próximo do intervalo de valores da biblioteca final, que vai de 97,6% a 99,9%. Para todos os valores de *e-value*, foram encontradas proporções de imunoglobulina acima de 90%.

As bibliotecas V_L do conjunto Roche apresentaram diminuição gradativa da fração de imunoglobulinas encontradas pelo BLAST para valores menores de *e-value*, assim como as bibliotecas do conjunto Illumina S1 (Figura 16). A biblioteca inicial apresentou proporção de 71,7% a 98,3% de imunoglobulinas. No entanto, somente as execuções com *e-value* de 10^{-10} e de 10^{-5} encontraram mais de 90% de imunoglobulinas na biblioteca

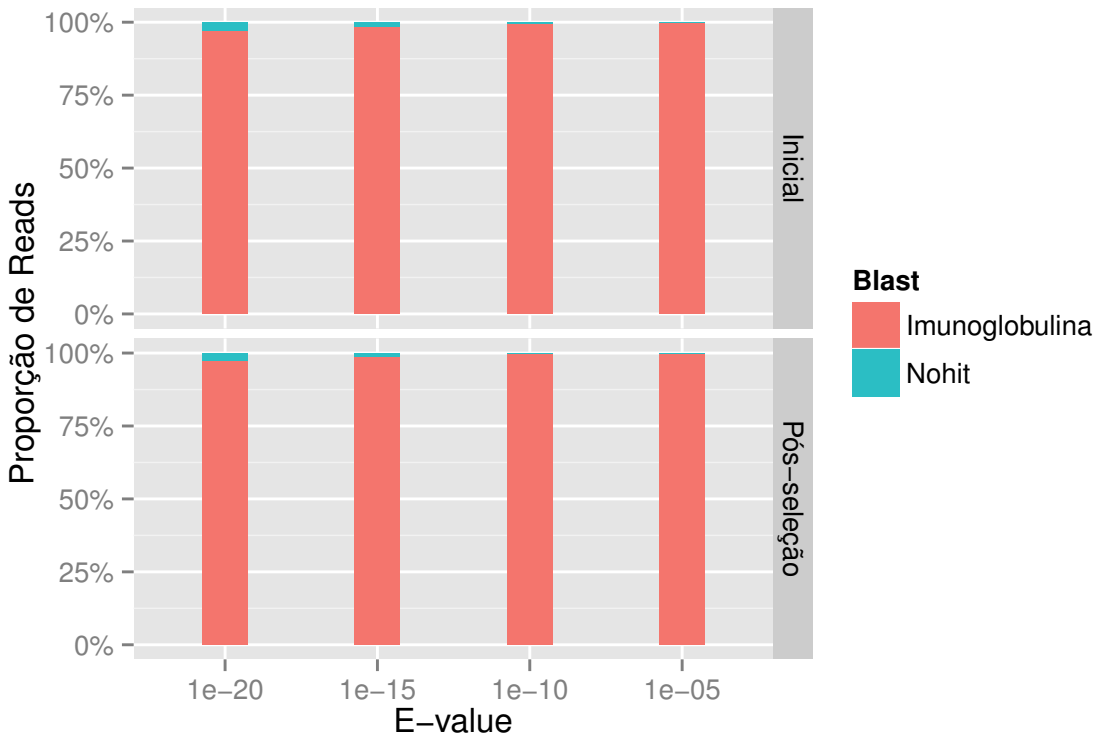

Figura 15: Proporção de imunoglobulinas nas bibliotecas inicial e final de V_H do conjunto 454 Roche, para diferentes valores de *e-value*.

inicial de V_L.

A biblioteca final de V_L do conjunto Roche apresenta mais de 90% de imunoglobulinas para os valores de *e-value* de 10^{-10} e de 10^{-5}. Porém, para os valores de 10^{-20} e de 10^{-15}, a maioria das sequências não possui nenhum *hit* contra o banco de *germlines*, sendo a porcentagem de imunoglobulinas correspondente a, respectivamente, 0,36% e 1,8%.

Existem diferentes evidências para supor que os dados desta biblioteca em especial passaram por algum tipo de problema na fase de bancada, durante os experimentos de *phage display*. A primeira evidência consiste no fato de que não foram encontradas sequências candidatas de V_L do conjunto Roche. Embora alguns clones de fato tenham sido amplificados, isto é, enriquecidos da biblioteca inicial para final, nenhum deles foi reconhecido como imunoglobulina pelo Abnum. O segundo indício é a drástica redução do tamanho da biblioteca na etapa de tradução, em que foi traduzido apenas 0,56% da biblioteca filtrada. Como terceira evidência tem-se os resultados de uma análise manual,

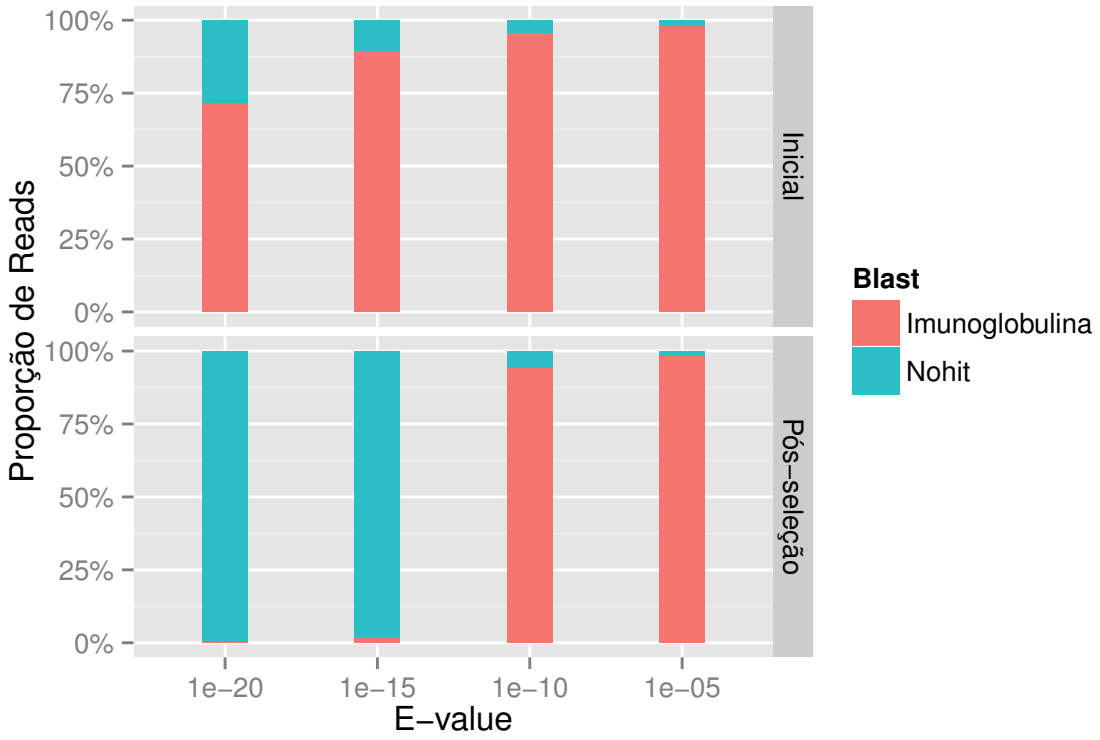

Figura 16: Proporção de imunoglobulinas nas bibliotecas inicial e final de V_L do conjunto 454 Roche, para diferentes valores de *e-value*.

em que as sequências foram traduzidas pela ferramenta Transeq do pacote EMBOSS (The European Molecular Biology Open Software Suite) (Rice *et al.*, 2000), a partir dos quais foi possível notar que existe um clone altamente amplificado que possui deleções. Finalmente, como último argumento, a amplificação destes clones espúrios também foi observada por análise de sequenciamento Sanger. Dessa maneira, independente da plataforma de sequenciamento ou do método de análise de bioinformática, esta biblioteca final de V_L está comprometida.

Com relação ao conjunto Illumina S2, as bibliotecas inicial e final de V_H apresentaram proporção notavelmente alta de imunoglobulinas entre os diferentes valores de *e-value*, de modo que todas as execuções encontraram aproximadamente 99,9% de *hits* de imunoglobulina para ambas as bibliotecas (Figura 17).

Quanto às bibliotecas V_L do conjunto Illumina S2, estas apresentaram diminuição gradual da fração de imunoglobulinas para valores gradativamente menores de *e-value*, e ainda com variações de proporção de imunoglobulinas bastante similares. Na biblioteca

inicial, a proporção de *hits* de imunoglobulinas variou de 85,4% a 99,9%, para os valores de *e-value* de 10^{-20} a 10^{-5}, respectivamente (Figura 18). Já a biblioteca final apresentou um intervalo de 86% a 99,9%, para o mesmo intervalo de valores de *e-value*.

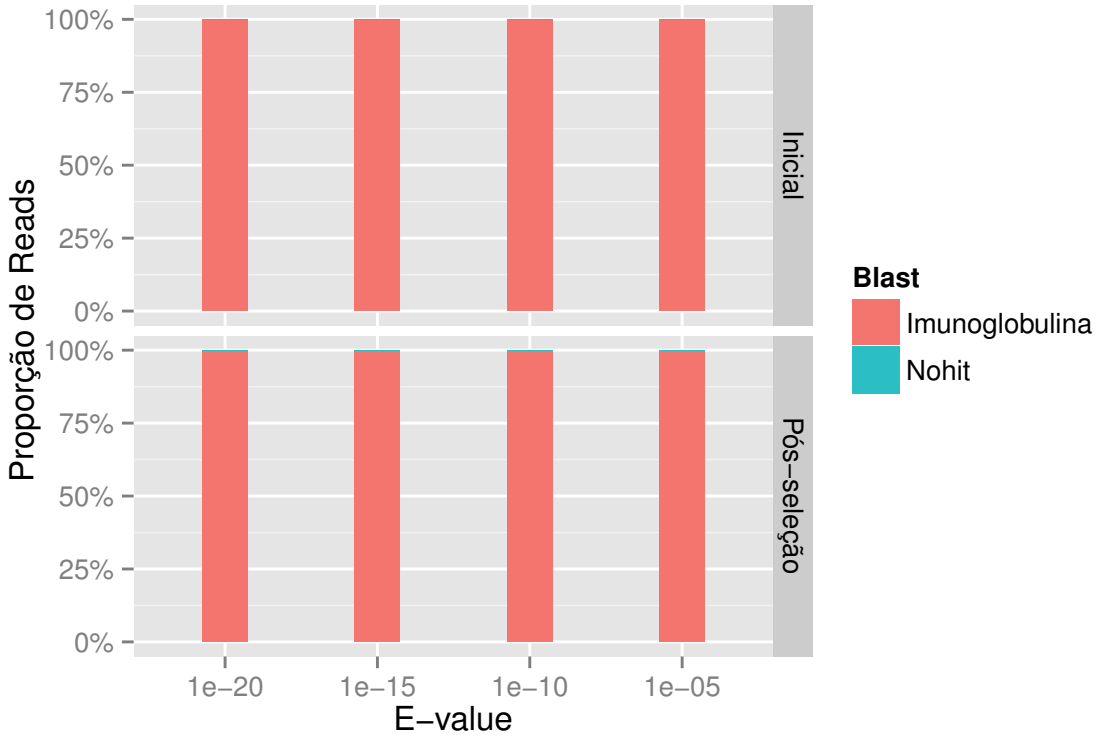

Figura 17: Proporção de imunoglobulinas nas bibliotecas inicial e final de V_H do conjunto Illumina S2, para diferentes valores de *e-value*.

Considerando que os três conjuntos utilizaram a mesma fonte de sequências de domínio variável nos experimentos de *phage display*, e diferenciam-se pelo antígeno utilizado nos ciclos de seleção, seria razoável comparar as bibliotecas iniciais de V_H e V_L entre os diferentes conjuntos. Desse modo, os conjuntos Illumina S1, 454 Roche e Illumina S2 podem ser vistos como 3 amostras da biblioteca original. Embora o número de amostras seja pequeno, e não seja possível estender suposições para a biblioteca original, existem observações sobre as amostras que podem ser aqui descritas.

A primeira observação consiste no fato de que existe uma proporção de imunoglobulinas maior nas bibliotecas iniciais de V_H que nas bibliotecas iniciais de V_L, o que pode ser constatado a partir da comparação entre os intervalos de proporções de imunoglobulinas, mais restritos e mais elevados em V_H que em V_L (Figuras 13 a 18).

A segunda refere-se às bibliotecas iniciais de V_L, as quais apresentam diminuição gradual da fração de *hits* para imunoglobulina concomitante a diminuição dos valores de *e-value*. Para tais bibliotecas, quanto mais exigente o *e-value*, menor a quantidade de sequências identificadas como imunoglobulinas. Os *e-values* de 10^{-10} e 10^{-5} permitiram encontrar acima de 90% de sequências com *hits* para imunoglobulinas, em todas as amostras de V_L. Diante disso, nota-se que a identificação de imunoglobulinas é dependente de *e-value* para as bibliotecas iniciais de V_L, dependência esta que não ocorre para as bibliotecas iniciais de V_H. Supõe-se que seja mais uma evidência de que as bibliotecas iniciais de V_L realmente possuam uma fração menor de sequências de imunoglobulinas que as bibliotecas de V_H.

Os conjuntos Illumina S1 e S2 permitem uma comparação mais equivalente pois utilizaram além da mesma fonte de sequências de domínio variável, a mesma plataforma de sequenciamento. Tanto nas bibliotecas iniciais de V_H quanto nas de V_L, as proporções de imunoglobulinas encontradas são mais altas no conjunto Illumina S2.

Embora ambos os conjuntos apresentem qualidade média por base adequada para a maioria dos *reads*, isto é, qualidade PHRED acima de 20, o conjunto S1 perde uma quantidade maior de sequências na etapa de tradução. A biblioteca inicial de V_H do conjunto S1 tem apenas 17,5% de sequências traduzidas e dotadas de assinatura de domínio variável. Já a biblioteca inicial de V_H do conjunto S2 tem 75,6% das sequências traduzidas e contendo marcas de anticorpo.

Apesar de não serem tão discrepantes as proporção de sequências traduzidas das bibliotecas iniciais de V_L dos conjuntos S1 e S2, o conjunto S1 ainda possui quantidade menor, 77,8% de sequências traduzidas, enquanto o conjunto S2 apresenta 85% de sequências traduzidas. Sendo assim, os resultados da etapa de tradução corroboram a ideia de que as bibliotecas iniciais do conjunto S2 possuem uma fração maior de sequências identificadas como imunoglobulinas, o que é válido tanto para V_H quanto para V_L.

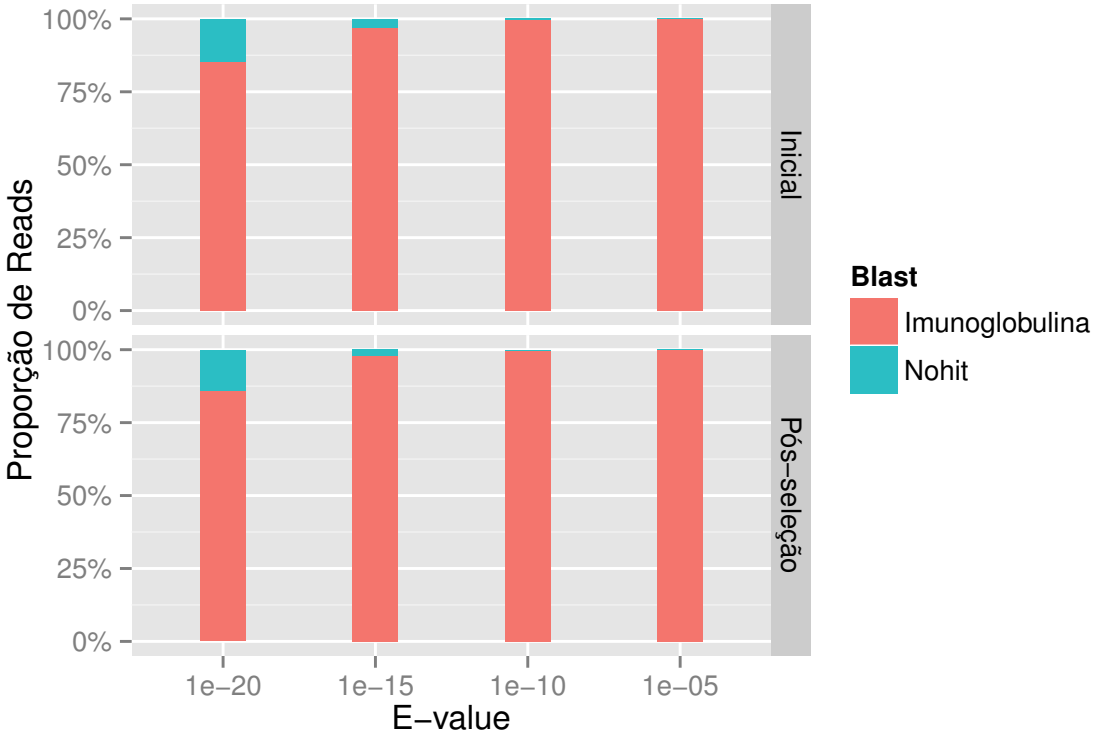

Figura 18: Proporção de imunoglobulinas nas bibliotecas inicial e final de V_L do conjunto Illumina S2, para diferentes valores de *e-value*.

3.3 Distâncias entre resíduos canônicos do domínio variável

Foram obtidos dois tipos de conjuntos tanto para V_H quanto para V_L. Um dos conjuntos contem sequências *germline*, e o outro possui sequências já recombinadas, dotadas de CDR3 e FR4. Como mostra a Tabela 3, os conjuntos de sequências recombinadas é consideravelmente maior que os de *germlines*. Tal discrepância é coerente com o fato de existir um número limitado de linhagens que geram toda a diversidade possível de anticorpos da espécie humana, em contrapartida ao número gigantesco de possíveis combinações dos segmentos gênicos V, D, J (V_H) ou V e J (V_L).

Tabela 3: Tamanhos das bibliotecas usadas na análise de distância

Biblioteca	**Número de sequências**
VH_G	44
VH_R	39914
VL_G	36
VL_R	14559

G: *germlines*. R: recombinados.

Ambos os conjuntos, *germlines* e recombinados, apresentaram uma distribuição de sequências em intervalos similares de distâncias entre os resíduos de cisteína. Tal observação é válida para V_H e para V_L. Como mostra a Figura 19, a maioria das sequências germlines de V_H manteve distâncias dentro de um intervalo de 71 a 76 resíduos, e as sequências recombinadas, dentro de um intervalo de 71 a 77 resíduos. Diante disso, no programa de tradução, o intervalo de resíduos admitido entre as duas cisteínas de V_H foi definido entre 70 e 78 resíduos. Ressalta-se ainda que embora todas as sequências constituintes do pico de 69 resíduos possuam duas cisteínas, tal distância foi desconsiderada, pois a maioria das sequências (97,14 %) corresponde a anticorpos artificiais, derivados de um único trabalho, como por exemplo a sequência depositada no GenBank com o GI 58222213. Assim, seria razoável pensar que tal distância é específica para este tipo de sequências sintéticas, e não um padrão comum em domínios variáveis de cadeia pesada.

No intuito de comparar o intervalo encontrado e definido no presente método com as distâncias obtidas pelo grupo de Bioinformática da UCL, criador do Abnum, foi realizada a soma dos intervalos das regiões CDR1, FR2, CDR2 e FR3, os quais são

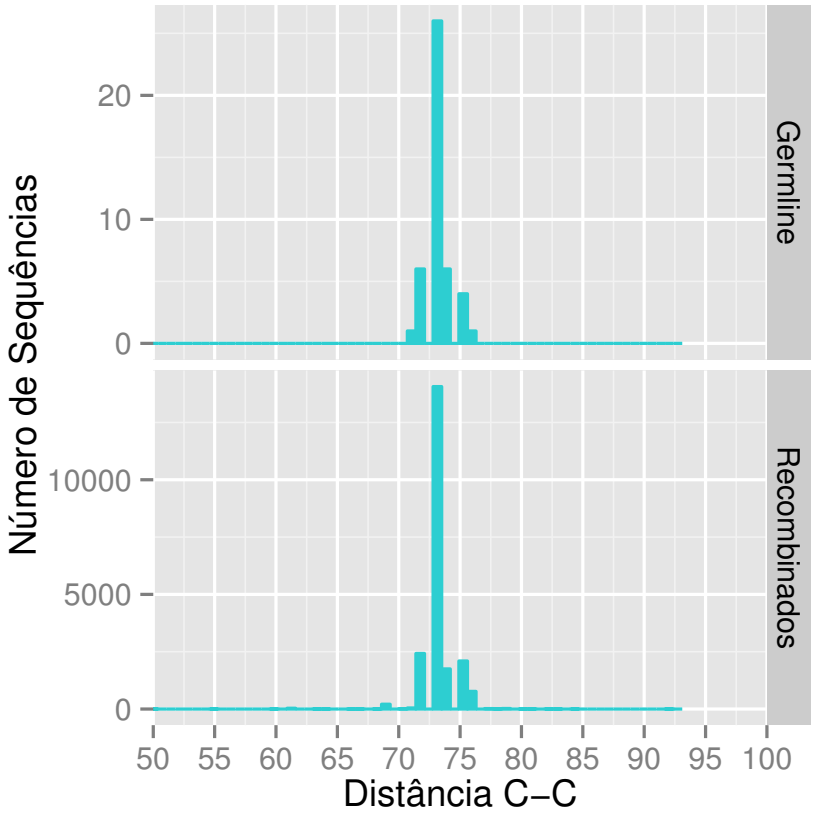

Figura 19: Distribuição de sequências de acordo com a distância entre os resíduos de cisteína de V_H.

apresentados no trabalho de 2008 do grupo (Abhinandan & Martin, 2008). Tais regiões foram escolhidas para comparação em virtude do programa de tradução buscar pelos dois resíduos de cisteína que delimitam o início da CDR1 e o início da CDR3, isto é, o conjunto de regiões consecutivas, da CDR1 até a FR3.

Os valores mínimo e máximo de resíduos constituintes de tais regiões, observados no banco de sequências de Kabat[22], pelo grupo da UCL, foram usados para estimar valores mínimo e máximo entre os dois resíduos de cisteínas do domínio variável. Assim, calculou-se um intervalo por meio da soma dos valores mínimos de cada região e da soma dos valores máximos de cada região. Desse modo, o intervalo usado no presente método para cadeia pesada, de 70 a 78 resíduos, está dentro do intervalo calculado a partir das distâncias observadas pelo grupo da UCL, que é de 51 a 84 resíduos.

[22]Banco de sequências de Kabat: é o banco de sequências de anticorpos humanos e murinos, a partir do qual foi criado o esquema de numeração Kabat, baseado apenas na variabilidade das sequências (Wu & Kabat, 1970).

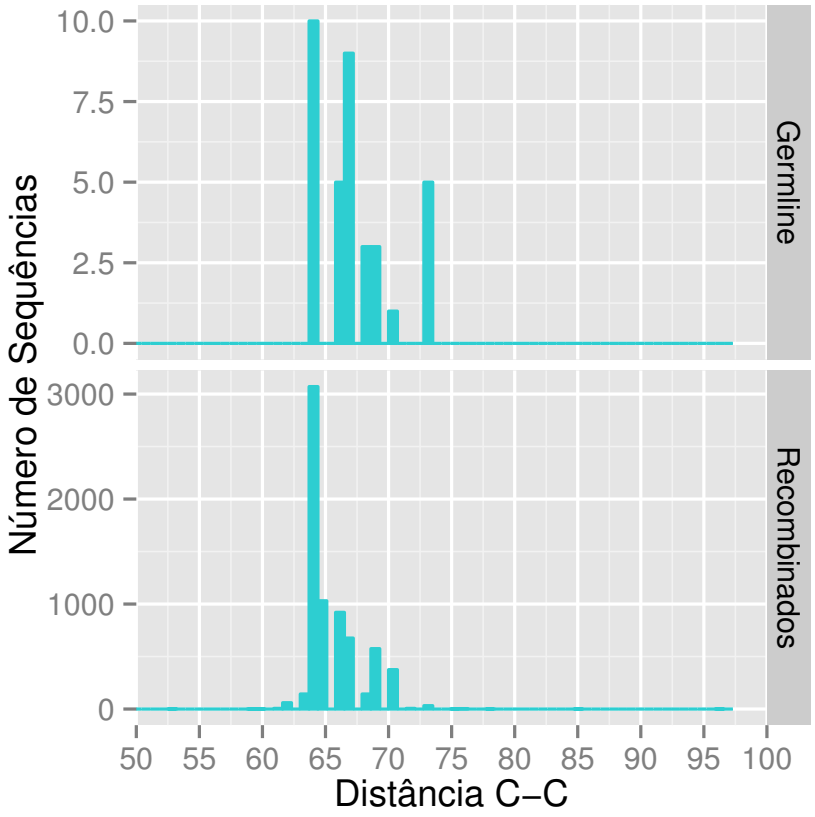

Figura 20: Distribuição de sequências de acordo com a distância entre os resíduos de cisteína de V_L.

Quanto a V_L, a maioria das sequências *germline* apresenta-se em um intervalo de distâncias de 64 a 73 resíduos, já as recombinadas mantiveram-se entre 62 e 74 resíduos (Figura 20). Assim, foi escolhido o intervalo de 62 a 74 resíduos entre as duas cisteínas de V_L, para o programa de tradução. O intervalo de número de resíduos entre as cisteínas da cadeia leve, estimado por meio das distâncias descritas no trabalho do grupo da UCL, é de 56 a 85 resíduos e, portanto, inclui o intervalo utilizado no presente método.

As distâncias usadas na tradução para encontrar o conjunto de regiões delimitado pelos dois resíduos de cisteína, apresentam-se mais restritas em relação ao citado na literatura, afinal o banco de Kabat é consideravelmente mais heterogêneo que as sequências utilizadas na análise de distância do presente trabalho, incluindo sequências um pouco mais longas e também de origem murina (Abhinandan & Martin, 2008). Todavia, os intervalos aqui definidos para V_H e para V_L estão incluídos nas distâncias mencionadas no trabalho de 2008, e concordam com distâncias encontradas nas *germlines*, as quais

dão origem a todas as sequências recombinadas.

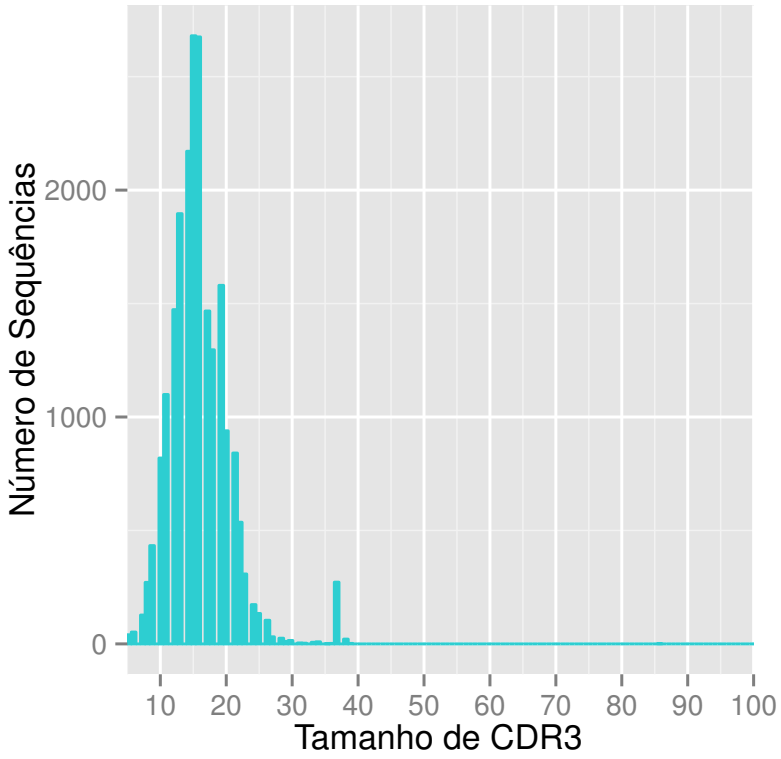

Figura 21: Distribuição de sequências de acordo com o tamanho da CDR3 de V_H.

No que diz respeito ao tamanho de CDR3, o intervalo de V_H (de 5 a 28 resíduos), mostrou-se mais amplo que o de V_L (de 5 a 13 resíduos) (Figuras 21 e 22). Tal diferença concorda com o fato de existir maior variação em V_H que em V_L, em virtude de V_H contar com a junção de três segmentos gênicos, V, D e J, enquanto V_L conta com apenas dois segmentos, V e J. Na Figura 21, que representa a distribuição de sequências de acordo com o tamanho da CDR3 de V_H, nota-se 1 pico de distâncias afastado da maioria, de 37 resíduos.

Esta distância foi desconsiderada pois supõe-se que seja específica para as sequências sintéticas, já que 100% das sequências do pico foram produzidas pelo mesmo trabalho (Doria-Rose *et al.*, 2014). Assim, foram definidos para o programa de tradução, os intervalos de 5 a 30 resíduos, para CDR3 de V_H e de 5 a 15 resíduos, para CDR3 de V_L. Tais intervalos estão consideravelmente próximos dos observados pelo grupo da UCL, os quais são de 2 a 30 resíduos para CDR3 de V_H, e 4 a 18 resíduos, para CDR3

de V_L (Abhinandan & Martin, 2008).

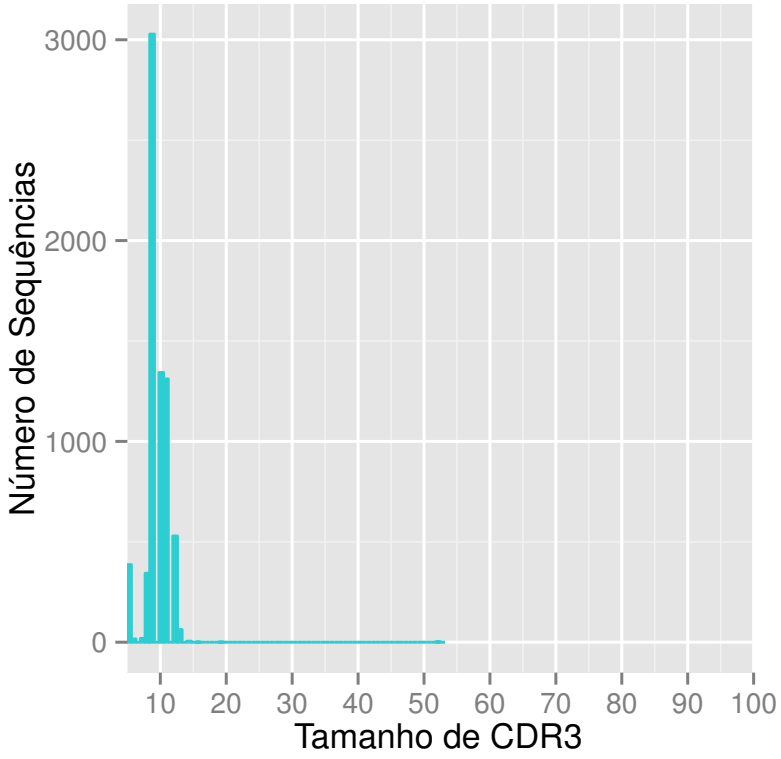

Figura 22: Distribuição de sequências de acordo com o tamanho da CDR3 de V_L.

Ressalta-se ainda que existem sequências de outras espécies tais como fragmentos artificiais de camelo, tubarão e aves, que podem apresentar resíduos de cisteínas não usuais dentro da CDR3 (Wu *et al.*, 2012; Harmsen *et al.*, 2000; Stanfield *et al.*, 2004). No entanto, as distâncias usadas neste método aplicam-se somente a sequências humanas de imunoglobulinas, visto que a análise de distâncias utilizou somente sequências humanas, e estas possuem frequência bem mais baixa de cisteínas não canônicas (1,6%) (Wu *et al.*, 2012). Sendo assim, embora restritas para casos gerais de sequências humanas de domínio variável, as distâncias usadas neste trabalho demonstraram-se válidas e coerentes com o descrito na literatura.

3.4 Otimização de programas

No presente método, os programas *translateab9* e *frequency_counter3.pl* foram otimizados. Embora a primeira versão do *translateab*, desenvolvida em Perl, fosse capaz de

traduzir as sequências e aplicar o primeiro critério do método, o programa apresentou tempos de execução inviáveis para bibliotecas NGS. Em virtude disso, foi desenvolvido um programa C, *translateab9*, cujos tempos de execução são muito menores que os da versão Perl (Figura 23). As maiores bibliotecas foram traduzidas em cerca de 2 a 3 horas pela versão Perl, e em no máximo 5 minutos pela versão C.

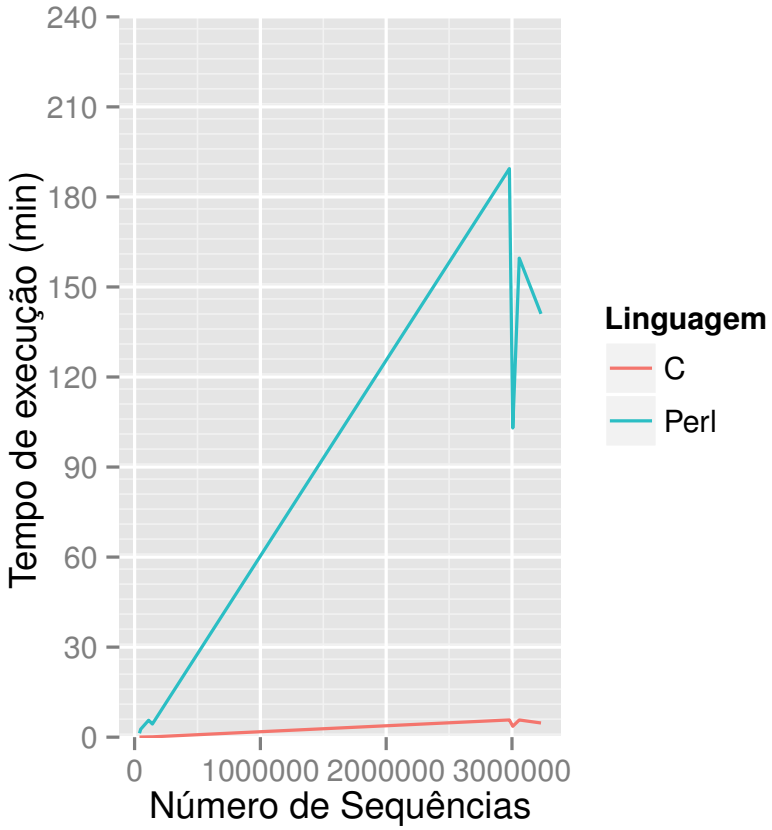

Figura 23: Tempos de execução do programa *translateab*, nas versões Perl e C.

Seria esperado que o tempo de execução fosse proporcional ao número de sequências de entrada. No entanto, como apresentado na Figura 23, ocorreram casos em que bibliotecas menores levaram mais tempo para serem traduzidas. Isto se deve ao fato de que as execuções foram simultâneas entre si, e que outros processos de outros usuários estavam em execução no servidor durante os testes. Tantos processos simultâneos manipulando conjuntos de dados muito grandes multiplicam as trocas entre a memória principal e a memória cache e, portanto, sobrecarregam a memória cache e aumentam o tempo de processamento.

Quanto a eficiência do *translateab9*, esta deve-se a uma estratégia inteligente de

armazenamento do código genético em uma tabela de espalhamento[23], a qual permite a tradução de sequências de modo consideravelmente rápido. Além disso, o programa C busca por padrões com distâncias mais específicas (Tabela 4), cujos intervalos foram estabelecidos a partir da análise de distâncias que envolveu não somente *germlines* como também sequências recombinadas, em contrapartida à versão Perl que usava distâncias baseadas na observação do perfil de *germlines*.

Tabela 4: Distâncias entre resíduos canônicos do domínio variável

Padrão	**Distâncias**	V_H	V_L
C-C	(min,max)	(70,150)	(70,130)
C-C	(min1,max1)	(70,78)	(62,74)
CDR3	(min2,max2)	(5,30)	(5,15)

As distâncias (min,max) são usadas pela versão Perl do programa de tradução. As demais distâncias são usadas pela versão C. Min-max: distâncias mínima e máxima entre o primeiro resíduo de cisteína e a sequência canônica W/FGXG. Min1-max1: distâncias mínima e máxima entre os dois resíduos de cisteína do domínio variavel. Min2-max2: tamanhos mínimo e máximo da CDR3.

Com relação ao *frequency_counter3.pl*, a primeira versão foi desenvolvida em linguagem C, chamada *counter2*, e usava como estrutura de dados um vetor[24] de listas encadeadas[25]. O programa recebia como entrada o arquivo em formato `fasta` contendo a biblioteca traduzida, calculava a frequência relativa de clones de acordo com o número de *susbtrings* iguais, e imprimia uma lista ordenada de clones em ordem decrescente de frequência relativa, em um arquivo de saída. Como pode ser visto na Figura 24, esta versão demonstrou ser consideravelmente incompatível com automatização de análise de dados NGS. A estratégia do programa consistia em ler uma sequência de aminoácidos, ler sua *substring* contendo CDRs, buscar no vetor de registros uma *substring* igual a atual, aumentar a frequência bruta da sequência caso encontrasse uma *substring* igual, ou inicializar um novo registro caso a *substring* não fosse encontrada.

[23]Tabela de símbolos em que cada símbolo é associado a uma chave, por meio de uma função de espalhamento. Assim, é possível ter acesso direto ao símbolo tendo apenas o valor da chave e a função de espalhamento usada para preencher a tabela.

[24]Vetor: consiste numa estrutura de dados que armazena elementos em posições consecutivas da memória, sendo seu acesso sequencial.

[25]Lista encadeada: conjunto de registros "ligados"ou "encadeados" entre si por apontadores. Um apontador, por sua vez, é uma variável que armazena um endereço de memória, neste caso, o endereço de um registro.

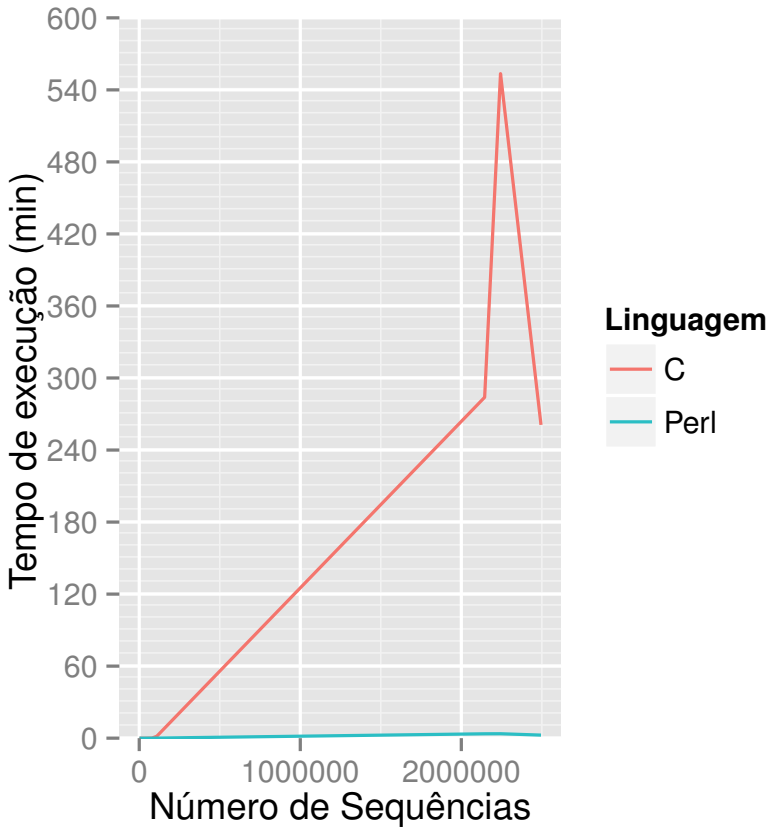

Figura 24: Tempos de execução do programa *frequency_counter3.pl* na versão Perl e *counter2* na versão C.

O acesso e ordenação do vetor tornam-se processos muito lentos, uma vez que seu tamanho é proporcional ao da biblioteca. Desse modo, foi desenvolvida uma versão em linguagem Perl, mas desta vez com um vetor de *hashes*[26]. Cada elemento do vetor armazena um *hash* com 10000 *hashes*. Então a busca de uma *substring* passa a ser mais rápida devido ao conjunto de busca ser menor, 10000 entradas por vez, e porque no *hash* a própria *substring* é usada como chave, então seu acesso é direto. A melhoria de estratégia pode ser notada pela brusca diminuição dos tempos de execução, os quais chegaram a atingir 9,22 horas na versão C, e caíram para no máximo 3,7 minutos na versão Perl (Figura 24). Nos testes do programa *counter2*, ocorreu a mesma situação dos testes da tradução: alguns conjuntos de dados menores que outros levaram mais tempo para serem processados em virtude da sobrecarga da memória cache.

[26] *Hash*: na linguagem Perl, corresponde uma estrutura que permite armazenar pares chave-valor de maneira não ordenada, em que a chave é uma *string*. Esta estrutura de dados já está previamente construída nas bibliotecas Perl (Cozens & Wainwright, 2000).

A eficiência dos programas *translateab9* e *frequency_counter3.pl* proporcionam a execução da análise completa com tempos curtos (Tabela 5). As quatro bibliotecas do conjunto Illumina S1, com tamanhos da ordem de 10^6, foram analisadas em aproximadamente 2 horas. Já as bibliotecas do conjunto 454 Roche, por serem menores, foram analisadas ainda mais rapidamente, em cerca de 4 minutos.

Tabela 5: Tempo de execução da análise completa

Conjunto de dados	**Biblioteca**	**Número de sequências***	**Tempo de execução (min)**
Illumina S1	V_H	9977325	47,6
Illumina S1	V_L	9863398	61,6
454 Roche	V_H	87284	1,3
454 Roche	V_L	252887	2,7

S*: soma total dos *reads* das bibliotecas inicial e final.

Na literatura, um estudo apresenta a análise de bibliotecas de *phage display* sequenciadas pela plataforma Illumina, que utiliza uma série de *scripts* MathLab (Matochko *et al.*, 2012). O processamento total da análise atinge de 6 a 8 horas, sem produzir resultados específicos sobre os clones candidatos. Outro trabalho, embora encontre candidatos baseados na frequência de clones (Ravn *et al.*, 2013), não foi automatizado e utiliza somente a frequência de clones como critério para detecção de candidatos e analisa apenas V_H. Dessa maneira, o presente método automatizado mostra-se compatível com a análise de bibliotecas NGS produzidas por *phage display*, não somente por sua capacidade em detectar clones candidatos usando os critérios de frequência de clones e assinatura de imunoglobulinas, como também pela eficiência em gerar diversos resultados sobre as bibliotecas de V_H e V_L.

3.5 Comparação entre BLAST e *translateab9*

A fim de avaliar os resultados obtidos pelo programa *translateab9*, as bibliotecas de sequências traduzidas foram comparadas aos arquivos de saída dos alinhamentos realizados pelo BLAST. Para tanto, foi desenvolvido um *script* Perl, *getidblast.pl*. Este *script* recebe como entrada a saída do BLAST e um arquivo contendo a lista de identificadores das sequências traduzidas pelo *translateab9*, e então calcula o número de sequências para as quais o BLAST encontrou *hits* de imunoglobulina, o número de sequências traduzidas, e o número de sequências identificadas em comum pelo BLAST e pelo *translateab9*.

A partir de tais valores, foram elaborados diagramas de Venn no intuito de comparar os conjuntos de sequências identificadas pelo BLAST e pelo *translateab9*. As bibliotecas de entrada para ambos os programas pertencem ao conjunto Illumina S1, o qual foi adotado para esta comparação por apresentar *reads* com maior qualidade (média de score PHRED acima de 20 para todas as bibliotecas), dentre os três conjuntos analisados neste trabalho. Ressalta-se que os alinhamentos usados para construir os diagramas de Venn possuem *e-value* abaixo de 10^{-20}. A escolha deste limite de *e-value* constitui uma tentativa de garantir a maior confiabilidade possível aos alinhamentos, dentro do intervalo de valores de *e-value* utilizados nas análises do BLAST sobre as bibliotecas NGS.

Por ser inadequado comparar os resultados do *translateab9* com as saídas do BLAST usando *e-values* diferentes, a comparação foi realizada usando o *e-value* de 10^{-20} para todas as bibliotecas.

Os diagramas de Venn referentes às bibliotecas de V_H denotam que tanto o BLAST quanto o *translateab9* identificaram uma quantidade maior de imunoglobulinas na bibliotecas finais (Figuras 25, 26, 27 e 28). O mesmo é demonstrado nos diagramas das bibliotecas de V_L. As Figuras 13 e 14 também corroboram tal observação sobre os alinhamentos do BLAST com *e-value* de 10^{-20}. Tais resultados concordam com o esperado para um experimento de *phage display* bem sucedido, pois à medida em que são realizados os ciclos de seleção de *phage display*, supõe-se que as sequências incapazes de se ligar ao antígeno de interesse sejam descartadas, e desse modo, espera-se que exista proporções gradativamente maiores de sequências de imunoglobulinas nas bibliotecas.

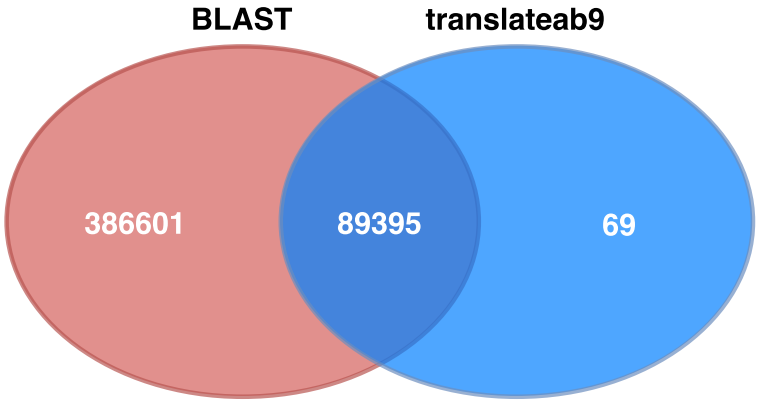

Figura 25: Diagrama de Venn representando o número de sequências identificadas como imunoglobulina da biblioteca inicial de V_H do conjunto Illumina S1.

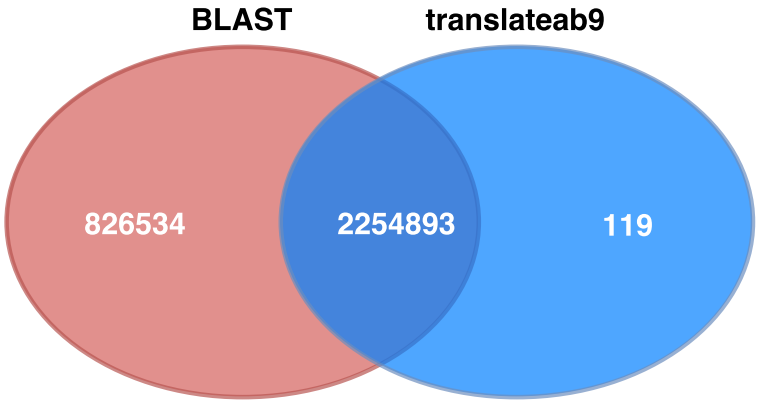

Figura 26: Diagrama de Venn representando o número de sequências identificadas como imunoglobulina da biblioteca final de V_H do conjunto Illumina S1.

Outro ponto a ser destacado compreende as intersecções entre o BLAST e o *translateab9*, isto é, o conjunto de sequências identificadas como imunoglobulina por ambos os programas. Nota-se que a intersecção entre os programas é maior nas bibliotecas finais de V_H e também de V_L. Tal discrepância origina-se pelo aumento da proporção de imunoglobulinas nas bibliotecas finais, como comentado acima, e também por questões de profundidade da amostragem. O sequenciamento é realizado a partir da amplificação por PCR dos genes de domínio variável isolados das bibliotecas de fagos. As partículas de fagos correspondentes a um dado clone podem constituir uma quantidade tão pequena que o clone não é amplificado na PCR, e portanto, não será visto no sequenciamento. Porém, este mesmo clone pode ser selecionado e amplificado ao longo dos ciclos de seleção de *phage display*. Então passará a ter uma quantidade de partículas suficiente para a amplificação anterior ao sequenciamento, e poderá ser de-

tectado na biblioteca sequenciada. Deste modo, existem clones detectados somente nas bibliotecas finais, em virtude da amostragem ser incapaz de alcançar toda a diversidade da biblioteca.

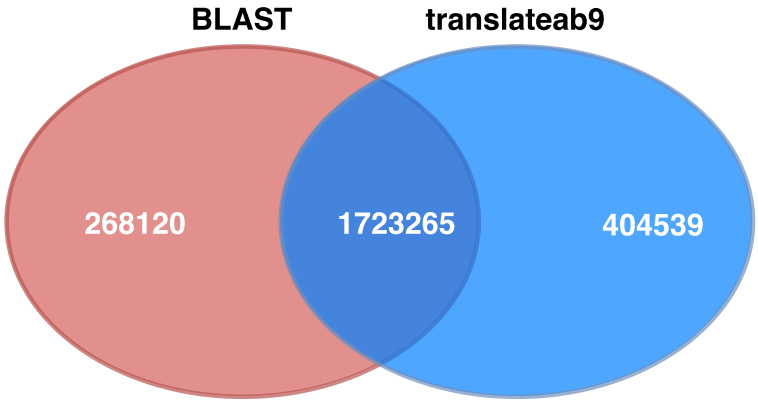

Figura 27: Diagrama de Venn representando o número de sequências identificadas como imunoglobulina da biblioteca inicial de V_L do conjunto Illumina S1.

Uma evidência disso é que o programa *find_duplicates7.pl* encontrou 838015 clones de V_H e 499676 clones de V_L presentes na biblioteca final e ausentes na biblioteca inicial. Tais valores são referentes a clones individuais, isto é, grupo de sequências que possuem uma dada *substring* em comum, e por conseguinte, o número de sequências é consideravelmente maior que o número de clones. Então, embora o número de clones da biblioteca final seja menor que o número de clones da biblioteca inicial, como resultado dos ciclos de seleção, o número de sequências de imunoglobulinas é maior na biblioteca final, devido à amplificação de uma parte dos clones.

A análise realizada pelo presente método permitiu constatar que em V_L ocorreu redução de 62% dos clones da biblioteca inicial para final, mas o número de sequências da biblioteca final, 2493387, é maior que o da biblioteca inicial, 2127804. Quanto à V_H a biblioteca inicial filtrada tem tamanho muito menor que a biblioteca final filtrada (511078 em comparação a 3203359 sequências), pois a maioria dos *reads* da biblioteca inicial possui menos de 300 pb, e portanto foram descartados na etapa de filtragem.

Com relação à eficácia do *translateab9*, este foi capaz de identificar um número de imunoglobulinas maior que o BLAST, para ambas as bibliotecas de V_L (Figuras 27 e 28). Para as bibliotecas de V_H o BLAST encontrou uma proporção maior de imunoglobulinas (Figuras 25 e 26). Todavia, dentre as sequências que o *translateab9* descartou das biblioteca inicial e final de V_H, 400039 e 770911, respectivamente, não continham dois

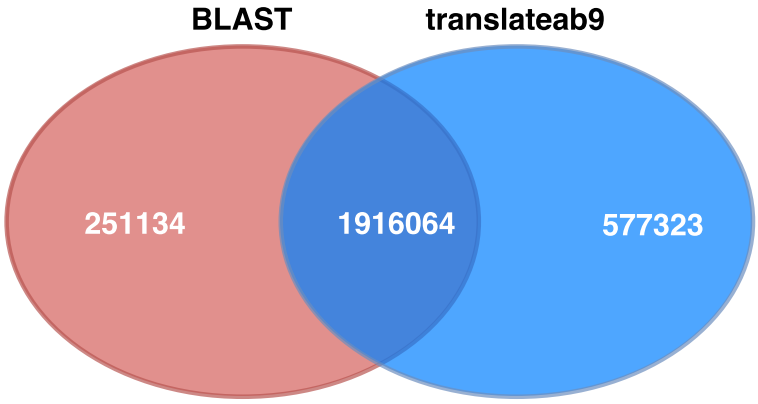

Figura 28: Diagrama de Venn representando o número de sequências identificadas como imunoglobulina da biblioteca final de V_L do conjunto Illumina S1.

resíduos de cisteína e/ou da CDR3. O restante das descartadas apresentava códons de parada em todas as fases de leitura. Dessa maneira, o BLAST identificou estas sequências como imunoglobulinas porque tinham similaridade com as *germlines*, sem garantir que as sequências fossem dotadas dos resíduos canônicos do domínio variável.

Além disso, o BLAST tem problemas para escolher a fase de leitura correta. Pode-se citar o caso de uma sequência que o *translateab9* encontrou a fase sem códons de parada, e que o BLAST escolheu uma fase com códons de parada, pois tinha *score* de similaridade maior com as *germlines* (Anexo C). Logo, o BLAST escolhe a fase de leitura de acordo com a similaridade calculada, já o *translateab9* escolhe a fase de acordo com o que ocorre no processo biológico, isto é, tradução da sequência mais longa sem códon de parada.

Com relação ao desempenho, o *translateab9* apresenta tempos de execução mais compatíveis com a automatização da análise de bibliotecas NGS (Figura 29), não chegando nem mesmo a 10 minutos, em contrapartida ao BLAST, que pode levar até mais de 10 horas para analisar bibliotecas da ordem de 10^6. Assim, o *translateab9* não somente é capaz de aplicar o primeiro critério de escolha de candidatos, e garantir que sejam escolhidas sequências candidatas dentre um conjunto que possua assinatura de anticorpo, como também apresenta tempos de execução consideravelmente menores que o BLAST.

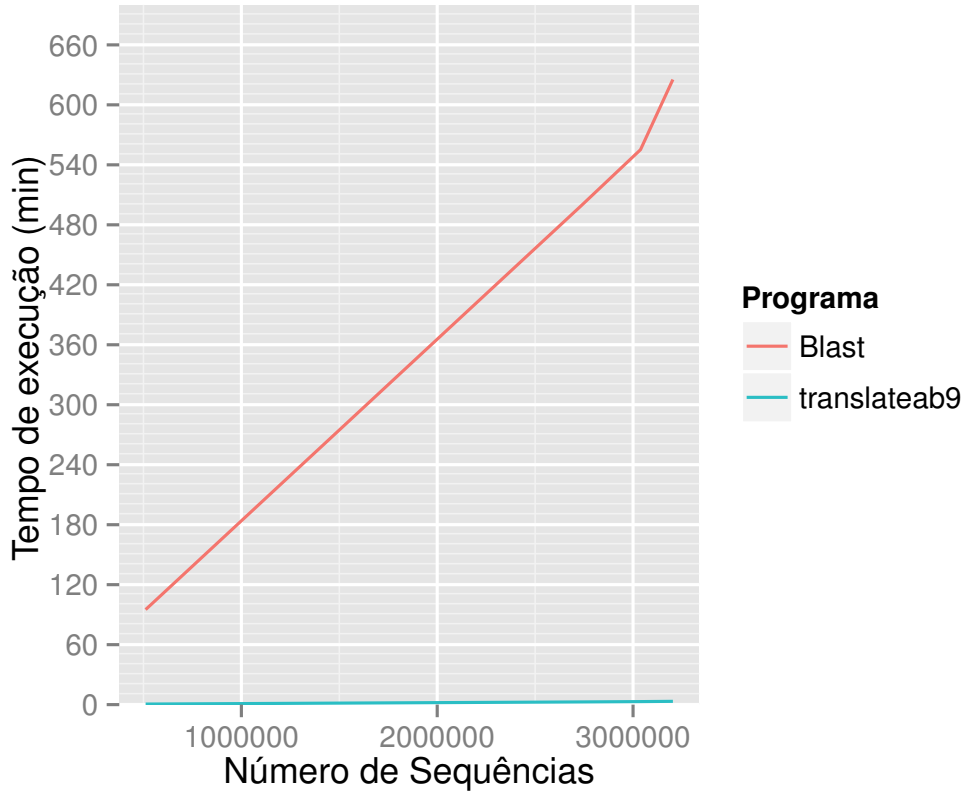

Figura 29: Comparação entre os tempos de execução do BLAST e do *translateab9*. Foram usadas as bibliotecas iniciais e finais de V_H e V_L do conjunto Illumina S1. O valor de *e-value* usado para o BLAST foi de 10^{-5}.

3.6 Diversidade das bibliotecas

No sentido de entender as mudanças na diversidade das bibliotecas, foi calculado o índice de Shannon para todas as bibliotecas dos conjuntos 454 Roche e Illumina S1. O número total de sequências de cada biblioteca pode ser visto na Tabela 6.

Tabela 6: Total de sequências das bibliotecas de *phage display*

Conjunto	**Biblioteca Inicial**	**Biblioteca Final**
VH Illumina	3006172	3230499
VL Illumina	2979342	3057825
VH 454 Roche	48595	38689
VL 454 Roche	111595	141407

No que diz respeito à diversidade das bibliotecas V_H do conjunto Illumina S1, os índices de Shannon permitem notar redução da incerteza sobre os clones na biblioteca final, para todos os valores de identidade usados nas execuções do CD_HIT (Tabela 7). Os valores de identidade de 85% e 95% destacaram-se por apresentar as maiores reduções dos índices de Shannon entre as bibliotecas inicial e final de V_H.

Tabela 7: Análise de diversidade de V_H do conjunto Illumina S1

Identidade(%)	$\boldsymbol{H_{R_0}}$	$\boldsymbol{H_{Rs}}$	**Redução(%)**
80	1,47	0,40	72,79
85	3,21	0,66	79,44
90	6,61	1,35	79,58
95	12,58	3,00	76,15
100	20,71	18,44	10,96

H: índice de Shannon. R_0: biblioteca inicial. Rs: biblioteca final.

Quanto às bibliotecas V_L do conjunto Illumina S1, estas também apresentaram redução de diversidade na biblioteca final, para todos os valores de identidade (Tabela 8). A maior redução do índice de Shannon pode observada para identidade de 80%.

Numa comparação mais ampla, se considerarmos os índices de Shannon como estimativa aproximada da diversidade, as reduções de entropia seriam proporcionais as

reduções de diversidade da biblioteca inicial para final, tanto de V_H quanto de V_L, corroborando o pressuposto de que a seleção de clones no experimento de *phage display* de fato ocorreu e foi bem sucedida. Os valores de entropia indicam que as bibliotecas V_H possuem maior incerteza sobre clones que as bibliotecas V_L.

Tabela 8: Análise de diversidade de V_L do conjunto Illumina S1

Identidade(%)	$\boldsymbol{H_{R_0}}$	$\boldsymbol{H_{Rs}}$	**Redução(%)**
80	0,51	0,085	83,33
85	0,80	0,16	80
90	1,7	0,44	74,12
95	4,63	1,35	70,84
100	19,11	15,86	17,01

H: índice de Shannon. R_0: biblioteca inicial. Rs: biblioteca final.

Com relação ao conjunto 454 Roche, observa-se redução da diversidade das bibliotecas V_H para todos os valores de identidade (Tabela 9). Ocoreu maior redução do índice de Shannon para o valor de identidade de 90%. Pode-se supor que as bibliotecas V_H foram selecionadas de maneira bem sucedida, considerando que houve redução de entropia e provavelmente, de diversidade para todos os valores de identidade.

Tabela 9: Análise de diversidade de V_H do conjunto 454 Roche

Identidade(%)	$\boldsymbol{H_{R_0}}$	$\boldsymbol{H_{Rs}}$	**Redução(%)**
80	0,86	0,14	83,72
85	2,38	0,25	89,5
90	5,47	0,39	92,87
95	10,66	0,86	91,93
100	15,41	5,6	63,66

H: índice de Shannon. R_0: biblioteca inicial. Rs: biblioteca final.

Finalmente, a análise de diversidade das bibliotecas V_L do conjunto 454 Roche gerou índices de Shannon mostrando que houve redução para todos os valores de identidade testados, assim como nas demais bibliotecas mencionadas (Tabela 10). A identidade de 95% apresentou maior redução do índice de Shannon. Diante dos problemas referentes à biblioteca final de V_L do conjunto 454 Roche, discutidos na seção 3.2, seria tendenciosa a comparação da diversidade desta biblioteca com as descritas acima, e devido a isso,

tais comparações não serão inferidas na presente seção.

Tabela 10: Análise de diversidade de V_L do conjunto 454 Roche

Identidade(%)	$\boldsymbol{H_{R_0}}$	$\boldsymbol{H_{Rs}}$	**Redução(%)**
80	0,37	0,044	88,11
85	0,80	0,071	91,13
90	1,98	0,14	92,936
95	4,89	0,33	93,25
100	14,25	3,54	75,16

H: índice de Shannon. R_0: biblioteca inicial. Rs: biblioteca final.

Diante do exposto, a entropia de Shannon demostrou ser uma medida de diversidade adequada para as bibliotecas analisadas, embora provavelmente sejam necessários outros tipos de testes estatísticos e dados mais completos sobre a eficiência da amplificação prévia ao sequenciamento, a fim de mitigar interpretações com viés de amostragem.

4 Considerações Finais

O presente trabalho apresenta um método *in silico* para detecção de sequências de imunoglobulinas selecionadas por tecnologia de *phage display*. Os critérios escolhidos para análise foram eficazes em detectar clones candidatos, pois a cada etapa do método o número de sequências é reduzido até que seja gerada uma lista das sequências mais frequentes, dotadas de assinatura de domínio variável de imunoglobulina.

Até então a literatura não tem registros de um método automatizado para encontrar clones selecionados por *phage display*, a partir de bibliotecas NGS. Além da automatização, esta abordagem tem como contribuições a eficiência, exigindo pouco tempo para obter diversos resultados sobre as bibliotecas de V_H e de V_L, bem como o uso de um critério biológico de análise que garante que as sequências candidatas de fato tenham sido reconhecidas como imunoglobulinas.

Apesar de promissor, o método apresenta duas limitações. A primeira diz respeito ao tipo de sequência analisada, cujas marcas podem ser identificadas de maneira eficaz se forem de origem humana. As distâncias estabelecidas entre resíduos canônicos de regiões do domínio variável são baseadas em sequências humanas. No entanto, existem sequências artificais de camelo e tubarão, e originais de galinha que apresentam cisteínas não usuais na CDR3 (Wu *et al.*, 2012). O programa *translateab9* muito provavelmente identificaria de maneira incorreta clones formados por sequências deste tipo, visto que a busca por expressão regular não considera a existência de cisteínas não usuais, já que sua frequência é consideravelmente baixa em humanos (aproximadamente 1.6%) (Wu *et al.*, 2012), e portanto, não são típicas de sequências humanas.

A segunda restrição corresponde ao fato de que a abordagem é pouco sensível a variações de sequências de aminoácidos. O programa *frequency_counter3.pl* recebe como entrada sequências de aminoácidos, e considera sequências como pertencentes a um mesmo clone caso possuam subsequências exatamente iguais. A subsequência abrange as regiões CDR1, FR2, CDR2, FR3 e CDR3, e por conseguinte, a identificação de clones permite diferenças entre as sequências somente nas regiões FR1 e FR4. Como consequência, a análise pode separar clones que na verdade são um clone só. No entanto, a identificação de clones baseada nesta subsequência apresenta a vantagem de permitir a análise da maioria das regiões que compõem o domínio variável, de V_H e de V_L, não

limitando-se a CDR3 de V_H, como tem sido descrito na literatura (Glanville *et al.*, 2009; Ravn *et al.*, 2010; Ravn *et al.*, 2013).

Com relação à escolha em utilizar busca exata e não alinhamentos, tal abordagem justifica-se pela redução do tempo de execução. Como comentado na subseção 3.5, o BLAST, considerado um dos programas mais rápidos de alinhamento, pode levar até mais de 10 horas para processar bibliotecas NGS, enquanto a análise completa de todas as bibliotecas pelo presente método não chega nem mesmo a 3 horas de processamento.

Além disso, um programa que execute somente alinhamento não garante a aplicação do critério de assinatura de anticorpo, como faz o *translateab9*. Embora alinhamentos lidem melhor com variações de sequências, tornariam difícil ou talvez inviável assegurar o reconhecimento de domínios variáveis, e assim, também implicariam em restrições de análise. Diante das limitações em ambas as estratégias, preferiu-se o desenvolvimento de um método rápido que, embora apresente pouca sensibilidade à variação de sequências, forneça resultados passíveis de serem analisados mais profundamente caso necessário.

5 Propostas Futuras

A fim de compartilhar o método com a comunidade acadêmica, o pacote de programas desenvolvido neste trabalho será disponibilizado para *download* gratuito, juntamente com um manual. Embora o método atualmente possua interface via linha de comando intuitiva, que permite gerar o arquivo de configuração, e executar o *script* de automatização, algumas melhorias poderiam ser implementadas posteriomente.

No intuito de facilitar a criação do arquivo de configuração e tornar mais agradável a experiência do usuário, pretende-se desenvolver uma arquivo *html* com função de formulário, no qual o usuário poderá escolher diretórios e arquivos por meio de interface gráfica. Uma vez criado o arquivo de configuração por meio do formulário *html*, o usuário poderá executar o método apenas indicando no terminal o caminho onde se encontra o arquivo de configuração.

Outro aspecto relevante diz respeito ao escopo de sequências para os quais o método é eficaz. Até então o método analisa somente sequências humanas, no entanto, tem-se como proposta futura incluir no programa *translateab9* expressões regulares que permitam identificar sequências de outras espécies e/ou artificiais, dotadas de resíduos de cisteína não usuais.

Como discutido anteriormente, a abordagem possui limitações quanto à sensibilidade a variações de resíduos de aminoácidos e, nesse sentido, seria interessante associar alguma medida de confiabilidade de identificação de clones, de modo que o usuário possa ter uma estimativa do quão confiável é o agrupamento das bibliotecas em clones.

Finalmente, espera-se desenvolver futuramente uma análise de diversidade mais completa, que envolva a classificação de *germlines* das bibliotecas inteiras e não somente das sequências candidatas. Dessa maneira seria possível produzir resultados sobre a distribuição do uso de *germlines* nas bibliotecas de *phage display*, prática bastante frequente nos estudos de diversidade de repertórios de imunoglobulinas.

Referências

Abhinandan, K.; Martin, A. C. 2008. Analysis and improvements to kabat and structurally correct numbering of antibody variable domains. *Molecular immunology*, Elsevier, v. 45, n. 14, p. 3832–3839.

Al-Lazikani, B.; Lesk, A. M.; Chothia, C. 1997. Standard conformations for the canonical structures of immunoglobulins. *Journal of molecular biology*, Elsevier, v. 273, n. 4, p. 927–948.

Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. 1990. Basic local alignment search tool. *Journal of molecular biology*, Elsevier, v. 215, n. 3, p. 403–410.

Andrews, S. 2012. *FastQC Project*. Disponível em: ⟨http://www.bioinformatics.babraham.ac.uk/projects/fastqc/⟩.

Aronesty, E. 2011. *ea-utils: Command-line tools for processing biological sequencing data*. ⟨http://code.google.com/p/ea-utils/⟩.

Aronesty, E. 2013. *Comparison of sequencing utility programs. Open Bioinform. J. 7: 1–8*.

Barbas, C. F. I.; Burton, D. R.; Scott, J. K.; Silverman, G. J. 2001. *Phage Display: A Laboratory Manual*. 1. ed. Cold Spring Harbor, New York: Cold Spring Harbor Laboratory Press.

Biosystems, A. 2009. *DNA Sequencing by Capillary Eletrophoresis*. Disponível em:⟨https://www3.appliedbiosystems.com/cms/groups/mcb_support/documents/generaldocuments/cms_041003.pdf⟩.

Blachman, N. 1968. A mathematical theory of communication. *IEEE Transactions on Information Theory*, v. 14, p. 27–31.

Branden, C.; Tooze, J. 1999. *Introduction to Protein Structure*. 2. ed. New York: Garland Publishing.

Brezski, R. J.; Jordan, R. E. 2010. Cleavage of iggs by proteases associated with invasive diseases: an evasion tactic against host immunity? In: TAYLOR & FRANCIS. *MAbs*. [S.l.], v. 2, n. 3, p. 212–220.

Christiansen, A.; Kringelum, J. V.; Hansen, C. S.; Bøgh, K. L.; Sullivan, E.; Patel, J.; Rigby, N. M.; Eiwegger, T.; Szépfalusi, Z.; Masi, F. D. *et al.* 2015. High-throughput sequencing enhanced phage display enables the identification of patient-specific epitope motifs in serum. *Scientific reports*, Nature Publishing Group, v. 5.

Christiansen, A.; Kringelum, J. V.; Hansen, C. S.; Bøgh, K. L.; Sullivan, E.; Patel, J.; Rigby, N. M.; Eiwegger, T.; Szépfalusi, Z.; Masi, F. D. *et al.* 2015. High-throughput sequencing enhanced phage display enables the identification of patient-specific epitope motifs in serum. *Scientific reports*, Nature Publishing Group, v. 5.

Coloma, M.; Clift, A.; Wims, L.; Morrison, S. L. 2000. The role of carbohydrate in the assembly and function of polymeric igg. *Molecular Immunology*, v. 37, n. 18, p. 1081 – 1090. ISSN 0161-5890. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S0161589001000244⟩.

Cozens, S.; Wainwright, P. 2000. *Beginning Perl*. [S.l.]: Wrox Press.

Dantas-barbosa, C.; Brigido, M. de M.; Maranhao, A. Q. 2012. Antibody phage display libraries: contributions to oncology. *International journal of molecular sciences*, Molecular Diversity Preservation International, v. 13, n. 5, p. 5420–5440.

Dias-neto, E.; Nunes, D. N.; Giordano, R. J.; Sun, J.; Botz, G. H.; Yang, K.; Setubal, J. C.; Pasqualini, R.; Arap, W. 2009. Next-generation phage display: integrating and comparing available molecular tools to enable cost-effective high-throughput analysis. *PloS one*, Public Library of Science, v. 4, n. 12, p. e8338.

Dias-Neto, E.; Nunes, D. N.; Giordano, R. J.; Sun, J.; Botz, G. H.; Yang, K.; Setubal, J. C.; Pasqualini, R.; Arap, W. 2009. Next-generation phage display: integrating and comparing available molecular tools to enable cost-effective high-throughput analysis. *PloS one*, Public Library of Science, v. 4, n. 12, p. e8338.

Doria-Rose, N. A.; Schramm, C. A.; Gorman, J.; Moore, P. L.; Bhiman, J. N.; Dekosky, B. J.; Ernandes, M. J.; Georgiev, I. S.; Kim, H. J.; Pancera, M. *et al.* 2014. Developmental pathway for potent v1v2-directed hiv-neutralizing antibodies. *Nature*, Nature Publishing Group, v. 509, n. 7498, p. 55–62.

Ecker, D. M.; Jones, S. D.; Levine, H. L. 2015. The therapeutic monoclonal antibody market. In: TAYLOR & FRANCIS. *MAbs*. [S.l.], v. 7, n. 1, p. 9–14.

Eisen, H. N. 2014. Affinity enhancement of antibodies: how low-affinity antibodies produced early in immune responses are followed by high-affinity antibodies later and in memory b-cell responses. *Cancer immunology research*, AACR, v. 2, n. 5, p. 381–392.

Elgert, K. D. 1998. *Immunology: Understanding the Immune System*. 1. ed. [S.l.]: John Wiley & Sons.

Ewing, B.; Hillier, L.; Wendl, M. C.; Green, P. 1998. Base-calling of automated sequencer traces usingphred. i. accuracy assessment. *Genome research*, Cold Spring Harbor Lab, v. 8, n. 3, p. 175–185.

Glanville, J.; Zhai, W.; Berka, J.; Telman, D.; Huerta, G.; Mehta, G. R.; Ni, I.; Mei, L.; Sundar, P. D.; Day, G. M. *et al.* 2009. Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 106, n. 48, p. 20216–20221.

Harmsen, M. M.; Ruuls, R. C.; Nijman, I. J.; Niewold, T. A.; Frenken, L. G.; Geus, B. de. 2000. Llama heavy-chain V regions consist of at least four distinct subfamilies revealing novel sequence features. *Molecular immunology*, Elsevier, v. 37, n. 10, p. 579–590.

Hert, D. G.; Fredlake, C. P.; Barron, A. E. 2008. Advantages and limitations of next-generation sequencing technologies: A comparison of electrophoresis and non-electrophoresis methods. *Electrophoresis*, Wiley Online Library, v. 29, n. 23, p. 4618–4626.

Holm, L.; Sander, C. 1998. Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, Oxford Univ Press, v. 14, n. 5, p. 423–429.

Illumina. 2011. *Quality Scores for Next-Generation Sequencing*. Disponível em: ⟨http://www.illumina.com/documents/products/technotes/technote_Q-Scores.pdf⟩.

Jost, L. 2006. Entropy and diversity. *Oikos*, Wiley Online Library, v. 113, n. 2, p. 363–375.

Jung, S.; Spinelli, S.; Schimmele, B.; Honegger, A.; Pugliese, L.; Cambillau, C.; Plückthun, A. 2001. The importance of framework residues H6, H7 and H10 in antibody heavy chains: experimental evidence for a new structure subclassification of antibody VH domains. *Journal of Molecular Biology*, v. 309, p. 701–716.

Kabat, E. A.; Wu, T. T.; Perry, H. M.; Gottesman, K. S.; Foeller, C. 1992. *Sequences of proteins of immunological interest*. [S.l.]: DIANE publishing.

Kay, B. K.; Winter, J.; Mccafferty, J. 1996. *Phage display of peptides and proteins: a laboratory manual*. [S.l.]: Academic Press.

Kircher, M.; Kelso, J. 2010. High-throughput dna sequencing–concepts and limitations. *Bioessays*, Wiley Online Library, v. 32, n. 6, p. 524–536.

Köler, G.; Milstein, C. 1975. Continuous culture of fused cells secreting antibody of predefined specificity. *Nature*, v. 256, n. 5517, p. 495–497.

Lefranc, M.-P.; Giudicelli, V.; Ginestoux, C.; Jabado-michaloud, J.; Folch, G.; Bellahcene, F.; Wu, Y.; Gemrot, E.; Brochet, X.; Lane, J. *et al.* 2009. Imgt®, the international immunogenetics information system®. *Nucleic acids research*, Oxford Univ Press, v. 37, n. suppl 1, p. D1006–D1012.

Li, W. 2015. *CD-HIT Users's Guide*. Disponível em: http://weizhongli-lab.org/lab-wiki/doku.php?id=cd-hit-user-guide.

Li, W.; Godzik, A. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, Oxford Univ Press, v. 22, n. 13, p. 1658–1659.

Li, W.; Jaroszewski, L.; Godzik, A. 2001. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, Oxford Univ Press, v. 17, n. 3, p. 282–283.

Madigan, M. T.; Martinko, J. M.; Dunlap, P. V.; Clark, D. P. 2009. *Microbiologia de brock*. [S.l.]: Artmed Editora.

Magurran, A. E. 2013. *Measuring biological diversity*. [S.l.]: John Wiley & Sons.

Maranhao, A.; Brigido, M. 2000. Expression of anti-z-dna single chain antibody variable fragment on the filamentous phage surface. *Brazilian Journal of Medical and Biological Research*, SciELO Brasil, v. 33, n. 5, p. 569–579.

Maranhão, A. Q.; Costa, M. B. W.; Guedes, L.; Moraes-vieira, P. M.; Raiol, T.; Brigido, M. M. 2013. A mouse variable gene fragment binds to dna independently of the bcr context: a possible role for immature b-cell repertoire establishment. *PloS one*, Public Library of Science, v. 8, n. 9, p. e72625.

Maranhão, A. Q.; Brígido, M. de M. 2000. Expression of anti-Z-DNA single chain antibody variable fragment on the filamentous phage surface. *Brazilian Journal of Medical and Biological Research*, v. 33, n. 5, p. 569–579.

Marchalonis, J. J.; Bernstein, R. M.; Shen, S. X.; Schluter, S. F. 1996. Emergence of immunoglobulin family: conservation in protein sequence and plasticity in gene organization. *Glicobiology*, v. 6, p. 657–663.

Mardis, E. R. 2013. Next-generation sequencing platforms. *Annual review of analytical chemistry*, Annual Reviews, v. 6, p. 287–303.

Martin, M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, v. 17, n. 1, p. 10–12. Disponível em: ⟨http://journal.embnet.org/index.php/embnetjournal/article/view/200⟩.

Masuda, K.; Sakamoto, K.; Kojima, M.; Aburatani, T.; Ueda, T.; Ueda, H. 2006. The role of interface framework residues in determining antibody vh/vl interaction strength and antigen-binding affinity. *FEBS Journal*, Wiley Online Library, v. 273, n. 10, p. 2184–2194.

Matochko, W. L.; Chu, K.; Jin, B.; Lee, S. W.; Whitesides, G. M.; Derda, R. 2012. Deep sequencing analysis of phage libraries using Illumina platform. *Methods*, Elsevier, v. 58, n. 1, p. 47–55.

Mayer, A. L.; Donovan, R. P.; Pawlowski, C. W. 2014. Information and entropy theory for the sustainability of coupled human and natural systems. *Ecology and Society*, v. 19, n. 3, p. 11.

Metzker, M. L. 2010. Sequencing technologies-the next generation. *Nature reviews genetics*, Nature Publishing Group, v. 11, n. 1, p. 31–46.

Myllykangas, S.; Buenrostro, J.; Ji, H. P. 2012. Overview of sequencing technology platforms. In: *Bioinformatics for high throughput sequencing*. [S.l.]: Springer. p. 11–25.

Naylor, M.; Capra, J. D. 1999. Mutational status of ig vh genes provides clinically valuable information in b-cell chronic lymphocytic leukemia. *Blood*, Am Soc Hematology, v. 94, n. 6, p. 1837–1839.

Owen, J. A.; Punt, J.; Stranford, S. A.; Jones, P. 2013. *Kuby Immunology*. 7. ed. New York: W. H. Freeman and Company.

Porter, R. 1958. Separation and isolation of fractions of rabbit gamma-globulin containing the antibody and antigenic combining sites. Nature Publishing Group.

Prabakaran, P.; Streaker, E.; Chen, W.; Dimitrov, D. S. 2011. 454 antibody sequencing-error characterization and correction. *BMC research notes*, BioMed Central Ltd, v. 4, n. 1, p. 404.

R Core Team. 2015. *R: A Language and Environment for Statistical Computing*. Vienna, Austria. Disponível em: ⟨https://www.R-project.org⟩.

Raghavan, A. K. *Sequence and structural analysis of antibodies*. Tese (Doutorado) — UCL (University College London), 2009.

Ravn, U.; Didelot, G.; Venet, S.; Ng, K.-T.; Gueneau, F.; Rousseau, F.; Calloud, S.; Kosco-vilbois, M.; Fischer, N. 2013. Deep sequencing of phage display libraries to support antibody discovery. *Methods*, Elsevier, v. 60, n. 1, p. 99–110.

Ravn, U.; Gueneau, F.; Baerlocher, L.; Osteras, M.; Desmurs, M.; Malinge, P.; Magistrelli, G.; Farinelli, L.; Kosco-vilbois, M.; Fischer, N. 2010. By-passing in vitro screening next generation sequencing technologies applied to antibody display and in silico candidate selection. *Nucleic acids research*, Oxford Univ Press, v. 38, n. 21, p. e193–e193.

Rice, P.; Longden, I.; Bleasby, A. *et al.* 2000. Emboss: the european molecular biology open software suite. *Trends in genetics*, [Amsterdam, The Netherlands: Elsevier Science Publishers (Biomedical Division)], c1985-, v. 16, n. 6, p. 276–277.

Rizzi, E.; Lari, M.; Gigli, E.; Bellis, G. D.; Caramelli, D. 2012. Ancient dna studies: new perspectives on old samples. *Genet Sel Evol*, v. 44, p. 21. Material Suplementar.

Sanger, F.; Nicklen, S.; Coulson, A. R. 1977. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 74, n. 12, p. 5463–5467.

Schmieder, R.; Edwards, R. 2011. Quality control and preprocessing of metagenomic datasets. *Bioinformatics (Oxford, England)*, v. 27, n. 6, p. 863–864. ISSN 1367-4811. PMID: 21278185. Disponível em: ⟨http://www.ncbi.nlm.nih.gov/pubmed/21278185⟩.

Sciences, . L. 2012. *How genome sequencing is done ?* Disponível em: ⟨http://www.454.com/downloads/news-events/how-genome-sequencing-is-done_FINAL.pdf⟩.

Scientific, T. 2015. *Single Stranded Templates for PyroSequencing*. Disponível em: ⟨https://www.thermofisher.com/br/en/home/life-science/dna-rna-purification-analysis/napamisc/capture-of-biotinylated-targets/single-stranded-templates-for-pyrosequencing.html#fig3⟩.

Setubal, J. C.; Meidanis, J.; Setubal-meidanis. 1997. *Introduction to computational molecular biology*. [S.l.]: PWS Pub.

Sievers, F.; Wilm, A.; Dineen, D.; Gibson, T. J.; Karplus, K.; Li, W.; Lopez, R.; Mcwilliam, H.; Remmert, M.; Söding, J. *et al.* 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, Wiley Online Library, v. 7, n. 1.

Sompayrac, L. 2012. *How the Immune System Works*. 4. ed. Chichester: Willey-Blackwell: John Wiley & Sons.

Squizzato, S.; Park, Y. M.; Buso, N.; Gur, T.; Cowley, A.; Li, W.; Uludag, M.; Pundir, S.; Cham, J. A.; Mcwilliam, H. *et al.* 2015. The ebi search engine: providing search and retrieval functionality for biological data from embl-ebi. *Nucleic acids research*, Oxford Univ Press, p. gkv316.

Stanfield, R. L.; Dooley, H.; Flajnik, M. F.; Wilson, I. A. 2004. Crystal structure of a shark single-domain antibody v region in complex with lysozyme. *Science*, American Association for the Advancement of Science, v. 305, n. 5691, p. 1770–1773.

Tramontano, A.; Chotia, C.; Lesk, A. M. 1990. Framework residue 71 is a major determinant of the position and conformation of the second hypervariable region in the VH domains of immunoglobulins. *Journal of Molecular Biology*, v. 215, p. 175–182.

Walsh, G. 2007. *Pharmaceutical Biotechnology: concepts and applications*. 1. ed. Chichester: John Wiley & Sons.

Wang, L.-F.; Yu, M. 2004. Epitope identification and discovery using phage display libraries: applications in vaccine development and diagnostics. *Current drug targets*, Bentham Science Publishers, v. 5, n. 1, p. 1–15.

Wang, Y.; Jackson, K. J.; Sewell, W. A.; Collins, A. M. 2008. Many human immunoglobulin heavy-chain ighv gene polymorphisms have been reported in error. *Immunology and cell biology*, Nature Publishing Group, v. 86, n. 2, p. 111–115.

Willats, W. G. T. 2002. Phage display: practicalities and prospects. *Plant Molecular Biology*, v. 50, n. 6, p. 837–854.

Williams, A. F.; Barclay, A. N. 1988. The immunoglobulin superfamily-domains for cell surface recognition. *Annual Reviews Immunology*, v. 6, p. 381–405.

Wu, L.; Oficjalska, K.; Lambert, M.; Fennell, B. J.; Darmanin-sheehan, A.; Shúilleabháin, D. N.; Autin, B.; Cummins, E.; Tchistiakova, L.; Bloom, L. *et al.* 2012. Fundamental characteristics of the immunoglobulin VH repertoire of chickens in comparison with those of humans, mice, and camelids. *The Journal of Immunology*, Am Assoc Immnol, v. 188, n. 1, p. 322–333.

Wu, T. T.; Kabat, E. A. 1970. An analysis of the sequences of the variable regions of Bence Jones proteins and myeloma light chains and their implications for antibody complementarity. *The Journal of experimental medicine*, v. 132, n. 2, p. 211–250.

Ye, J.; Ma, N.; Madden, T. L.; Ostell, J. M. 2013. Igblast: an immunoglobulin variable domain sequence analysis tool. *Nucleic acids research*, Oxford Univ Press, p. gkt382.

ANEXO A

Matriz de identidade referente ao alinhamento das sequências candidatas de V_H do conjunto 454 Roche

1: IY4FGGO08JMUKM|FOLD-CHANGE_129.3723	100.00	98.89	45.22	45.22	44.90	45.22	44.59
2: IY4FGGO08JOYJN|FOLD-CHANGE_82.8988	98.89	100.00	45.08	45.22	44.90	45.22	44.44
3: IY4FGGO08JR57V|FOLD-CHANGE_33335.3485	45.22	45.08	100.00	96.41	96.41	96.41	96.69
4: IY4FGGO08JI6F1|FOLD-CHANGE_33.9131	45.22	45.22	96.41	100.00	99.45	99.45	98.62
5: IY4FGGO08I5WQ6|FOLD-CHANGE_30.1450	44.90	44.90	96.41	99.45	100.00	99.45	98.62
6: IY4FGGO08JSGPB|FOLD-CHANGE_22.6088	45.22	45.22	96.41	99.45	99.45	100.00	98.62
7: IY4FGGO08I6IQ8|FOLD-CHANGE_20.0967	44.59	44.44	96.69	98.62	98.62	98.62	100.00

ANEXO B

Gráfico de proporção de *reads* com tamanho adequado

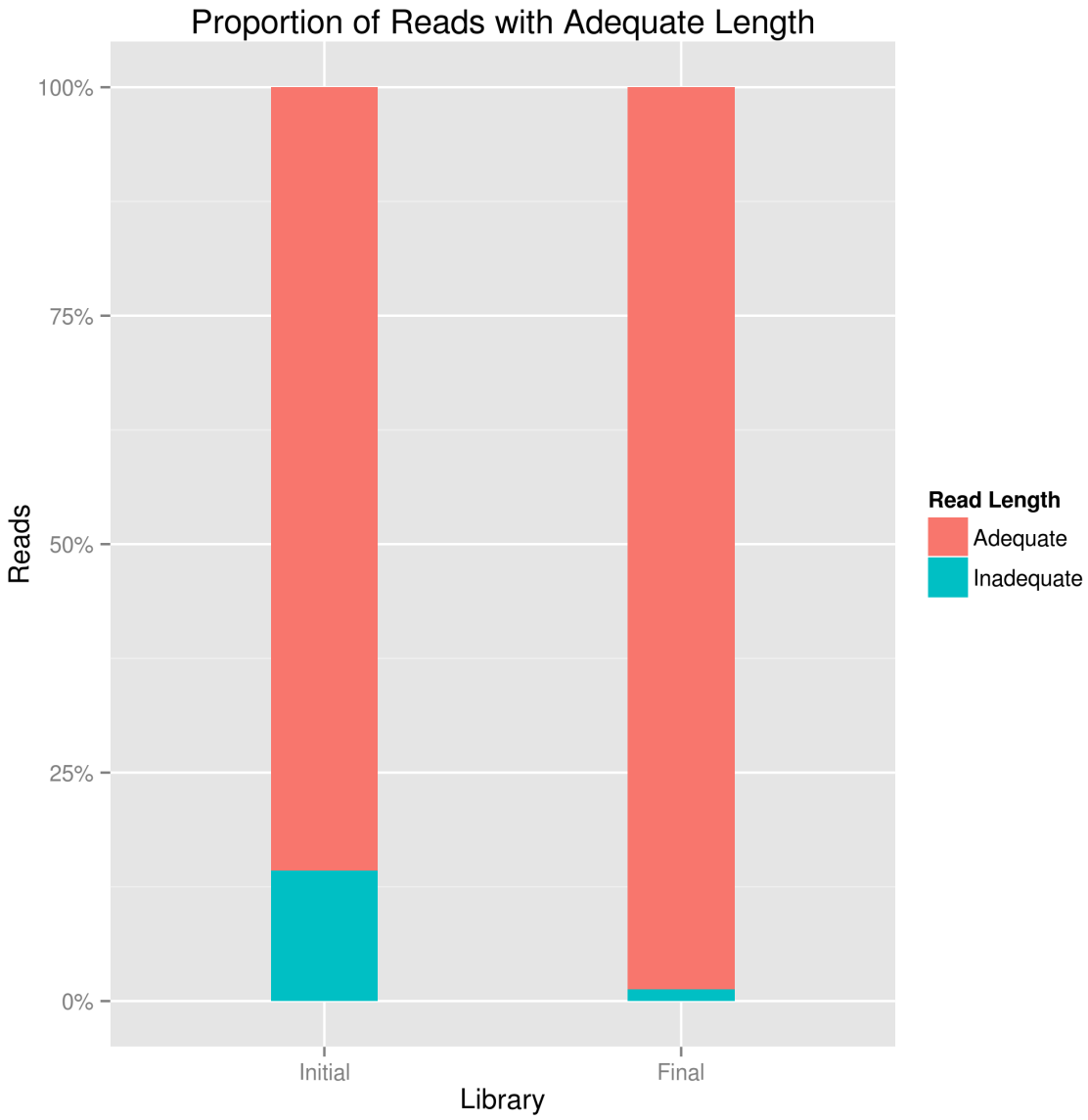

Figura 30: *Reads* das bibliotecas V_L do conjunto Illumina S1.

Gráfico de número de *reads* por etapa

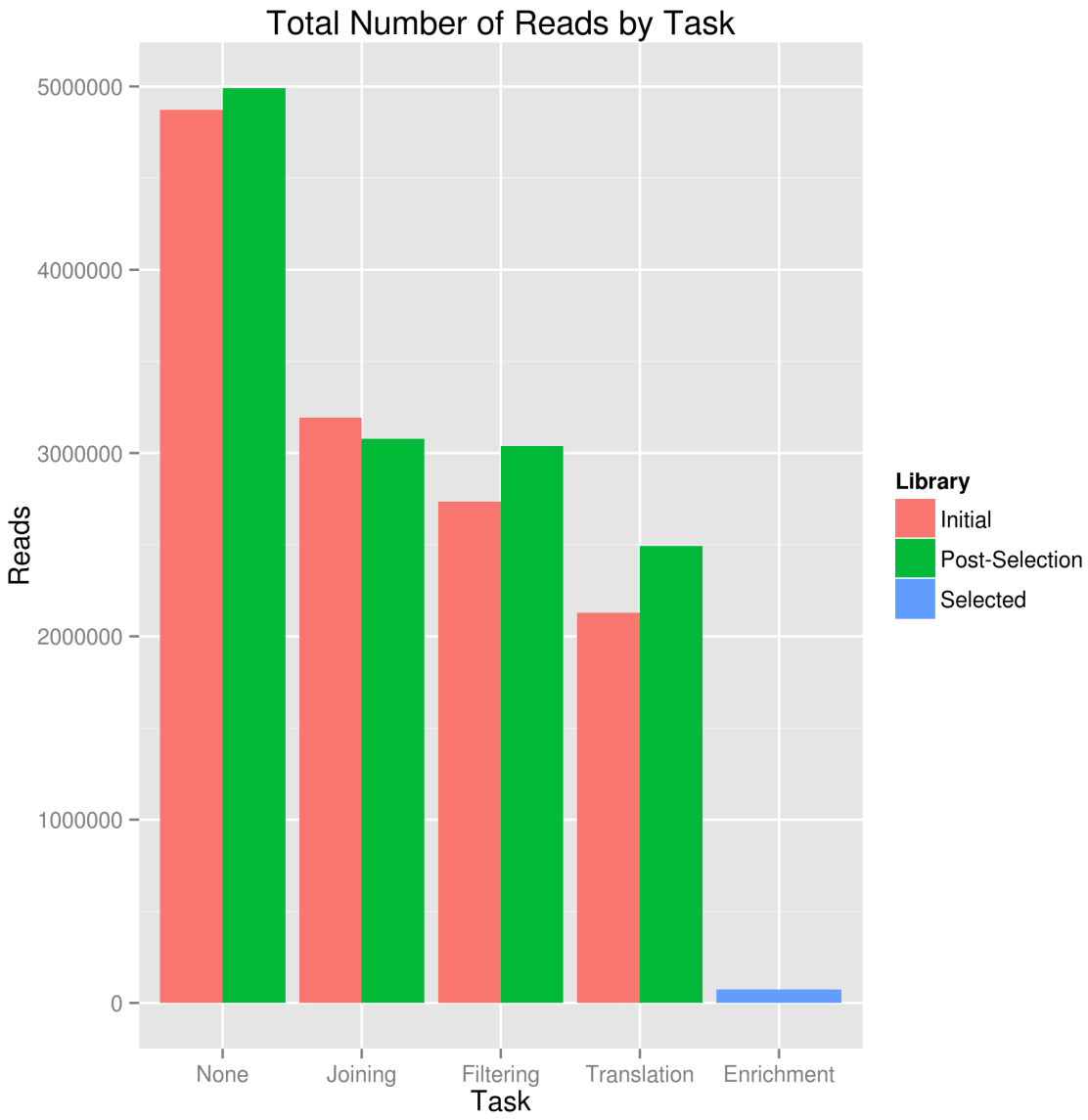

Figura 31: *Reads* das bibliotecas V_L do conjunto Illumina S1.

ANEXO C

Alinhamento de melhores *hits* com uma sequência da biblioteca final de VH do conjunto Illumina S1

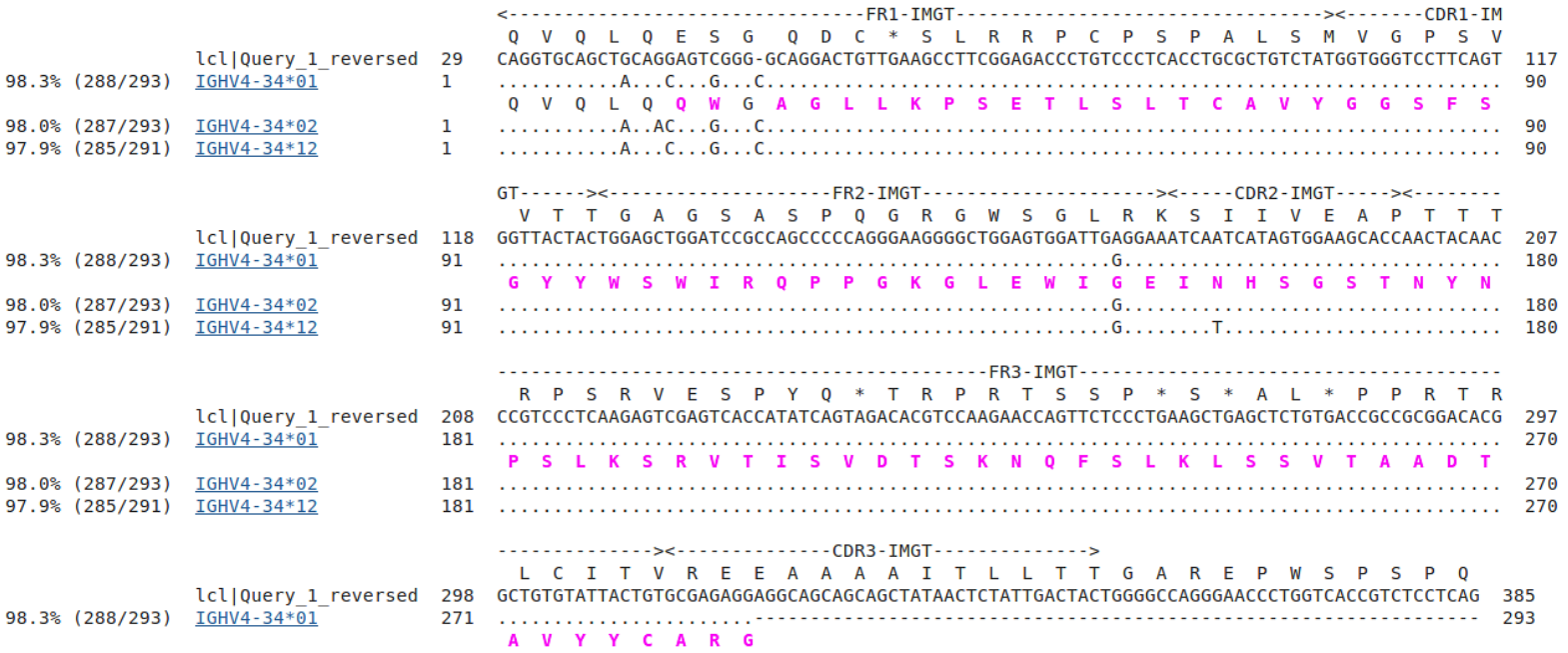

Figura 32: Alinhamento executado pela ferramenta IgBlast. Asteriscos representam códons de parada.